

**POLITECNICO DI MILANO**

**Scuola di Ingegneria Industriale e dell'Informazione**

**Laurea in Computer Science and Engineering**



**REALTÀ VIRTUALE PER LA VISUALIZZAZIONE  
INTERATTIVA DI RICOSTRUZIONI TRIDIMENSIONALI  
DA TOMOGRAFIA AD EMISSIONE DI POSITRONI**

**Relatore: Prof. Pierluca Lanzi**

**Co-Relatrice: Ing. Erica Stella**

**Tesi di:**

**Federico Reale**

**Matricola: 899299**

**Anno Accademico 2019-2020**

# Ringraziamenti

Ed eccomi qui, giunto ad un traguardo importante della mia vita, posto alla fine di un viaggio iniziato al termine delle superiori, che mi ha regalato le più svariate emozioni.

Arrivato alla fine di questo percorso vorrei ringraziare il Professor Lanzi, che mi ha mostrato come la professionalità e il rigore, possano andare di pari passo con la disponibilità, la serenità ed il divertimento nel campo lavorativo.

Ringrazio anche Erica, mia correlatrice. Con i tuoi consigli hai dato quel tocco di esperienza in più, che da solo non avrei potuto avere.

Vorrei ringraziare il Politecnico di Milano, scuola di vita che mi ha insegnato e dato tanto e alla quale sarò eternamente grato.

Un grazie speciale va a tutta la mia famiglia: a mia mamma, a mio papà, a mia sorella, ai miei nonni e a tutti gli altri parenti. Ci siete sempre stati in questi anni, dimostrandomi affetto, vicinanza ed aiuto nei momenti in cui ve l'ho chiesto.

Ringrazio anche tutti coloro che sono stati miei compagni di corso in questi anni di università: una parte di questo è anche vostra, per me avete rappresentato uno stimolo a non arrendermi e a cercare di dare sempre il massimo; assieme ci siamo divertiti, abbiamo sofferto, abbiamo gioito ed abbiamo passato momenti memorabili.

Ringrazio anche coloro che fanno e hanno fatto parte della mia vita in questi anni: dagli amici del calcio presenti e passati, ai compagni del Real Birretta, agli ex compagni delle superiori, all'oratorio, a Mombello e a tutti quelli con cui ho condiviso una parte del mio cammino: ognuno di voi nel suo piccolo mi ha aiutato ad arrivare al raggiungimento di questo obiettivo. Ed infine ci siete voi, amici di sempre, uno dei pilastri fondamentali della mia vita. Con voi ho condiviso praticamente tutto, i momenti tristi, quelli felici, gli interminabili viaggi in auto per andare in vacanza, le vacanze indimenticabili, le notti a Vararo, le grigliate e pure le semplici giornate passate in compagnia. Siete stati e sarete sempre il motore immobile, che mi da energia e mi permette di andare avanti. Semplicemente GRAZIE!



# Abstract

Computed tomography is a radiodiagnostic investigation technique which, by means of a computer-generated analysis of the attenuation of an X-ray beam passing through a body section, is capable of reproducing sectional and three-dimensional images. The introduction of the DICOM format (Digital Imaging and COmmunication in Medicine), the standard format for saving these images, has made it possible to obtain examinable and transferable files containing medical information, giving birth to the Medical Imaging.

The term *Medical Imaging* refers to the generic process through which it is possible to create a visual representation of the function of some organs or tissues and to dare to examine the areas of an organism not visible from the outside. It tries to reveal the internal structures hidden by the skin and bones, in addition to diagnosing and treating diseases. Moreover, it also establishes a database of the normal anatomy and physiology of an organism, to allow the identification of anomalies. The visualization of medical images therefore represents a fundamental weapon, whose use can be effective in the field of research, diagnosis and medical planning.

Today, with modern technologies, we are able not only to access simple two-dimensional images, but we are capable of obtaining three-dimensional volumetric reconstructions, which allow us to have new and different approaches for observing the organism.

If we combine this with the important advances made by virtual reality in the last decade (advances that have made this technology easily accessible and available to the public), we can now transport the visualization on a new front: Virtual Reality Medical Imaging.

The purpose of this paper is to present a prototype, also available for medium/low range viewers, which, starting from a series of computed tomography (CT) or a Positron Emission Tomography/Computed Tomography (PET/CT) images in DICOM format relating to a single anatomical plane, allows, in virtual reality, the visualization of the images deriving from the reconstruction of the three main anatomical planes (axial, sagittal and coronal) and

the examination of the 3D model obtained from the same series of files.

The test results showed good performances, in relation to short-term uses, on low-end viewers such as Oculus GO, envisaging an optimal use of the software on mid-range all-in-one viewers such as Oculus Quest, which would allow medical visualization in virtual reality to be more easily accessible, both in the medical field and in private use.



# Sommario

La tomografia computerizzata è una tecnica di indagine radiodiagnostica che, mediante un'analisi generata a computer dell'attenuazione di un fascio di raggi X passanti attraverso una sezione corporea, è in grado di riprodurre le immagini in sezione e quelle tridimensionali. L'introduzione del formato DICOM (Digital Imaging and COmmunication in Medicine), il formato standard per il salvataggio di queste immagini, ha permesso di ottenere dei file esaminabili e trasferibili contenenti informazioni mediche, dando così i natali al Medical Imaging. Con il termine *Medical Imaging* ci si riferisce al generico processo attraverso il quale è possibile creare una rappresentazione visiva della funzione di alcuni organi o tessuti e dare la possibilità di esaminare aree di un organismo non visibili dall'esterno. Esso cerca di rivelare le strutture interne nascoste dalla pelle e dalle ossa, oltre a diagnosticare e curare le malattie. Inoltre, istituisce anche un database della normale anatomia e fisiologia di un organismo, per consentire l'identificazione di anomalie. La visualizzazione di immagini mediche rappresenta quindi una fondamentale arma, il cui impiego può risultare efficace nel campo della ricerca, della diagnosi e della pianificazione medica.

Oggi, con le moderne tecnologie, siamo in grado non solo di accedere alle semplici immagini bidimensionali, bensì siamo capaci di ottenere ricostruzioni volumetriche tridimensionali, che permettono di avere nuovi e diversi approcci per l'osservazione dell'organismo.

Se uniamo questo, agli importanti progressi fatti dalla realtà virtuale nell'ultima decade (progressi che hanno reso questa tecnologia di facile accesso e reperibilità al pubblico), possiamo ora trasportare la visualizzazione su un nuovo fronte: il Virtual Reality Medical Imaging.

Lo scopo di questo elaborato è quello di presentare un prototipo, disponibile anche per visori di fascia medio/bassa, che, partendo da una serie di immagini di tomografia computerizzata (CT) o di tomografia computerizzata ad emissione di positroni (PET/CT) in formato DICOM relative ad un solo piano anatomico, permetta, in realtà virtuale, la visualizzazione delle immagini derivanti dalla ricostruzione dei tre principali piani anatomici (assiale, sagit-

tale e coronale) e l'esaminazione del modello 3D ricavato sempre dalla stessa serie di file.

I risultati dei test hanno mostrato una buona resa, relativamente ad utilizzi di breve durata, su visori di fascia economica come Oculus GO, prospettando quindi un'ottimale impiego del software su visori all-in-one di fascia media come Oculus Quest, cosa che permetterebbe alla visualizzazione medica in realtà virtuale di essere più facilmente accessibile, sia in campo medico sia in uso privato.





# Indice

<b>Ringraziamenti</b>	<b>II</b>
<b>Sommario</b>	<b>VII</b>
<b>Elenco delle figure</b>	<b>XII</b>
<b>Lista delle abbreviazioni</b>	<b>XIV</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 La tomografia computerizzata . . . . .	1
1.2 La PET/CT . . . . .	5
1.3 Il progetto di tesi . . . . .	6
1.4 Struttura della tesi . . . . .	7
<b>2 Stato dell'arte</b>	<b>9</b>
2.1 Lo standard DICOM . . . . .	9
2.1.1 L'estensione .dcm . . . . .	9
2.1.2 I Value Representation . . . . .	10
2.1.3 I DICOM tags . . . . .	12
2.2 La realtà virtuale . . . . .	18
2.2.1 Che cos'è la realtà virtuale . . . . .	18
2.2.2 Evoluzione della RV negli anni . . . . .	19
2.2.3 Effetti della RV sulla salute . . . . .	24
2.2.4 Le applicazioni della RV in ambito medico . . . . .	26
2.3 I visori Oculus . . . . .	28
2.4 Alcuni software esistenti per la visualizzazione di immagini di TC in formato DICOM . . . . .	30

<b>3</b>	<b>Materiali e metodi</b>	<b>36</b>
3.1	Pydicom, Numpy e la ricostruzione multiplanare . . . . .	36
3.2	DicomToMesh e l'algoritmo Marching Cubes . . . . .	38
3.3	Unity . . . . .	41
3.3.1	AssetBundle e Batchmode . . . . .	42
3.3.2	Coroutines e Multithreading . . . . .	42
3.3.3	Lightweight Rendering Pipeline . . . . .	45
3.3.4	Baked Lights . . . . .	49
3.3.5	Oculus Integration . . . . .	49
3.3.6	Cross Section Shader Graph . . . . .	55
3.4	Fellow Oak Dicom . . . . .	56
<b>4</b>	<b>Scenario Applicativo</b>	<b>58</b>
4.1	Preparazione dei dati . . . . .	58
4.2	Il visualizzatore . . . . .	63
4.3	Discussione . . . . .	69
<b>5</b>	<b>Conclusioni e Sviluppi futuri</b>	<b>73</b>
	<b>Bibliografia</b>	<b>75</b>

