



Ricardo Filipe Henriques Ribeiro

Licenciatura em Ciências de Engenharia Biomédica

**Desenvolvimento de uma interface de visualização
de conectividade cerebral usando realidade virtual
e controlo por gestos**

Dissertação para obtenção do Grau de Mestre em
Engenharia Biomédica

Orientador: Hugo Alexandre Ferreira, Professor Doutor,
Faculdade de Ciências da Universidade de Lisboa

Júri:

Presidente: Prof. Doutora Carla Maria Quintão Pereira

Arguente: Prof. Doutor Mário António Basto Forjaz Secca

Vogal: Prof. Doutor Hugo Alexandre Ferreira



Novembro, 2014



Ricardo Filipe Henriques Ribeiro

Licenciatura em Ciências de Engenharia Biomédica

Desenvolvimento de uma interface de visualização de conectividade cerebral usando realidade virtual e controlo por gestos

Dissertação para obtenção do Grau de Mestre em
Engenharia Biomédica

Dissertação apresentada na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa para obtenção do Grau de Mestre em Engenharia Biomédica.

A presente dissertação foi desenvolvida em colaboração com o Instituto de Biofísica e Engenharia Biomédica (IBEB/FCUL).

Orientador: Professor Doutor Hugo Alexandre Ferreira

Desenvolvimento de uma interface de visualização de conectividade cerebral usando realidade virtual e controlo por gestos

Copyright © 2014 – Todos os direitos reservados. Ricardo Filipe Henriques Ribeiro, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Para os que estiveram sempre comigo e me querem o bem

Obrigado!

Agradecimentos

Mais uma meta que se atinge, mais um objetivo cumprido. Foi sempre esse o espírito para seguir com este caminho para a frente. Quando mais difícil de obter melhor sabe a vitória.

A presente dissertação decorreu na Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa e no Instituto de Biofísica e Engenharia Biomédica da Faculdade de Ciências da Universidade de Lisboa. Antes de mais quero agradecer a ambas as instituições pelas oportunidades que me deram ao longo da minha vida académica.

Em segundo lugar quero agradecer ao meu Orientador, Hugo Ferreira, por todo o apoio que deu e ideias que surgiram de longas conversas. Pela integração que me ofereceu no IBEB e em vários outros projetos. Agradeço ter-me recebido de tão boa vontade e pela confiança demonstrada ao longo de todo o tempo.

Agradeço a todos os docentes do IBEB que de certa forma me ajudaram na minha estadia. Um especial obrigado à Dona Beatriz que todos os dias nos começa por alegrar a manhã e quando era preciso algo, lá vinha a Dona Beatriz socorrer.

Quero deixar um agradecimento a todos os que tiveram paciência para me aturar no tempo que passaram pelo laboratório. Um especial agradecimento ao João Sousa, sem as ideias e apoio dado desde o início, esta ida para o IBEB não se tinha concretizado. Depois disto temos umas belas ponchas para festejar :P. Deixo outro agradecimento ao Filipe Valadas, o estagiário mais exemplar que passou pelo IBEB. Agradeço pela amizade que começou e que já deu frutos.

Ao pessoal do *Airsoft*, os jogos de fim-de-semana foram essenciais para aliviar a cabeça e apontar para mais uma semana de trabalho

À minha família, especialmente aos meus pais, por me oferecerem esta oportunidade e ver que vale sempre a pena bater com a cabeça às vezes. Porque água mole em pedra dura tanto bate até que fura.

Por fim, deixo o especial agradecimento à minha companheira de guerra desde o início, que graças a ela tive iniciativa para concluir o curso e ver que vale sempre a pena ter alguém a nosso lado com o mesmo objetivo. Obrigado Inês Neiva (Paixão :P);

FCT Fundação para a Ciência e a Tecnologia

MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Este trabalho foi financiado pela Fundação para a Ciência e Tecnologia (FCT) e Ministério da Ciência e Educação (MCE) Portugal (PIDDAC) integrado nos projetos PTDC/SAU-ENB/120718/2010 e PEst-OE/SAU/UI0645/2014.

Resumo

A visualização de dados médicos complexos, em particular de dados imagiológicos, de forma intuitiva, simples e completa é um desafio. Com as tecnologias de aquisição de imagem médica usadas atualmente é possível ter acesso a um vasto leque de informação relevante para diagnóstico e tratamento de patologias. Assim sendo, é fundamental que a visualização dessa mesma informação seja feita de uma forma rápida e intuitiva.

No presente trabalho desenvolveu-se uma nova ferramenta de visualização de imagens de ressonância magnética e de conectividade cerebral. Esta ferramenta contém em si uma interface amigável e intuitiva para o utilizador, fazendo uso de duas tecnologias emergentes, a realidade virtual e o reconhecimento gestual. No desenvolvimento da interface fez-se ainda uso de uma tecnologia menos usual na área médica, a *Game Engine Unity* 3D, usada convencionalmente na programação de jogos de vídeo.

A interface oferece um ambiente imersivo 3D da anatomia e conectividade cerebrais, fazendo uso de óculos de realidade virtual. Nesta interface o utilizador consegue navegar no interior no cérebro, fazendo uma interação com as estruturas com recurso ao controlo do Leap Motion, um dispositivo ótico de reconhecimento gestual.

Esta interface de "navegação cerebral" será útil no futuro para a comunidade médica, em particular para a formação de neurocirurgiões graças ao reconhecimento gestual. Além disso, a interface fornece informações sobre as estruturas cerebrais e vários parâmetros de dados de imagens e de conectividade cerebral, tornando-se adequada para o ensino e investigação em neurociências.

Palavras-chave: Realidade Virtual, Controlo Gestual, *Game Engine, Unity*, Navegação Cerebral, Conectividade.

Abstract

The visualization of complex medical data, in particular imaging data, in an intuitive, simple and comprehensive way is a challenge. With the technologies of medical imaging currently used it is possible to have access to a wide range of relevant information for the diagnosis and treatment of diseases. Therefore, it is crucial that the visualization of this information is done in a quick and intuitive way.

In this work a novel tool for visualizing Magnetic Resonance Images and derived brain connectivity data was developed. This tool contains a user-friendly interface which makes use of two emerging technologies, virtual reality and the gesture recognition. In the development of the interface it was also used the Unity 3D Game Engine, conventionally used for video game programming.

The interface provides an immersive 3D environment of the brain anatomy and connectivity, by making use of virtual reality glasses. In this interface, the user is able to navigate inside the brain, making an interaction with brain structures with the Leap Motion device, an optical gesture recognition device.

This "brain navigation" interface will be useful in the future for the medical community, in particular in the training of neurosurgeons thanks to the gesture recognition capabilities. In addition, the interface provides information about brain structures and various parameters of image data, and brain connectivity, thus making it suitable for teaching and research in neuroscience.

Keywords: Virtual Reality, Gesture Control, Game Engine, Unity, Brain Navigation, Connectivity.

Índice de conteúdos

Agradecimentos	vii
Resumo	ix
Abstract.....	xi
Índice de conteúdos	xiii
Índice de tabelas	xvii
Índice de figuras	xix
Siglas e Acrónimos.....	xxiii
1 Introdução.....	1
1.1 Motivação.....	1
1.2 Estrutura da Dissertação.....	2
2 Conectividade cerebral.....	5
2.1 Conectividade cerebral	6
2.1.1 Introdução à conectividade cerebral.....	6
2.1.2 Caracterização e métricas de conectividade.....	6
2.2 Pacotes de software de visualização de dados de imagem médica.....	9
3 Sistemas de Realidade Virtual e Interfaces Pessoa Máquina	15
3.1 Realidade Virtual.....	15
3.2 Primeiros sistemas de realidade virtual.....	15
3.3 Head Mounted Display.....	18
3.3.1 <i>Oculus Rift</i>	18
3.3.1.1 Desenvolvimento de <i>Software</i>	21
3.3.2 <i>Wireless Personal 3D Viewer</i>	21
3.3.2.1 Desenvolvimento de <i>Software</i>	23

3.4	Aplicações da Realidade Virtual.....	23
3.4.1	Realidade Virtual em Medicina	23
3.5	Interfaces Pessoa Máquina.....	26
3.5.1	Leap Motion	26
3.5.1.1	Desenvolvimento de aplicações	27
3.5.1.2	Funcionamento do dispositivo	28
3.5.1.3	Desenvolvimento do Leap Motion	29
3.5.2	<i>Virtuix Omni Natural Motion Interface</i>	29
4	Desenvolvimento de aplicações 3D	31
4.1	Game Engines.....	31
4.1.1	<i>Unity</i> 3D	33
4.1.1.1	Interface de Utilização.....	34
4.1.2	<i>Unreal Engine</i>	36
4.1.2.1	Interface de Utilização.....	36
4.2	<i>Unity</i> 3D vs. <i>Unreal Engine</i>	38
5	Metodologia de trabalho.....	41
5.1	Processamento das Imagens.....	42
5.1.1	Preparação das imagens estruturais.....	42
5.1.2	Preparação dos dados de conectividade	44
5.2	Desenvolvimento da aplicação	44
5.3	Estrutura da aplicação.....	45
5.3.1	Interface de visualização de dados estruturais	46
5.3.2	Interface de visualização de dados de conectividade.....	48
5.4	Controlo gestual	53
6	Resultados obtidos.....	55

6.1	Interface de visualização de imagens estruturais.....	55
6.2	Interface de visualização de conectividade	58
6.3	Pontos de destaque da aplicação.....	64
7	Conclusão.....	65
7.1	Limitações da aplicação.....	65
7.2	Trabalho futuro.....	66
7.3	Conclusões finais	67
8	Bibliografia.....	69
Anexo I – Segmentação Cortical do Freesurfer.....		79
I.I	- Atlas Destrieaux.....	79
I.II	- Atlas Desikan-Killiany-Tourville	82
Anexo II – Segmentação Cortical segundo AAL.....		85

Índice de tabelas

Tabela 1.1 – Estrutura da dissertação	3
Tabela 4.1 – Comparação entre o <i>Unity</i> e o <i>Unreal Engine</i> . Adaptado de [59]	38
Tabela I.I – Identificação das estruturas da parcelização segundo o atlas <i>Destrieaux</i> [83]	81
Tabela I.II – Identificação das estruturas da parcelização segundo o atlas Desikan-Killiany-Tourville [84], [85].....	83
Tabela II.I – Lista de labels das ROI's do atlas AAL [86].....	87

Índice de figuras

Figura 2.1 - Representação gráfica das três redes neuronais e as correspondentes matrizes de conectividade ou de adjacência. Para orientar o sentido das redes as setas indicam o sentido do fluxo de informação. Adaptado de [4].....	7
Figura 2.2 – Processo para criação de um grafo de conectividade. Adaptado de [10].....	8
Figura 2.3 – Esquema das métricas de conectividade estrutural que se pode usar [13]	9
Figura 2.4 – Interface do MRIcro [15]	10
Figura 2.5 – Janelas de visualização do <i>software</i> OsiriX [18].....	11
Figura 2.6 – Visualização de dados no TrackVis [22]	12
Figura 2.7 – Interface da aplicação BrainCAT [29]	13
Figura 2.8 – Conectograma obtido pela MIBCA. Nos anéis exteriores para o interior temos as ROI's, Padrão de Captação de Valor, média de difusão, o volume da ROI, espessura cortical e o grau nó calculado a partir conectividade estrutural [28]	14
Figura 3.1 – Sensorama, sistema criado por Morton Heilig [32].....	16
Figura 3.2 – Na imagem à esquerda encontra-se o sistema <i>Headsign</i> desenvolvido pela Philco Corporation. [33] Na imagem à direita o sistema <i>Ultimate Display</i> [35].....	16
Figura 3.3 – <i>EyePhone</i> em conjunto com <i>DataGlove</i> , o primeiro HMD comercial desenvolvido pela empresa VPL [33].....	17
Figura 3.4 – Primeiro <i>Development Kit</i> fornecido. Na imagem da esquerda, é a vista geral dos <i>Oculus Rift</i> , a central a vista da interface, a da direita é a caixa de controlo [41]....	18
Figura 3.5 – Componentes dos <i>Oculus Rift</i> . A imagem em cima à esquerda é a vista do ecrã. A imagem em cima à direita ilustra a ligação entre o ecrã e o sistema Head Tracking. A imagem em baixo ilustra a <i>board Oculus Tracker V2</i> , responsável pelo sistema <i>Head Tracking</i> . A vermelho está indicado um microcontrolador STMicroelectronics 32F103C8 ARM Cortex-M3 com 72Mhz CPU. A amarelo está indicado um compasso digital tri-axial Honeywell HMC5983 para compensação do giroscópio. A laranja está indicado um	

controlador de posição Invensense MPU-6000 de seis eixos, em que associa giroscópio e acelerómetro. [41]	19
Figura 3.6 – Aplicação do efeito barril e efeito almofada à imagem [42].....	20
Figura 3.7 – Caixa de controlo <i>Oculus Rift</i> com respetivas portas de conectividade [41] 20	
Figura 3.8– HMZ-T3W da Sony. Na imagem em cima à esquerda, é a vista geral do HMZ-T3W, na imagem em cima à direita a vista da interface do utilizador, na imagem em baixo a caixa de controlo do HMD [43]	22
Figura 3.9 – Sistema de realidade virtual de visualização de tractografia [48]	24
Figura 3.10 – Robot para cirurgia de laparoscopia. Requer pequenas incisões entre 0.5 a 1.5 cm de comprimento no paciente, onde pequenas câmaras e instrumentos cirúrgicos são inseridos [49]	24
Figura 3.11 – Sistemas de coordenadas do Leap Motion e campo de visão [64], [65]	27
Figura 3.12 - Representação das métricas obtidas pelo Leap Motion na ponta dos dedos [64]	28
Figura 3.13 – Representação das métricas obtidas pelo Leap Motion na palma da mão [64]	28
Figura 3.14 – Integração de <i>Oculus Rift</i> e Leap Motion lançada no final de Agosto 2014 [66]	29
Figura 3.15 – Utilização do sistema <i>Virtuix Omni</i> , juntamente com <i>Oculus Rift</i> [67]	30
Figura 3.16 – Sapatos do <i>Virtuix Omni</i> [61]	30
Figura 4.1 - Estrutura Modular de uma <i>Game Engine</i> (ver explicação no texto) [70].....	32
Figura 4.2 – Sistema CaveUT criado com uma <i>Game Engine</i> [72]	33
Figura 4.3 – Interface da Game Engine <i>Unity</i> [75].....	34
Figura 4.4 – Interface Editor do <i>Unreal Engine</i> . Adaptado de [79]	37
Figura 4.5 – <i>Glass Brain</i> desenvolvido em <i>Unity</i> no Sandler Neurosciences Center da Universidade da Califórnia [81]	39
Figura 5.1 – Interface menu da aplicação.....	45

Figura 5.2 – Menu para visualização de imagens estruturais.....	46
Figura 5.3 – Fluxograma da metodologia da interface de visualização de imagens estruturais	47
Figura 5.4 - – Interface de visualização de conectividade	48
Figura 5.5 – GUI do PickAtlas. À esquerda encontramos os atlas disponíveis para extração de máscaras, à direita as máscaras escolhidas [92].....	50
Figura 5.6 – Parcelizações efetuadas nos diferentes atlas cerebrais usados. A – Parcelização do AAL. B – Parcelização cortical HOA. C – Parcelização do TA. D – Parcelização DCA. E – Parcelização DKCA. Imagens obtidas com recurso ao <i>software</i> ITK-SNAP [94].....	51
Figura 5.7 - Fluxograma da metodologia da interface de visualização da matriz de conectividade	52
Figura 5.8 – Forma de utilização do <i>Touchless</i> , aplicação que simula um ecrã tátil virtual [95].....	53
Figura 5.9 – Lista de gestos pré-definidos pela API do Leap Motion [64]	54
Figura 6.1 – Janela de seleção de ficheiros adaptada de [89] para as necessidades do projeto	56
Figura 6.2 – Seleção de estruturas a visualizar pelo utilizador	56
Figura 6.3 – Erro apresentado ao utilizador quando não existe vista selecionada	57
Figura 6.4 – Renderização das zonas subcorticais do Freesurfer em ecrã de computador	57
Figura 6.5 - Renderização das zonas subcorticais no HMD	58
Figura 6.6 - Janelas de seleção de ROI's. A – Seleção de ROI's do atlas DKCA. B – Seleção de ROI's do atlas DCA. C – Seleção de ROI's do atlas AAL. D – Seleção de ROI's do atlas TA.	59
Figura 6.7 – Visualização em PC de ROI's analisadas do atlas AAL. Nas duas imagens superiores foram selecionadas as ROI's com as labels de 1 a 8. Nas duas imagens inferiores tem-se a visualização de todas as ROI's do atlas AAL.....	60
Figura 6.8 – Visualização da cena no ecrã do HMD	61
Figura 6.9 – Erro apresentado ao utilizador quando não existe vista selecionada	61

Figura 6.10 – Erro apresentado ao utilizador quando não existe matriz de conectividade	62
Figura 6.11 – Visualização da matriz de conectividade de um sujeito de estudo, com estatísticas globais da rede.....	62
Figura 6.12 – Navegação pelo grafo de conectividade com visualização das estatísticas do nodo consideradas para o estudo.....	63
Figura 6.13 – Visualização no HMD do grafo com navegação pelos nodos	63
Figura 6.14 – Navegação pelo interior do grafo com o HMD	64
Figura I.I – Parcelização cortical segundo o atlas <i>Destrieaux</i> [83]	79
Figura I.II – Parcelização cortical segundo o atlas Desikan-Killiany-Tourville [84]	82
Figura II.I – ROI's do AAL no plano em axial. Valores no canto inferior direito de cada fatia indica a coordenada estereotáxica z em milímetros (z=19 mm a z=75 mm) [91]	85
Figura II.II – ROI's do AAL no plano em axial. Valores no canto inferior direito de cada fatia indica a coordenada estereotáxica z em milímetros (z=-43 mm a z=13 mm) [91] ...	86

Siglas e Acrónimos

2D - Bidimensional

3D – Tridimensional

AAL - *Automated Anatomical Labeling*

API - Interface de Programação de Aplicações (do inglês *Application Programming Interface*)

BCI - *Brain Computer Interfaces*

BMP - *Bitmap Image*

CMOS - *Complementary Metal-Oxide-Semiconductor*

CPU - Unidade Central de Processamento (do inglês *Central Processing Unit*)

CRT – Tubo de Raios Catódicos (do inglês *Cathode Ray Tube*)

CT - Tomografia Computorizada (do inglês *Computed Tomography*)

DCA - Dextrieux Cortical Atlas

DICOM - *Digital Imaging and Communications in Medicine*

DKCA - Desikian Killiany Cortical Atlas

DTI – Imagem de Tensor de Difusão (do inglês *Diffusion Tensor Imaging*)

EEG – Eletroencefalografia (do inglês *Electroencephalography*)

FLAIR - *Fluid attenuated inversion recovery*

fMRI - Imagem de Ressonância Magnética Funcional (do inglês *Functional Magnetic Resonance Imaging*)

FoV – Campo de Visão (do inglês *Field of Vision*)

FPS – Tiro em Primeira Pessoa (do inglês *First-Person Shooter*)

FSA - Freesurfer Subcortical Atlas

FSL - FMRIB *Software Library*

GPU - Unidade de Processamento Gráfico (do inglês *Graphics Processing Unit*)

GUI - *Graphical User Interface*

HMD - *Head Mounted Display*

HMZ-T3W - Wireless Personal 3D Viewer

HOA - Harvard Oxford Atlas

IBEB - Instituto de Biofísica e Engenharia Biomédica

JPEG - *Joint Photographic Experts Group*

LCD - *Liquid Crystal Display*

MIBCA - *Multimodal Image Brain Connectivity Analysis*

MNI - *Montreal Neurological Institute*

MRI – Imagem de Ressonância Magnética (do inglês *Magnetic Resonance Imaging*)

NIFTI – *Neuroimaging Informatics Technology Initiative*

OLED - *Organic Light-Emitting Diode*

OS – Sistema Operativo (do inglês *Operating System*)

PC – Computador Pessoal (do inglês *Personal Computer*)

PET - Tomografia por Emissão de Positrões (do inglês *Positron Emission Tomography*)

ROI – Região de Interesse (do inglês *Region Of Interest*)

SDK – Conjunto de desenvolvimento de *software* (do inglês *Software Development Kit*)

TA - Talairach Atlas

USB - *Universal Serial Bus*

VR – Realidade Virtual (do inglês *Virtual Reality*)

1

1 Introdução

Ao longo da evolução tecnológica, o ser humano sempre ambicionou chegar mais longe, procurando sempre superar as metas anteriormente atingidas. Este facto constatou-se em todas as áreas do saber e a imagem médica não foi uma exceção. Desde os tempos da antiguidade que o ser humano tem uma curiosidade tremenda sobre o seu próprio corpo, sendo que as primeiras imagens anatómicas internas foram obtidas por autópsias. Desde então, a imagem médica tem vindo a evoluir num sentido a ser minimamente invasiva, tendo sempre em vista a obtenção de imagens precisas mas ao mesmo tempo interativas.

1.1 Motivação

A visualização de dados médicos complexos, e em particular de dados imagiológicos, de forma intuitiva, simples e completa é um desafio. Atualmente a visualização de dados de imagem médica é feita a partir de *softwares* de processamento e análise de dados, sendo que muitas vezes esses mesmos não se apresentam como sendo multiplataforma nem de utilização intuitiva.

Com recurso ao uso de novas tecnologias, como a robótica e a realidade virtual, o utilizador estende a perceção dos cinco sentidos, representando mais do que o real das coisas. Para tal, existem três conceitos que estão interligados e que são responsáveis pela expansão de sentidos [1]:

- Imersão: o utilizador fica envolvido no meio envolvente. Isto é visto como um fator mental, com a experiência das imagens 3D. No entanto, também pode ser uma experiência física com recurso a visualizadores estereoscópicos e plataformas de gravidade [1];
- Navegação: o utilizador consegue navegar pelo mundo virtual [1];
- Interação: o utilizador consegue controlar o mundo virtual de uma forma simples e intuitiva como se se tratasse do meio real [1].

Com base nestes conceitos, foi vista uma oportunidade de criar uma nova ferramenta de visualização de dados de imagiologia por ressonância magnética (do inglês *Magnetic Resonance Imaging*, MRI), uma ferramenta intuitiva, usando duas tecnologias emergentes, a realidade virtual e o reconhecimento gestual.

O principal objetivo deste trabalho tinha em mente o desenvolvimento de uma interface que permitisse a visualização, quer no monitor de computador quer num sistema realidade virtual. Pretendia-se fazer uma renderização da superfície cerebral e estruturas subcorticais com sobreposição de volumes de interesse, mapas funcionais, grafos derivados de estudos de conectividade cerebral e outros dados de interesse.

1.2 Estrutura da Dissertação

O trabalho aqui apresentado foi desenvolvido numa parceria entre o Instituto de Biofísica e Engenharia Biomédica (IBEB), situado na Faculdade de Ciências da Universidade de Lisboa e a Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

A presente dissertação encontra-se dividida em oito capítulos resumidos na tabela 1.1. No primeiro capítulo é feita a introdução e motivação para a realização da dissertação. No segundo capítulo introduz-se a MRI a conectividade cerebral e *softwares* de visualização de dados. No terceiro capítulo é feita uma descrição das tecnologias de realidade virtual e controlo gestual, visto ser um ponto fundamental da dissertação. O quarto capítulo incide sobre *Game Engines*, *softwares* desenvolvidos para facilitar a criação de aplicações 3D, com ferramentas de renderização de imagem. O quinto capítulo descreve a metodologia de tra-

balho seguida para a criação da aplicação proposta, desde a obtenção de imagens anatómicas, à criação da interface proposta. No sexto capítulo faz-se a análise do trabalho realizado e resultados obtidos. No sétimo capítulo é feita a conclusão, limitações do trabalho realizado e futuro trabalho a desenvolver. Por fim no oitavo capítulo é possível encontrar toda a bibliografia consultada para a realização da dissertação.

Conceitos teóricos	Metodologia	Resultados	Conclusão
<ul style="list-style-type: none">• Conectividade cerebral• Realidade Virtual• Controlo Gestual• Game Engines	<ul style="list-style-type: none">• Desenvolvimento da aplicação	<ul style="list-style-type: none">• Interface desenvolvida• Análise da interface	<ul style="list-style-type: none">• Limitações do trabalho desenvolvido• Trabalho futuro a desenvolver• Conclusão Final

Tabela 1.1 – Estrutura da dissertação

2

2 Conectividade cerebral

A ressonância magnética é das tecnologias mais utilizadas para obtenção de imagens do corpo humano. Apesar das suas inúmeras aplicações é na área das neurociências que esta técnica tem especial importância. A partir das imagens obtidas e com recurso aos inúmeros *softwares* existentes no mercado é possível fazer um estudo extensivo do cérebro, nomeadamente estudos estruturais, funcionais entre outros.

Atualmente existem vários programas de processamento de dados de imagem cerebral, entre eles o Freesurfer [2] e o FMRIB *Software Library* (FSL) [3]. No entanto, uma das desvantagens destes é a nível de visualização de dados, sendo necessário experiência alargada no funcionamento do *software*.

Uma das áreas das neurociências que tem vindo a ser mais explorada é a da conectividade cerebral. Têm sido feitos estudos de conectividade de forma a estabelecer padrões entre doenças neurológicas e a conectividade funcional e estrutural entre as diferentes regiões do cérebro [4].

Neste contexto foi percebida uma oportunidade de criar uma interface inovadora de visualização de dados estruturais e funcionais de MRI de uma forma intuitiva e eficaz, sendo nesta ideia que a presente dissertação está assente.

Em seguida será feito uma breve descrição sobre a conectividade cerebral e *softwares* existentes de análise de dados de conectividade.

2.1 Conectividade cerebral

2.1.1 Introdução à conectividade cerebral

Ao longo dos últimos anos a investigação em neurociências tem vindo a crescer. Tal facto é derivado das novas tecnologias que permitem avaliar estrutura e atividade cerebral de uma forma não invasiva. Atualmente uma das áreas que tem sofrido maior avanço e que tem despertado maior interesse junto da comunidade científica é a conectividade cerebral [4], [5].