# Elenco delle figure

1.1	Macchinario per eseguire esami a TC . . . . .	2
1.2	Emissione dei raggi X dal tubo radiogeno . . . . .	2
1.3	Le 4 generazioni di CT a confronto . . . . .	3
1.4	I valori di alcuni tessuti e sostanze nella HS . . . . .	3
1.5	Due esempi di finesra . . . . .	4
1.6	Applicazione del windowing sulla stessa immagine . . . . .	4
1.7	Le due immagini rappresentate degli scan PET/CT. Si può notare come il radiofarmaco reagisca con le cellule tumorali presenti nel polmone destro, marcando in modo ben definito la zona affetta da carcinoma . . . . .	6
1.8	I tre piani anatomici . . . . .	7
2.1	Lo stereoscopio di Charles Wheatstone . . . . .	20
2.2	Lo stereoscopio di David Brewster . . . . .	20
2.3	L'illusione Haunted swing . . . . .	20
2.4	Gli occhiali stereoscopici per televisione . . . . .	22
2.5	Sensorama . . . . .	22
2.6	Il primo HMD . . . . .	22
2.7	Sistema "Sword Of Damocle" . . . . .	22
2.8	Il sistema VIVED . . . . .	23
2.9	I gradi di libertà . . . . .	29
2.10	I visori della famiglia Oculus . . . . .	30
2.11	RadiAnt DICOM Viewer . . . . .	31
2.12	3D Slicer . . . . .	32
2.13	Dicom Viewer XR . . . . .	33
2.14	Dicom VR . . . . .	34

2.15 ChimeraX . . . . .	35
3.1 La ricostruzione volumetrica . . . . .	38
3.2 La ricostruzione multiplanare . . . . .	38
3.3 Marching Cubes Algorithm 1 . . . . .	40
3.4 Marching Cubes Algorithm 2 . . . . .	40
3.5 Marching Cubes Algorithm 3 . . . . .	41
3.6 Ciclo di vita degli script in Unity . . . . .	44
3.7 Diagramma della una rendering pipeline . . . . .	46
3.8 Il mapping dei controller Oculus . . . . .	54
3.9 Cross Section Shader Graph . . . . .	55
3.10 Cross Section Shader Graph 2 . . . . .	56
4.1 L'applicativo python 1 . . . . .	59
4.2 L'applicativo python 2 . . . . .	60
4.3 L'applicativo python 3 . . . . .	61
4.4 L'applicativo python 4 . . . . .	62
4.5 Il Dicom Viewer 1 . . . . .	64
4.6 Il Dicom Viewer 2 . . . . .	65
4.7 Il Dicom Viewer 3 . . . . .	66
4.8 Il Dicom Viewer 4 . . . . .	68

# Lista delle abbreviazioni

AB	AssetBundle
AE	AfterEffects
BL	Baked Lights
CAT	Computed Axial Tomography
CPU	Central Processing Unit
CRT	Coroutine
CT	Computed Tomography
DICOM	Digital Imaging and COmmunications in Medicine
DOF	Degree Of Freedom
FOV	Field Of View
FRP	Forward Rendering Path
GDL	Gradi Di Libertà
GPU	Graphic Processing Unit
HMD	Head Mounted Display
HS	Hounsfield Scale
HU	Hounsfield Unit
LZMA	Lempel-Ziv-Markov Algorithm
MR	Magnetic Resonance

MRI	Magnetic Resonance Imaging
PET	Positron Emission Tomography
RA	Realtà Aumentata
RM	Realtà Mista
RV	Realtà Virtuale
TAC	Tomografia Assiale Computerizzata
TC	Tomografia Computerizzata
UI	User Interface
UWP	Universal Windows Platform
VM	Value Multiplicity
VR	Value Representation
WC	Window Center
WL	Window Level
WW	Window Width
XR	X Reality (o Cross Reality)





# Capitolo 1

## Introduzione

### 1.1 La tomografia computerizzata

In radiologia, la tomografia computerizzata è una tecnica di indagine radiodiagnostica che, mediante un'analisi generata a computer dell'attenuazione di un fascio di raggi X passanti attraverso una sezione corporea, è in grado di riprodurre le immagini in sezione e quelle tridimensionali.

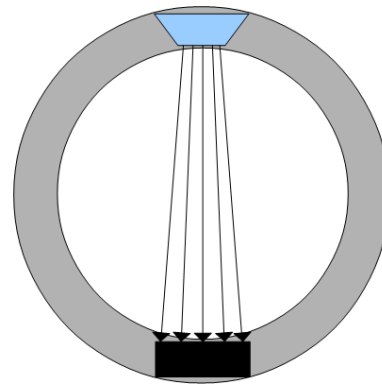
Grazie ad essa siamo in grado di controllare ogni parte del corpo (encefalo, polmoni, fegato, pancreas, reni, utero, vasi arteriosi e venosi, muscoli, ossa e articolazioni) per la diagnosi e lo studio di tumori e di numerose altre patologie.

Ha inoltre rappresentato il primo metodo per acquisire, in modo non invasivo, le immagini dell'interno del corpo umano, non distorte dalla sovrapposizione di strutture anatomiche distinte [28]. Pur condividendone il mezzo di trasmissione, questo tipo di esame si differenzia dalle comuni radiografie, infatti produce immagini di contrasto molto più elevato. Durante gli anni '70, fu un enorme passo verso l'avanzamento delle possibilità diagnostiche in medicina [29]. Per via di questa differenza, ora è possibile visualizzare all'interno della stessa immagine non solo le ossa, ma anche tutti i cosiddetti "soft-tissue" (tessuti molli), tra i quali ad esempio fegato, e polmoni.

Inizialmente questo studio veniva condotto lungo un solo asse, quello assiale, per questo tale procedura in principio prendeva il nome di TAC (o CAT); ai giorni d'oggi tale acronimo è caduto in disuso per via dei moderni macchinari multistrato, i quali, grazie ad una tecnica di acquisizione delle immagini a spirale, sono in grado di ricostruire perfino modelli tridimensionali.



**Fig. 1.1:** *Macchinario per eseguire esami a TC*



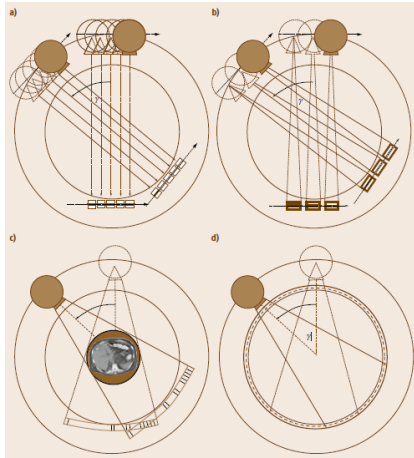
**Fig. 1.2:** *Emissione dei raggi X dal tubo radiogeno*

Il procedimento di acquisizione dei dati avviene tramite il macchinario mostrato in Figura 1.1: inizialmente il paziente si trova sdraiato sul lettino, che si muove orizzontalmente e passa attraverso l'apertura circolare di spessore non superiore a 50 cm, condizione che evita al paziente reazioni di claustrofobia. Durante lo spostamento del soggetto, il tubo radiogeno ruota attorno ad esso emettendo raggi X (Figura 1.2), i quali attraversano le superfici corporee, fino ad arrivare ad un gruppo di sensori situato in posizione diametralmente opposta alla sorgente emittente.

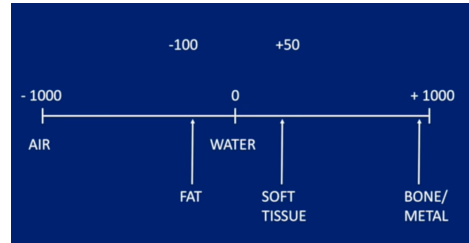
In questo tragitto i raggi incontrano superfici, tessuti e altre sostanze di diversa densità, ne rilevano le varie intensità e inviano il tutto al centro computazionale, che assegna a quella specifica zona il valore riscontrato.

Le varie generazioni di CT hanno portato a varie modifiche e migliorie legate alla composizione, al procedimento ed al movimento delle onde elettromagnetiche all'interno del tubo: si è partiti dall'avere un unico fascio, che puntava ad un unico sensore in grado di spostarsi solo linearmente e con il quale bisognava ripetere la procedura di acquisizione a diverse angolazioni, fino ad avere raggi a fascio di cono ed un intero anello di sensori.

Godfrey Newbold Hounsfield e Allan Cormack, entrambi premi Nobel per la medicina nel 1979, sono quelli che in principio hanno contribuito alla nascita e allo sviluppo di questa tecnica; il loro approccio è tutt'ora utilizzato nelle moderne apparecchiature per la TC. Il loro principale lavoro era focalizzato sui valori di attenuazione registrati dal fascio di raggi X: ogni  $\mu$  ottenuto veniva posizionato all'interno di una scala adimensionale, secondo la seguente



**Fig. 1.3:** Le 4 generazioni di CT a confronto



**Fig. 1.4:** I valori di alcuni tessuti e sostanze nella HS

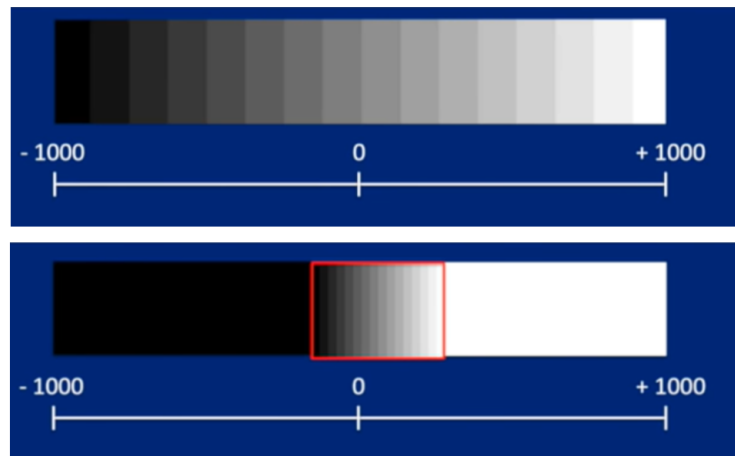
formula:

$$CT_{value} = \frac{\mu - \mu_{water}}{\mu_{water}} * 1000$$

questa scala prende il nome direttamente dal primo dei due scienziati e viene rappresentata con l'acronimo HU.

La HS gioca un ruolo importante nei software dedicati alla visualizzazione di immagini o alla ricostruzione di modelli 3D derivanti da TC. Per mostrare a video uno scan ci si basa principalmente su due fattori: il tipo di scala (HS nel nostro caso) e la finestra (più comunemente detta window). La finestra ha due caratteristiche principali, l'ampiezza (detta WW) e il centro (detto WL o WC). Prendiamo per esempio la HS riportata in Figura 1.4: una WW di 2000 e un WL pari a 0, stanno a significare che ad ogni valore di radiodensità, da -1000 a 1000, sarà assegnato un colore appartenente alla scala dei grigi, partendo dal nero, che rappresenta un materiale di bassa densità, fino al bianco per quelli di intensità alta; tutto ciò che sarà fuori scala verrà rappresentato in bianco, `rgb(255,255,255)`, se superiore a 1000, in nero, `rgb(0,0,0)`, se inferiore a -1000.

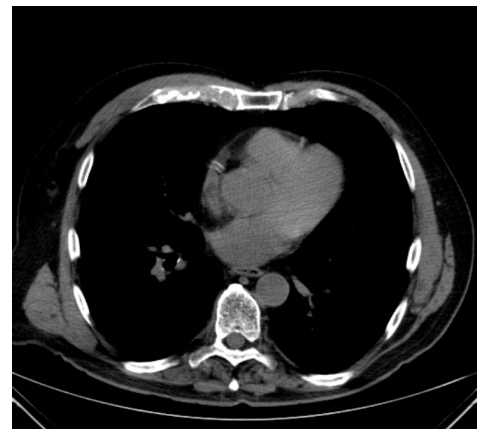
Giocando con questi elementi esiste la possibilità di "modificare" l'immagine da osservare, per concentrarsi di più su particolari aspetti. Questo procedimento prende il nome di windowing e consiste nel variare i valori WW e WL, così facendo si riescono a mettere in risalto diversi tessuti all'interno della stessa foto. Questo risulta molto utile ai fini delle diagnosi, infatti con un solo esame è possibile dare attenzione a tutti i diversi tessuti del corpo umano, potendo avere una migliore e più dettagliata visione del problema ricercato, da diversi punti di vista.



**Fig. 1.5:** Due esempi di finestra, la prima con  $WW=2000$  e  $WL=0$ , la seconda invece con  $WW=400$  e  $WL=50$ ; si può notare come tutti i valori minori di  $-150$  (che nella precedente window venivano mostrati con diversi colori), saranno tutti rappresentati di nero, la stessa cosa accade con il bianco per i valori maggiori di  $250$ . I valori compresi tra  $-150$  e  $250$  invece, avranno diverse sfumature e saranno più facilmente distinguibili (cosa più difficile invece nella window precedente)



(a)  $WW=1500$ ,  $WL=-400$



(b)  $WW=360$ ,  $WL=70$

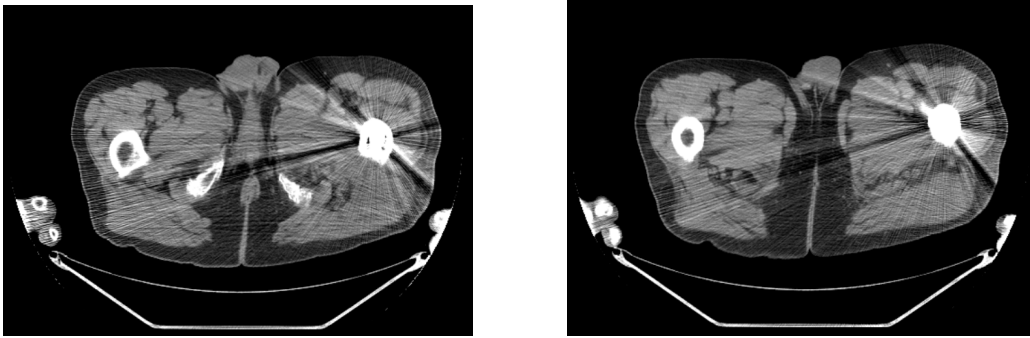
**Fig. 1.6:** Applicazione del windowing sulla stessa immagine

## 1.2 La PET/CT

La PET/CT è una metodica di diagnostica per immagini, applicata in numerosi campi di medicina, ma con principale utilizzo in oncologia. Questa tecnica permette uno studio approfondito di diverse tipologie tumorali [22]. Questo esame, oltre alla potenza espressiva della TC, sfrutta anche i vantaggi della PET.

La PET è una procedura che prevede l'impiego di un radiofarmaco debolmente radioattivo iniettato per via endovenosa. Tale sostanza andrà a reagire con le cellule tumorali, agendo come un segnalatore, evidenziando la zona colpita e svelando altre informazioni importanti sull'attività metabolica del tumore in studio (Figura 1.7). Diverse tipologie di tumore necessitano di diversi radiofarmaci. Quelli attualmente in uso sono:

- **18F-FDG** (FluoroDesossiGlucosio): il suo comportamento simile al glucosio (zucchero) lo rende perfetto per l'analisi delle neoplasie. La maggior parte di esse, infatti, utilizza una grande quantità di zuccheri per mantenersi attiva, perciò l'FDG tenderà ad accumularsi intensamente all'interno della zona tumorale, permettendo così la visualizzazione nelle immagini PET/CT e lo studio del metabolismo glucidico del carcinoma. Questo farmaco è il più utilizzato in questo ambito ed è reperibile in tutti i centri di medicina nucleare che dispongono di apparecchiature per effettuare PET/CT. È inoltre usato anche in ambito non oncologico, per esempio nello studio di infezioni o di vasculiti.
- **11C-Colina**: è un radiofarmaco che necessita di essere prodotto "in loco", per cui è disponibile solo in alcuni centri di medicina nucleare provvisti di ciclotrone. Viene utilizzato principalmente per lo studio della neoplasia prostatica.
- **68Ga-DOTATOC**: è un radiofarmaco utilizzato per lo studio dei tumori neuroendocrini. Questo tracciante, analogo di somatostatina, ha la caratteristica di legarsi a dei recettori presenti sulla superficie delle cellule del tumore, permettendo la sua visualizzazione.
- **18F-DOPA**: è un radiofarmaco usato per la diagnostica di alcune forme tumorali che metabolizzano la dopamina.



**Fig. 1.7:** Le due immagini rappresentate degli scan PET/CT. Si può notare come il radiofarmaco reagisca con le cellule tumorali presenti nel polmone destro, marcando in modo ben definito la zona affetta da carcinoma

### 1.3 Il progetto di tesi

DICOM<sup>®</sup> (Digital Imaging and COmmunications in Medicine) è lo standard internazionale per immagini mediche ed informazioni correlate. Permette lo scambio di dati medici, conservando per le informazioni trasmesse, tutte le qualità necessarie per l'uso clinico.

DICOM<sup>®</sup> è implementato in quasi tutti i dispositivi di radiologia, imaging cardiologico e radioterapia (raggi X, TC, risonanza magnetica, ultrasuoni, ecc.) e sempre più in uso anche in dispositivi di altri settori medici come l'oftalmologia e l'odontoiatria. Con centinaia di migliaia di dispositivi di imaging medicale in uso, DICOM<sup>®</sup> è uno degli standard di messaggistica sanitaria più diffusi al mondo. Ci sono letteralmente miliardi di immagini DICOM<sup>®</sup> attualmente in uso per le cure cliniche.

Dalla sua prima pubblicazione nel 1993, DICOM<sup>®</sup> ha rivoluzionato la pratica della radiologia, consentendo la sostituzione della pellicola radiografica con un flusso di lavoro completamente digitale. Proprio come Internet è diventata la piattaforma usata dalle nuove applicazioni di informazione in ambito medico; ha inoltre fatto nascere avanzate app di imaging medico che "hanno cambiato il volto della medicina clinica". Dal pronto soccorso, allo stress cardiaco, alla rilevazione del carcinoma mammario, DICOM<sup>®</sup> è lo standard che fa funzionare l'imaging medico per i medici e per i pazienti.

È riconosciuto dall'International Organization for Standardization come lo standard ISO 12052 [18].

Tale standard è stato sviluppato per venire incontro all'esigenza di produttori e utilizzatori nella realizzazione di interconnessioni tra apparecchiature diagnostiche differenti, attraverso reti standardizzate [21].

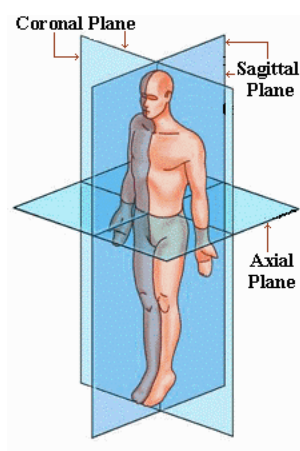
Ai giorni d'oggi sono presenti differenti software che permettono la visualizzazione di immagini mediche in formato DICOM, ma la caratteristica che li accomuna quasi tutti è la restrizione a due dimensioni. È comunque possibile la visualizzazione di un modello tridimensionale, ma essa è confinata in uno schermo di due dimensioni.

Al momento in commercio, sono praticamente assenti applicazioni che permettono l'osservazione di volumi DICOM di due o tre dimensioni, in ambienti virtuali tridimensionali. I pochi visualizzatori di RV esistenti, legati all'ambito DICOM, o sono progetti in via di sviluppo, oppure sono correlati a dispositivi di fascia alta, che oltre agli elevati costi, necessitano pure di supporti computazionali aggiuntivi con particolari specifiche tecniche.