Conectividade cerebral é um conceito abrangente, encontrando-se dividido em três tipos de conectividade: estrutural, funcional e efetiva. [4]. A conectividade estrutural corresponde à conexão anatómica entre regiões cerebrais, tipicamente correspondendo aos tratos de substância branca [4], [5]. A conectividade funcional é definida como a correlação temporal entre acontecimentos neurofisiológicos espacialmente remotos, analisando eventos entre de neurónios distribuídos por grupos e áreas diferentes [5], [6]. A conectividade efetiva deriva de causas diretas da influência que uma região cerebral produz noutra [4], [7].

2.1.2 Caracterização e métricas de conectividade

As redes neuronais são complexas, partilhando um certo número de características comuns com redes de outros sistemas físicos e biológicos [8]. Uma rede neuronal consiste numa estrutura de conexão, na qual o processamento se encontra distribuído por um grande número de pequenas unidades de processamento (nodos) densamente interligadas (conexões) [4], [9]. Os nodos representam regiões cerebrais, enquanto as conexões representam conexões anatómicas, funcionais e efetivas [8]. As conexões têm um papel fundamental no estudo das redes neuronais, uma vez que são elas que fornecem uma forma de caracterização de rede.

As redes podem ser representadas graficamente ou na sua forma matricial numa matriz quadrada, onde uma conexão é representada por um elemento da matriz.

O grafo da rede pode classificar-se em *directed* ou *undirected* e também em *weighted* ou *unweighted* [4], [5], [8]. No grafo *unweighted*, cada conexão tem a mesma força, sendo que caso a conexão exista esta é representada por 1, caso contrário por 0. No grafo *weighted*

as conexões são ponderadas (por exemplo o número de fibras que ligam duas ROI's). No grafo *undirected* as conexões são bidirecionais, isto é a sua matriz de conectividade apresenta simetria relativamente à diagonal principal. No grafo *directed* a matriz de conectividade não é simétrica dado que as conexões são unidirecionais [4], [7], [8]. A seguinte figura ilustra três tipos de redes:

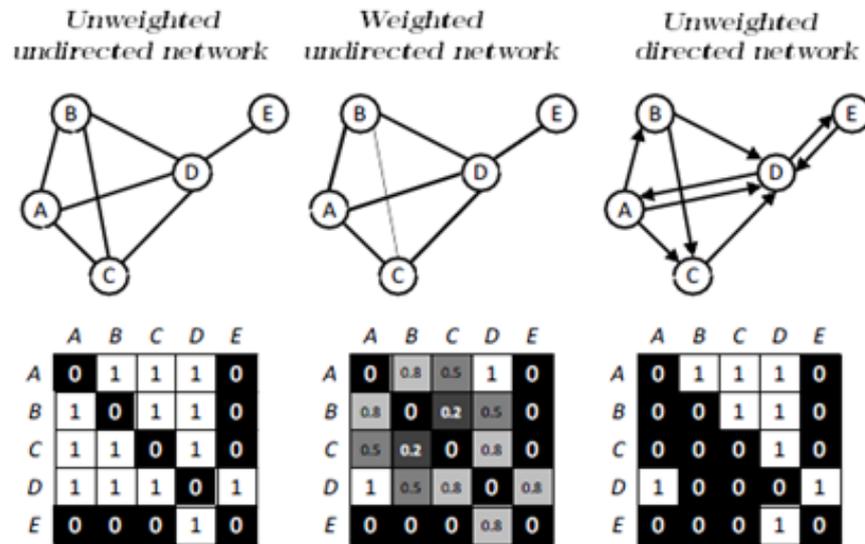


Figura 2.1 - Representação gráfica das três redes neuronais e as correspondentes matrizes de conectividade ou de adjacência. Para orientar o sentido das redes as setas indicam o sentido do fluxo de informação. Adaptado de [4].

Obter uma rede neuronal de conexões é um processo realizado em fases. Este pode variar de acordo com o estudo que se realiza, mas maioritariamente pode ser dividido em 5 fases, ilustradas na figura 2.2:

- Fase 1: Aquisição dos dados de MRI. É feito um exame de acordo com os protocolos de aquisição para o tipo de estudo que se pretende fazer. É feita posteriormente uma reconstrução de sinal sendo os dados armazenados no formato DICOM [10].
- Fase 2: Os dados anteriormente obtidos são processados por *softwares* de imagem médica, como por exemplo o Freesurfer ou SPM [11], segmentando o cérebro em substância branca e cinzenta e líquido cefalorraquidiano [10].
- Fase 3: São criadas regiões de interesse (ROI) normalizadas. Esta partição do cérebro em ROI's é feita com recurso a atlas padronizados que contêm

como que uma etiquetagem de cada ROI. É feito então um registo dessas etiquetas no cérebro segmentado [10].

- Fase 4: Dados de fMRI e de tensor de difusão são, em paralelo, também processados com recurso a *softwares* como FSL, SPM ou TrackVis [12]. São obtidas séries temporais filtradas a partir de dados de fMRI e tensores de tractografia a partir de dados de difusão [10].
- Fase 5: Usando a partição da fase 3 e os dados da fase 4 são obtidas relações entre ROI's resultantes de correlações temporais, relações causais através de dados de fMRI e ligações estruturais através da tractografia. São obtidas, por fim, matrizes e grafos de conectividade funcional, efetiva ou estrutural respetivamente [10].

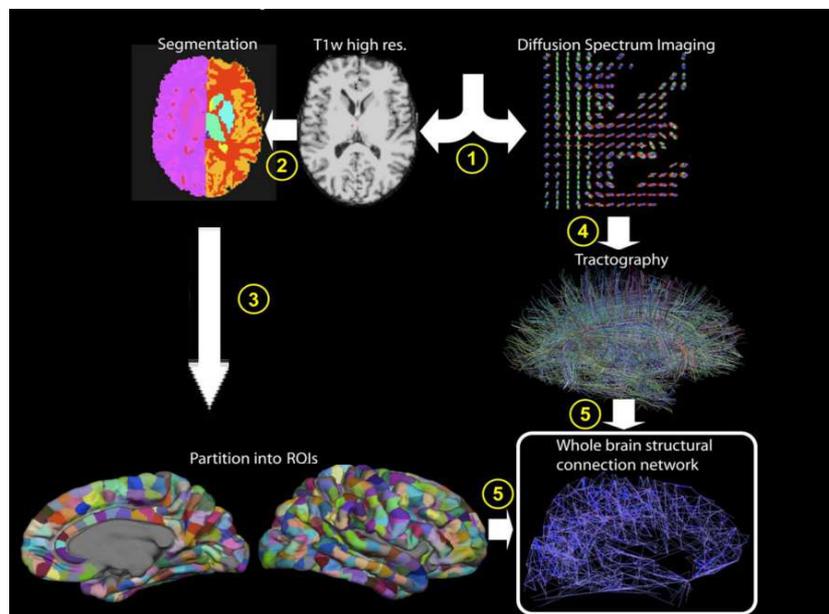


Figura 2.2 – Processo para criação de um grafo de conectividade. Adaptado de [10]

A partir dos dados de conectividade das redes neuronais podem ser extraídas certas métricas. Essas métricas permitem determinar a topologia da rede e a sua respetiva eficiência. Existem várias métricas que se podem considerar, sendo que a sua relevância de utilização varia de acordo com o estudo que se está a considerar. Nesta dissertação para desenvolvimento de *software* foram considerados dados de conectividade estrutural. A variedade de métricas que se podem usar estão apresentadas figura 2.3:



Figura 2.3 – Esquema das métricas de conectividade estrutural que se pode usar [13]

A análise as métricas de conectividade tem-se revelado uteis dado que:

- Permite uma caracterização global das redes neuronais [14];
- Permite a análise de relações entre conectividade funcional e estrutural [14];
- Permite comparar entre parâmetros de redes de diferentes populações de indivíduos, de forma a detetar alterações associadas a doenças neuropsiquiátricas [14];

2.2 Pacotes de *software* de visualização de dados de imagem médica

Existem alguns pacotes de *softwares* disponíveis que fornecem uma visão 3D de dados anatómicos e funcionais de MRI. O MRICro é um dos mais populares visualizadores de imagens médicas. Neste *software* pode-se fazer uma renderização de superfícies ou volumes cerebrais com os respetivos mapas estatísticos.

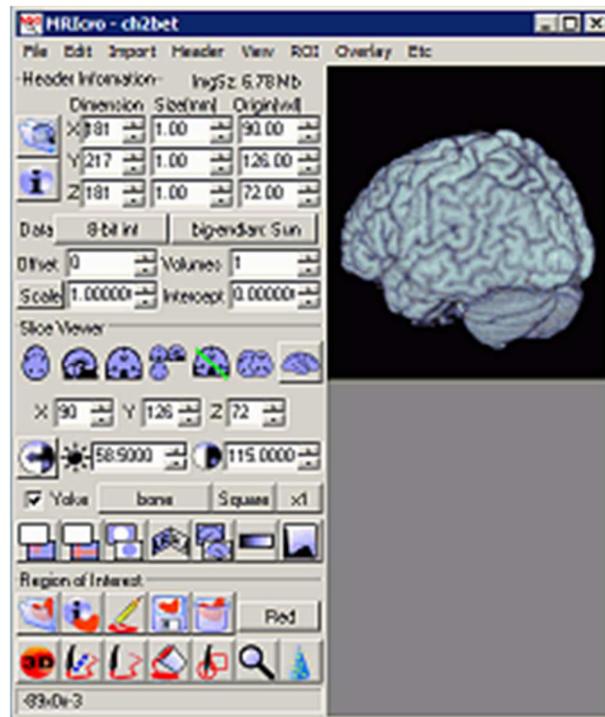


Figura 2.4 – Interface do MRIcro [15]

As principais características de destaque da aplicação são [15]:

- Converte imagens médicas para os formatos mais comuns de *software* de análise estatística como o SPM;
- Permite visualizar imagens já processadas;
- Permite criar ROI's em 3D;
- Permite a sobreposição de vários atlas de ROI's;
- Permite fazer ajustes de imagens para *softwares* de processamento como o SPM;
- Permite exportação de imagens para os formatos mais acessíveis como BMP ou JPEG.

Outros pacotes como o 3DView, Slicer, DataViewer3D, ou FSLView estão providos das mesmas capacidades [16].

Outro *software* que se destaca é o OsiriX. Este trata-se de um *software* gratuito disponível apenas para Mac OSX e iOS [17]. Este *software* pode ser usado para visualizar conjuntos de imagens médicas no formato DICOM e extrair informação visual de imagens

de CT, MRI, Tomografia por Emissão de Positrões (do inglês *Positron Emission Tomography*, PET), etc. A navegação pelo *software* é conseguida com recurso a três janelas principais, como é possível ver na figura 2.7. Na primeira janela, a *Data Window*, estão as imagens importadas para a aplicação. A segunda janela, a *Viewer Window*, permite ao utilizador visualizar e manipular conjuntos de imagens em 2D. A terceira janela, a *3D Volume Rendering*, o utilizador tem acesso à imagem 3D gerada pelo conjunto de imagens 2D [18].

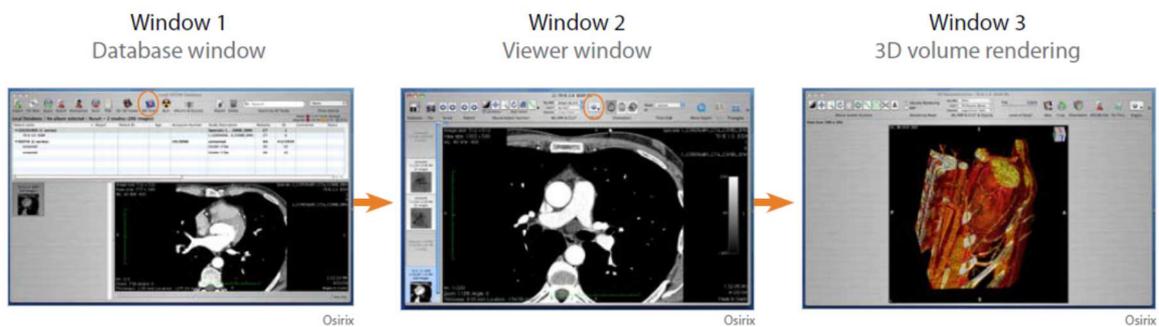


Figura 2.5 – Janelas de visualização do *software* OsiriX [18]

Um ponto forte desta aplicação é a possibilidade do utilizador desenvolver *plugins* para juntar às funcionalidades deste programa. Um exemplo disso foi um *plugin* desenvolvido por L.C. Ebert onde as imagens DICOM são controladas por gestos com recurso a camara Kinect da Microsoft [19]. Este trabalho pode ser encontrado em mais pormenor na referência [20].

Múltiplos *softwares* de análise de redes cerebrais estão disponíveis gratuitos na *internet*. Estes *softwares* estão desenvolvidos em linha de comandos nas mais populares linguagens como Matlab e Python, bem como em versões com interface gráfica [21].

O TrackVis da figura 2.6 é um *software* que permite visualizar e analisar dados de tractografia [12]. As principais características de destaque deste *software* são:

- É multiplataforma, estando disponível para Windows, Mac OS X e Linux [12];
- Tem diferentes ferramentas de análise disponíveis para tractografia [12];
- Diferentes modos de fazer a renderização de imagem [12];
- Ajuste de parâmetros em tempo real [12];
- Análise estatística escalar de tratos e ROI's [12];

- Permite a comparação de vários conjuntos de dados, permitindo comparação entre indivíduos [12];

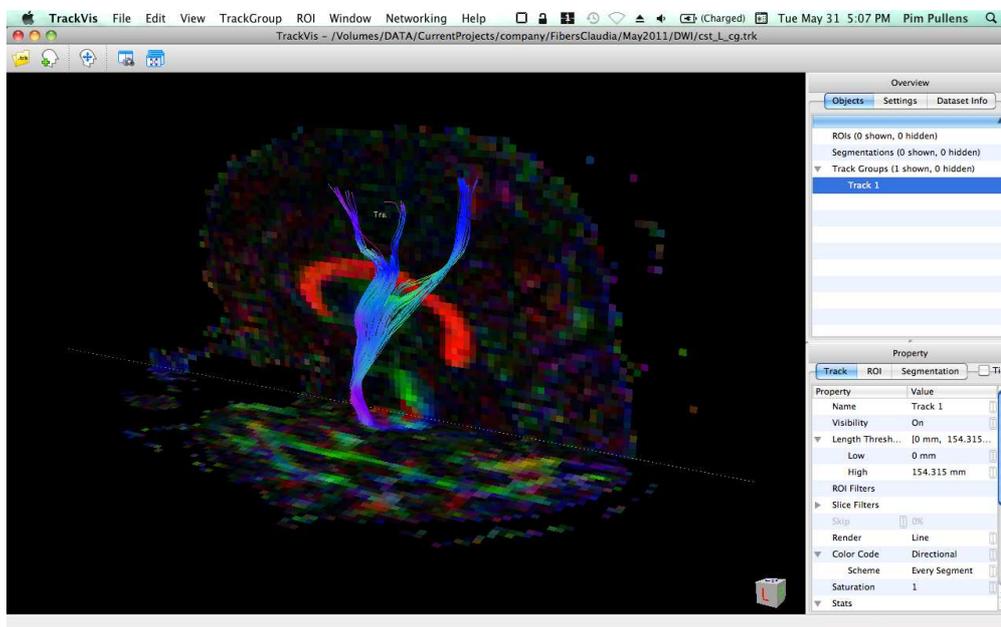


Figura 2.6 – Visualização de dados no TrackVis [22]

Disponível para Matlab [23], temos, por exemplo, a *toolbox Functional Connectivity Toolbox* [24]. Esta *toolbox* é utilizada para realizar análises de conectividade funcional, utilizando a maioria das abordagens mais usadas pela comunidade científica para identificação de dependências funcionais entre séries de temporais de fMRI obtidas de duas ou mais regiões cerebrais [25]. Outra *toolbox* que se destaca é a *Brain Connectivity Toolbox* [26]. Esta *toolbox* fornece ferramentas para caracterização de conjunto de dados estruturais e funcionais [26].

Existem vários programas disponíveis que suportam visualização de conectividade cerebral, como por exemplo o BrainCAT [27] e a *toolbox Multimodal Image Brain Connectivity Analysis* (MIBCA) [28]. O BrainCAT é uma ferramenta para análise de conectividade cerebral de imagens de ressonância magnética funcional, desenvolvido na Universidade do Minho. O BrainCAT foi criado com o objetivo de tornar o processamento de dados mais fácil e mais rápido e minimizando a necessidade de intervenção do utilizador [29]. A figura 2.7 ilustra a interface de utilização de esta ferramenta.

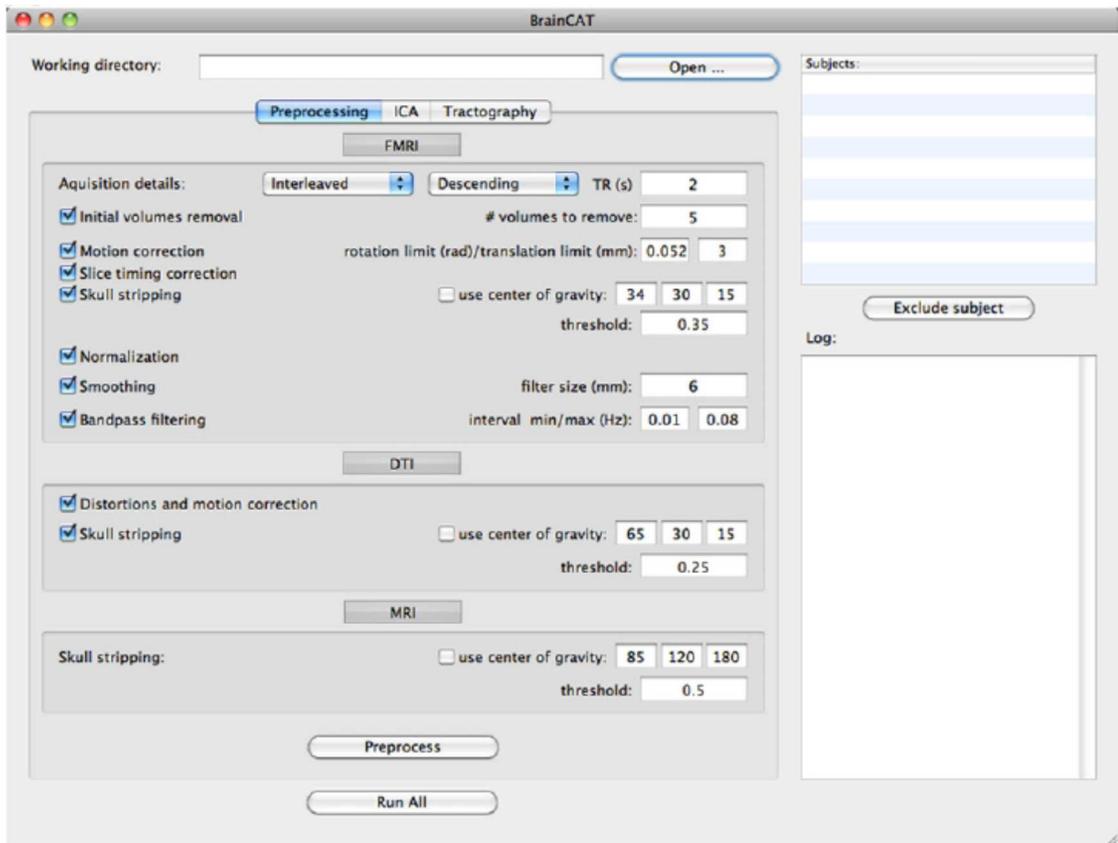


Figura 2.7 – Interface da aplicação BrainCAT [29]

A *toolbox* MIBCA consiste num conjunto de ferramentas de análise totalmente automatizada conectividade que oferece pré-processamento de dados, análise de conectividade e análise de teoria dos grafos de imagens multimodais de MRI anatómica, funcional, de difusão e PET. Esta ferramenta foi desenvolvida no IBEB, tendo já demonstrado bons resultados na análise de dados de doentes com Alzheimer [28]. Esta ferramenta permite a visualização da conectividade em forma de um conectograma, como o ilustrado na figura 2.8.

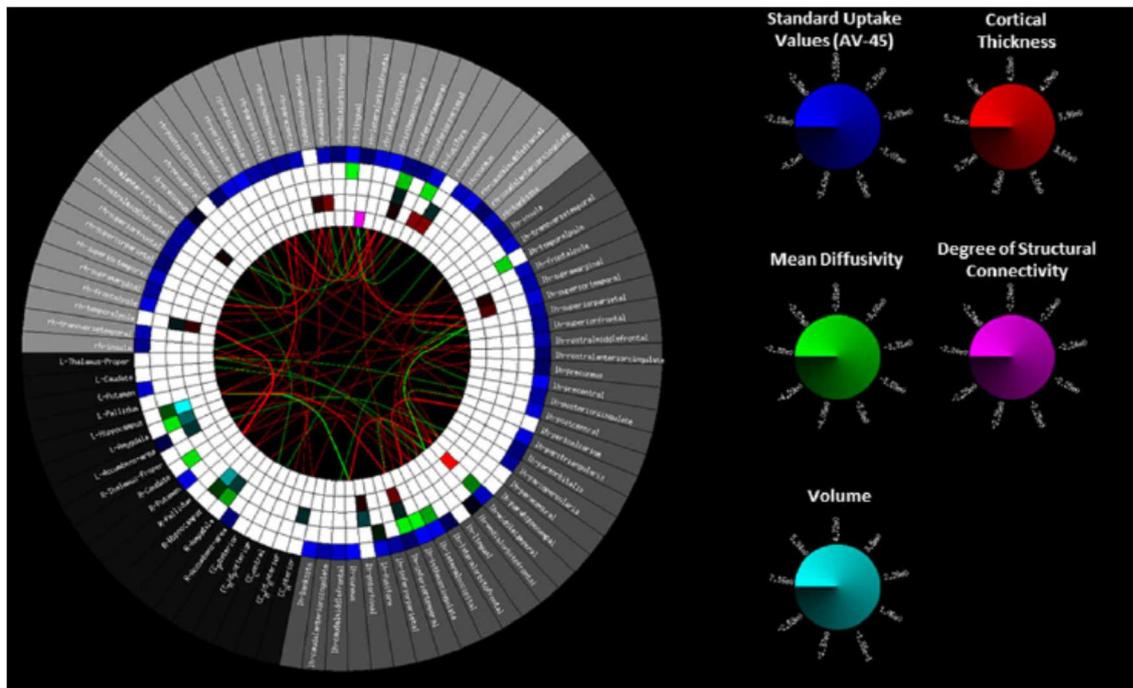


Figura 2.8 – Conectograma obtido pela MIBCA. Nos anéis exteriores para o interior temos as ROI's, Padrão de Captação de Valor, média de difusão, o volume da ROI, espessura cortical e o grau nó calculado a partir conectividade estrutural [28]

Assim sendo, na dissertação partimos com o objetivo de criar um *software* que permitisse visualizar imagens médicas em ambiente de realidade virtual controlado por gestos com acesso aos dados estatísticos e dados de conectividade. Esta abordagem ainda não é possível nos *softwares* de imagem médica existentes, levando-nos a entrar no campo da inovação, combinado a realidade virtual e controlo gestual à imagem médica. A aplicação desenvolvida será explicada no capítulo 5.

3

3 Sistemas de Realidade Virtual e Interfaces Pessoa Máquina

3.1 Realidade Virtual

A realidade virtual é definido pela criação de um mundo computacional tridimensional, no qual a pessoa está imersa. É uma simulação eletrónica de ambientes artificiais através de óculos de *head-mounted systems* e roupa conectada a computadores, que permite ao utilizador interagir de forma realista e imersiva em ambientes tridimensionais [30].

Os sistemas de realidade virtual têm ganho adeptos ao longo dos anos, tendo tido um maior desenvolvimento junto da indústria dos jogos. Estes estão constantemente a tentar fornecer uma experiência mais realista aos jogadores de forma a estes sentirem que fazem parte do jogo. Foi neste contexto que esta dissertação surgiu, fornecer um ambiente imersivo para visualização de dados de conectividade cerebral

3.2 Primeiros sistemas de realidade virtual

A realidade virtual surgiu com o objetivo de ir além das formas padronizadas, como o teclado do computador e o rato, de interação com o mundo virtual. Assim sendo, esta tecnologia está a evoluir num sentido de tornar essa interação mais natural como no mundo real.

O desenvolvimento da realidade virtual considera-se que começou na década de 50 do século XX. Morton Heilig criou em 1957 o Sensorama, um simulador multisensor que a partir de um filme previamente gravado em estéreo, fazia como que um “aumento” do mesmo, permitindo ao utilizador usufruir de uma experiência realista do filme sentindo na sua face o vento, a vibração, bem como os cheiros e ainda uma percepção espacial do som [31].



Figura 3.1 – Sensorama, sistema criado por Morton Heilig [32]

O primeiro *Head Mounted Display* (HMD) foi desenvolvido por engenheiros da empresa Philco Corporation. Este foi chamado de *Headsight* (ver figura 3.2) e foi desenvolvido com o objetivo de ser usado por pilotos de helicópteros que necessitassem de ver o ambiente que os rodeava à noite. Em 1965, Ivan Sutherland criou o *Ultimate Display* ou também chamado de *Sword of Damocles*, um HMD ligado a um computador que permitia visualizar um mundo virtual. No entanto, este sistema ainda se apresentava como sendo de grandes dimensões ocupando grande parte da sala e encontrando-se suspenso [31], [33]–[35].