Questo lavoro di tesi ha come scopo quello di creare un prototipo di visualizzatore in RV di immagini 2D e volumi 3D provenienti da file DICOM, che sia disponibile anche per visori di fascia inferiore, permettendo così di aprire nuove possibilità e direzioni nel campo della visualizzazione medica.

## 1.4 Struttura della tesi

Il lavoro di tesi ha avuto come scopo quello della creazione, nell'ambito della realtà virtuale, di un visualizzatore per i risultati della TC o della PET/CT che, partendo da una serie di immagini in formato DICOM (tutte relative allo stesso piano anatomico), dia la possibilità di poter osservare gli scan 2D del corpo dai tre diversi punti di vista (piano assiale, piano sagittale e piano coronale) e mostri la ricostruzione tridimensionale della parte esaminata del soggetto.



**Fig. 1.8:** I tre piani anatomici

---

Lo sviluppo è stato suddiviso in due parti. La prima parte è consistita nella creazione di un applicativo scritto in python che, partendo dalle immagini DICOM: ricostruisce il modello 3D, gli scan dei tre diversi piani e carica tutto su di un server. La seconda parte ha riguardato la realizzazione in unity di un app per la RV (specifica per visori della famiglia Oculus) che, scaricando i dati dal server, ne permette la visualizzazione. L'elaborato è organizzato in sei capitoli. Questo capitolo è stato dedicato alla presentazione e alla descrizione in termini generali della TC e della PET/CT, ciò che permette a questa tesi di poter essere messa in atto. Nel capitolo 2 vengono presentati lo standard DICOM (il formato standard internazionale per trasmettere, archiviare, recuperare, stampare, elaborare e visualizzare informazioni di imaging medico), la realtà virtuale ed alcuni software per la visualizzazione di immagini mediche (sia di RV che non) già esistenti. Il capitolo 3 si concentra sulle librerie, i framework e le tecniche utilizzate per lo sviluppo del lavoro in oggetto. Nel capitolo 4 si trovano i risultati finali del lavoro, mostrati attraverso alcune schermate. Il capitolo 5 tratta la discussione relativa ai test di usabilità fatti. Nel sesto ed ultimo capitolo infine, sono riportate le considerazioni conclusive ed i possibili sviluppi futuri.



## Capitolo 2

# Stato dell'arte

### 2.1 Lo standard DICOM

Siccome questo standard è in continua evoluzione ed espansione, in questa sezione verrà presentata la versione in vigore durante il periodo di sviluppo della tesi (versione copyright © 2020 NEMA) e saranno trattati solamente gli aspetti che sono risultati necessari e direttamente collegati al progetto oggetto di questo elaborato.

Poiché la parte iniziale del lavoro partiva da una serie di file salvati in formato .dcm (formato DICOM), l'aspetto principale con il quale è stato necessario fronteggiarsi è stato: com'è organizzata la struttura di un file DICOM?

#### 2.1.1 L'estensione .dcm

Il suffisso .dcm rappresenta l'estensione per file contenenti dati organizzati secondo lo standard DICOM.

Il file DICOM è formato da un *dataset* contenente una sequenza di *data element* che racchiudono le informazioni inserite all'interno del file. Ogni *data element* è formato da diversi campi che lo caratterizzano: TAG, VR, VM, VL, Description/Name VF.

- TAG (etichetta): identifica univocamente il data element come (Gruppo, Elemento all'interno del gruppo);
- VR, Value Representation: descrive come rappresentare il valore che segue;
- VM, Value Multiplicity: descrive quanti valori sono codificati da questo elemento;

- VL, Value Length: lunghezza del campo contenente il valore che segue;
- Description/Name: contiene una breve descrizione del dato contenuto;
- VF, Value Field: campo che contiene il valore da trasmettere.

Quindi un'immagine o un segnale è contraddistinto da un TAG specifico ed il contenuto è inserito all'interno del corrispettivo value field. Vediamo ad esempio un estratto da un file DICOM:

- (0010,0040) CS 1 2 Patient's Sex m
- (0010,1010) AS 1 4 Patient's Age 018Y
- (0010,1020) DS 1 4 Patient's Size 1.78
- (0010,1030) DS 1 2 Patient's Weight 78

Come si vede, un file DICOM è composto da molti elementi e non solo dall'immagine o dal segnale biomedico.

Dal sito ufficiale sono consultabili le tabelle contenenti tutti i VR e tutti i tag, rispettivamente agli indirizzi [http://dicom.nema.org/dicom/2013/output/chtml/part05/sect\\_6.2.html](http://dicom.nema.org/dicom/2013/output/chtml/part05/sect_6.2.html) e [http://dicom.nema.org/medical/dicom/current/output/html/part06.html#chapter\\_6](http://dicom.nema.org/medical/dicom/current/output/html/part06.html#chapter_6); nei successivi paragrafi verranno elencati e descritti solamente i VR e i tag che sono risultati utili ai fini della visualizzazione o per la ricostruzione multiplanare.

### 2.1.2 I Value Representation

- **AS (Age String):** Una stringa di caratteri con uno dei seguenti formati: nnnD, nnnW, nnnM, nnnY; dove nnn deve contenere il numero di giorni per D, settimane per W, mesi per M o anni per Y. Esempio: "018M" rappresenterebbe un'età di 18 mesi.
- **CS (Code String):** Una stringa di caratteri con spazi iniziali o finali (20H) non significativi.
- **DA (Date):** Una stringa di caratteri nel formato AAAAMMGG; dove AAAA deve contenere l'anno, MM deve contenere il mese e GG deve contenere il giorno,

interpretato come una data del sistema di calendario gregoriano. Esempio: "19930822" rappresenta il 22 Agosto 1993.

- **DS (Decimal String):** Una stringa di caratteri che rappresentano un numero in virgola fissa o un numero in virgola mobile. Un numero in virgola fissa deve contenere solo i caratteri 0-9 con un "+" o "-" iniziale facoltativo e un "." facoltativo per segnare il punto decimale. Un numero in virgola mobile deve essere indicato come definito in ANSI X3.9, con una "E" o "e" per indicare l'inizio dell'esponente. Le stringhe decimali possono essere riempite con spazi iniziali o finali. Gli spazi incorporati non sono ammessi.
- **IS (Integer String):** Una stringa di caratteri che rappresenta un numero intero in base 10 (decimale), deve contenere solo i caratteri da 0 a 9, con un "+" o "-" iniziale facoltativo. Può contenere spazi iniziali e/o finali. Gli spazi incorporati non sono ammessi. L'intero  $n$ , rappresentato, deve essere compreso nell'intervallo:  
$$-2^{31} \leq n \leq 2^{31} - 1.$$
- **LO (Long String):** Una stringa di caratteri che può essere riempita con spazi iniziali e/o finali. Il codice carattere 5CH (BACKSLASH "\") in ISO-IR 6) non deve essere presente, poiché viene utilizzato come delimitatore tra valori nei tag aventi VM maggiore di 1. La stringa non deve avere caratteri di controllo ad eccezione di ESC.
- **OB (Other Byte String):** Una stringa di byte in cui la codifica del contenuto è specificata dalla sintassi di trasferimento stabilita. OB è un VR insensibile all'ordinamento dei byte Little/Big Endian (vedere [19]). La stringa di byte deve essere riempita con un singolo valore byte NULL finale (00H) quando necessario per raggiungere una lunghezza pari.
- **OW (Other Word String):** Una stringa di parole a 16 bit in cui la codifica del contenuto è specificata dalla sintassi di trasferimento stabilita. OW è un VR che richiede lo scambio di byte all'interno di ogni parola, quando si cambia tra l'ordinamento di byte Little Endian e Big Endian (vedere [19]).
- **PN (Person Name):** Una stringa di caratteri codificata utilizzando una convenzione a 5 componenti. Il codice carattere 5CH (BACKSLASH "\") in ISO-IR

6) non deve essere presente, poiché viene utilizzato come delimitatore tra valori nei tag aventi VM maggiore di 1. La stringa può essere riempita con spazi finali. Per l'uso ordinario, i cinque componenti nel loro ordine di occorrenza sono: complesso di nomi di famiglia (cognomi), complesso di nomi dato, secondo nome, prefisso di nome, suffisso di nome.

- **SS (Signed Short):** Numero intero binario con segno lungo 16 bit nella forma del complemento di 2. Rappresenta un numero intero  $n$  nell'intervallo:  $-2^{15} \leq n < 2^{15} - 1$ .
- **TM (Time):** Una stringa di caratteri nel formato HHMMSS.FFFFFFFF; dove HH contiene ore (intervallo "00" - "23"), MM contiene minuti (intervallo "00" - "59"), SS contiene secondi (intervallo "00" - "60") e FFFFFFFF contiene una parte frazionaria di un secondo di appena 1 milionesimo di secondo (intervallo "000000" - "999999"). Viene utilizzato un orologio di 24 ore. La mezzanotte deve essere rappresentata solo da "0000" poiché "2400" violerebbe l'intervallo di ore. La stringa può essere riempita con spazi finali. Non sono ammessi spazi iniziali e incorporati. Uno o più dei componenti MM, SS o FFFFFFFF possono essere non specificati a condizione che ogni componente a destra di un componente non specificato sia anch'esso non specificato, il che indica che il valore non è preciso alla precisione di quei componenti mancanti. Il componente FFFFFFFF, se presente, deve contenere da 1 a 6 cifre. Se FFFFFFFF non è specificato, il precedente "." non deve essere incluso. Esempi: "070907.0705" rappresenta 7 ore, 9 minuti e 7.0705 secondi. "1010" rappresenta 10 ore e 10 minuti. "021" non è un valore valido.
- **US (Unsigned Short):** Numero intero binario senza segno lungo 16 bit. Rappresenta il numero intero  $n$  nell'intervallo:  $0 \leq n < 2^{16}$ .

### 2.1.3 I DICOM tags

- **(0008,0005) Specific Character Set:**
  - set di caratteri che espande o sostituisce il set grafico di base. Richiesto se viene utilizzato un set di caratteri espanso o sostitutivo. Vedere [11] per i termini definiti.

- VR: CS (Code String)
- VM: 1-n
- **(0008,0020) Study Date:**
  - contiene la data in cui l'esame è stato effettuato
  - VR: DA (Date)
  - VM: 1
- **(0008,0030) Study Time:**
  - contiene l'orario in cui l'esame è stato effettuato
  - VR: TM (Time)
  - VM: 1
- **(0010,0010) Patient's Name:**
  - contiene, se i file non sono anonimizzati, il nome del paziente su cui l'esame è stato eseguito
  - VR: PN (Person Name)
  - VM: 1
- **(0010,0020) Patient ID:**
  - contiene, se i file non sono anonimizzati, l'identificatore del paziente su cui l'esame è stato eseguito
  - VR: LO (Long String)
  - VM: 1
- **(0010,0040) Patient's Sex:**
  - se presente, contiene il sesso del paziente (M, Male, F, Female, O Other)
  - VR: CS (Code String)
  - VM: 1
- **(0010,1010) Patient's Age:**

- se presente, contiene l'età del paziente
- VR: AS (Age String)
- VM: 1
- **(0018,0050) Slice Thickness:**
  - spessore nominale dello scan, in mm.
  - VR: DS (Decimal String)
  - VM: 1
- **(0018,0088) Spacing Between Slices:**
  - distanza tra le scan, in mm. La distanza tra due sezioni viene misurata da centro a centro.
  - VR: DS (Decimal String)
  - VM: 1
- **(0020,0013) Instance Number:**
  - numero che identifica questa immagine (questo attributo era denominato Image Number nelle versioni precedenti di questo standard)
  - VR: IS (Integer String)
  - VM: 1
- **(0020,0032) Image Position (Patient):**
  - Le coordinate x, y e z dell'angolo in alto a sinistra (centro del primo voxel trasmesso) dell'immagine, in mm. Vedere [13] per ulteriori spiegazioni.
  - VR: DS (Decimal String)
  - VM: 3
- **(0020,1041) Slice Location:**
  - posizione relativa del piano dell'immagine, espressa in mm. Vedere [14] per ulteriori spiegazioni.
  - VR: DS (Decimal String)

- VM: 1
- **(0028,0002) Samples Per Pixel:**
  - numero di campioni (piani) in questa immagine. Vedere [15] per ulteriori spiegazioni.
  - VR: US (Unsigned Short)
  - VM: 1
- **(0028,0004) Photometric Interpretation:**
  - specifica l'interpretazione prevista dei dati pixel. Vedere [16] per ulteriori spiegazioni.
  - VR: CS (Code String)
  - VM: 1
- **(0028,0010) Rows:**
  - indica il numero di righe dell'immagine
  - VR: US (Unsigned Short)
  - VM: 1
- **(0028,0011) Columns:**
  - indica il numero di colonne dell'immagine
  - VR: US (Unsigned Short)
  - VM: 1
- **(0028,0030) Pixel's Spacing:**
  - distanza fisica nel paziente tra il centro di ciascun pixel, specificata da una coppia numerica - spaziatura di riga adiacente in mm (delimitatore) spaziatura di colonna adiacente in mm - Esempio: "1.375/1.345". Vedere [7] per ulteriori spiegazioni.
  - VR: DS (Decimal String)
  - VM: 2

- **(0028,0100) Bits Allocated:**

- numero di bit allocati per ciascun campione di pixel. Ogni campione deve avere lo stesso numero di bit assegnati. I bit assegnati devono essere 1 o multipli di 8. Vedi [20] per ulteriori spiegazioni.
- VR: US (Unsigned Short)
- VM: 1

- **(0028,0101) Bits Stored:**

- numero di bit memorizzati per ciascun campione di pixel. Ogni campione deve avere lo stesso numero di bit memorizzati. Vedi [20] per ulteriori spiegazioni.
- VR: US (Unsigned Short)
- VM: 1

- **(0028,0102) High Bit:**

- bit più significativo per i pixel dei campioni di dati. Ogni campione deve avere lo stesso valore di High Bit. Il valore deve essere uno in meno di quello in Bits Stored. Vedi [20] per ulteriori spiegazioni.
- VR: US (Unsigned Short)
- VM: 1

- **(0028,0103) Pixel Representation:**

- rappresentazione dei dati nei campioni di pixel. Ogni campione deve avere la stessa rappresentazione in pixel. 0000H intero senza segno, 0001H Complemento a 2.
- VR: US (Unsigned Short)
- VM: 1

- **(0028,0120) Pixel Padding Value:**

- valore singolo o un limite (compreso) in un intervallo di valori utilizzati in un'immagine, per passare al formato rettangolare o per segnalare allo sfondo



che può essere soppresso. Vedere [12] per ulteriori spiegazioni. Obbligatorio se è presente Pixel Padding Range Limit (0028,0121) oppure se Pixel Data (7FE0,0010) o Pixel Data Provider URL (0028,7FE0) sono presenti.

- VR: US (Unsigned Short), SS (Signed Short)
- VM: 1

- **(0028,1050) Window Center:**

- centro della finestra per la visualizzazione. Vedere [9] per ulteriori spiegazioni. Richiesto se la sequenza VOI LUT (0028,3010) non è presente. Potrebbe essere presente diversamente.
- VR: DS (Decimal String)
- VM: 1-n

- **(0028,1051) Window Width:**

- larghezza della finestra per la visualizzazione. Vedere [10] per ulteriori spiegazioni. Richiesto se è presente Window Center (0028,1050).
- VR: DS (Decimal String)
- VM: 1-n

- **(0028,1052) Rescale Intercept:**

- il valore  $b$  nella relazione tra i valori memorizzati ( $SV$ ) e le unità di output ( $OU$ ).  $OU = m * SV + b$ . Se il primo valore del tag (0008,0008) è ORIGINAL e il terzo non è LOCALIZER e l'acquisizione CT multi-energia (0018,9361) è assente o NO, le unità di uscita devono essere Unità Hounsfield (HU).
- VR: DS (Decimal String)
- VM: 1

- **(0028,1053) Rescale Slope:**

- $m$  nell'equazione specificata in Rescale Intercept (0028,1052).
- VR: DS (Decimal String)
- VM: 1

- **(0028,1054) Rescale Type:**

- specifica le unità di output di Rescale Slope (0028,1053) e Rescale Intercept (0028,1052). Vedere [8] per i termini definiti e ulteriori spiegazioni. Richiesto se il tipo di riscatto non è HU o Multi-energy CT Acquisition (0018,9361) è YES. Potrebbe essere presente diversamente.
- VR: LO (Long String)
- VM: 1

- **(7FE0,0010) Pixel Data:**

- flusso di dati dei campioni di pixel che compongono l'immagine. Vedere [17] per ulteriori spiegazioni. Richiesto se Pixel Data Provider URL (0028,7FE0) non è presente.
- VR: OB (Other Byte String) o OW (Other Word String)
- VM: 1

## 2.2 La realtà virtuale

### 2.2.1 Che cos'è la realtà virtuale

Il termine realtà virtuale (RV) è comunemente usato dai media popolari per descrivere mondi immaginari che esistono solo nei computer e nelle nostre menti [23].

Una prima definizione la possiamo trovare in [40], dove gli autori fanno un'analisi semantica delle due parole, prese separatamente, definendo *realtà* come: *"lo stato o la qualità di essere reali. Qualcosa che esiste indipendentemente dalle idee che la riguardano. Qualcosa che costituisce un'entità reale o effettiva distinta da quello che è semplicemente apparente."*, mentre *virtuale* come: *"essere nell'essenza o nell'effetto, ma non nei fatti"*.

Rileggendo queste due spiegazioni, però, si può facilmente notare come siano abbastanza contraddittorie l'una con l'altra e risulta quindi difficile pensare che possano far parte di uno stesso concetto. Una più completa descrizione, è presente in [26], dove la RV viene espressa come: *un ambiente artificiale che viene sperimentato attraverso*

*stimoli sensoriali (come immagini e suoni), forniti da un computer e in cui le proprie azioni determinano parzialmente ciò che accade nell'ambiente.*

Da queste frasi si può desumere come l'obiettivo principale della RV, sia quindi quello di cercare di generare digitalmente, quindi in modo astratto, un ambiente che possa essere percepito come reale da un utente esterno che ne viene in contatto.

Pensiamo ad esempio di sederci sul divano del nostro soggiorno e di vedere in televisione un video che mostra una stanza di un museo; ora, per tutta la durata del filmato, saremo in grado di percepire che quello che stiamo vedendo in TV (la stanza del museo), si trova in posto diverso da quello nel quale ci troviamo attualmente (il soggiorno). La RV ha come scopo proprio quello di eliminare questa percezione. Con la strumentazione necessaria per l'emulazione virtuale, saremo in grado quindi, sempre dal divano del nostro soggiorno, di vedere le stesse immagini della stanza del museo, percependo però la sensazione di essere direttamente all'interno della stanza stanza stessa e non da un'altra parte.

L'aspetto che risulta quindi fondamentale in questo ambito, è rappresentato dalla gestione degli stimoli psicofisici ai quali veniamo sottoposti durante un'esperienza in RV. Poiché l'ambiente che stiamo simulando deve essere percepito da noi come reale, anche gli stimoli che riceviamo interagendo con esso devono essere congrui a quelli che noi reputiamo reali (questo concetto verrà trattato con maggiore chiarezza nella sezione 2.2.3).

L'esperienza ideale, dovrebbe essere in grado di far dimenticare totalmente al fruitore di trovarsi fisicamente in un luogo differente da quello che sta percependo, simulando con precisione assoluta tutte le stimolazioni alle quali sta venendo sottoposto.