Figura 3.2 – Na imagem à esquerda encontra-se o sistema *Headsight* desenvolvido pela Philco Corporation. [33] Na imagem à direita o sistema *Ultimate Display* [35]

Em 1980 foi quando a realidade virtual sofreu um grande avanço. Na altura Scott Fisher da NASA, lidera o grupo de investigação de realidade virtual *Ames Research Center* a desenvolver o seu próprio *hardware* e *software*. A equipa desenvolveu o seu *Virtual Interface Environment Workstation* e respetivas luvas de interação com o mundo virtual. Um dos elementos da equipa, Jaron Lanie, continuou a trabalhar na luva, desenvolvendo a famosa *DataGlove* com a empresa VPL. Esta empresa associou mais tarde esta luva ao seu primeiro HMD comercial, o *EyePhone* [31], [35]. Este sistema está ilustrado na figura 3.3.



Figura 3.3 – *EyePhone* em conjunto com *DataGlove*, o primeiro HMD comercial desenvolvido pela empresa VPL [33]

A realidade virtual continuou a ser popular nos anos 90, mas toda a propaganda em volta desta tecnologia teve um efeito negativo em si, levando a um decréscimo de popularidade. O público sentia que os sistemas de realidade virtual não correspondia às suas expectativas, começando assim a perder um pouco o interesse.

Nos últimos 20 anos o desenvolvimento dos HMD disparou, pois as empresas têm associado novas tecnologias a estes sistemas. Estes sistemas têm convergido para um uso móvel e com recurso a tecnologia sem fios. Assim sendo, têm-se associado a estes HMD reconhecimento de voz, reconhecimento gestual, seguimento do olhar (do inglês *eye tracking*) e interfaces cérebro computador (do inglês *Brain Computer Interfaces*, BCI) [33].

3.3 Head Mounted Display

Um HDM é um *headset* que normalmente incorpora dois ecrãs de cristais líquidos (do inglês *Liquid Crystal Display*, LCD) ou de tubos de raios catódicos (do inglês *Cathode Ray Tube*, CTR), cada um com óticas para permitir a visão estereoscópica. Estes sistemas têm um *tracker*, normalmente sensores inerciais, para que a imagem se possa mover de acordo com a posição atual da cabeça do utilizador, de forma a dar uma melhor ilusão de realidade aquando num ambiente virtual [36], [37]. Estes sistemas são característicos de serem bastante imersivos na sua experiência de utilização.

De momento no mercado encontramos vários sistemas HDM, cada um com as suas vantagens e desvantagens. Os HDM que de momento se destacam são os *Oculus Rift*, da empresa OculusVR [38] e os *Wireless Personal 3D Viewer*, da empresa Sony [39].

3.3.1 Oculus Rift

Os *Oculus Rift*, ilustrados na figura 3.4, surgiram de uma campanha *Kickstarter* lançada pela empresa Oculus VR. Em Agosto de 2012 foi anunciado que quem financiasse o projeto com \$300, receberia a primeira versão de desenvolvimento, sendo que foi aí que este dispositivo teve a sua expansão. Este *headset* foi desenhado para os jogos de vídeo, mantendo sempre em vista a sua usabilidade e conforto para o utilizador [40].



Figura 3.4 – Primeiro *Development Kit* fornecido. Na imagem da esquerda, é a vista geral dos *Oculus Rift*, a central a vista da interface, a da direita é a caixa de controlo [41]

Este primeiro *Development Kit* é composto por um ecrã LCD com resolução de 1280x800 pixéis, com 640x800 por cada olho. O sistema possui um sistema de *head tracking*

com sensores de orientação com nove graus de liberdade, onde inclui giroscópio, acelerómetro e magnetómetro. A figura 3.5 ilustra os componentes dos *Oculus Rift*.

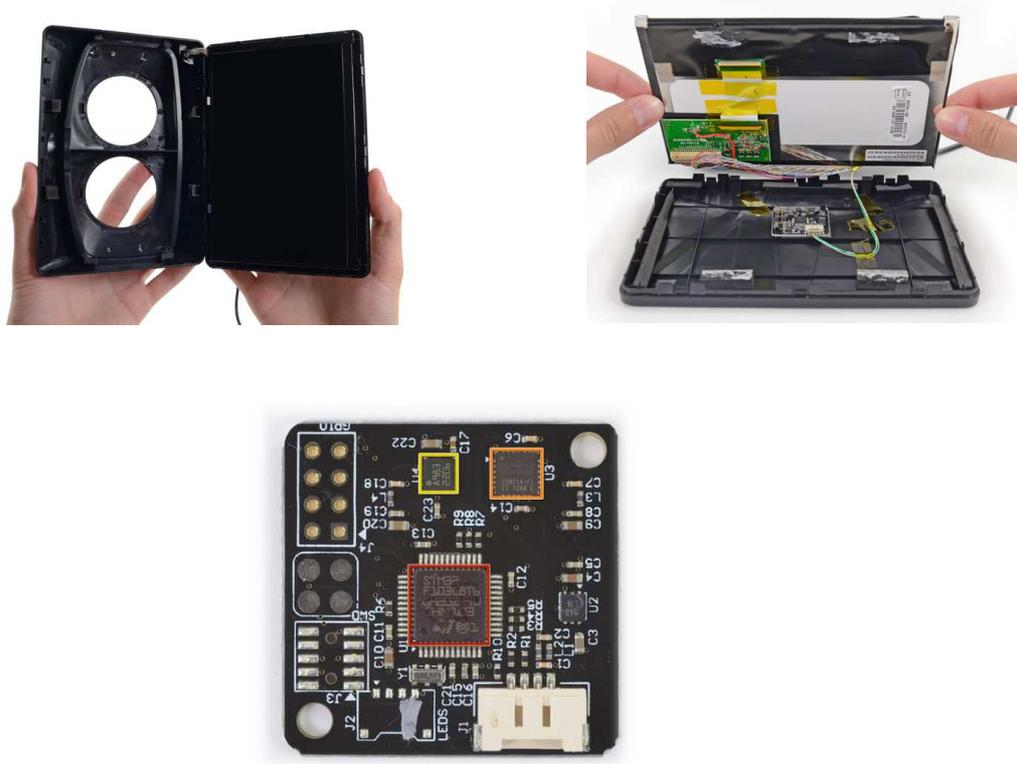


Figura 3.5 – Componentes dos *Oculus Rift*. A imagem em cima à esquerda é a vista do ecrã. A imagem em cima à direita ilustra a ligação entre o ecrã e o sistema Head Tracking. A imagem em baixo ilustra a *board Oculus Tracker V2*, responsável pelo sistema *Head Tracking*. A vermelho está indicado um microcontrolador STMicroelectronics 32F103C8 ARM Cortex-M3 com 72Mhz CPU. A amarelo está indicado um compasso digital tri-axial Honeywell HMC5983 para compensação do giroscópio. A laranja está indicado um controlador de posição Invensense MPU-6000 de seis eixos, em que associa giroscópio e acelerómetro. [41]

Passando para o campo de visão que o dispositivo permite, este apresenta-se com um campo de visão (FoV) com mais de 90 graus horizontais. O objetivo era preencher completamente o campo de visão do utilizador, fazendo como que um bloqueio do mundo real, fornecendo assim uma experiência muito mais imersiva no mundo virtual. A imagem para cada olho é vista no ecrã com distorção barril (ver imagem 3.6), distorção essa que é corrigida por efeito de almofada (ver imagem 3.6) com recurso a um conjunto de lentes, gerando assim uma imagem esférica [42].

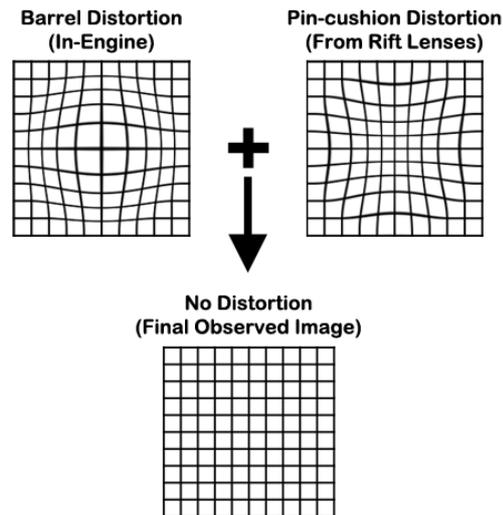


Figura 3.6 – Aplicação do efeito barril e efeito almofada à imagem [42]

Os *Oculus Rift* utilizam tecnologia de localização de movimento personalizada. Assim, é possível obter uma latência de *head tracking* bastante baixa em todos os 360 graus, permitindo explorar facilmente o mundo virtual que envolve o utilizador. Cada movimento da cabeça é processado em tempo real criando uma experiência intuitiva, natural e imersiva [38].

Este dispositivo pode ser usado em qualquer PC que tenha instalado o sistema operativo Windows ou Mac OS. Como forma de comunicação, os *Oculus Rift* estão interligados a uma caixa de controlo. Esta caixa é um dispositivo muito simples, contendo apenas 5 botões, um para ligar o sistema e os restantes para controlo de brilho e luminosidade.



Figura 3.7 – Caixa de controlo *Oculus Rift* com respetivas portas de conectividade [41]

Como é possível ver pela figura 3.7, a caixa de controlo possui uma porta HDMI, DVI, Mini-USB e DC-in. A entrada de imagem nos *Oculus Rift* é feita com DVI, no entanto com o uso da porta HDMI, a caixa de controlo converte o sinal da entrada HDMI em DVI. Para isso tira partido da sua *motherboard* [38], [41].

3.3.1.1 Desenvolvimento de *Software*

Para desenvolver aplicações que usem os *Oculus Rift* é necessário comprar o *Development Kit* disponível à venda no *website* da empresa por \$350, fazendo o download do *Software Development Kit* (SDK). O SDK está disponível para desenvolvimento em ambiente Windows ou Mac OS. A nível de linguagens de programação, o SDK está preparado para ser integrado em projetos desenvolvidos em C++. No entanto, existe também disponível um pacote de integração do SDK com uma *Game Engine* bastante comum no desenvolvimento de jogos, o *Unity*. Este pacote vem preparado para ser integrado em projetos desta plataforma tanto desenvolvidos em C# como em *Javascript*.

Nesta dissertação optou-se por se desenvolver a aplicação usando o *Unity*. Os motivos desta escolha serão explicados mais à frente.

3.3.2 *Wireless Personal 3D Viewer*

O *Wireless Personal 3D Viewer*, conhecido simplesmente por Sony HMZ-T3W, é o HMD desenvolvido pela empresa Sony, lançado em Novembro de 2014. Este está ilustrado na figura 3.8.



Figura 3.8– HMZ-T3W da Sony. Na imagem em cima à esquerda, é a vista geral do HMZ-T3W, na imagem em cima à direita a vista da interface do utilizador, na imagem em baixo a caixa de controlo do HMD [43]

Um dos pontos fortes deste sistema é o facto de ter três modos de ligar ao HMD, por via *wireless*, com fios e ainda via *Smartphone*, o que permite uma maior mobilidade do utilizador e opção de escolha. Está equipado com um ecrã de tecnologia *Organic Light-Emitting Diode* (OLED) de 1220x720 *pixels*, com um FoV de 45°, com capacidade para imagem em alta definição e em 3D [43], [44].

Segundo a Sony usar o HMZ-T3W é como ter uma sala de cinema multiplex com som *surround 7.1* e um ecrã virtual de 750” visível a 20 metros de distância. Os dois ecrãs OLED criam a ilusão de uma única grande projeção [39]. Um dos pontos fracos deste sistema é que não possui *head tracking*, ou seja o utilizador não consegue controlar o cenário com um simples movimento de cabeça.

3.3.2.1 Desenvolvimento de *Software*

O HMZ-T3W pode ser comprado no *website* da Sony, estando disponível na Europa por 1299,00€, sendo dos sistemas mais caros no mercado. Este preço é justificado pela tecnologia de visualização usada e pela tecnologia sem fios, o que é inovador na área de realidade virtual. O desenvolvimento de aplicações para este HDM não foi considerada, não havendo qualquer tipo de SDK de desenvolvimento, o que torna este HDM menos apelativo para a indústria de desenvolvimento de aplicações.

3.4 Aplicações da Realidade Virtual

O tema da realidade virtual já faz parte do nosso quotidiano, em que muita gente sabe o que é, mas no entanto não sabe ao certo a sua utilidade. Desde muito cedo que a realidade virtual está associada aos jogos de vídeo. No entanto, as suas aplicações vão muito mais além dessa finalidade. As principais áreas onde estes sistemas se destacam é a nível de aplicações militares, na medicina e investigação. Dado o carácter da dissertação, em seguida irá ser feita uma descrição das aplicações em medicina.

3.4.1 Realidade Virtual em Medicina

A análise de imagens médicas em 3D é um campo fascinante, onde é visto como uma forma de melhorar tanto o diagnóstico como a terapia [45]. Atualmente existem quatro áreas na medicina onde a realidade virtual tem tido um papel emergente: na área de assistência à preparação e realização de cirurgias; na educação médica e treino clínico; visualização de dados de imagem médica e reabilitação [46].

Na área de preparação e realização de cirurgias, a realidade virtual surge como uma forma de otimizar as intervenções médicas. Muitas áreas como a cirurgia estereotáxica, o tratamento não é possível sem uma preparação prévia com recurso a computador: é feito um planeamento médico onde são testadas diferentes metodologias de aproximação ao problema, com previsão do possível resultado final, como por exemplo a estética final do pa-

ciente [47]. Outro exemplo do uso de realidade virtual na cirurgia é um trabalho desenvolvido por Chun-Yi Lo et al [48], que desenvolveram um ambiente de realidade virtual para visualizar massas tumorais em integração com visualização de dados de imagem de tensor de difusão (do inglês *Diffusion Tensor Imaging*, DTI) e anatomia peritumoral. Usando o controlo 3D da aplicação é possível usando este trabalho fazer uma escolha do melhor caminho cirúrgico a seguir. A equipa para este projeto usa um projetor adaptado em conjunto com óculos polarizados como na figura 3.9. O controlo da visão é todo feito em computador [48].



Figura 3.9 – Sistema de realidade virtual de visualização de tractografia [48]

Na cirurgia assistida por robótica, a realidade virtual também se destaca. Neste caso a cirurgia é realizada por um dispositivo robótico controlado pelo médico-cirurgião, como na figura 3.10. Com recurso a sistemas de realidade virtual, o cirurgião consegue controlar os movimentos robóticos, em particular movimentos pequenos e delicados que seriam difíceis de realizar pelo próprio cirurgião.



Figura 3.10 – Robot para cirurgia de laparoscopia. Requer pequenas incisões entre 0.5 a 1.5 cm de comprimento no paciente, onde pequenas câmaras e instrumentos cirúrgicos são inseridos [49]

A educação médica e treino clínico são das áreas mais promissoras em realidade virtual na área médica. Os sistemas de realidade virtual oferecem uma oportunidade única para o treino cirúrgico, onde os profissionais põem em prática de forma repetida inúmeros procedimentos clínicos de forma a minimizar o erro.

Uma empresa bastante conceituada nesta área é a *Simbionix*. Esta empresa cria variados sistemas de simulação clínica de cirurgia e avaliação médica. Para mais informação pode-se consultar a referência [50].

Este tipo de tecnologia médica traz vantagens para o clínico e futuros doentes tais como:

- Mais eficiente que praticar com doentes reais, eliminando o risco de lesões acidentais por parte dos clínicos menos experientes [51];
- Os sistemas de realidade virtual estão providos de formas para avaliar e medir a competência técnica, bem como sua evolução, dando incentivo ao clínico para aprimorar as suas competências [51];
- Estes sistemas trazem vantagens éticas, uma vez que permite fazer uma prática exaustiva de procedimentos médicos, levando a que quando haja a transição para humanos o clínico tenha já mais confiança e segurança [51].

Na reabilitação clínica, a realidade virtual tem sido desenvolvida para auxiliar na reabilitação de doentes com lesões cerebrais. Está descrito na literatura que um doente com lesões cerebrais tem dificuldade em interagir com o ambiente que o rodeia, desde o ambiente familiar a situações normais rotineiras, levando a uma maior dificuldade de reabilitação. No entanto, estudos sugerem que este efeito pode ser revertido a partir de um “enriquecimento ambiental”, levando a um melhoramento de funções de memória e concentração [52], [53]. A realidade virtual oferece a possibilidade de desenvolver meios específicos e adequados para interação com o ambiente, consoante as necessidades do doente. Acima de tudo, a intimidade do doente é respeitada, visto que com o uso de esta tecnologia, a interação ambiental é feita diretamente com o doente com recursos a HMD ou ecrãs, diminuindo assim a necessidade de grandes equipas de reabilitação que muitas vezes deixa o doente desconfortável [53].

A realidade virtual também é usada para o tratamento de doenças psiquiátricas como fobias, tais como a aracnofobia (fobia de aranhas) [54], claustrofobia (fobia de espaços fechados) [55], acrofobia (fobia a alturas) [56], [57] e fobia social [58]. Com os sistemas de realidade virtual os médicos colocam o doente em contacto faseado com a sua fobia de modo a que estes a consigam superar. Esta técnica apresenta bons resultados, havendo já clínicas especializadas na área. Exemplo disso é o *Virtual Reality Medical Center* em São Diego que faz uso da realidade virtual para ajudar os seus pacientes nas mais variadas fobias [59].

3.5 Interfaces Pessoa Máquina

Os sistemas realidade virtual idealmente integram na sua utilização métodos de interação com meio virtual que vão muito mais para além do que o convencional rato e teclado. As interfaces utilizadas não podem interferir com a imersão do utilizador no meio virtual, mas ao mesmo tempo têm que ser simples e intuitivas de usar. Da pesquisa realizada, encontrou-se que de momento existem dois sistemas que têm sido integrados com os sistemas realidade virtual. Um deles é o *Leap Motion*, da empresa Leap Motion, que se trata de um dispositivo capaz de reconhecer gestos [60]. O segundo dispositivo trata-se do *Virtuix Omni Natural Motion Interface* [61].

Em seguida é feita uma descrição dos dois produtos, onde se apresentará as suas especificações bem como os seus pontos fortes e fracos.

3.5.1 Leap Motion

O Leap Motion é um controlador desenvolvido pela empresa do mesmo nome. Começou a ser comercializado em 2013, podendo ser adquirido no *website* da empresa por um custo de 89.99€. O Leap Motion foi desenvolvido inicialmente na vertente dos jogos de vídeo, estando agora a sofrer uma expansão na sua aplicabilidade. A ligação do dispositivo é feita via USB, tendo *drivers* disponíveis para sistema operativo Windows, Mac OS e Linux.

Este dispositivo faz o reconhecimento de mãos, dedos e objetos que tenham a forma de um cilindro. Tecnicamente são conhecidos poucos detalhes sobre a construção do dispositivo [62]. No entanto, fazendo uma observação do dispositivo desmontado, constata-se que este tem na sua composição duas câmaras de tecnologia *complementary metal-oxide-semiconductor* (CMOS) em associação com três emissores de infravermelho [63]. Tendo isto por base, verifica-se que se trata de um dispositivo de *optical tracking*, baseado num princípio de visão estereoscópica. O aparelho cria um campo de visão baseado em coordenadas cartesianas com a forma de pirâmide invertida, com um campo de visão entre 2,5 cm e 60 cm acima do dispositivo, como ilustrado na figura 3.11 [63], [64].

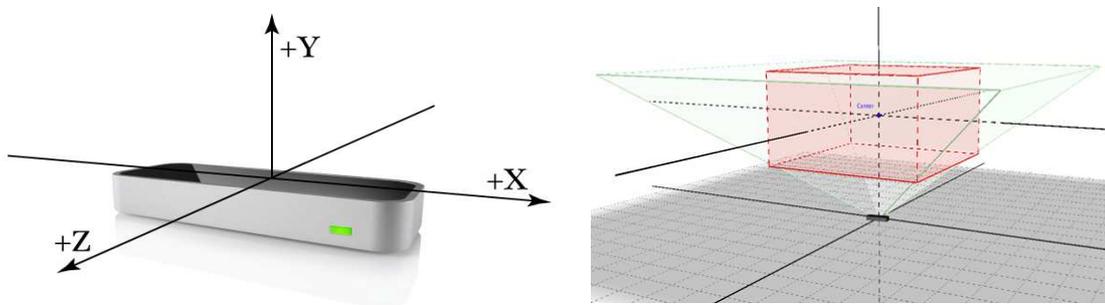


Figura 3.11 – Sistemas de coordenadas do Leap Motion e campo de visão [64], [65]

3.5.1.1 Desenvolvimento de aplicações

Para desenvolver aplicações com recurso ao Leap Motion é necessário o *hardware* da empresa e *software* da mesma. Outra forma de ter acesso ao Leap Motion é a partir da compra do computador HP Envia 17 Leap Motion SE Notebook ou do teclado HP Leap Motion. Isto resultou de uma parceria entre a empresa Leap Motion e a HP, em que o dispositivo iria ser integrado em PC's como se fosse um rato ou um teclado.

No *website* da empresa encontra-se disponível de forma gratuita o SDK do Leap Motion. Este SDK pode ser integrado em projetos desenvolvidos em *JavaScript*, *C++*, *Java*, *Python* e *Objective-C*. Tendo em conta que se trata de um dispositivo inicialmente

criado para jogos, existe também uma versão de integração para *Unity*, a *Game Engine* anteriormente referida.

Na presente dissertação recorreu-se à versão de integração em *Unity* de forma a fazer uma articulação com a parte do projeto desenvolvida para os *Oculus Rift* e o *Leap Motion*.

3.5.1.2 Funcionamento do dispositivo

O Leap Motion é um dispositivo que consegue fazer localização espacial dos dedos, ou objetos semelhantes a dedos, e da palma da mão, como se verifica nas figuras 3.12 e 3.13. Quando o dispositivo deteta uma mão, dedo ou objeto é atribuída uma identificação, fornecendo *update* da informação a cada *frame*, dando para cada *frame* a informação referente à posição, tempo em *frame*, velocidade e ângulo de cada um dos dedos e palma da mão. Em cada *frame* é possível recolher informação referente até duas mãos, sendo aconselhando não ultrapassar mais do que esse número [19], [64].



Figura 3.12 - Representação das métricas obtidas pelo Leap Motion na ponta dos dedos [64]

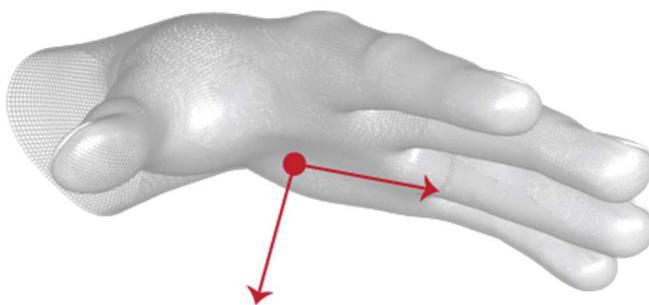


Figura 3.13 – Representação das métricas obtidas pelo Leap Motion na palma da mão [64]

3.5.1.3 Desenvolvimento do Leap Motion

O sistema Leap Motion dado o seu carácter inovador no mercado encontra-se em constante atualização. De momento existem duas versões de SDK. Neste trabalho foram consideradas a primeira versão dos SDK's dado que a segunda versão, onde foi incluído *Skeletal Tracking* e acesso ao *Raw Data*, apenas foi dado ao público em Agosto, quase no final do desenvolvimento do presente trabalho não havendo muito tempo para explorar o novo SDK.

Recentemente foi lançada também integração do dispositivo com sistemas de realidade virtual, como o da figura 3.14, podendo ser montado em conjunto com os *Oculus Rift* descritos anteriormente [66].

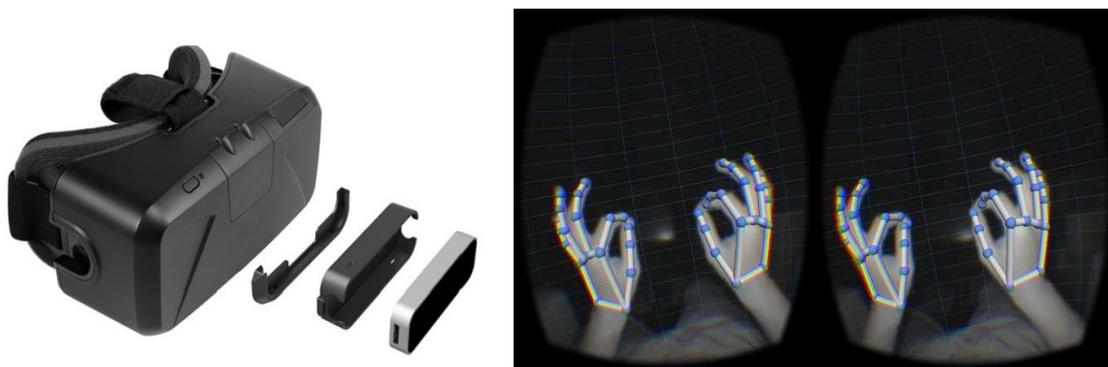


Figura 3.14 – Integração de *Oculus Rift* e Leap Motion lançada no final de Agosto 2014 [66]

Esta integração irá permitir um melhoramento significativo no uso do dispositivo em aplicações de realidade virtual, expandindo assim o espaço de rastreamento das mãos para qualquer parte que o utilizador esteja a olhar.

3.5.2 *Virtuix Omni Natural Motion Interface*

O *Virtuix Omni* é a primeira interface de realidade virtual que permite ao utilizador movimentar-se no cenário virtual de uma forma natural. Com esta interface é dada a oportunidade ao utilizador de sentir-se como nunca antes num ambiente totalmente imersivo, navegando pelo mesmo como se fosse andar ou a correr na vida real. Este projeto começou

a partir de uma campanha *Kickstarter* com mais de 3000 contribuições angariando mais de um milhão de dólares [61].



Figura 3.15 – Utilização do sistema *Virtuix Omni*, juntamente com *Oculus Rift* [67]

A *Omni* envolve todo o corpo numa experiência virtual como nenhum outro sistema. Como se vê na figura 3.15, o sistema *Omni* utiliza uma plataforma côncava que permite, uma marcha suave e natural, o que origina como que um “passeio” imersivo na realidade virtual. Junto com o sistema, vem incluído um par de sapatos, como dos da figura 3.16, com solas de baixo atrito de forma a facilitar o movimento e a sua deteção pelo sistema [61], [67].