## **2.2.2 Evoluzione della RV negli anni**

### **Il 1800**

Uno degli strumenti che può essere considerato un precursore per la realtà virtuale, fu lo stereoscopio o stereovisore, inventato nel 1832 da Sir Charles Wheatstone (Figura 2.1). È un dispositivo ottico, dotato di lenti, che, utilizzando specchi angolati a 45° per riflettere le immagini nell'occhio sia dal lato destro che da quello sinistro, permetteva di dare a delle immagini bidimensionali, un'illusione di profondità.

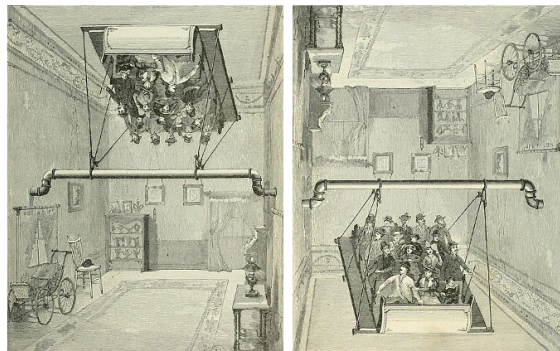
David Brewster, circa nel 1851, riuscì a crearne una versione portatile più piccola ed intuitiva (Figura 2.2), che riscosse molto successo.



**Fig. 2.1:** *Lo stereoscopio di Charles Wheatstone*



**Fig. 2.2:** *Lo stereoscopio di David Brewster*



**Fig. 2.3:** *L'illusione Haunted swing: la prima foto rappresenta la percezione che le persone hanno, mentre la seconda mostra la reale situazione della stanza*

Verso il 1860 il prototipo di Brewster poteva essere trovato sul mercato nelle più svariate versioni, alcune, per esempio, in cartone autoassemblato con immagini in movimento controllate a mano. Come si può facilmente notare, il design di questo strumento è concettualmente lo stesso del View-Master del 20° secolo e delle moderne Google Cardboard.

Un'altra importante nota risalente a questo periodo, sebbene apparentemente possa sembrare non direttamente legata alla RV, è datata 1895; in quell'anno alla fiera di

metà inverno a San Francisco veniva presentata quella che sarà considerata fino ai giorni nostri una delle più avvincenti dimostrazioni tecniche di un'illusione: l'Haunted Swing. Essa consiste nell'avere delle persone sedute in una piattaforma all'interno di una camera e dare a loro la percezione che stiano ruotando sottosopra, mentre in realtà a ruotare sono solo le pareti della stanza (Figura 2.3). Questo gioco tra percezione e realtà, come già detto nella sezione precedente, rappresenta una delle "ragioni d'essere" alla base della RV.

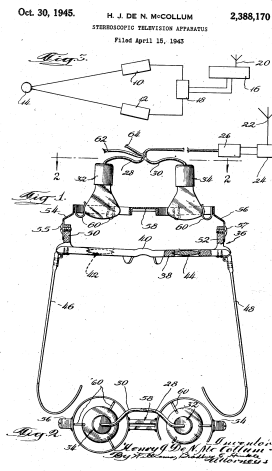
## Il 1900

Se nel secolo precedente abbiamo potuto osservare la nascita di alcune idee e strumentazioni che potevano essere considerate precursori della RV, ma non appartenenti ad essa, in questo invece, possiamo effettivamente collocare la nascita delle prime apparecchiature di emulazione virtuale tridimensionale. L'idea che funse da lancio, risale al 1935 ed è da attribuire a Stanley G. Weinbaum, scrittore e autore di fantascienza statunitense. Nella sua opera *"Pygmalion's Spectacles"*, il personaggio principale, Dan Burke, incontra un professore elfo, Albert Ludwig, che ha inventato un paio di occhiali con i quali "vedere": *"un film che da vista e suono [...] gusto, olfatto e tatto. [...] Sei nella storia, parli con le ombre (personaggi) e loro rispondono e invece di essere su uno schermo, la storia parla di te e tu ci sei dentro."* [39].

Probabilmente ispirati da questa storia, nel 1945 nacquero i primi occhiali stereoscopici per televisione (Figura 2.4) grazie a Henry e Thelma McCollum, mentre successivamente Morton Heilig brevettò due tipologie di display uno montato su testa (1950s) e uno fisso (1992). Il primo HMD, mostrato in Figura 2.6, consentiva un campo visivo orizzontale e verticale di 140° ed era fornito di auricolari stereo e ugelli di scarico dell'aria che davano un senso di brezza a diverse temperature, così come la possibilità di percepire i profumi.

Il suo display fisso (Figura 2.5) chiamato Sensorama e destinato alla visione di film immersivi, era fornito viste stereoscopiche a colori con un ampio campo visivo, suoni stereo, inclinazione del sedile, vibrazioni, odori e vento [24].

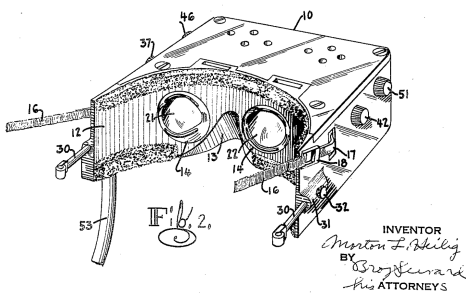
Un altro passo in avanti fu fatto nel 1961, quando venne introdotto un HMD in grado di tracciare la posizione della testa: il display, collegato ad una telecamera situata in una stanza diversa, ad una rotazione del capo faceva corrispondere una rotazione della



**Fig. 2.4:** Gli occhiali stereoscopici per televisione di Henry e Thelma McCollum



**Fig. 2.5:** Sensorama, il display fisso per la visione immersiva di film, brevettato da Heilig nel 1992



**Fig. 2.6:** L'HMD brevettato da Heilig nel 1960



**Fig. 2.7:** Sistema "Sword of Damocles": il primo HMD con tracciamento della testa che mostrava immagini generate a computer

camera, in modo che l'utente potesse vedere esattamente come se si trovasse direttamente nella stanza (questo rappresentò il primo sistema di telepresenza funzionante al mondo). Solamente un anno dopo, IBM, introdusse anche il tracciamento delle dita, brevettando (con lo scopo di sostituire gli input da tastiera) un guanto che riconosceva per ogni dito della mano quattro posizioni diverse, portando così i possibili input a 1,048,576 ( $4^5$ ).

Dobbiamo aspettare solo tre anni per vedere il primo HMD che utilizza il tracciamento della testa e visualizza immagini generate al computer; è il 1965 e il sistema in questione, inventato da Ivan Sutherland, porta il nome di *"Sword of Damocles"* (Spada di Damocle), Figura 2.7.

Nel 1985, Scott Fisher, insieme ad altri ricercatori della NASA, diede vita a VIVED (Virtual Visual Environment Display) Figura 2.8, il primo HMD stereoscopico con tracciamento della testa e con un ampio campo visivo, producibile commercialmente. Questo strumento si rifaceva ad una maschera facciale per sub, equipaggiata con dei display presi da due Citizen Pocket TV. Grazie al sistema Convolvotron era dotata anche di suoni 3D localizzati.

Il fatto che questo sistema poteva essere prodotto ad un prezzo relativamente conveniente, rappresentò l'inizio della nascita dell'industria RV.



**Fig. 2.8:** Il sistema VIVED, creato dalla NASA

### Gli anni 2000

I primi anni 2000 hanno rappresentato il cosiddetto *"inverno della RV"*, l'iniziale entusiasmo nato ad inizio anni '90 era lentamente scemato, poiché lo sviluppo di nuove ed innovative tecnologie aveva subito un rallentamento. In questo periodo però, sono proseguiti gli studi e la comunità VR ha iniziato ad orientarsi verso una progettazione centrata sull'uomo, con enfasi sullo studio delle percezioni degli utenti.

I primi frutti di questi studi arrivano nel 2006, quando Mark Bolas (dell'USC MxR Lab) e Ian McDowall (di Fakespace Labs) creano un HMD con un campo visivo più ampio:

non più 140°, ma 150°. Secondo gli studi di Bolas, questa nuova apparecchiatura, chiamata Wide5, grazie al suo maggior FOV permetteva agli utenti di giudicare con maggiore precisione le distanze quando camminano verso un obiettivo. Alla conferenza IEEE VR 2012 tenutasi a Orange County, in California, questo dispositivo vinse il premio come miglior demo ed entrò a far parte del progetto open source del MxR Lab, che è il precursore della maggior parte degli HMD moderni.

In quel periodo Palmer Luckey, un membro di quel laboratorio, conosce John Carmack (ora CTO di Oculus VR) con il quale fonda Oculus VR. Nel 2012 i due lanciano la campagna Oculus Rift Kickstarter, ottenendo molto successo. Da quel momento le aziende tornano a vedere il potenziale della realtà virtuale ed iniziano ad investire in essa. Nel 2014 Facebook, acquisisce Oculus per 2 miliardi, dando così il via ad una nuova era per la RV.

Attualmente, svariate aziende e molteplici versioni di visori per RV sono presenti sul mercato.

### 2.2.3 Effetti della RV sulla salute

Come detto in precedenza, la RV "gioca" con le nostre percezioni, cercando di "illudere" la nostra mente per farci credere di essere realmente all'interno dell'ambiente digitale. Il cervello però, se viene "ingannato" male, o addirittura troppo bene, tende a reagire, con effetti che si ripercuotono sulla persona stessa. È perciò importante per uno sviluppatore di RV, tenere conto delle problematiche che possono incorrere sull'utilizzatore, durante e dopo la fruizione dell'esperienza.

#### Motion Sickness

*"La cinetosi o malattia del movimento o chinetosi, comunemente chiamata mal d'auto o anche mal di mare, mal d'aria, mal di treno, include una serie di disturbi che si manifestano quando ci si trova su un mezzo di trasporto [...] è provocata dall'invio al cervello di messaggi contrastanti da parte degli organi principalmente coinvolti nell'equilibrio: orecchio interno, occhi, muscoli e articolazioni. In particolare, l'orecchio interno invia al cervello un segnale diverso rispetto a quello che perviene tramite gli occhi. Ciò genera confusione e conseguente malessere."* [6]. All'interno della RV questa problematica può presentarsi principalmente a causa di due fattori:



- **Scene Motion:** Tutto ciò che concerne un movimento dell'ambiente circostante, che avviene però in modo distorto dal normale (per esempio un movimento che non risulta fluido, ma a scatti). Può essere di due tipi:
  - *intenzionale:* richiesto dall'esperienza (perché in RV non siamo costretti a simulare mondi realmente esistenti e che rispettino le normali leggi fisiche)
  - *non intenzionale:* dovuto a limitazioni della tecnologia, come ad esempio latenza o calibrazione imprecisa (campo visivo non corrispondente, distorsione ottica, ecc.)
- **Vection:** è la sensazione di movimento del corpo nello spazio, prodotta esclusivamente dalla stimolazione visiva [4].

Quando ci muoviamo nella realtà, riceviamo due tipi di stimolazioni: quelle visive e quelle fisiche. Nella RV ci si può trovare frequentemente nella situazione in cui l'utente fisicamente è fermo (nella realtà), ma visivamente (all'interno dell'ambiente digitale) si sta muovendo; tale evento causa una discrepanza tra ciò che viene visto e ciò che viene fisicamente provato. Questo fenomeno però, non sempre causa motion sickness. È stato riscontrato, infatti, che questa differenza di percezioni sfocia in un problema quando scendono in campo le accelerazioni. I principali "sensori" del nostro corpo in grado di captare i cambiamenti di velocità, si trovano nell'apparato vestibolare (sistema preposto al controllo dell'equilibrio) e prendono il nome di otoliti (piccolissimi agglomerati calcarei, inglobati in una matrice gelatinosa localizzata nell'orecchio interno). Si è notato che, a velocità costanti, la dissonanza tra le informazioni inviate dall'occhio e da questi organelli non è limitante ai fini del "benessere" dell'utente. È importante sottolineare con attenzione, che i movimenti rotatori (se pur presi a velocità angolare costante), sono sempre soggetti ad accelerazione.

### Flickering

Detto anche sfarfallio, si verifica quando l'immagine che vediamo risulta intermittente. Due esempi chiarificatori di questo concetto possono essere: un'immagine su un monitor continuamente attraversata verticalmente da bande orizzontali scure, oppure un'immagine che a schermo non viene mostrata completamente nitida e ci sembra

stia compiendo dei piccoli micro-movimenti.

Può provocare nausea, vertigini, mal di testa, panico, confusione e, in rari casi, convulsioni e perdita di coscienza [27].

I fattori che possono provocare questo fenomeno sono sia di natura tecnologica, come un errato frame rate e/o una capacità hardware limitata, che anche di natura grafica, come ad esempio la presenza di luci lampeggianti o ad intermittenza.

### **Aftereffects e Readaptation**

Una parentesi, la cui importanza non è da non sottovalutare, va dedicata a tutti quegli effetti collaterali che possono insorgere una volta che l'esperienza in RV è terminata, al momento del ritorno al mondo reale. Alcuni di questi sono: instabilità percettiva del mondo, disturbi alla locomozione e alla postura, mancanza di coordinazione occhio-mano, sonnolenza, disorientamento e flashback. Fortunatamente nella quasi totalità dei casi la durata di questi sintomi è breve, al massimo di un paio ore.

Il riadattamento alla normalità (decadimento degli AE) può avvenire in due modalità:

- **decadimento naturale:** astensione da qualsiasi attività (ad esempio rilassarsi con occhi chiusi, cercando di compiere meno movimenti possibili). Questo approccio può ridurre l'insorgenza degli AE ma può prolungare il periodo di riadattamento.
- **decadimento attivo:** comporta l'uso di attività mirate nella vita reale, volte a ricalibrare i sistemi sensoriali influenzati dalle esperienze di RV (ad esempio attività di coordinamento occhio-mano). Tali attività possono aumentare la probabilità di comparsa degli AE, ma ne accelerano il processo di riadattamento.

#### **2.2.4 Le applicazioni della RV in ambito medico**

La RV trova diversi sbocchi in ambito medico. Le sue applicazioni toccano diverse aree della medicina. Le tipologie più popolari sono:

- **i giochi medici virtuali riabilitativi:** aiutano ad alleviare il dolore o mirano ad migliorare e velocizzare i processi di riabilitazione.

Al momento, le diverse applicazioni per riabilitazione in RV esistenti, sono legate

principalmente alla ripresa della corretta funzionalità delle mani. A tal proposito [33] e [41] propongono due sistemi di RV che mirano a favorire il processo di *hand rehabilitation* prima citato.

- **trattamento immersivo dell'ambiente virtuale:** esistono diverse applicazioni mirate alla cura di disturbi psicologici e fobie.

La RV ha dato un grande contributo al trattamento della fobia. In questo modo i pazienti possono essere esposti alle loro paure in modo graduale [1]. *"In alcuni casi, gli specialisti ritengono che evitare il fenomeno che causa la fobia sia un trattamento sufficiente; in altri, l'esatto contrario, essere gradualmente esposti all'oggetto della paura, può portare a una cura. Dobbiamo iniziare a cercare altri metodi psicoterapici, innovativi, in ordine per aiutare le persone a superare le loro immense paure e migliorare la loro qualità di vita."* [31].

In [30] è contenuta una panoramica temporale sull'efficacia dell'uso della RV per il trattamento dei disturbi d'ansia.

- **educazione medica in RV:** applicazioni dedite all'insegnamento ed alla formazione degli studenti di medicina. Rappresenta al momento la più grande area di applicazione della RV in ambito medico.

L'uso del sistema di supporto RV per dimostrare e spiegare simultaneamente lo stato chirurgico, rende più facile per gli studenti comprendere e mitigare la fatica e la distrazione dell'osservazione a lungo termine [2].

In [32] e [25] è possibile trovare due esempi pratici di come la RV è stata applicata nel campo della formazione medica.

Esiste poi un altro ramo nel quale la RV sta iniziando ad espandersi: la visualizzazione di immagini mediche.

Il bisogno di evoluzione che ha portato la RV ad entrare in quest'area, è dovuto al fatto che: *"i software tradizionali di visualizzazione di immagini mediche non consentono di visualizzare sezioni 2D ad angoli arbitrari e le tecniche di rendering 3D non acquisiscono adeguatamente le informazioni di profondità"* [3].

Il quid in più che la RV può dare in quest'ambito, è senza dubbio la possibilità di osservare una ricostruzione 3D all'interno di un ambiente tridimensionale. Questo fornisce una diversa modalità di osservazione che, può migliorare ed ottimizzare i processi di

diagnosi e pianificazione medica. In [38] a questo proposito è riportato: *"Tra gli altri risultati, abbiamo riscontrato un livello molto elevato di efficacia dell'interfaccia RV, nell'identificare le fratture superficiali sui CT scan della testa."*

## 2.3 I visori Oculus

Attualmente esistono in commercio svariate tipologie di visori, ognuna delle quali con caratteristiche peculiari.

Dando maggiore attenzione a quelli della famiglia Oculus (poiché quelli con e per i quali è stato svolto il lavoro, sono di questo brand), possiamo estrapolare dai moderni apparecchi, due caratteristiche che ce ne permettono una classificazione.

- **supporto computazionale:** alcuni necessitano di essere collegati ad un PC per poter lavorare, mentre altri funzionano autonomamente. I primi solitamente, sfruttando la potenza computazionale dei computer, possono creare esperienze di elevata complessità, interazione e risoluzione grafica.
- **DOF:** questo tipo di differenziazione si basa sui gradi di libertà offerti. Ogni grado di libertà indica che un particolare movimento (ad esempio la rotazione della testa attorno all'asse verticale), viene captato dai sensori dello strumento. Rotazioni e traslazioni sono le tipologie che vengono tracciate e sono in ordine:
  - *Pitch:* rotazione attorno all'asse X (per il tracciamento della testa equivale al movimento che si fa per annuire)
  - *Yaw:* rotazione attorno all'asse Y (per il tracciamento della testa equivale al movimento che si fa per dire no)
  - *Roll:* rotazione attorno all'asse Z (per il tracciamento della testa equivale all'appoggiare un orecchio su di una spalla e spostare la testa, mantenendo sempre lo sguardo in avanti, finché l'altro orecchio si posa sull'altra spalla)
  - *Translate X:* spostamento lungo l'asse X (per il tracciamento della testa equivale a spostare la testa a destra e a sinistra)
  - *Translate Y:* spostamento lungo l'asse Y (per il tracciamento della testa equivale a spostare la testa in alto e in basso)

- *Translate Z*: spostamento lungo l'asse Z (per il tracciamento della testa equivale a spostare la testa avanti e indietro)

Esistono quindi dispositivi che lavorano a 3 DOF, riconoscendo solamente i tre movimenti rotatori, e a 6 DOF, identificando anche i movimenti traslatori.