Figura 3.16 – Sapatos do *Virtuix Omni* [61]

Segundo o fabricante, a *Omni* é capaz de adquirir movimentos como andar, correr, saltar e até mesmo sentar, tudo isso em 360 graus. De momento, a *Omni* só funciona em PC, com o respetivo SDK, mas é esperado estar brevemente disponível para outras plataformas [68]. Em conjunto com o *Omni*, o utilizador necessita de um HMD, como os *Oculus Rift*, para visualizar o meio virtual [61].

4

4 Desenvolvimento de aplicações 3D

O desenvolvimento de aplicações em 3D é um dos grandes desafios da indústria de *software*. Este processo é bastante demorado e requer *hardware* e *software* de ponta. A única forma de obter um meio realista e bastante similar ao real é com recurso a tecnologia que muitas vezes está fora do orçamento tanto de empresas como de particulares interessados no desenvolvimento de aplicações de realidade virtual. No início do século XXI, os mecanismos de alta performance gráfica e de realidade virtual apenas estavam disponíveis aos grandes laboratórios e para o exército que os usava no treino de soldados. No entanto, a expansão do mercado dos jogos levou a uma expansão do *hardware* dos computadores pessoais, levando a uma melhoria bastante significativa da performance [69], [70].

4.1 *Game Engines*

O desenvolvimento de aplicações realistas levou à necessidade da criação de ferramentas auxiliares que fossem independentes do *hardware* da máquina, mas que ao mesmo tempo oferecessem uma boa performance. Assim sendo, nesta perspetiva surgiram as *Game Engines*. Este tipo de plataformas são *frameworks* de *software* que incluem tipicamente ferramentas de código modular, como por exemplo *renderer* gráfico 2D e 3D, sistemas de simulação de eventos físicos, deteção de colisões, *scripting* de animação, som, inteligência

artificial, sistemas de gestão de memória, entre outras funcionalidades, que vieram agilizar a produção de jogos em 3D bem como de aplicações para realidade virtual [70], [71].

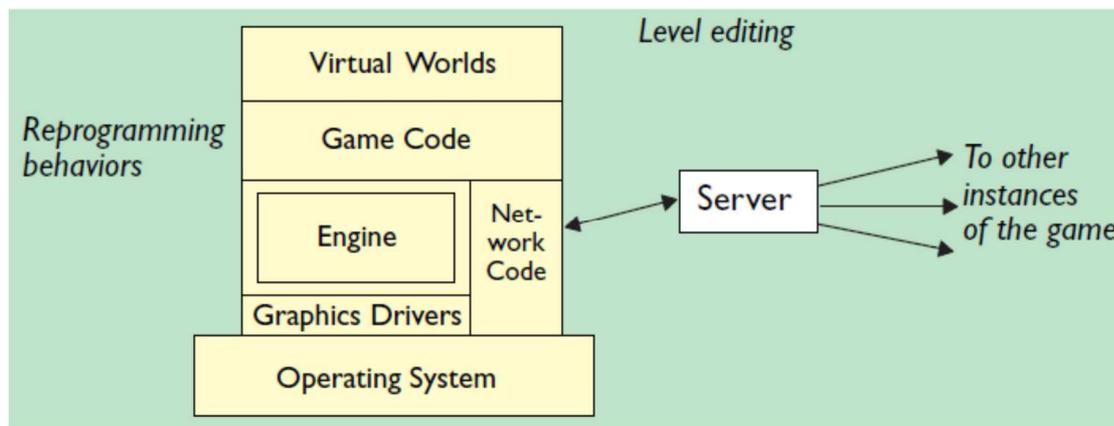


Figura 4.1 - Estrutura Modular de uma *Game Engine* (ver explicação no texto) [70]

O desenvolvimento de jogos e aplicações 3D/VR pode ser organizada em níveis como se pode ver na figura 4.1. No primeiro nível estão os cenários/mundos virtuais nos quais o utilizador interage. Estes são produzidos normalmente pelos *designers* com ferramentas de modelação 3D como o AutoCAD e 3ds Max por exemplo. No nível abaixo encontra-se o código de jogo, no qual estão programados os mecanismos de comportamento desde os mais básicos aos mais complexos. Em seguida, segue-se a parte mais importante da aplicação, o mecanismo de renderização de imagem. Neste grupo está incorporada toda a lógica de visualização em 3D. Esta parte maioritariamente é vista como uma caixa negra, uma vez que estes mecanismos são proprietários de empresas, não estando em *Open Source* para o usuário alterar. Associada a esta parte, estão associados os drivers gráficos, que são responsáveis por fazer a tradução das instruções vindas dos mecanismos de renderização. Esta parte é feita maioritariamente por bibliotecas de auxílio, sendo as mais comuns o OpenGL e o DirectX. Visto serem *Open Source*, estas bibliotecas podem ser facilmente integradas em sistemas de realidade virtual como os HMD [70].

Tendo em conta as várias características de uma *Game Engine*, estas podem ser usadas muitas vezes no âmbito da investigação científica, nomeadamente na área de visualização computacional, interfaces pessoa máquina, simulação de eventos e realidade virtual. Na literatura é possível encontrar já projetos de realidade virtual que se servem de *Game Engines* para serem bem-sucedidos. Exemplo disso é o sistema CaveUT da figura

4.2, um projeto *Open Source* criado pelo Departamento de Ciências da Informação da Universidade de Pittsburgh [72], que utiliza a tecnologia de *Game Engines* para criar um sistema de realidade virtual baseado em projeções imersivas de uma forma barata e acessível [73].

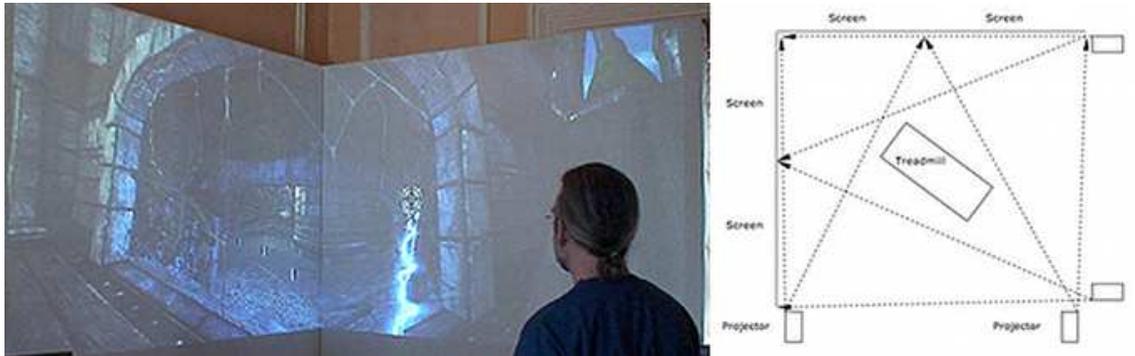


Figura 4.2 – Sistema CaveUT criado com uma *Game Engine* [72]

Existem várias *Game Engines* disponíveis no mercado, variando tanto a nível de preços, como de performance e módulos incluídos. Duas das mais usadas são o *Unity* e o *Unreal Engine*. As duas são *Game Engines* poderosas, sendo que se destacam em áreas diferentes. Em seguida irá fazer-se uma breve descrição das duas *Game Engines* e a comparação entre elas.

4.1.1 *Unity* 3D

O *Unity* 3D é uma multiplataforma desenvolvida pela empresa *Unity Technologies* para auxílio na criação de sistemas de jogo. Esta aplicação foi criada em 2005 sendo que desde então tem vindo a crescer e atualmente é bastante popular indo já na sua versão 5. Este *software* pode ser executado em computadores com Windows OS e Mac OS.

O *Unity* inclui em si uma *Game Engine* e uma interface de desenvolvimento, o *MonoDevelop*. Esta plataforma permite desenvolver jogos e aplicações para as mais variadas plataformas, como por exemplo plataformas Desktop, consolas de jogos como a *Play Station3* e plataformas móveis como *Android*. Ao todo é suportada a implementação para mais de 17 plataformas diferentes [74].

O *Unity* já é usado há vários anos, contando com inúmeros jogos produzidos, muitos deles com vários prêmios. A lista de jogos feita pode ser encontrada no *website* <http://Unity3d.com/gallery/made-with-Unity/>.

Esta plataforma foi desenvolvida de uma forma simples mas ao mesmo tempo eficaz de modo a que tanto utilizadores experientes como os mais novos consigam tirar o máximo partido desta *Game Engine*, podendo dar largas à sua criatividade não estando limitados à programação. Esta aplicação permite criar todo o tipo de jogos desde *First-Person Shooter* (FPS) a jogos plataforma passando inclusive por jogos *RolePlay Online* [75].

Os *scripts* em *Unity* utilizam linguagens de programação de alto nível. As linguagens que a aplicação suporta são o *C#*, *JavaScript* e *Boo*. A mais complexa é o *C#* para a qual é aconselhada o utilizador ter algum conhecimento de programação orientada objetos. As restantes são as mais aconselháveis para programadores menos experientes.

4.1.1.1 Interface de Utilização

A interface do *Unity* foi desenvolvida de modo a ser agradável à vista do utilizador, sendo fácil de utilizar. Todavia, não foi deixado de parte que esta tinha que conter uma enorme panóplia de ferramentas de modo a tornar a programação de uma aplicação mais eficiente.

Ao iniciar a aplicação é apresentado ao utilizador o seguinte ecrã:



Figura 4.3 – Interface da Game Engine *Unity* [75]

Em seguida faz-se a descrição da interface apresentada:

- Com o número 1 temos a vista intitulada de *Scene*. Nesta vista o utilizador pode visualizar e modificar a cena de jogo que se encontra a criar. É de referir que nesta vista apenas se consegue alterar propriedades físicas dos objetos como escala e posição [71], [75], [76];
- Com o número 2 temos a secção intitulada de *Hierarchy*, onde o utilizador tem a lista de todos os objetos criados na *Scene* e qual a sua organização e dependência entre si [71], [75], [76];
- Com o número 3 temos o *Project*, uma secção que permite uma vista sobre a pasta de jogo, onde estão presentes desde novos objetos a scripts para adicionar à cena [71], [75], [76];
- Com o número 4 temos o *Inspector*, onde é possível ver as propriedades dos objetos selecionados e adicionar-lhes scripts que definam as suas regras de comportamento quando a aplicação está a ser executada [71], [75], [76].

Esta interface pode ser customizada a gosto do utilizador, podendo assim adaptá-la às suas necessidades. Quando se cria um novo projeto é dado ao utilizador a opção de carregar para o projeto recursos pré-criados que o auxiliarão ao longo da criação do seu projeto. À semelhança de muitas aplicações, esta possui uma loja *online* onde é possível adquirir de forma gratuita ou pagar recursos extra para utilização pessoal em projetos.

Para utilizadores que se iniciem em *Unity*, este tem no seu *website* uma secção completamente dedicada à aprendizagem. Lá é possível encontrar tutorias, documentação, aprendizagem ao vivo e recursos básicos para utilização do *software*. Isto é uma prova que a companhia tem interesse em tornar este produto acessível a todos e oferecer uma melhor experiência de utilização.

Com isto, o *Unity* torna-se uma ferramenta poderosa para quem pretenda desenvolver aplicações tanto para jogos como quem pretenda criar ambientes de realidade virtual.

4.1.2 *Unreal Engine*

O *Unreal Engine* é um *software* desenvolvido pela empresa Epic Games. Este *software* surgiu em 1998 para a criação de um jogo FPS intitulado Unreal, sendo o seu uso difundido desde essa altura, indo já na versão 4. Este *software* pode ser executado em computadores com Windows OS e Mac OS.

Este *software* já foi vencedor de vários prémios entre as *Game Engines* existentes, destacando-se nas áreas de renderização 3D, desenvolvimento de jogos para plataformas moveis, filmes digitais entre outras [77]. Semelhante à *Game Engine* anterior, esta traz consigo muitos sistemas integrados, como sistemas de renderização, sistemas de inteligência artificial, ferramentas de cinemática entre outros, fazendo deste *software* uma *Game Engine* bastante poderosa para desenvolvimento de jogos e ambientes de realidade virtual.

Esta plataforma permite desenvolver jogos e aplicações para as mais variadas plataformas, como por exemplo plataformas Desktop, consolas de jogos como a Play Station3 e plataformas móveis como Android. Ao todo é suportada a implementação em mais de 10 plataformas diferentes [78].

A plataforma está toda desenvolvida em C++, sendo que para criação de aplicações estas têm que ser programadas neste tipo de linguagem. Tal facto pode ser visto como um entrave a programadores inexperientes uma vez que esta linguagem já requer um certo nível de conhecimento em programação.

4.1.2.1 Interface de Utilização

Ao iniciar a aplicação, o utilizador depara-se com uma interface organizada, amigável de utilizadores inexperientes. Esta oferece um tutorial de aprendizagem na primeira utilização que é ideal para ficar a conhecer as metodologias do *software*.



Figura 4.4 – Interface Editor do *Unreal Engine*. Adaptado de [79]

Em seguida faz-se a descrição da interface apresentada:

- Com o número 1 temos os *Models*. Nesta área estão geometrias e modelos pré-definidos de objetos para o utilizador juntar ao seu cenário criado.
- Com o número 2 é permitido ao utilizador escolher o tipo de projeto que está a realizar, desde a criação de um nível de um jogo à edição de imagem
- Com o número 3 temos a secção *Details*. Aqui é permitido ao utilizador ver os detalhes associados aos objetos em cenário, desde posições à iluminação no cenário. Estes detalhes podem ser editados nesta área.
- Com o número 4 temos o *Content Browser*. Nesta área é permitida uma vista sobre a pasta de jogo, onde estão presentes desde novos objetos a scripts para adicionar à cena;
- Com o número 5 temos o *ViewPort*. Nesta secção o utilizador pode observar o cenário, ou nível como é chamado na documentação, que se encontra a criar. Aqui pode variar as posições das câmaras e objetos em cenário.

É de destacar que para novos utilizadores, no *website* é possível encontrar todos os tipos de tutoriais para se iniciar no projeto pretendido. No entanto uma desvantagem é que não possui uma loja *online* de recursos adicionais a usar para desenvolver ideias. Isto pode ser visto como uma desvantagem em que o utilizador tem que fazer tudo de raiz.

4.2 *Unity* 3D vs. *Unreal Engine*

Colocando lado a lado estas duas *Game Engines* verifica-se que ambas têm os seus pontos fortes e pontos fracos. Fazendo uma comparação entre ambas obtemos a tabela 4.1:

<i>Game Engine</i>	<i>Unity</i>	<i>Unreal Engine</i>
Preço	Gratuito na versão Trial, com \$75 na versão Pro	\$19 por mês + 5% das vendas do produto desenvolvido
Plataformas para desenvolvimento		
Windows	Sim	Sim
Mac	Sim	Sim
Linux	Sim	Não
Android	Sim	Sim
IOS	Sim	Sim
Web	Necessita de <i>plugin</i>	Não
Características		
Desenvolvimento corporativo	Não	Não
Mundos ilimitados	Não	Não
Interface C++	Não	Sim
Editor de GUI	Não	Não
Loja Online Integrada	Sim	Não

Tabela 4.1 – Comparação entre o *Unity* e o *Unreal Engine*. Adaptado de [80]

Analisando o quadro verifica-se que ambas as *Game Engines* são bastante semelhantes. No entanto optou-se pelo *Unity*. Esta escolha assentou nos recursos existentes, nomeadamente os *Oculus Rift* e *Leap Motion*, em que existia já suporte para esta *Game Engine*. Outro motivo pelo qual o *Unity* se destaca como sendo preferível para a aplicação a desen-

volver deve-se ao fato de já existirem scripts que ajudam a conseguir o tipo de visão necessária para um HMD, ao contrário do *Unreal Engine*. No capítulo seguinte será feita uma análise das vantagens da utilização do *Unity* na aplicação.

Na área da investigação já existem aplicações que usam o *Unity*, sendo uma prova que as *Game Engines* são muito mais que plataformas para desenvolver jogos. Um exemplo dessas aplicações é o *Glass Brain*, uma aplicação desenvolvida no Sandler Neurosciences Center da Universidade da Califórnia. A aplicação é um visualizador de atividade e conectividade cerebral desenvolvida em *Unity*. O sistema é feito em tempo real com uso a eléctrodos de alta densidade de eletroencefalografia (EEG). Cada cor representa uma fonte de sinal em diferentes frequências (ondas alfa, beta, gama e *theta*). Na base da aplicação estão imagens de MRI para gerar imagens de alta resolução em 3D do cérebro de cada indivíduo, e imagens de DTI para fazer a reconstrução dos tratos de substância branca [81]. Na figura 4.5 está representado a utilização do sistema com um indivíduo de estudo:



Figura 4.5 – *Glass Brain* desenvolvido em *Unity* no Sandler Neurosciences Center da Universidade da Califórnia [81]

Este projeto serviu de motivação para continuar com o desenvolvimento da dissertação em *Unity*. No capítulo seguinte será feita uma análise das vantagens da utilização do *Unity* na dissertação.

5

5 Metodologia de trabalho

O presente trabalho tinha como objetivo o desenvolvimento de uma interface inovadora de visualização de conectividade cerebral obtida a partir de MR, com controlo gestual. Assim, a ideia de ter um sistema de controlo tipo “*Minority Report*” é uma realidade cada vez mais próxima, sendo uma das motivações para o desenvolvimento de trabalhos nesta área.

A visualização tridimensional do cérebro é vista como uma forma mais realista de analisar dados obtidos em exames médicos. Esta forma de visualização revela-se como meio complementar de análise para os médicos mais experientes, mas no entanto assume um papel relevante e inovador no ensino dos novos médicos, ajudando-os assim na sua formação.

Este tipo de aplicação terá grande relevância na área de investigação científica, conduzindo a uma melhor compreensão do cérebro, tanto a nível estrutural como funcional.

Para o desenvolvimento do protótipo da aplicação, o problema foi partido em partes, passando por várias fases ao longo do seu desenvolvimento. A primeira fase de desenvolvimento passou por uma escolha das plataformas a usar e respetivas ferramentas de desenvolvimento. Como tal, visto ter como principal objetivo a aplicação ser direcionada para a realidade virtual, usaram-se os *Oculus Rift*. Para a parte de controlo gestual, fez-se uso do controlador Leap Motion.

5.1 Processamento das Imagens

Visto que se pretendia desenvolver uma interface de visualização de dados de imagens cerebrais, antes de partir para a implementação da aplicação foi necessário fazer um processamento de imagens. Este passo é necessário para obter dados passíveis de serem integrados na aplicação e estabelecer metas e metodologias de implementação. Seguidamente irá se fazer uma explicação de como os dados da aplicação foram processados para serem integrados na aplicação desenvolvida.

5.1.1 Preparação das imagens estruturais

As imagens utilizadas para o desenvolvimento da interface provieram do serviço de imagiologia do Hospital Santa Maria. Estas imagens encontravam-se em formato DICOM, estando ponderadas em T1. Foi então feita uma conversão das mesmas para NIFTI, visto ser mais fácil trabalhar com este tipo de imagem com os meios disponíveis.

De forma a ter uma imagem anatómica do cérebro do indivíduo, teve-se de realizar um processamento das imagens. Este processamento foi feito com auxílio do *software* Freesurfer [2], onde se fez a introdução dos dados. Cada processamento levou em média 12 horas, sendo que neste processamento foi feita uma segmentação do cérebro em substância cinzenta, branca e líquido cefalorraquidiano, tendo por base dois mapas anatómicos, com uma divisão entre as regiões corticais e as regiões subcorticais. Posteriormente realizou-se a parcelização do cérebro. A nível das regiões corticais, esta parcelização foi feita com base no atlas anatómicos desenvolvidos por Destrieux et al, e no atlas anatómico desenvolvido por Desikan-Killiany-Tourville [82]. Ambos os mapas dividem o cérebro no hemisfério esquerdo e direito, sendo que cada hemisfério é depois parcelizado. O atlas Destrieux faz uma divisão de cada hemisfério em 74 áreas distintas [83], enquanto o Desikan-Killiany-Tourville faz uma divisão em 34 áreas distintas [84], [85]. Para pormenor sobre as áreas de cada mapa e nome de regiões pode consultar as referências e o Anexo I. A região subcortical, está fica dividida em 35 áreas distintas.

Com isto, obteve-se um ficheiro para cada hemisfério e atlas usado, onde estava contida a parcelização cerebral, e os respetivos ficheiros com as informações anatómicas relevantes.

As imagens obtidas no entanto apenas podiam ser lidas por *softwares* de imagem médica, encontrando-se limitado o seu manuseamento e respetiva identificação singular de cada região anatómica. Para resolver este problema, recorreu-se a técnicas de produção de cenários para jogos e *softwares* de modelação 3D.

Na pesquisa realizada encontrou-se um projeto chamado Brainerd criado por A. M. Winkler [86], consistindo num *website* onde o autor faz partilhas de pesquisas sobre imagem cerebral. Um dos trabalhos consistiu em converter imagens obtidas dos atlas do Freesurfer num tipo de ficheiro que *softwares* de modelação conseguissem fazer leitura. O formato escolhido foi o Wavefront .obj. Este formato é um ficheiro com definições de geometria do objeto, ideal para modelação 3D e aplicações de realidade virtual.

Assim sendo, aplicou-se o protocolo de conversão de segmentações obtidas em Free-surfer para um formato de modelação 3D. Este protocolo usou ferramentas de um pacote de processamento de dados de imagem cerebral a nível de áreas desenvolvido pela Universidade de Oxford. Este pacote pode ser encontrado em [87].

De forma a tornar este processo de conversão autónomo e mais rápido e de forma a poder ser usado para todos os dados dos atlas do Freesurfer, foi realizado um *script* em Matlab [23] responsável por converter os dados para Wavefront. A ideia que se seguiu foi de separar a superfície cortical em regiões individuais de forma a permitir que cada uma fosse visualizada individualmente no futuro. Assim sendo, aplicou-se um algoritmo que separou as diferentes parcelizações em ficheiros individuais, sendo-lhes aplicado posteriormente outro algoritmo para conversão para Wavefront.

No final do procedimento, tínhamos para cada parcelização efetuada sobre cada atlas anatómico, um ficheiro diferente de modelação 3D. De forma a verificar a fidelidade do processo aplicado, os ficheiros convertidos foram visualizados com recurso ao *software* de modelação gráfica Blender [86], [88].

Este processo pode ser usado para qualquer tipo de parcelização efetuada pelo Freesurfer. Isto repetiu-se para os dados de imagens subcorticais.

5.1.2 Preparação dos dados de conectividade

As métricas e matrizes de conectividade foram obtidas a partir de dados de tensor de difusão. Os dados de conectividade estrutural foram cedidos pelo professor Hugo Ferreira [13], podendo ser obtidos com recurso ao *software* MIBCA [28]. As matrizes de conectividade foram binarizadas para fins de ilustração do grafo cerebral a visualizar por realidade virtual.

5.2 Desenvolvimento da aplicação

Ao longo da pesquisa de formas de desenvolver uma aplicação 3D, convergiu-se bastante para as plataformas *gaming* e ferramentas de desenvolvimento de jogos. Foi nesta perspetiva que se encontrou o *Unity* 3D. Esta ferramenta acabou por ser a escolhida devido às suas funcionalidades. O seu ponto forte encontra-se na renderização de ambientes 3D. Isto foi visto como uma mais-valia para o desenvolvimento da aplicação, uma vez que cenários de realidade virtual são uma combinação de diferentes objetos 3D.

Outro fator que nos levou a usar o *Unity* foi o facto deste ser multiplataforma. Muitas vezes no desenvolvimento de *software* uma das limitações encontra-se no tipo de plataforma em que este pode ser executado. No entanto, com o *Unity* esta limitação é ultrapassada, sendo que a aplicação pode ser compilada para todas as versões de sistema operativo, oferecendo a um futuro utilizador uma maior escolha sobre a arquitetura da máquina a correr a aplicação

Outro fator que motivou a escolha do *Unity* deveu-se à documentação da API da aplicação existente nos *Oculus Rift* e no Leap Motion. Ambas as aplicações ofereciam suporte ao desenvolvimento de aplicações em *Unity*. Tal facto foi um fator decisivo sobre a plataforma de desenvolvimento.

A nível de linguagem de programação escolhida, toda a aplicação foi desenvolvida em C#, dado que atualmente existe um grande suporte da comunidade *online* utilizadora do *Unity*, com muitas ferramentas de auxílio desenvolvidas nessa linguagem.

Seguidamente, irá descrever-se o percurso seguido ao longo do desenvolvimento da aplicação.

5.3 Estrutura da aplicação

A aplicação está dividida em seis áreas diferentes, como se pode verificar pela figura abaixo. O primeiro menu é responsável por guiar o utilizador para uma interface de visualização de imagem estrutural. O segundo e o terceiro guiam o utilizador para uma interface de visualização de dados de conectividade estrutural, nomeadamente do grafo da matriz e das métricas de conectividade e também de outros dados estatísticos. O quarto menu é responsável por testar a ligação do dispositivo de reconhecimento gestual Leap Motion. O quinto menu fornece a informação sobre o projeto desenvolvido.



Figura 5.1 – Interface menu da aplicação

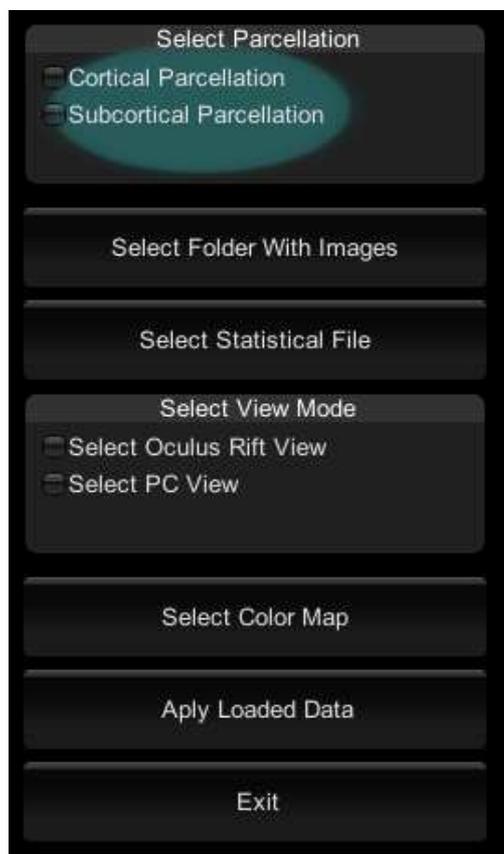
5.3.1 Interface de visualização de dados estruturais

O primeiro menu está direcionado para a visualização de imagens estruturais de MRI. Nesta secção teve-se como objetivo implementar uma interface simples, mas no entanto inovadora de visualização estrutural. Esta interface foi desenvolvida em forma de cascata, isto é, a sua utilização é feita de cima para baixo, requerendo que todos os parâmetros sejam preenchidos. A figura 5.2 mostra uma imagem do menu para visualização da estrutura cerebral.

Esta interface foi desenvolvida a pensar de forma a ser intuitiva. Optou-se por deixar o utilizador importar um conjunto de imagens cerebrais que se encontrassem numa pasta do seu computador, bem como o ficheiro contendo toda a informação estrutural das imagens em questão.

Apesar do objetivo ser a visualização em *Oculus Rift*, optou-se por integrar a visualização em 2D no ecrã do PC e em 3D em *Oculus Rift*. Esta decisão foi tomada para que quem não possua uma HMD não seja excluído da utilização da aplicação.

Figura 5.2 – Menu para visualização de imagens estruturais



O fluxograma da figura 5.3 descreve como a interface foi pensada:

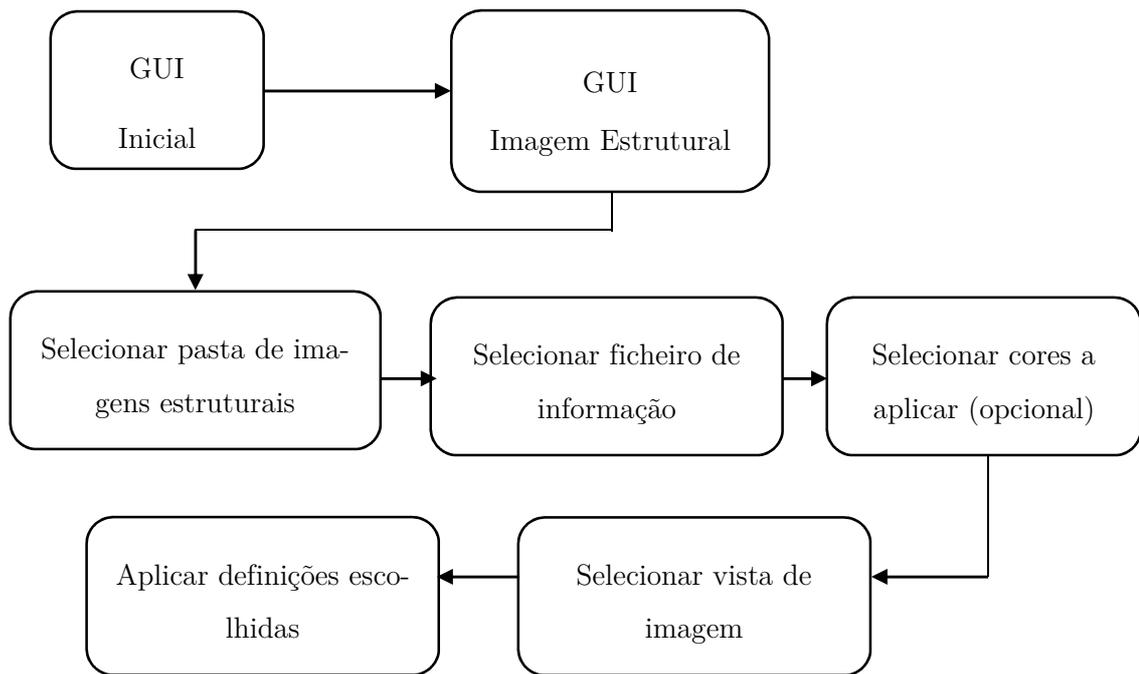


Figura 5.3 – Fluxograma da metodologia da interface de visualização de imagens estruturais

Uma das limitações do *Unity* está na importação de dados. Quando a aplicação está a correr, por predefinição o *Unity* não permite a introdução de novos elementos à cena criada. Este foi um desafio que teve que ser ultrapassado, uma vez que queremos que os dados vistos em cena sejam dinâmicos e que possam ser usados para mais do que um conjunto de dados sem ter que se estar a alterar o código da aplicação inicialmente criada.

Assim sendo, criou-se um método de importação de imagens um pouco diferente do convencional. Como ponto de partida foi criada uma interface de busca de ficheiros no computador. Esta interface foi baseada numa outra que pode ser encontrada na referência [89]. Inicialmente esta interface apenas permitia fazer uma pesquisa de ficheiros ao longo do PC, mas no entanto não realizava a importação do objeto selecionado para a cena criada. Como tal, esta interface foi associada a um script que permitia a instanciação de novos objetos à cena renderizada, deixando assim a aplicação dinâmica [90]. Esta importação é realizada sobre objetos *Wavefront*, sendo que cada objeto importado é instanciado individualmente, criando assim um novo elemento à cena. A interface usada e respetiva importação será vista em pormenor no capítulo seguinte.

5.3.2 Interface de visualização de dados de conectividade

Esta segunda parte da aplicação está direcionada à visualização de dados de conectividade, nomeadamente o grafo obtido com aplicação da matriz de conectividade e respetivas estatísticas de cada ROI usada.

No desenvolvimento da aplicação, usaram-se dados de conectividade estrutural para teste. Contudo, a aplicação é genérica, podendo visualizar dados de conectividade funcional e efetiva desde que as matrizes de conectividade apropriadas sejam fornecidas. A figura 5.4 mostra uma imagem do menu para visualização do grafo de conectividade.

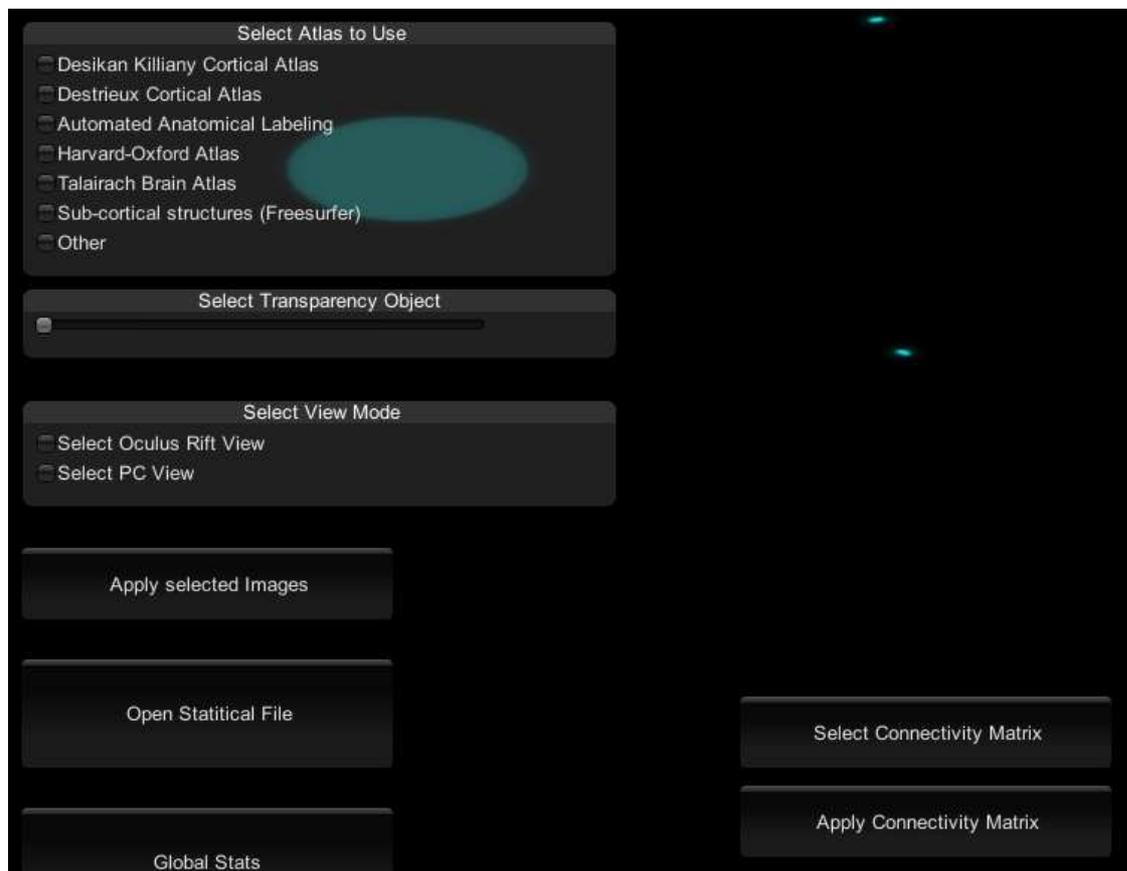


Figura 5.4 - – Interface de visualização de conectividade

Há semelhança da interface da secção anterior, também esta interface também foi desenvolvida em forma de cascata, isto é, a sua utilização é feita de cima para baixo, requerendo que todos os parâmetros sejam preenchidos.

Desenvolveu-se também na interface a possibilidade de visualização de matrizes de conectividade, obtidas de acordo com vários atlas.

Assim sendo criaram-se objetos de cada região individual de cada atlas. Estas imagens derivadas dos atlas são normalizadas/genéricas e podem ser utilizadas com dados de conectividade específicos de um indivíduo ou resultados estatísticos de grupo. Também é possível a utilização de imagens estruturais de um indivíduo específico. Contudo o processamento destes dados e a sua importação para a aplicação são morosos. Esta temática será abrangida em mais pormenor na secção seguinte.

As imagens referentes ao atlas Desikian Killiany Cortical Atlas (DKCA) e Dextrieux Cortical Atlas (DCA) como dito anteriormente são atlas usados pelo Freesurfer. O Freesurfer Subcortical Atlas (FSA) consiste na extração das zonas subcorticais a partir do processamento em Freesurfer. Visto o Freesurfer ser um *software* bastante usado em investigação em imagem neurológica, pensou-se ser uma mais-valia para a aplicação a desenvolver suportar a análise de dados provenientes desse *software*.

O *Automated Anatomical Labeling* (AAL) é um atlas cerebral que é muito comum na comunidade científica na análise de dados T1 e fMRI. O atlas foi desenvolvido a partir da parcelização de imagem normalizadas de alta resolução de MRI T1, fornecidas pelo *Montreal Neurological Institute* (MNI). Foi desenvolvido por um grupo de investigação francês podendo ser encontrado no artigo "*Automated Anatomical Labeling of activations in SPM using a Macroscopic Anatomical Parcellation of the MNI MRI single-subject brain*" o processo como foi obtido [91]. Este atlas está disponível para o SPM e foi extraído a partir de uma *toolbox* do Matlab, mostrada na figura 5.5, que permite gerar máscaras de regiões de interesse, o PickAtlas [92]. Este atlas faz uma parcelização do cérebro em 116 ROI's.

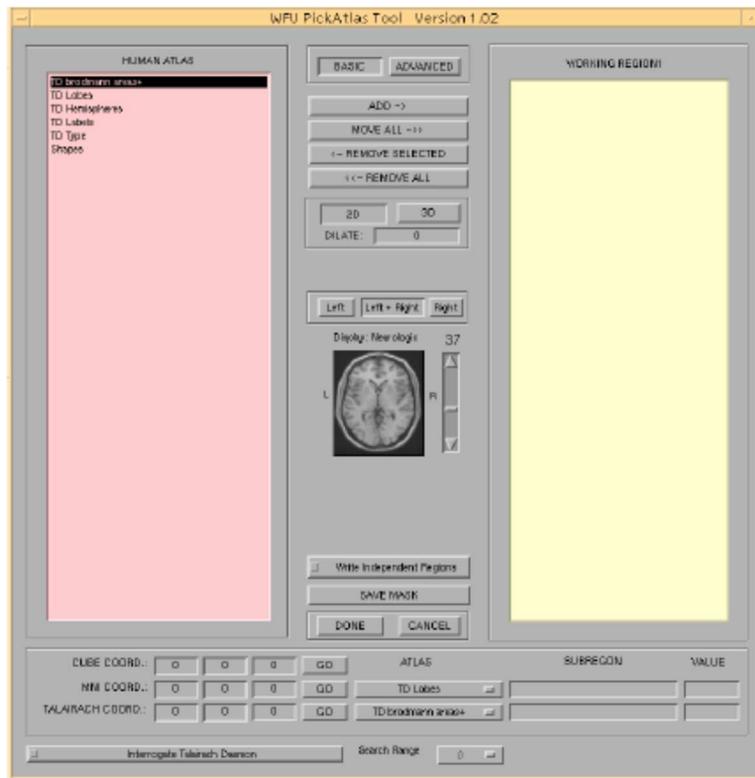


Figura 5.5 – GUI do PickAtlas. À esquerda encontramos os atlas disponíveis para extração de máscaras, à direita as máscaras escolhidas [92].

O *Harvard Oxford Atlas* (HOA) é um atlas presente no FSL. É um atlas probabilístico que parceliza o cérebro em 48 zonas corticais e 21 subcorticais. Este atlas deriva de imagens de MRI T1 de 21 indivíduos saudáveis do sexo masculino e 16 indivíduos saudáveis do sexo feminino que foram individualmente segmentados. No fim as imagens segmentadas foram combinadas de forma a criar um mapa de probabilidades para cada região [93]. O *Talairach Atlas* (TA) tal como o anterior, existe disponível no FSL, sendo um atlas mais complexo usado para análises mais exigentes, visto que este parceliza o cérebro em 1105 estruturas diferentes [93].

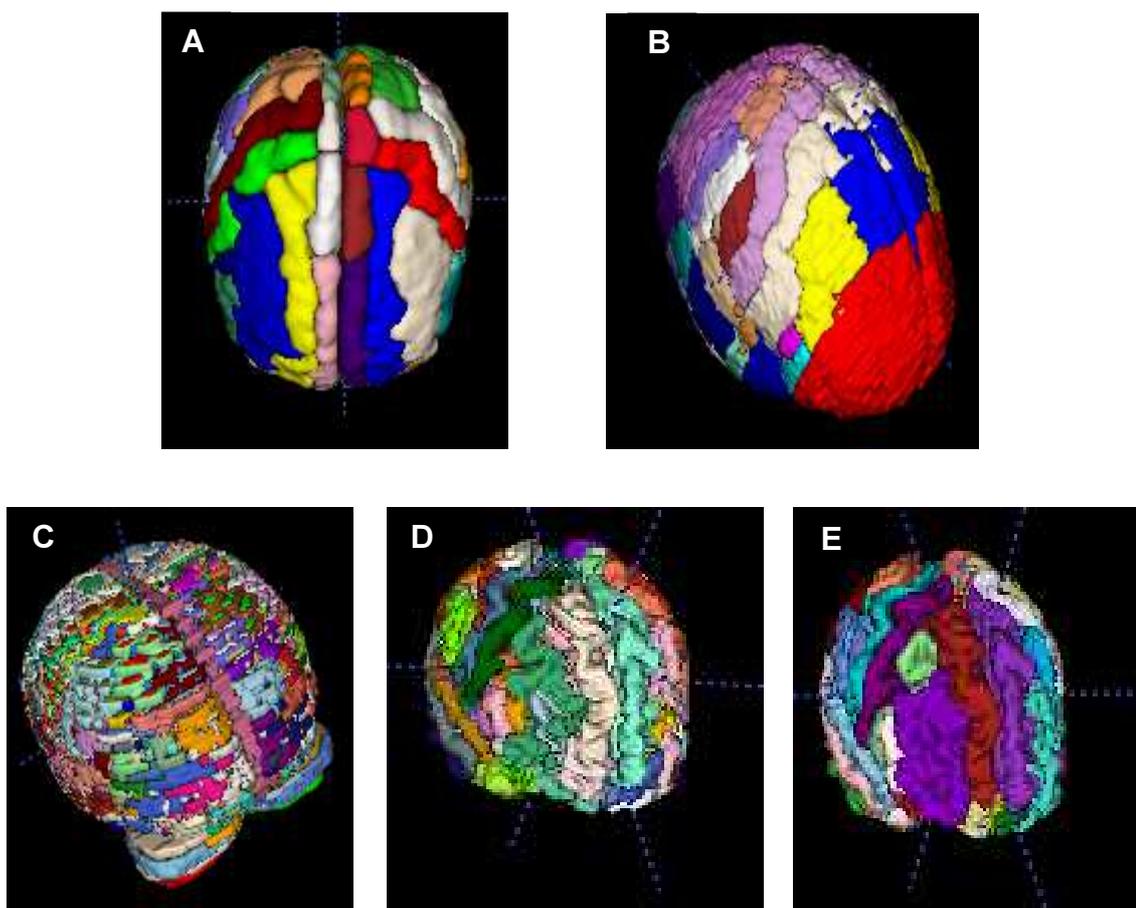


Figura 5.6 – Parcelizações efetuadas nos diferentes atlas cerebrais usados. A – Parcelização do AAL. B – Parcelização cortical HOA. C – Parcelização do TA. D – Parcelização DCA. E – Parcelização DKCA. Imagens obtidas com recurso ao *software* ITK-SNAP [94]

Todas as imagens foram convertidas para o formato Wavefront de forma a conseguir integrá-las com melhor facilidade na aplicação desenvolvida. Isto, mais uma vez foi feito com recurso a script de Matlab de forma a automatizar o processo e torna-lo genérico para todas as imagens parcelizadas.

Com recurso a estes mapas, esta aplicação pode ser usada em dados provenientes de variadas fontes, oferecendo uma maior liberdade ao utilizador.

O fluxograma da imagem 5.7 descreve como a interface foi pensada:

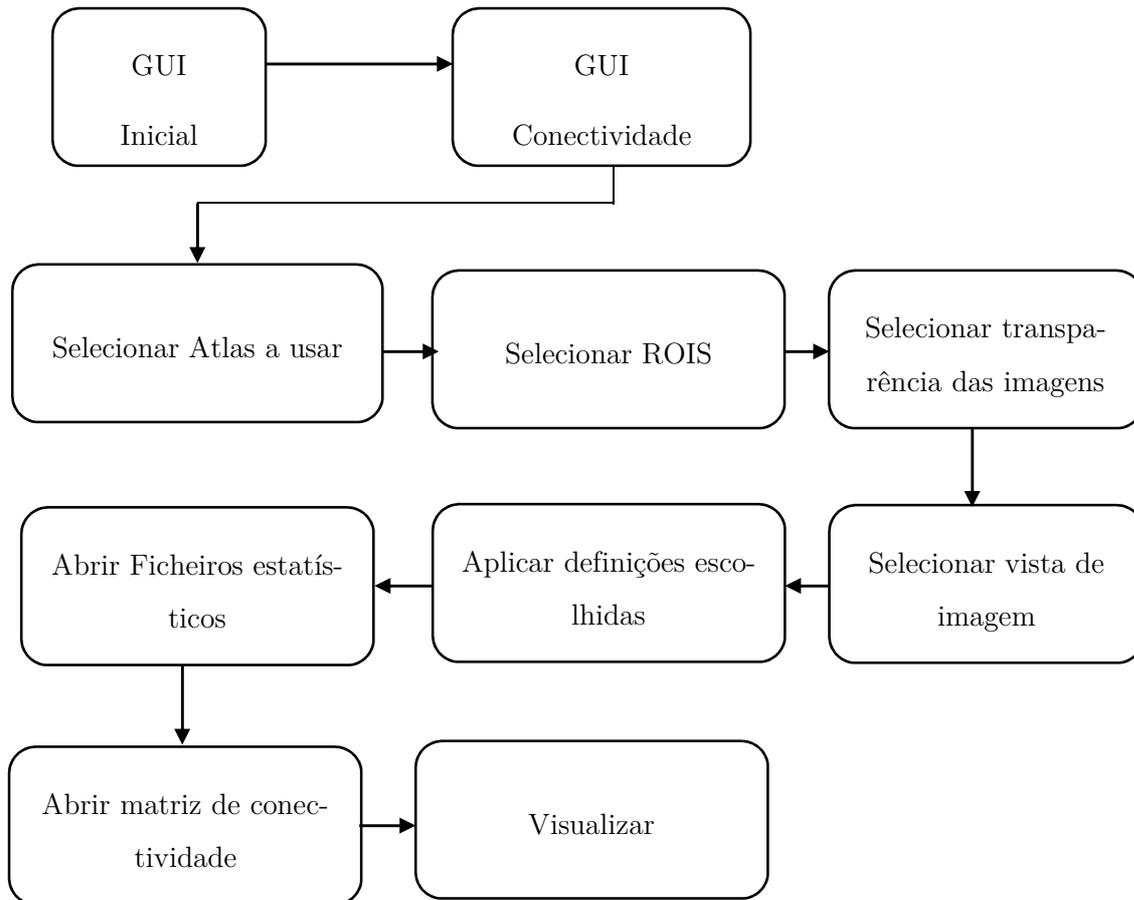


Figura 5.7 - Fluxograma da metodologia da interface de visualização da matriz de conectividade

Para visualização do resultado final, o utilizador pode optar por fazer a mesma no ecrã do computador ou caso contrário no HMD *Oculus Rift*. Esta escolha de opções foi feita a pensar de forma ao utilizador não ficar limitado no tipo de vista a ter na utilização da aplicação. Na aplicação de forma a suportar a visão em realidade virtual, teve-se de criar uma nova vista para o utilizador. Nesta vista foram incorporados os algoritmos responsáveis por criar o meio de realidade virtual. O algoritmo mais importantes que foi incluído nesta nova cena foi o de distorção de câmara, responsável por fazer a renderização da cena no HMD com a imagem distorcida que será posteriormente corrigida pelas lentes para criar a sensação de imagem estereoscópica.

Com a aplicação da metodologia acima apresentada é possível obter uma imagem 3D em ambiente de realidade virtual, onde é permitido ao utilizador fazer uma navegação virtual pelo cérebro e interagir com as suas diferentes estruturas. Nesta interação, o

utilizador pode usar o seu rato do computador juntamente com o teclado ou então usar o dispositivo Leap Motion. O uso do Leap Motion será explicado de seguida.

5.4 Controlo gestual

Para além de se pretender uma interface de visualização inovadora, quis-se ir um pouco mais longe na aplicação e juntar-lhe funcionalidade de navegação por gestos suportada por uma tecnologia bastante recente, o Leap Motion. De forma a interagir com a aplicação podem ser usadas duas formas. A primeira consiste em substituir o rato do computador por um rato virtual usando o Leap Motion. Este rato é simulado graças a uma aplicação, o *Touchless*. Esta aplicação cria um ecrã virtual do tipo *touch screen*, fazendo uma simulação do controlo do rato [95]. Esta aplicação encontra-se desenvolvida para Windows e Mac OSX.

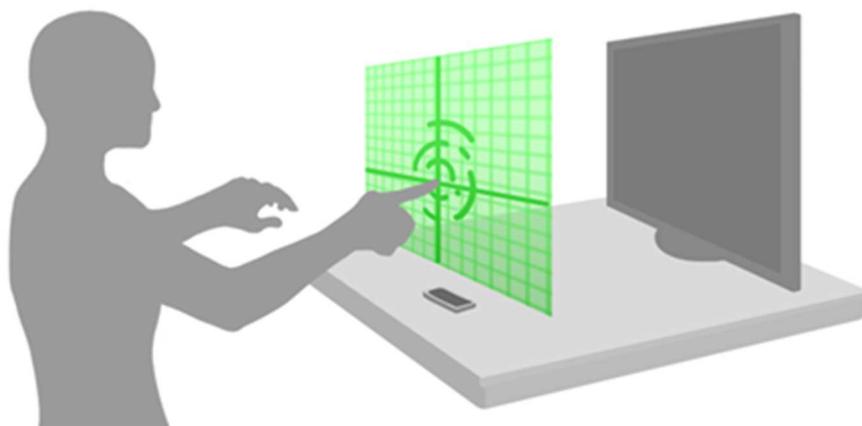


Figura 5.8 – Forma de utilização do *Touchless*, aplicação que simula um ecrã tátil virtual [95]

Esta aplicação funciona da seguinte forma:

- Movendo o dedo para o campo de visão do Leap Motion como se observa na figura 5.8, a aplicação fica ativa pronta a utilizar;
- Para realizar simular o *click* do rato, move-se o dedo para a frente até o cursor no ecrã ficar verde, fazendo de seguida um movimento como se tocas-se num ecrã *touch* de um *smartphone*.

- Para fazer *scroll* basta mover o dedo para cima e para baixo.

Com este simples aplicativo consegue-se controlar toda a nossa interface sem o recurso ao rato.

A outra metodologia de tornar a interface controlada por gestos consistiu em usar gestos da API do Leap Motion. A escolha de usar gestos pré-definidos deveu-se ao facto de o dispositivo ainda ser muito recente sendo que os algoritmos de aprendizagem de gestos ainda funcionam de forma sub-ótima. Implementar um novo gesto no dispositivo é uma tarefa muito morosa. No entanto, num futuro espera-se que essa limitação seja ultrapassada.

O Leap Motion contém na sua API quatro gestos pré-definidos, o *SwipeGesture*, o *CircleGesture*, o *ScreenTapGesture* e o *KeyTapGesture*. Esses gestos estão representados na figura abaixo.

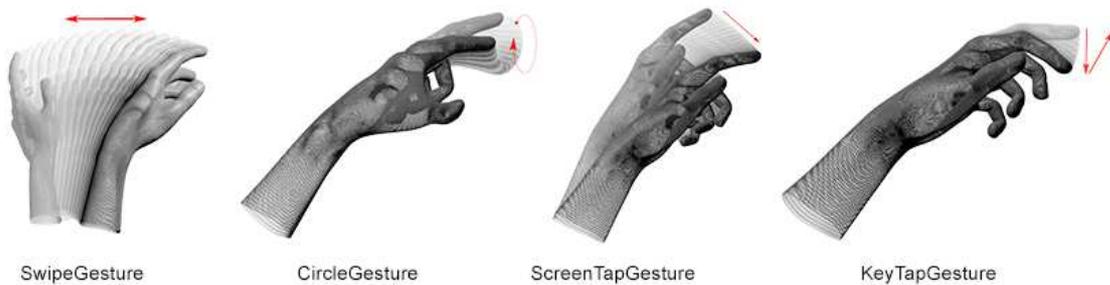


Figura 5.9 – Lista de gestos pré-definidos pela API do Leap Motion [64]

Com uma a utilização destes gestos é possível ao utilizador da aplicação desenvolvida controlar a sua navegação pelo cenário de realidade virtual sem o recurso ao rato.

6

6 Resultados obtidos

Com base em toda a metodologia descrita no capítulo anterior, foi implementada a aplicação proposta. Foram várias as etapas passadas ao longo do desenvolvimento com muitos desafios a serem ultrapassados. Seguidamente, serão apresentados os resultados obtidos ao longo do desenvolvimento da aplicação, assim como o seu funcionamento.

6.1 Interface de visualização de imagens estruturais

Nesta interface, como dito anteriormente, um dos desafios que se teve de resolver foi a importação de objetos com a aplicação a correr. Assim sendo, foi adaptada a interface da figura 6.1 às nossas necessidades, quer de importação de ficheiros individuais, quer de importação pastas completas:

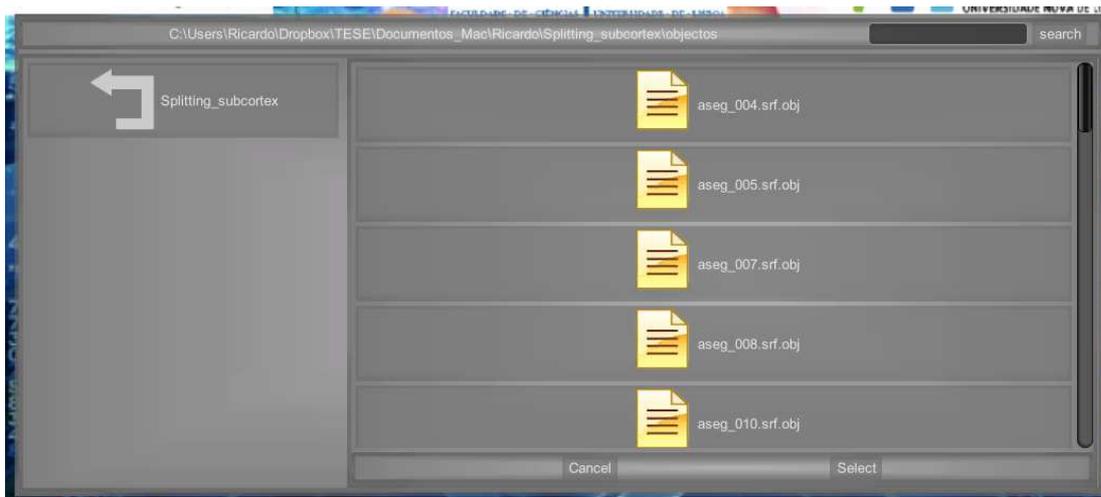


Figura 6.1 – Janela de seleção de ficheiros adaptada de [89] para as necessidades do projeto

Deste modo é dado ao utilizador total liberdade sobre os dados estruturais a visualizar. A limitação destes dados é que têm de se encontrar em formato Wavefront, limitação essa que pode ser ultrapassada com o script criado em Matlab para conversão.

Ao realizar a importação da pasta contendo as imagens estruturais, são apresentados na GUI de utilização todas as estruturas que podem ser importadas para visualização.

O utilizador tem a opção de importar todas as estruturas ou apenas as desejadas. Uma limitação encontrada nesta importação é que esta é bastante lenta. Isto deve-se ao facto de haver uma reconstrução de imagem. A aplicação tem que realizar a leitura do ficheiro e processar os dados neles contidos. Depois disto, tem que criar uma nova instância do objeto criado para que este se torne



visível ao utilizador. Isto tudo a nível computacional é bastante pesado, requerendo por vezes um grande número de renderizações. Uma solução para melhorar a performance seria limitar o número de polígonos que definem o objeto Wavefront. Isto é feito à custa de uma redução da qualidade de imagem. Esta situação foi explorada, no entanto os resultados obtidos não foram satisfatórios,

Figura 6.2 – Seleção de estruturas a visualizar pelo utilizador

uma vez que com a redução da qualidade da renderização das zonas corticais, no momento da visualização era criada uma imagem amorfa sem contornos de um cérebro. Assim sendo, optou-se por deixar a aplicação mais lenta, mas que ao mesmo tempo oferece uma boa qualidade de visualização.

Quando o utilizador tenta fazer a visualização e não escolheu a opção de ecrã de computador ou de realidade virtual, é apresentado uma janela de erro a alertar o utilizador, como se pode ver na figura 6.3.



Figura 6.3 – Erro apresentado ao utilizador quando não existe vista seleccionada

Em seguida é apresentado o resultado obtido de uma renderização em ecrã de PC da zona subcortical de um indivíduo de teste:

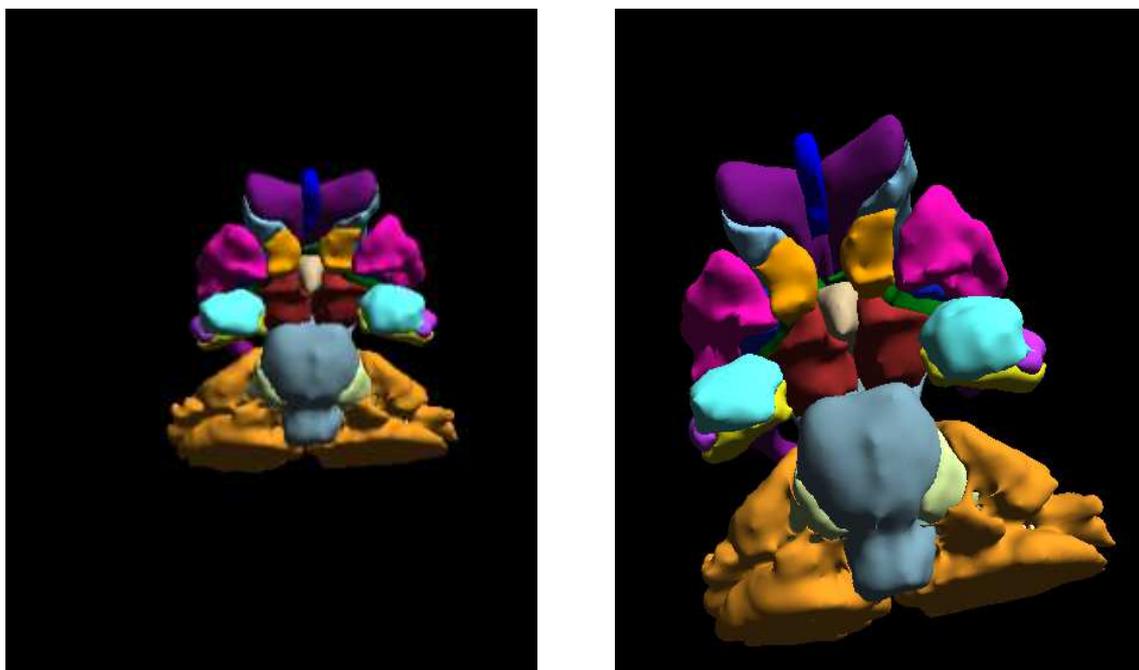


Figura 6.4 – Renderização das zonas subcorticais do Freesurfer em ecrã de computador

Como é possível verificar na figura 6.4, existe uma individualização de cada estrutura, sendo que o utilizador pode interagir com cada estrutura de modo a conseguir aceder às informações estruturais de cada uma

Visto que a aplicação está direccionada para a realidade virtual, também é possível fazer a visualização num sistema de HMD. A imagem obtida encontra-se na figura 6.5.

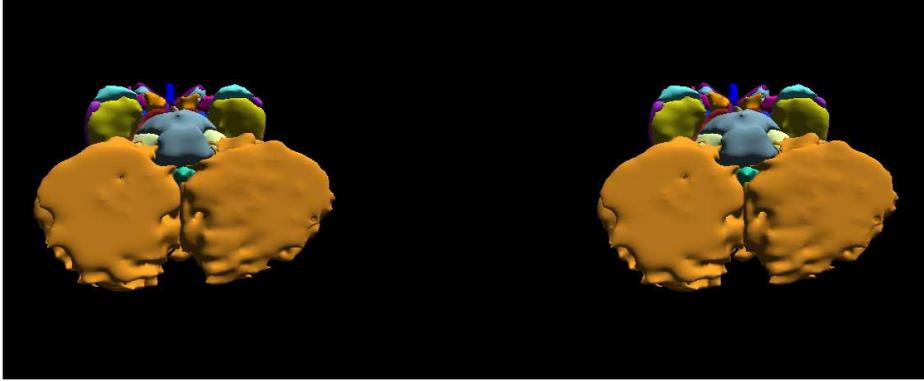


Figura 6.5 - Renderização das zonas subcorticais no HMD

A imagem à primeira vista parece semelhante à anterior, no entanto, esta visualização está preparada para os *Oculus Rift*. Esta parece replicada uma vez que é feita uma renderização para cada olho. Isto é visto no display do HMD sendo que a imagem depois é convertida para visão estereoscópica com o recurso a algoritmos internos e com lentes do HMD.

6.2 Interface de visualização de conectividade

Na presente interface tirou-se partido da importação de ficheiros implementada anteriormente. Neste caso, em vez de importação de pastas, pretendia-se fazer a importação de ficheiros contento as estatísticas de conectividade e a respetiva matriz de conectividade. Usou-se assim uma interface semelhante à apresentada na figura 6.1 adaptada à seleção de ficheiros.

Como referido no capítulo anterior, esta interface foi implementada de modo a suportar o uso de vários atlas. O utilizador ao chegar a este menu, a primeira escolha que deve fazer é a do atlas considerado no estudo.

Ao selecionar o atlas pretendido, o utilizador pode optar por usar todas as ROI's desse atlas ou fazer uma seleção manual das ROI's.

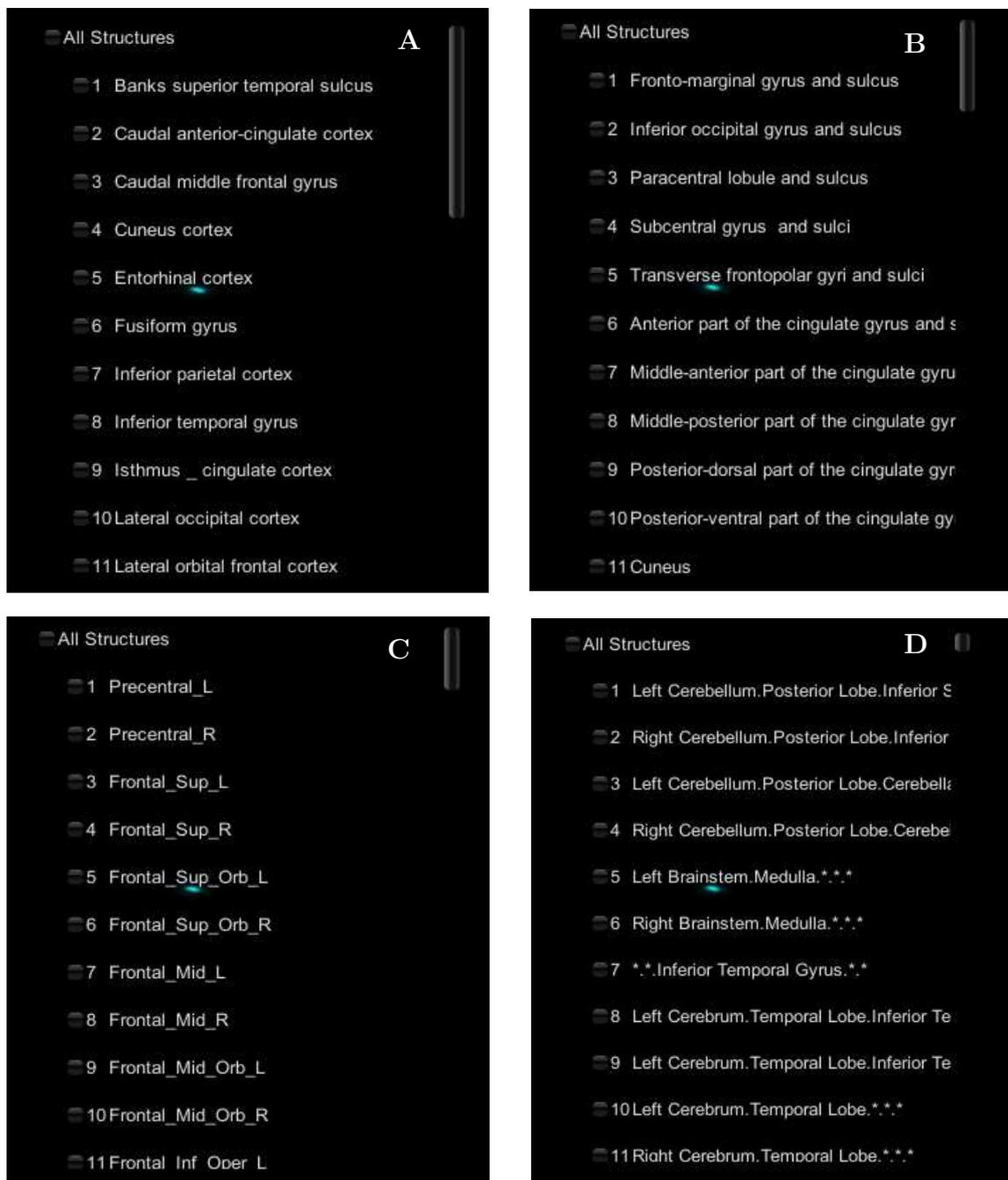


Figura 6.6 - Janelas de seleção de ROI's. A – Seleção de ROI's do atlas DKCA. B – Seleção de ROI's do atlas DCA. C – Seleção de ROI's do atlas AAL. D – Seleção de ROI's do atlas TA.

Nos exemplos a seguir será usado o atlas AAL (ver Anexo II), com recurso a dados de conectividade. O utilizador pode escolher usar apenas algumas das ROI's disponíveis.

No entanto, de forma a facilitar a orientação espacial do utilizador é feita uma sobreposição com o mapa total, dando ênfase às áreas seleccionadas pelo utilizador com opção de variação de transparência. Tal está demonstrado na seguinte figura:

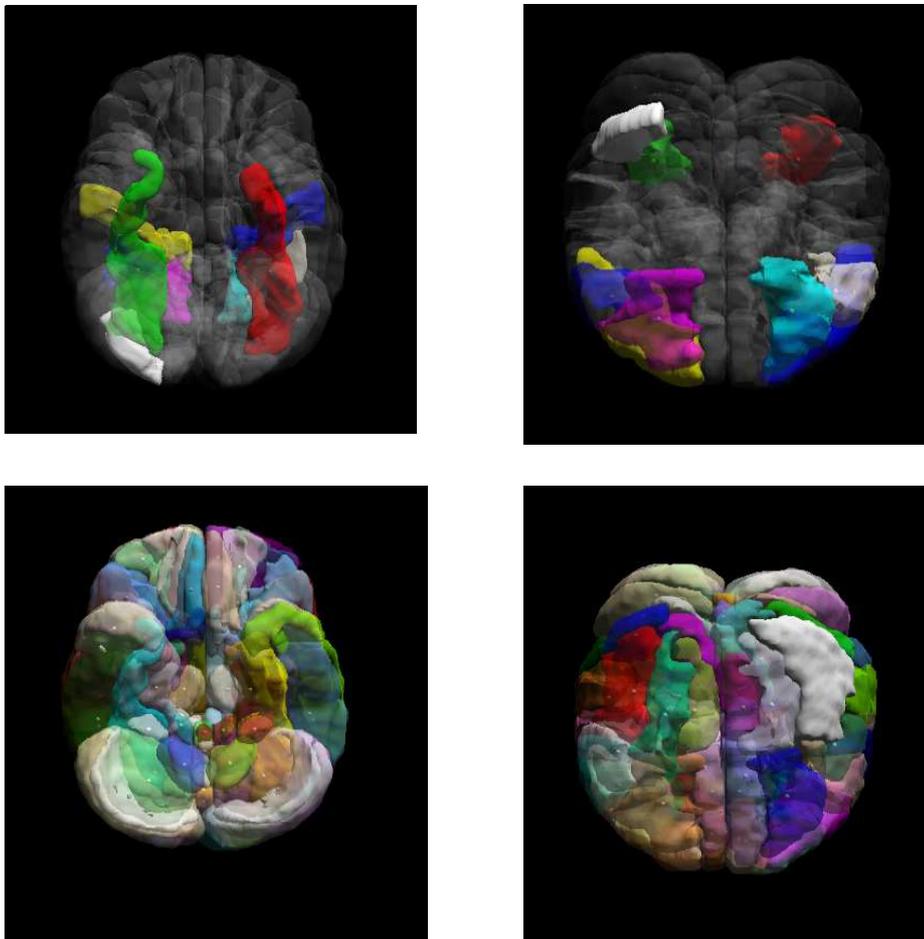


Figura 6.7 – Visualização em PC de ROI's analisadas do atlas AAL. Nas duas imagens superiores foram seleccionadas as ROI's com as labels de 1 a 8. Nas duas imagens inferiores tem-se a visualização de todas as ROI's do atlas AAL

À semelhança da secção anterior, esta visualização pode ser vista num HMD, onde o sistema se adapta às configurações necessárias. Com o uso dos *Oculus Rift*, o utilizador fica com uma visão imersiva do cérebro, sentindo-se num ambiente 3D fazendo como que uma navegação pelas diferentes áreas do cérebro. Na figura 6.8 mostra-se a vista do HMD para as ROI's todas do AAL. É de salientar que nas imagens aqui mostradas é difícil ter uma percepção total do tipo de imagens que se obtêm com a visualização no HMD, derivado da visão estereoscópica e das lentes usadas.

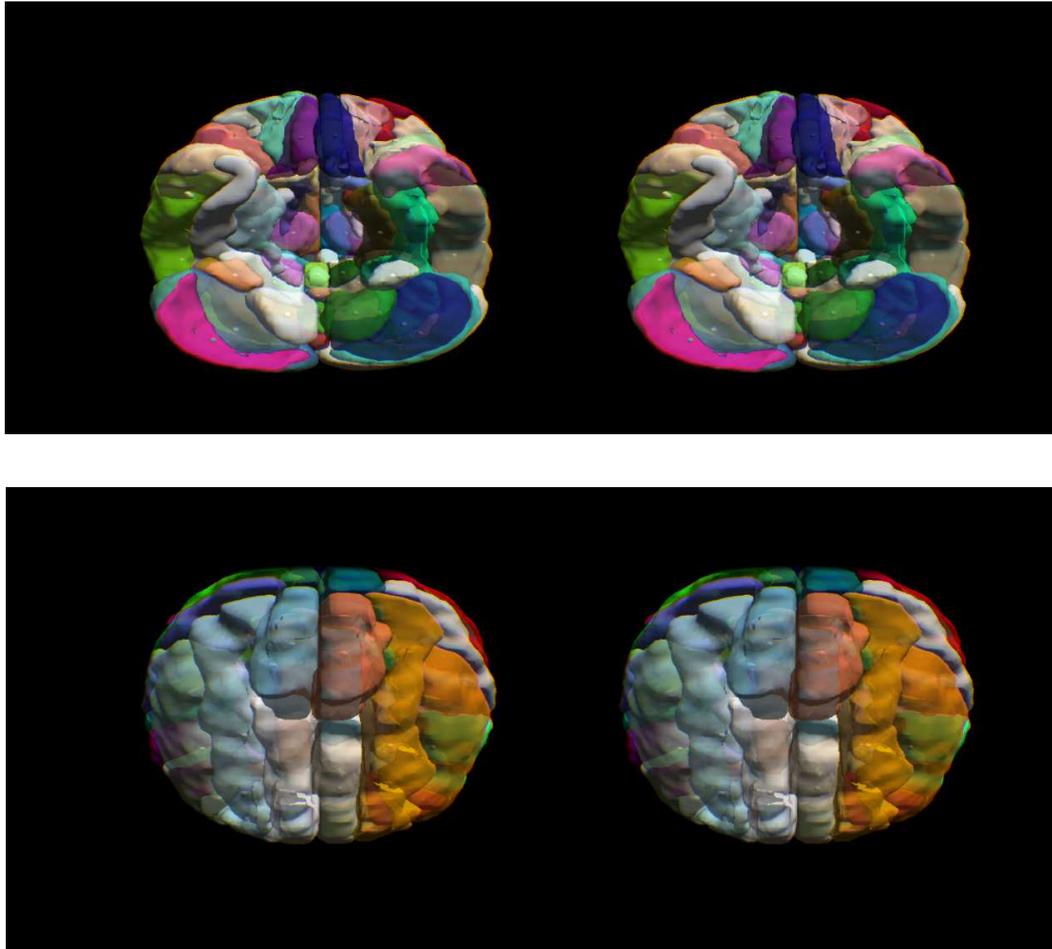


Figura 6.8 – Visualização da cena no ecrã do HMD

Quando o utilizador tenta fazer a visualização de um atlas, mas este ou não se encontra selecionado ou não escolheu a opção de ecrã de computador ou de realidade virtual, é apresentado uma janela de erro a alertar o utilizador, como se pode ver na figura 6.9. Assim sendo o utilizador tem hipótese de corrigir o erro em falha.

Após a seleção das ROI's de interesse, o utilizador pode selecionar os ficheiros contendo as estatísticas relevantes das análises efetuadas. Uma das limitações mas que num futuro pode ser ultrapassada é que as estatísticas depois de organizadas em Excel têm que ser exportadas para arquivo de texto .txt. Idealmente seria usar diretamente o ficheiro do Excel e espera-se no

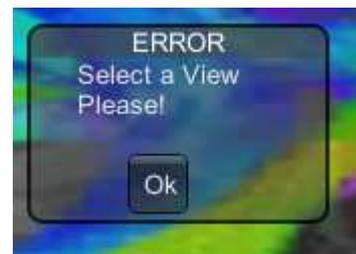


Figura 6.9 – Erro apresentado ao utilizador quando não existe vista selecionada

futuro ter essa funcionalidade ativa. Quando o utilizador aplica uma matriz de conectividade mas esta não se encontra já incluída no sistema é apresentada uma mensagem de erro a alertar do sucedido de forma ao utilizador corrigir a situação, como se pode ver na figura 6.10.



A aplicação faz assim uma renderização do atlas de ROI's com uma sobreposição com o grafo de conectividade. Esta sobreposição permite ao utilizador ter uma melhor noção das áreas e regiões cerebrais que se consideraram para o estudo, tendo esta aplicação especial importância para o ensino na área das neurociências e em investigação científica, fornecendo uma melhor perspectiva dos dados estudados.

Figura 6.10 – Erro apresentado ao utilizador quando não existe matriz de conectividade

Na figura 6.11 encontra-se a renderização em ecrã de computador dum estudo de conectividade funcional com ROI's seleccionadas do atlas AAL:

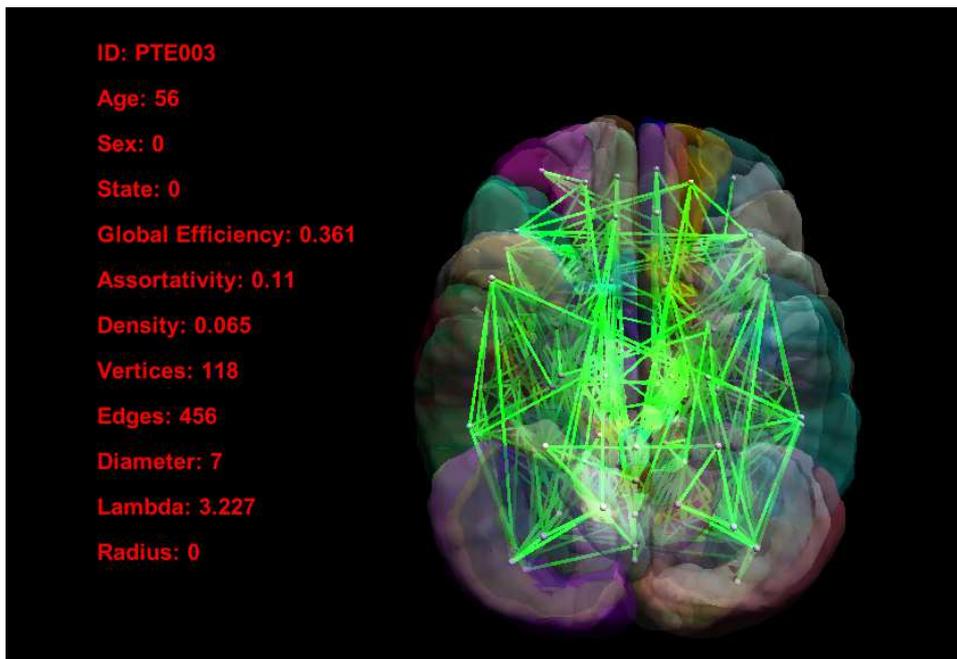


Figura 6.11 – Visualização da matriz de conectividade de um sujeito de estudo, com estatísticas globais da rede

Nesta vista, o utilizador consegue interagir com cada área individual, podendo aceder à informação estatística desse nodo específico da rede. Para aceder o utilizador por optar por usar o próprio rato ou se pretender usar a aplicação em modo gestual utilizar o Leap Motion em conjunto com o *Touchless*.

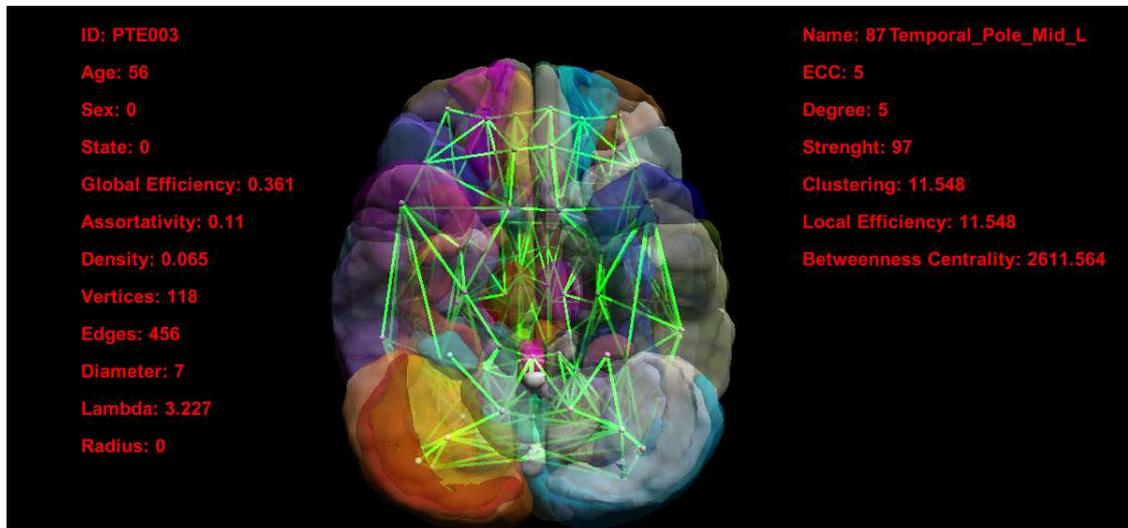


Figura 6.12 – Navegação pelo grafo de conectividade com visualização das estatísticas do nodo consideradas para o estudo

Repetindo o processo anterior, mas escolhendo uma visualização para o HMD, o utilizador fica com uma vista tridimensional do grafo, sentindo-se a “navegar” pelo cérebro. Uma das vantagens de usar o HMD é o facto deste permitir a visão estereoscópica e o facto de possuir em si sensores inerciais. Com isto, o utilizador pode fazer uma navegação mais natural pelo grafo uma vez no seu interior. Apenas com um virar da cabeça tem uma visão das conexões que rodeia um certo nodo.

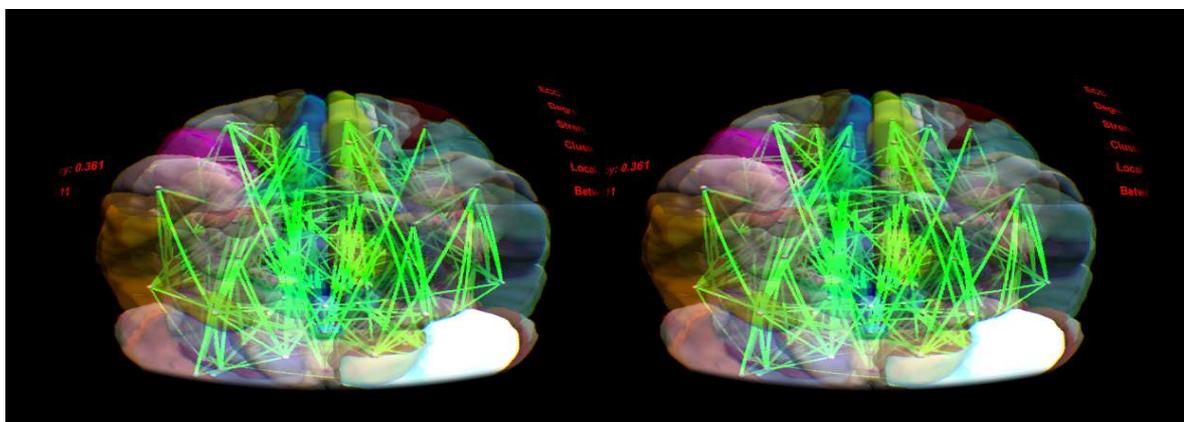


Figura 6.13 – Visualização no HMD do grafo com navegação pelos nodos

É possível fazer uma navegação pelo interior do grafo, tendo uma noção da sua dimensão e do efeito da visão estereoscópica, como se verifica na figura 6.14.

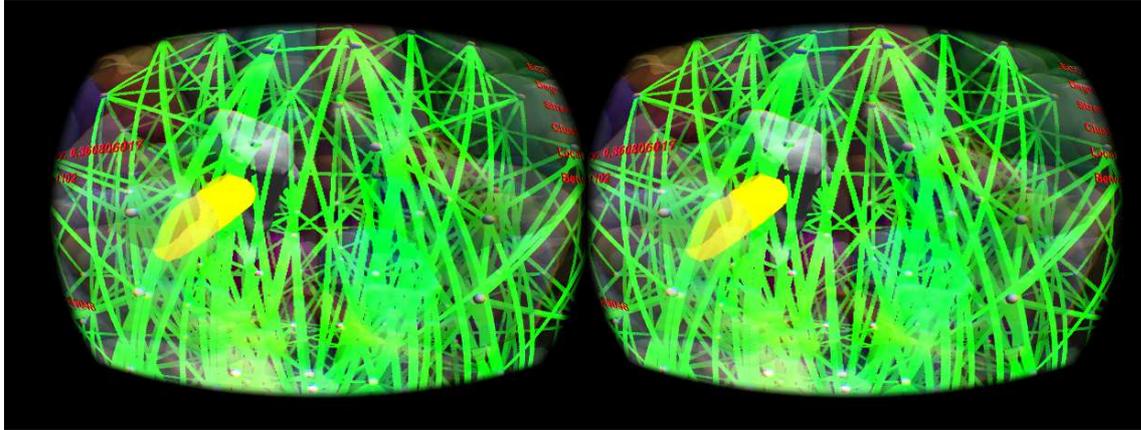


Figura 6.14 – Navegação pelo interior do grafo com o HMD

6.3 Pontos de destaque da aplicação

Analisando a aplicação desenvolvida pode-se destacar vários fatores de inovação. Os principais destaques e qualidades que encontramos na aplicação são:

- Visualização no mesmo ecrã de regiões cerebrais, conectividade e informação do grafo em estudo;
- Renderização 3D das estruturas cerebrais do indivíduo;
- Capacidade de suportar dados de vários atlas;
- Visualização de dados funcionais e estruturais;
- Interação gestual.

Tudo isto faz desta aplicação uma forte ferramenta para investigação e ensino, que com desenvolvimento futuro se pode tornar ainda mais inclusiva no tipo de dados e no tipo de exame estudado.

7

7 Conclusão

Este projeto foi desenvolvido sempre com a ideia que fosse inovador na área de visualização de dados de imagem médica, mas que ao mesmo tempo fosse útil tanto para clínicos como para a investigação e ensino. De um modo geral conseguiu-se desenvolver uma aplicação prometedora, que com as melhorias previstas fazer no futuro irá ser certamente oferecer uma forma revolucionária de visualizar imagem médica.

Neste trabalho conseguiu-se cruzar duas áreas, a imagem médica e a criação de jogos de vídeo, que muitas vezes não estão associadas uma à outra, provando que a Engenharia Biomédica é uma área transversal a muitas outras, conseguindo ser integrada nas mais variadas aplicações.

7.1 Limitações da aplicação

Um grande fator limitativo foi o tempo de desenvolvimento da dissertação. No entanto, como o tempo dado tirou-se o melhor partido disso e desenvolveu-se um protótipo de uma aplicação que pode vir a ser bastante revolucionária na área de visualização de imagem médica.

Uma das principais limitações de momento da aplicação encontra-se no controlo gestual. De momento esta usa uma emulação do rato do PC e os gestos nativos da aplicação

para interagir com a cena de visualização. Idealmente todo o cenário virtual seria controlada por gestos, oferecendo ao utilizador a opção de personalizar gestos a seu gosto, de forma a tornar a navegação no cenário de realidade virtual mais natural para si, uma vez que nem todas as pessoas usam o mesmo tipo de gestos.

Outra limitação referida anteriormente encontra-se no tipo de ficheiros que podem ser incluídos na aplicação. Esta está limitada ao formato *Wavefront* para os objetos e *.txt* para os dados, mas no trabalho futuro estas limitações poderão ser ultrapassadas. A nível da visualização de imagens estruturais, esta necessita de ser melhorada, uma vez que apesar de se conseguir fazer a visualização, ainda existem alguns problemas com a interação com as estruturas, como por exemplo dificuldade em usar o cursor em cima das mesmas. No futuro esta limitação será ultrapassada.

A nível de HMD, uma das limitações encontra-se no fato deste não ser *wireless*, o que muitas vezes limita o movimento e navegação natural do utilizador no cenário criado.

7.2 Trabalho futuro

O presente trabalho será certamente para continuar a desenvolver, uma vez que as tecnologias aqui usadas são bastante recentes e o potencial de utilização interessante.

A nível estético, no futuro a GUI será melhorada de forma a tornar-se mais simples para o utilizador.

Na aplicação um ponto que pode vir a ser melhorado é a nível da importação de dados. Seria desejável que a aplicação no futuro permitisse uma inclusão de outros tipos de ficheiros para além do *txt*. Seria útil também para a aplicação fazer uma migração dos algoritmos usados em Matlab para a aplicação em si, para que o processamento das imagens fosse todo feito na mesma aplicação.

Relativamente ao dispositivo Leap Motion, no futuro o controlo gestual espera-se que possa vir a ser desenvolvido, com a implementação de novos gestos e melhor controlo da aplicação. O objetivo é conseguir controlar toda a aplicação só com o movimento gestual, substituindo interfaces convencionais como o rato e o teclado do computador, mas ao mesmo tempo permitindo ao utilizador personalizar a interface a seu gosto.

Seria também interessante implementar na aplicação um sistema de comparação de métricas de conectividade entre sujeitos de estudo de forma ao utilizador detetar mais facilmente as alterações relevantes dessas métricas.

A nível do desenvolvimento da aplicação, espera-se que no futuro as restantes aplicações que surjam com base nesta possam e vir a incluir:

- A visualização de dados provenientes de outras modalidades de imagem médica, como CT, PET ou EEG;
- Desenvolvimento uma solução para ser usada *in vivo* na sala operatória;
- Um simulador de neurocirurgia mais realista;
- Uma aplicação para outros sistemas e órgãos, como por exemplo para o sistema nervo central e autónomo, para o aparelho digestivo e articulações;
- Adaptação do sistema para o HMD de realidade aumentada como por exemplo *Google Glass*.

A nível do HMD, no decorrer do trabalho futuro seria interessante haver uma adaptação para outro tipo de HMD sem fios para que a experiência de realidade virtual possa ser mais realista e natural.

7.3 Conclusões finais

Para concluir, o objetivo proposto no início da dissertação de criação de uma interface de visualização de conectividade em realidade virtual foi de modo global conseguido.

Este trabalho serviu para integrar várias noções apreendidas ao longo do curso a nível de imagem médica e programação orientada a objetos, conseguindo constatar que engenharia biomédica se cruza com as mais variadas áreas que muitas vezes parecem distintas. A forma como lidamos com este cruzamento de áreas varia de acordo com a pessoa e seus gostos, mostrando que a nível profissional podemos trabalhar nas mais variadas áreas.

Esta dissertação serviu para me fazer crescer a nível profissional e ao mesmo tempo estabelecer áreas nas quais gostaria de evoluir com a minha carreira futura. Ao longo desta dissertação foram feitos contactos com investigadores que trabalham com conectividade e realidade virtual, tendo o trabalho desenvolvido sido muito bem aceite. Este fato revela-se

como sendo gratificante para mim uma vez que vejo que existe interesse por parte da comunidade científica na evolução da imagem médica para a visualização em realidade virtual.

Nota: A aplicação desenvolvida neste trabalho foi apresentada no LXLMS 2014 – Demo Day no dia 24 Julho 2014, no Instituto Superior Técnico da Universidade de Lisboa, com o nome “*BrainVR: the brain connectivity navigator*”



8

8 Bibliografia

- [1] J. Marescaux, J. M. Clément, V. Tasseti, C. Koehl, S. Cotin, Y. Russier, D. Mutter, H. Delingette, and N. Ayache, “Virtual reality applied to hepatic surgery simulation: the next revolution.,” *Ann. Surg.*, vol. 228, no. 5, pp. 627–34, Nov. 1998.
- [2] “Freesurfer.” [Online]. Available: <http://freesurfer.net/>. [Accessed: 10-Mar-2014].
- [3] T. Glatard, C. Lartizien, B. Gibaud, R. F. da Silva, G. Forestier, F. Cervenansky, M. Alessandrini, H. Benoit-Cattin, O. Bernard, S. Camarasu-Pop, N. Cerezo, P. Clarysse, A. Gaignard, P. Hugonnard, H. Liebgott, S. Marache, A. Marion, J. Montagnat, J. Tabary, and D. Friboulet, “A virtual imaging platform for multi-modality medical image simulation.,” *IEEE Trans. Med. Imaging*, vol. 32, no. 1, pp. 110–8, Jan. 2013.
- [4] F. Ferreira-Santos, “Complex Network Analysis of Brain Connectivity: An Introduction,” *Lab. Neuropsychophysiology - Univ. Porto*, Porto, 2012.
- [5] J. Sousa, “Análise da conectividade estrutural na doença de Parkinson,” Tese de Mestrado, Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologia, Lisboa, 2013.
- [6] A. Fingelkurts and S. Kähkönen, “Functional connectivity in the brain - is it an elusive concept?,” *Neurosci. Biobehav. Rev.*, no. 8, pp. 827–836, 2005.

- [7] O. Sporns, *Networks of the Brain*. Cambridge: Massachusetts Institute of Technology, 2011.
- [8] M. Rubinov and O. Sporns, “Complex network measures of brain connectivity: uses and interpretations,” *Neuroimage*, vol. 52, no. 3, pp. 1059–69, Sep. 2010.
- [9] P. Menezes, “O Que é uma Rede Neuronal?” [Online]. Available: <http://home.isr.uc.pt/~paulo/PROJ/NN95/node24.html>. [Accessed: 06-Aug-2014].
- [10] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J. Honey, V. J. Wedeen, and O. Sporns, “Mapping the structural core of human cerebral cortex,” *PLoS Biol.*, vol. 6, no. 7, p. e159, Jul. 2008.
- [11] E. T. Bullmore and D. S. Bassett, “Brain graphs: graphical models of the human brain connectome,” *Annu. Rev. Clin. Psychol.*, vol. 7, pp. 113–40, Jan. 2011.
- [12] “TrackVis.” [Online]. Available: trackvis.org/. [Accessed: 09-Sep-2014].
- [13] C. Ferra, H. A. Ferreira, P. G. Pereira, R. Manaças, and A. Andrade, “Conectividade estrutural do cérebro: diferenças entre um cérebro normal e um cérebro com patologia,” *Saúde Tecnol. ed. online*, p. P. e29–e38, 2014.
- [14] G. S. P. Pamplona, “Conectividade funcional no cérebro: Uma análise das associações com desempenho intelectual e atenção sustentada usando imagens por ressonância,” Tese de Mestrado, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da Universidade de São Paulo, São Paulo, 2014.
- [15] “MRIcro.” [Online]. Available: <http://www.mccauslandcenter.sc.edu/mricro/mricro/mricro.html>. [Accessed: 01-Sep-2014].
- [16] M. Pyka, M. Hertog, R. Fernandez, S. Hauke, D. Heider, U. Dannlowski, and C. Konrad, “fMRI data visualization with BrainBlend and Blender,” *Neuroinformatics*, vol. 8, no. 1, pp. 21–31, Mar. 2010.
- [17] “OsiriX Imaging Software.” [Online]. Available: <http://www.osirix-viewer.com/Downloads.html>. [Accessed: 08-Sep-2014].
- [18] Tonya Limberg, “Osirix as a Resource,” Chicago, 2008.

- [19] C. Rodrigues, “Touchless autopsy report,” Tese Mestrado, Faculdade de Ciências e Tecnologia - Universidade Nova de Lisboa, Lisboa, 2013.
- [20] 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,” *J. Forensic Radiol. Imaging*, vol. 1, no. 1, pp. 10–14, Jan. 2013.
- [21] M. Rubinov and O. Sporns, “Complex network measures of brain connectivity: uses and interpretations.,” *Neuroimage*, vol. 52, no. 3, pp. 1059–69, Sep. 2010.
- [22] “BrainVoyager Support.” [Online]. Available: <http://support.brainvoyager.com/diffusion-weighted-imaging/70-interaction-with-other-software/348-convert-trackvis-trk-file-to-brainvoyager-fbr.html>. [Accessed: 10-Sep-2014].
- [23] “MATLAB - The Language of Technical Computing.” [Online]. Available: <http://www.mathworks.com/products/matlab/>. [Accessed: 22-Sep-2014].
- [24] D. Zhou, W. K. Thompson, and G. Siegle, “MATLAB toolbox for functional connectivity.,” *Neuroimage*, vol. 47, no. 4, pp. 1590–607, Oct. 2009.
- [25] “Functional Connectivity Toolbox.” [Online]. Available: <https://sites.google.com/site/functionalconnectivitytoolbox/>. [Accessed: 05-Sep-2014].
- [26] “Brain Connectivity Toolbox.” [Online]. Available: <https://sites.google.com/site/bctnet/>. [Accessed: 05-Sep-2014].
- [27] “BrainCAT.” [Online]. Available: <http://www.icvs.uminho.pt/research-scientists/neurosciences/resources/braincat>. [Accessed: 22-Sep-2014].
- [28] A. S. Ribeiro, L. M. Lacerda, N. A. da Silva, and H. A. Ferreira, “Multimodal imaging brain connectivity analysis (MIBCA) toolbox: preliminary application to Alzheimer’s disease,” in *EJNMMI Physics 2014 1(Suppl 1):A61*.
- [29] P. Marques, J. M. Soares, V. Alves, and N. Sousa, “BrainCAT - a tool for automated and combined functional magnetic resonance imaging and diffusion tensor imaging brain connectivity analysis,” *Front. Hum. Neurosci.*, vol. 7, no. November, p. 794, Jan. 2013.

- [30] J. Steuer, “Defining Virtual Reality: Dimensions Determining Telepresence,” *J. Commun.*, vol. 42, no. 4, pp. 73–93, Dec. 1992.
- [31] T. Mazuryk and M. Gervautz, “Virtual reality-history, applications, technology and future,” Report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria, 1996.
- [32] “Inventor in the field of virtual reality.” [Online]. Available: <http://www.mortonheilig.com/InventorVR.html>. [Accessed: 07-Aug-2014].
- [33] “HMD – History and objectives of inventions.” [Online]. Available: <http://glassdevelopment.wordpress.com/2014/04/17/hmd-history-and-objectives-of-inventions/>. [Accessed: 07-Aug-2014].
- [34] “How did virtual reality begin?” [Online]. Available: <http://www.vrs.org.uk/virtual-reality/beginning.html>. [Accessed: 06-Aug-2014].
- [35] S. Burke, “The History of Virtual Reality & The Future: Rift, Omni, STEM, castAR.” [Online]. Available: www.gamersnexus.net/guides/1208-history-of-virtual-reality-and-future. [Accessed: 07-Aug-2014].
- [36] R. Holloway and A. Lastra, “Virtual environments: A survey of the technology,” Eurographics, Chapel Hill, 1993.
- [37] “The Virtual Reality Headset.” [Online]. Available: <http://www.vrs.org.uk/virtual-reality/headset.html>. [Accessed: 07-Aug-2014].
- [38] O. VR, “Oculus Rift - Virtual Reality Headset for Immersive 3D Gaming.” [Online]. Available: <http://www.oculusvr.com/rift/>. [Accessed: 01-Aug-2014].
- [39] “HMZ-T3W | Óculos de Realidade Virtual | Sony PT.” [Online]. Available: <http://www.sony.pt/electronics/produtos-oculos-realidade-virtual/hmz-t3w>. [Accessed: 13-Aug-2014].
- [40] M. Bolas, P. Hoberman, P. Luckey, J. Iliff, N. Burba, I. McDowall, and D. M. Krum, “Open virtual reality,” *2013 IEEE Virtual Real.*, pp. 183–184, Mar. 2013.
- [41] “Oculus Rift Teardown.” [Online]. Available: <https://www.ifixit.com/Teardown/Oculus+Rift+Teardown/13682>. [Accessed: 13-Aug-2014].

- [42] N. Whiting, “Integrating the Oculus Rift into Unreal Engine 4.” [Online]. Available: http://gamasutra.com/blogs/NickWhiting/20130611/194007/Integrating_the_Oculus_Rift_into_Unreal_Engine_4.php. [Accessed: 05-Aug-2014].
- [43] “Sony HMZ-T3W review.” [Online]. Available: http://www.trustedreviews.com/sony-hmz-t3w-personal-3d-viewer_Gadget_review.
- [44] “Sony HMZ-T3W Head Mounted Display transports you to your own private world (hands-on).” [Online]. Available: <http://www.cnet.com/products/sony-hmz-t3w-head-mou>. [Accessed: 13-Aug-2014].
- [45] N. Ayache, “Medical computer vision, virtual reality and robotics,” *Image Vis. Comput.*, vol. 13, no. 4, pp. 295–313, May 1995.
- [46] R. M. Satava, “Medical applications of virtual reality.,” *J. Med. Syst.*, vol. 19, no. 3, pp. 275–80, Jun. 1995.
- [47] G. Szekely and R. M. Satava, “Virtual reality in medicine,” *Bmj*, vol. 319, no. 7220, pp. 1305–1305, Nov. 1999.
- [48] C.-Y. Lo, Y.-P. Chao, K.-H. Chou, W.-Y. Guo, J.-L. Su, and C.-P. Lin, “DTI-based virtual reality system for neurosurgery.,” *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2007, pp. 1326–9, Jan. 2007.
- [49] “About Robotic Surgery.” [Online]. Available: <http://health.ucsd.edu/specialties/surgery/mis/about/Pages/about-robotic.aspx>. [Accessed: 02-Sep-2014].
- [50] “Symbionix.” [Online]. Available: <http://symbionix.com/company/>. [Accessed: 03-Sep-2014].
- [51] J. Skinovsky, T. Pr, M. Chibata, and D. E. D. Siqueira, “Virtual reality and robotics in surgery : Where we are and where we,” *Colégio Bras. Cir.*, vol. 35, 2008.
- [52] M. Caglio, L. Latini-Corazzini, F. D’Agata, F. Cauda, K. Sacco, S. Monteverdi, M. Zettin, S. Duca, and G. Geminiani, “Virtual navigation for memory rehabilitation in a traumatic brain injured patient.,” *Neurocase*, vol. 18, no. 2, pp. 123–31, Jan. 2012.

- [53] F. D. Rose, B. M. Brooks, and A. a Rizzo, "Virtual reality in brain damage rehabilitation: review.," *Cyberpsychol. Behav.*, vol. 8, no. 3, pp. 241–62; discussion 263–71, Jun. 2005.
- [54] A. S. Carlin, H. G. Hoffman, and S. Weghorst, "Virtual reality and tactile augmentation in the treatment of spider phobia: A case report," *Behav. Res. Ther.*, vol. 35, pp. 153–158, 1997.
- [55] C. Botella, R. M. Baños, C. Perpiñá, H. Villa, M. Alcañiz, and A. Rey, "Virtual reality treatment of claustrophobia: A case report," *Behav. Res. Ther.*, vol. 36, pp. 239–246, 1998.
- [56] L. F. Hodges, R. Kooper, T. C. Meyer, B. O. Rothbaum, D. Opdyke, J. J. de Graaff, J. S. Williford, and M. M. North, "Virtual environments for treating the fear of heights," *Computer (Long. Beach. Calif.)*, vol. 28, pp. 27–33, 1995.
- [57] B. O. Rothbaum, L. F. Hodges, R. Kooper, D. Opdyke, J. S. Williford, and M. North, "Effectiveness of computer-generated (virtual reality) graded exposure in the treatment of acrophobia," *Am. J. Psychiatry*, vol. 152, pp. 626–628, 1995.
- [58] E. Klinger and S. Bouchard, "Virtual Reality Therapy Versus Cognitive Behavior Therapy for Social Phobia: A Preliminary Controlled Study," ... *Behav.*, vol. 8, pp. 76–89, 2005.
- [59] "The Virtual Reality Medical Center." [Online]. Available: <http://www.vrphobia.com/>. [Accessed: 21-Sep-2014].
- [60] "Leap Motion." [Online]. Available: <https://www.leapmotion.com/>. [Accessed: 08-Jun-2014].
- [61] "Virtuix Omni." [Online]. Available: <http://www.virtuix.com/>. [Accessed: 08-Aug-2014].
- [62] "Leap Motion unveils contents of its own app store." [Online]. Available: <http://www.newscientist.com/blogs/onepercent/2013/02/leap-motion-app-store.html>. [Accessed: 14-Aug-2014].
- [63] 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, pp. 6380–93, Jan. 2013.

- [64] “API Overview.” [Online]. Available: https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html?proglang=current. [Accessed: 14-Aug-2014].
- [65] “Wonderland Labs.” [Online]. Available: <http://www.wonderlandlabs.com/blog/feedback>. [Accessed: 14-Aug-2014].
- [66] “Leap Motion Jumps Into Virtual Reality.” [Online]. Available: <http://www.pcmag.com/article2/0,2817,2465498,00.asp>. [Accessed: 01-Sep-2014].
- [67] “Insane Virtuix Omni VR treadmill can now be preordered for \$500.” [Online]. Available: <http://www.dvice.com/2013-8-12/insane-virtuix-omni-vr-treadmill-can-now-be-preordered-500>. [Accessed: 02-Sep-2014].
- [68] M. Carlsson, “Virtual Reality – Streaming and Concepts of Movement Tracking,” Bachelor thesis in Computer Science, Mälardalens Högskola, 2014.
- [69] D. Trenholme and S. P. Smith, “Computer game engines for developing first-person virtual environments,” *Virtual Real.*, vol. 12, no. 3, pp. 181–187, Mar. 2008.
- [70] M. Lewis and J. Jacobson, “Game Engines In Scientific Research,” *Commun. ACM*, vol. 45, pp. 27–31, 2002.
- [71] E. B. Passos, J. R. Silva Jr., F. E. C. Ribeiro, and P. T. Mourão, “Tutorial: Desenvolvimento de jogos com unity 3d,” in *VIII Brazilian Symposium on Games and Digital Entertainment*, 2009, pp. 1–30.
- [72] “CaveUT.” [Online]. Available: http://publicvr.org/html/pro_caveut.html. [Accessed: 10-Sep-2014].
- [73] J. Jacobson and M. Lewis, “Game Engine Virtual Reality with CaveUT,” *IEEE Comput. Soc.*, vol. 38, no. April, pp. 79–82, 2005.
- [74] “What is Unity?” [Online]. Available: <https://unity3d.com/pt/pages/what-is-unity>. [Accessed: 27-Aug-2014].
- [75] J. Suvak, “Getting Started with Unity,” *Learn Unity 3D Program. with UnityScript*, 2014.

- [76] “Learning the Interface.” [Online]. Available: <http://docs.unity3d.com/Manual/LearningtheInterface.html>. [Accessed: 27-Aug-2014].
- [77] “Unreal Engine.” [Online]. Available: <https://developer.nvidia.com/unrealengine>. [Accessed: 27-Aug-2014].
- [78] “Unreal Engine - Frequently Asked Questions (FAQ).” [Online]. Available: <https://www.unrealengine.com/faq>. [Accessed: 27-Aug-2014].
- [79] “Quick Start.” [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/QuickStart/1/index.html>. [Accessed: 27-Aug-2014].
- [80] “Game Engine Comparison.” [Online]. Available: <http://esenthel.com/?id=compare>. [Accessed: 27-Aug-2014].
- [81] “Glass Brain.” [Online]. Available: neuroscapelab.com/projects/glass-brain/. [Accessed: 03-Sep-2014].
- [82] “Cortical Parcellation.” [Online]. Available: <https://surfer.nmr.mgh.harvard.edu/fswiki/CorticalParcellation>. [Accessed: 04-Apr-2014].
- [83] C. Destrieux, B. Fischl, A. Dale, and E. Halgren, “Automatic parcellation of human cortical gyri and sulci using standard anatomical nomenclature.,” *Neuroimage*, vol. 53, no. 1, pp. 1–15, Oct. 2010.
- [84] R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, M. S. Albert, and R. J. Killiany, “An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest.,” *Neuroimage*, vol. 31, no. 3, pp. 968–80, Jul. 2006.
- [85] A. Klein and J. Tourville, “101 Labeled Brain Images and a Consistent Human Cortical Labeling Protocol,” *Front. Neurosci.*, vol. 6, no. December, p. 171, Jan. 2012.
- [86] A. M. Winkler, “Brainder.” [Online]. Available: <http://brainder.org>. [Accessed: 13-Feb-2014].

- [87] “Areal interpolation.” [Online]. Available: <http://brainder.org/download/areal/>. [Accessed: 18-Feb-2014].
- [88] “blender.org - Home of the Blender project.” [Online]. Available: <http://www.blender.org/>. [Accessed: 15-Apr-2014].
- [89] “C# File Browser.” [Online]. Available: <https://www.assetstore.unity3d.com/en/#!/content/18308>.
- [90] “[UPDATE] runtime obj importer.” [Online]. Available: <http://forum.unity3d.com/threads/update-runtime-obj-importer.43161/>.
- [91] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, “Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain.,” *Neuroimage*, vol. 15, no. 1, pp. 273–89, Jan. 2002.
- [92] J. Maldjian, “WFU PickAtlas User Manual v2 . 4,” pp. 1–13.
- [93] “FSL Atlases.” [Online]. Available: <http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases>. [Accessed: 26-Aug-2014].
- [94] “ITK-SNAP Home Page.” [Online]. Available: <http://www.itksnap.org/pmwiki/pmwiki.php>. [Accessed: 04-Apr-2013].
- [95] Leap Motion, “Touchless For Windows.” [Online]. Available: <https://airspace.leapmotion.com/apps/touchless-for-windows/windows>. [Accessed: 25-Aug-2013].

Anexo I – Segmentação Cortical do Freesurfer

I.I - Atlas Destrieaux

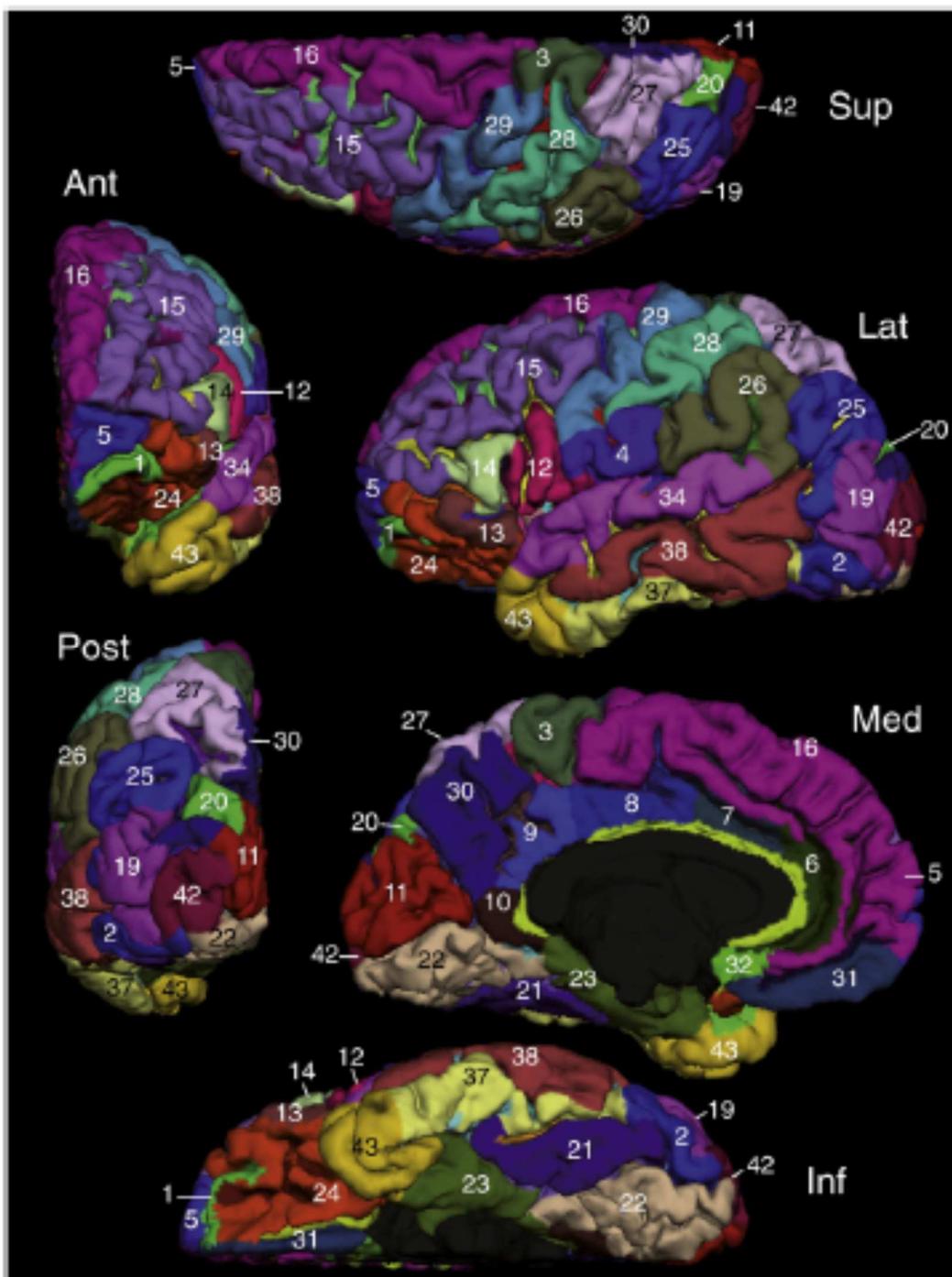


Figura I.I – Parcelização cortical segundo o atlas *Destrieaux* [83]

ID	Nome da Estrutura
1	Fronto-marginal gyrus and sulcus
2	Inferior occipital gyrus and sulcus
3	Paracentral lobule and sulcus
4	Subcentral gyrus and sulci
5	Transverse frontopolar gyri and sulci
6	Anterior part of the cingulate gyrus and sulcus
7	Middle-anterior part of the cingulate gyrus and sulcus
8	Middle-posterior part of the cingulate gyrus and sulcus
9	Posterior-dorsal part of the cingulate gyrus
10	Posterior-ventral part of the cingulate gyrus
11	Cuneus
12	Opercular part of the inferior frontal gyrus
13	Orbital part of the inferior frontal gyrus
14	Triangular part of the inferior frontal gyrus
15	Middle frontal gyrus
16	Superior frontal gyrus
17	Long insular gyrus and central sulcus of the insula
18	Short insular gyri
19	Middle occipital gyrus
20	Superior occipital gyrus
21	Lateral occipito-temporal gyrus
22	Lingual gyrus lingual part of the medial occipito-temporal gyrus
23	Parahippocampal gyrus, parahippocampal part of the medial occipito-temporal gyrus
24	Orbital gyri
25	Angular gyrus
26	Supramarginal gyrus
27	Superior parietal lobule
28	Postcentral gyrus
29	Precentral gyrus
30	Precuneus
31	Straight gyrus, Gyrus rectus
32	Subcallosal area, subcallosal gyrus
33	Anterior transverse temporal gyrus
34	Lateral aspect of the superior temporal gyrus
35	Planum polare of the superior temporal gyrus
36	Planum temporale or temporal plane of the superior temporal gyrus
37	Inferior temporal gyrus
38	Middle temporal gyrus
39	Horizontal ramus of the anterior segment of the lateral sulcus

40	Vertical ramus of the anterior segment of the lateral sulcus
41	Posterior ramus (or segment) of the lateral sulcus
42	Occipital pole
43	Temporal pole
44	Calcarine sulcus
45	Central sulcus
46	Marginal branch of the cingulate sulcus
47	Anterior segment of the circular sulcus of the insula
48	Inferior segment of the circular sulcus of the insula
49	Superior segment of the circular sulcus of the insula
50	Anterior transverse collateral sulcus
51	Posterior transverse collateral sulcus
52	Inferior frontal sulcus
53	Middle frontal sulcus
54	Superior frontal sulcus
55	Sulcus intermedius primus
56	Intraparietal sulcus and transverse parietal sulci
57	Middle occipital sulcus and lunatus sulcus
58	Superior occipital sulcus and transverse occipital sulcus
59	Anterior occipital sulcus and preoccipital notch
60	Lateral occipito-temporal sulcus
61	Medial occipito-temporal sulcus and lingual sulcus
62	Lateral orbital sulcus
63	Medial orbital sulcus
64	Orbital sulci
65	Parieto-occipital sulcus
66	Pericallosal sulcus
67	Postcentral sulcus
68	Inferior part of the precentral sulcus
69	Superior part of the precentral sulcus
70	Suborbital sulcus
71	Subparietal sulcus
72	Inferior temporal sulcus
73	Superior temporal sulcus
74	Transverse temporal sulcus

Tabela I.I – Identificação das estruturas da parcelização segundo o atlas *Destrieaux* [83]

I.II - Atlas Desikan-Killiany-Tourville

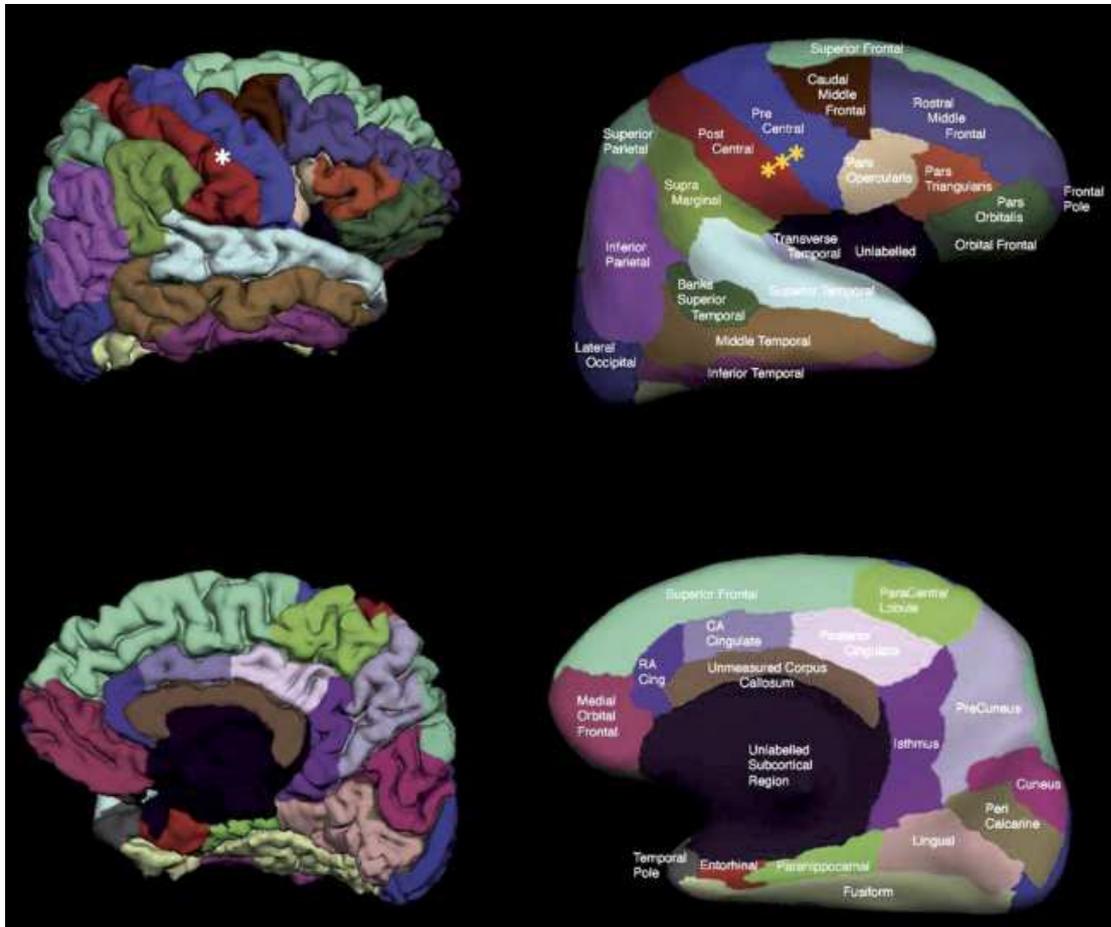


Figura I.II – Parcelização cortical segundo o atlas Desikan-Killiany-Tourville [84]

ID	Nome da Estrutura
1	Banks superior temporal sulcus
2	Caudal anterior-cingulate cortex
3	Caudal middle frontal gyrus
4	Cuneus cortex
5	Entorhinal cortex
6	Fusiform gyrus
7	Inferior parietal cortex
8	Inferior temporal gyrus
9	Isthmus cingulate cortex
10	Lateral occipital cortex
11	Lateral orbital frontal cortex
12	Lingual gyrus
13	Medial orbital frontal cortex
14	Middle temporal gyrus
15	Parahippocampal gyrus
16	Paracentral lobule
17	Pars opercularis
18	Pars orbitalis
19	Pars triangularis
20	Pericalcarine cortex
21	Postcentral gyrus
22	Posterior-cingulate cortex
23	Precentral gyrus
24	Precuneus cortex
25	Rostral anterior cingulate cortex
26	Rostral middle frontal gyrus
27	Superior frontal gyrus
28	Superior parietal cortex
29	Superior temporal gyrus
30	Supramarginal gyrus
31	Temporal pole
32	Transverse temporal cortex

Tabela I.II – Identificação das estruturas da parcelização segundo o atlas Desikan-Killiany-Tourville [84], [85]

Anexo II – Segmentação Cortical segundo AAL

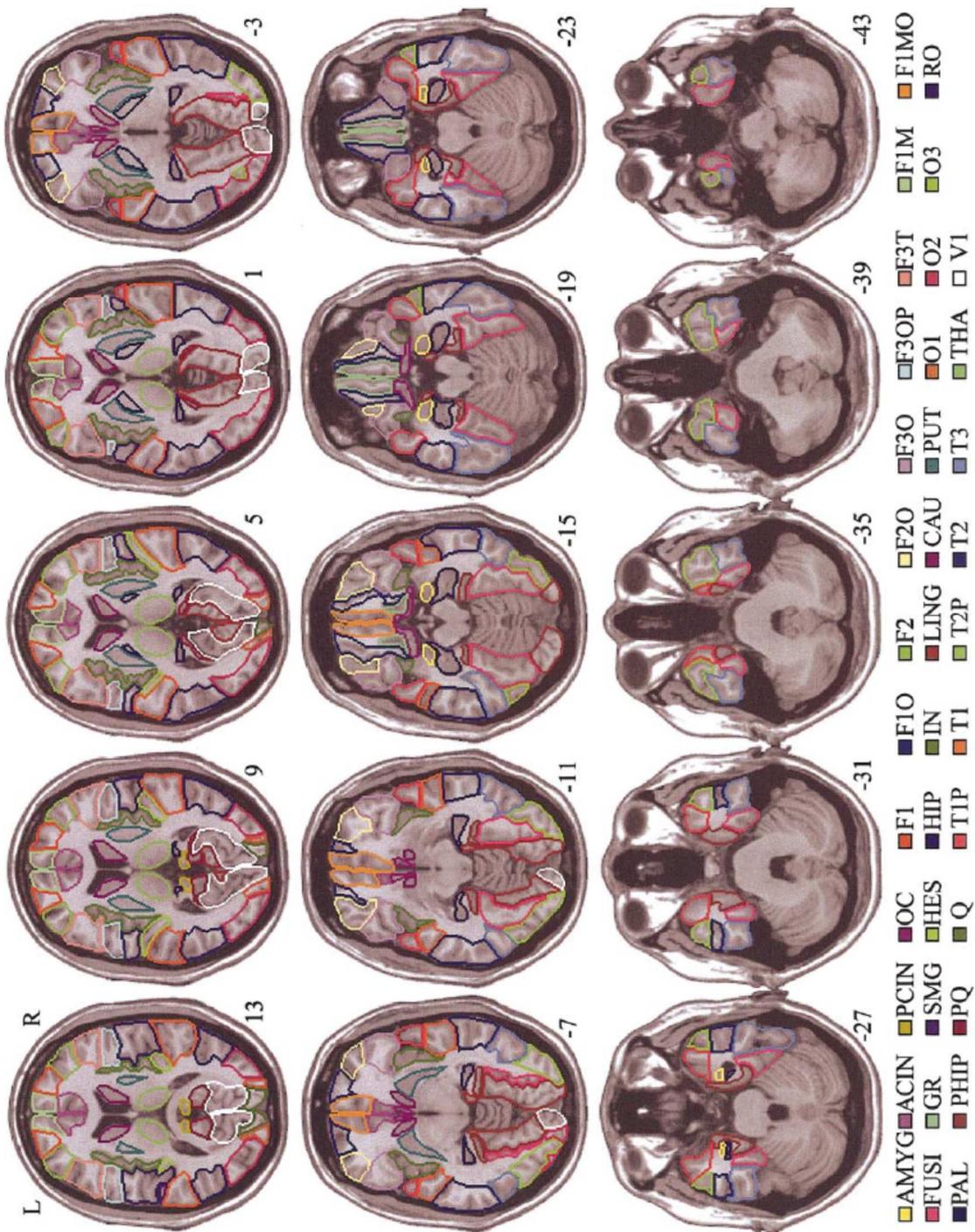


Figura II.I – ROI's do AAL no plano em axial. Valores no canto inferior direito de cada fatia indica a coordenada estereotáxica z em milímetros ($z=19$ mm a $z=75$ mm) [91]

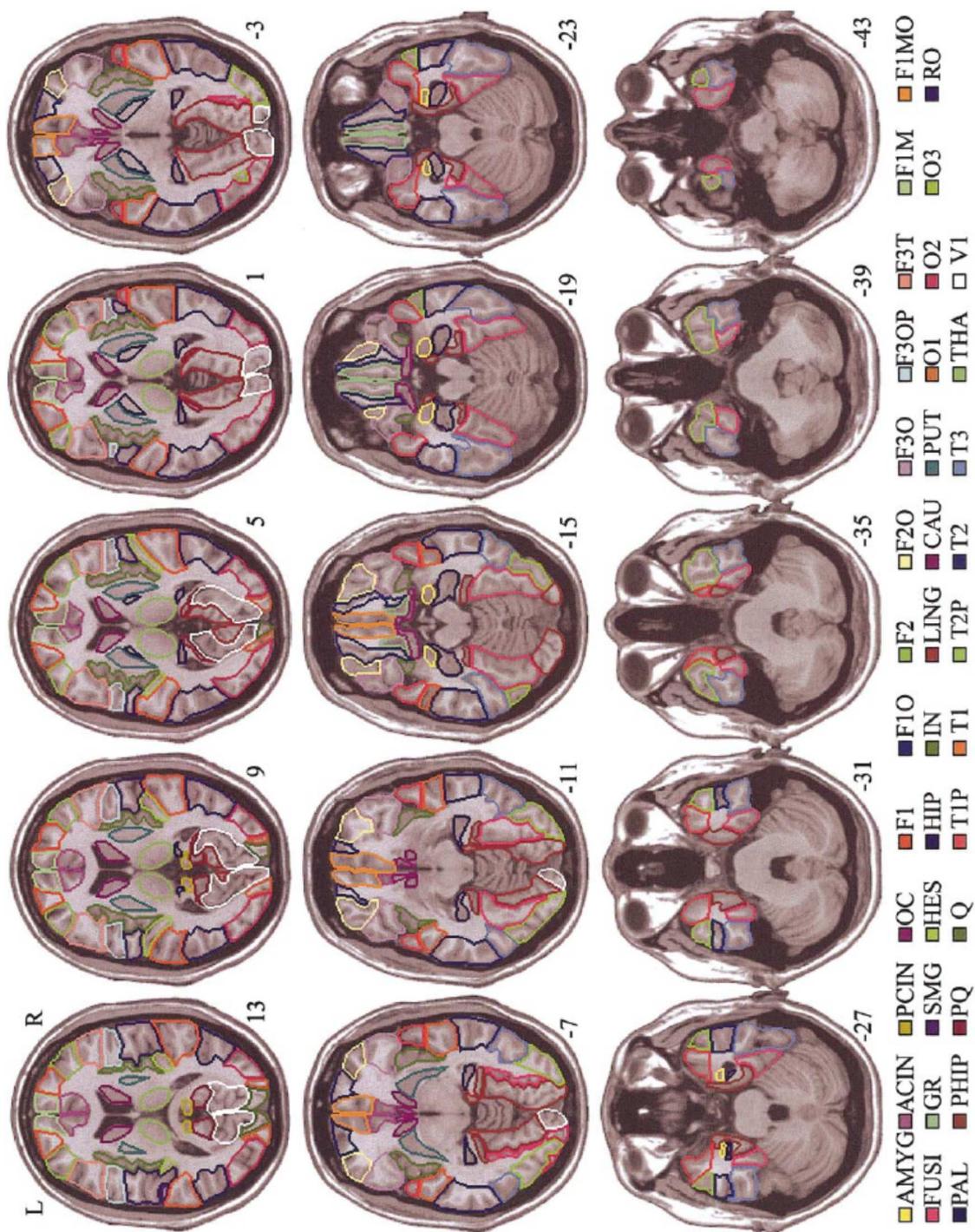


Figura II.II – ROI's do AAL no plano em axial. Valores no canto inferior direito de cada fatia indica a coordenada estereotática z em milímetros ($z=-43$ mm a $z=13$ mm) [91]

Anatomical description	Label
Central region	
Precentral gyrus	PRE
Postcentral gyrus	POST
Rolandic operculum	RO
Frontal lobe	
Lateral surface	
Superior frontal gyrus, dorsolateral	F1
Middle frontal gyrus	F2
Inferior frontal gyrus, opercular part	F3OP
Inferior frontal gyrus, triangular part	F3T
Medial surface	
Superior frontal gyrus, medial	F1M
Supplementary motor area	SMA
Paracentral lobule	PCL
Orbital surface	
Superior frontal gyrus, orbital part	F1O
Superior frontal gyrus, medial orbital	F1MO
Middle frontal gyrus, orbital part	F2O
Inferior frontal gyrus, orbital part	F3O
Gyrus rectus	GR
Olfactory cortex	OC
Temporal lobe	
Lateral surface	
Superior temporal gyrus	T1
Heschl gyrus	HES
Middle temporal gyrus	T2
Inferior temporal gyrus	T3
Parietal lobe	
Lateral surface	
Superior parietal gyrus	P1
Inferior parietal, but supramarginal and angular gyri	P2
Angular gyrus	AG
Supramarginal gyrus	SMG
Medial surface	
Precuneus	PQ
Occipital lobe	
Lateral surface	
Superior occipital gyrus	O1
Middle occipital gyrus	O2
Inferior occipital gyrus	O3
Medial and inferior surfaces	
Cuneus	Q
Calcarine fissure and surrounding cortex	V1
Lingual gyrus	LING
Fusiform gyrus	FUSI
Limbic lobe	
Temporal pole: superior temporal gyrus	T1P
Temporal pole: middle temporal gyrus	T2P
Anterior cingulate and paracingulate gyri	ACIN
Median cingulate and paracingulate gyri	MCIN
Posterior cingulate gyrus	PCIN
Hippocampus	HIP
Parahippocampal gyrus	PHIP
Insula	IN
Sub cortical gray nuclei	
Amygdala	AMYG
Caudate nucleus	CAU
Lenticular nucleus, putamen	PUT
Lenticular nucleus, pallidum	PAL
Thalamus	THA

Tabela II.I – Lista de *labels* das ROI's do atlas AAL [91]