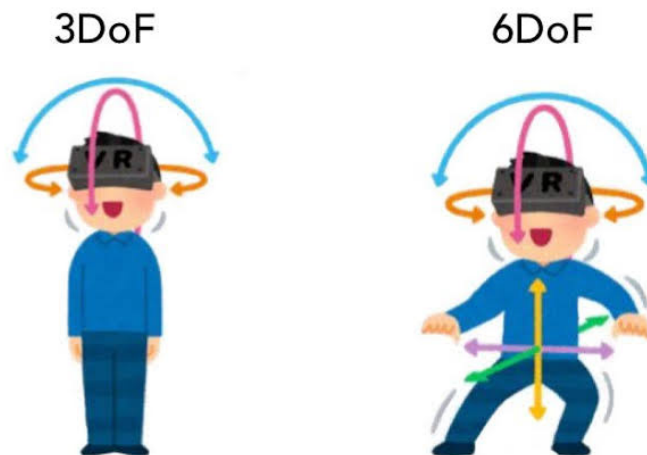


Fig. 2.9: I gradi di libertà

Il brand Oculus ha tre prodotti di punta per quanto riguarda l'emulazione virtuale:

- **Oculus GO:**

- all-in-one (che funziona senza ausilio di PC)
- 3 DOF (sia per il tracciamento della testa che per il tracciamento delle mani)
- con capacità hardware inferiori rispetto a suoi fratelli
- frequenza di aggiornamento 60 Hz
- utile principalmente per la visualizzazione in RV

- **Oculus Quest:**

- all-in-one (che funziona senza ausilio di PC), ma anche con la possibilità di funzionare supportato da computer
- 6 DOF (sia per il tracciamento della testa che per il tracciamento delle mani)
- capacità hardware ottime

- frequenza di aggiornamento 72 Hz
- utile principalmente per videogiochi all-in-one

- **Oculus Rift:**

- funziona solo se supportato da PC (con specifici requisiti minimi)
- 6 DOF (sia per il tracciamento della testa che per il tracciamento delle mani)
- capacità hardware ottime
- frequenza di aggiornamento 80 Hz
- utile principalmente per videogiochi per i quali è richiesto l'uso del computer



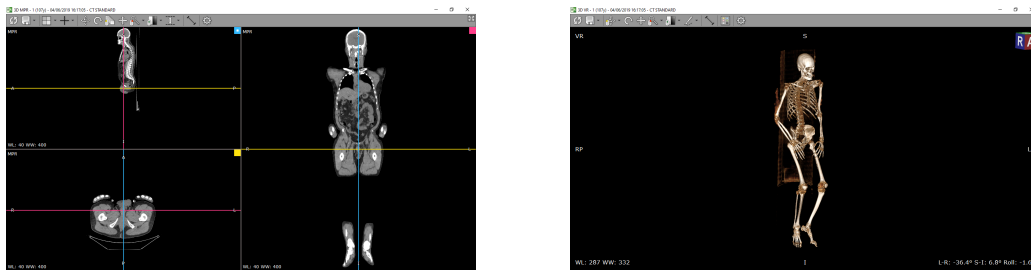
**Fig. 2.10:** *I visori della famiglia Oculus*

## 2.4 Alcuni software esistenti per la visualizzazione di immagini di TC in formato DICOM

In questa sezione sono presentati alcuni dei software di visualizzazione di immagini mediche in formato DICOM esistenti. Alcuni sono legati all'ambito 2D, quindi permettono l'osservazione degli scan e delle ricostruzioni tridimensionali su monitor bidimensionali, come ad esempio quelli di un computer; altri, invece, sono relativi all'ambito 3D della RV. Molti di questi esistono in commercio sia in versione gratuita che in versione a pagamento, mentre uno dei software di RV citati è un progetto in via di sviluppo, che ancora deve essere rilasciato.

### RadiAnt DICOM Viewer

È un programma che funziona su PC (quindi visioni bidimensionali), che partendo da file DICOM contenenti dati relativi ad un solo piano, è in grado di compiere sia le ricostruzioni multiplanari e che quelle del modello 3D. Ti permette inoltre di compiere specifiche operazioni sulle immagini.



(a) *Renderizzazione dei 3 piani*

(b) *Renderizzazione modello 3D*

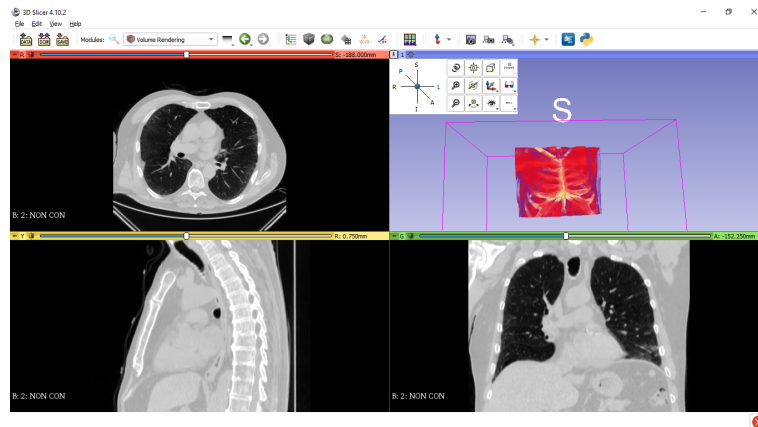
**Fig. 2.11:** *Il software per PC RadiAnt DICOM Viewer*

Supporta file DICOM di diversa natura, tra cui per esempio:

- Radiografia Digitale (CR, DX)
- Mammografia (MG)
- Tomografia Computerizzata (CT)
- Risonanza Magnetica (MR)
- Tomografia ad Emissione di Positroni PET-CT (PT)
- Ecografia (US)
- Angiografia Digitale (XA)

### 3D Slicer

3D Slicer è una piattaforma software open source per l'informatica medica, l'elaborazione delle immagini e la visualizzazione tridimensionale. È supportato dal National Institutes of Health e da una comunità di sviluppatori in tutto il mondo. Ha inoltre strumenti di elaborazione multiplatforma gratuiti e potenti.



**Fig. 2.12:** Una schermata presa durante l'utilizzo di 3D Slicer

Permette l'analisi (compresa la registrazione e la segmentazione interattiva) e la visualizzazione (incluso il rendering del volume) di immagini mediche e per la ricerca nella terapia guidata delle immagini.

È open source, gratuito e disponibile su più sistemi operativi: Linux, MacOSX e Windows.

È estensibile, con potenti funzionalità plug-in per l'aggiunta di algoritmi e applicazioni.

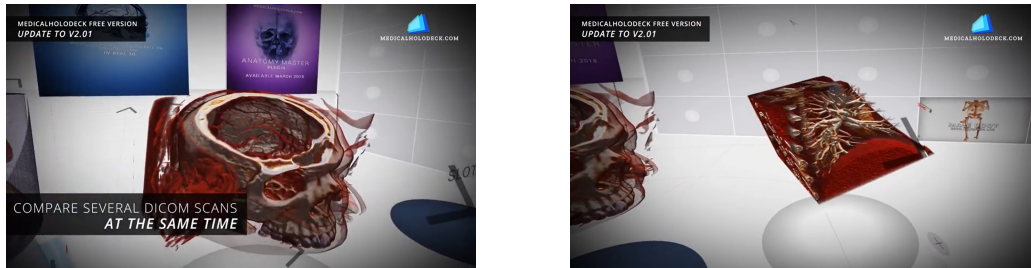
Supporta l'imaging multimodale tra cui MRI, CT, medicina nucleare e microscopia.

Non ci sono restrizioni sull'uso, ma Slicer non è approvato per uso clinico, è destinato alla ricerca.



## DICOM Viewer XR

Creato dai laboratori di Medicalholodeck per una pianificazione intuitiva della chirurgia spaziale, un'adeguata formazione medica ed una migliore informazione del paziente. Permette di studiare l'imaging medico in formato DICOM in realtà virtuale e aumentata ed usa strumenti 3D precisi per filtrare, tagliare e segmentare.



**Fig. 2.13:** Il software *Dicom Viewer XR*, creato dai laboratori di *MedicalHolodeck*, per la visualizzazione in RV di file DICOM

La piattaforma Medicalholodeck consente di eseguire diverse app XR mediche fianco a fianco e di trarre profitto dal confronto dei vari dati forniti. Questo dà la possibilità a DICOM Viewer XR di risultare efficace per la collaborazione di team virtuale.

Funziona su qualsiasi hardware di RV, su dispositivi mobili iOS e Android e su hardware di realtà aumentata come gli occhiali Magic Leap.

## DICOM VR

Nato per via della necessità fondamentale di lavorare con la crescente complessità dell'imaging medico volumetrico.

Questo progetto è stato recentemente finanziato da una sovvenzione del Brigham Research Institute.

Permette l'imaging volumetrico di CT o MRI.



**Fig. 2.14:** Due immagini dell'ambiente virtuale del software Dicom VR

Si pone come obiettivo quello di creare un sistema che consenta ai medici di visualizzare e manipolare in modo intuitivo le immagini mediche in un ambiente di RV, per migliorare la velocità, la sicurezza e la precisione dei trattamenti relativi al cancro.

### ChimeraX virtual reality

ChimeraX consente la visualizzazione e l'analisi di strutture e mappe di densità utilizzando cuffie per RV come HTC Vive, Vive Pro, Oculus Rift, Samsung Odyssey e Windows Mixed Reality.

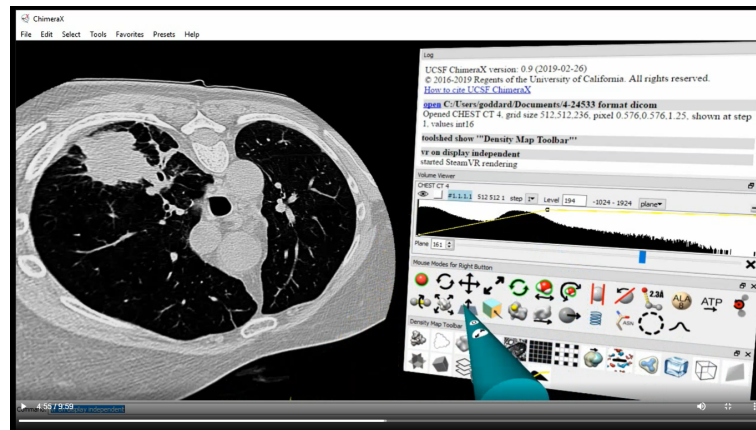


Fig. 2.15: Il software ChimeraX

Permette la creazione di sessioni con più persone.

Ha però alcune limitazioni legate ai requisiti minimi necessari del PC da usare come supporto computazionale, sulla tipologia di modello che si può visualizzare (per non incorrere in problemi legati al rate o al flickering).

## Capitolo 3

# Materiali e metodi

### 3.1 Pydicom, Numpy e la ricostruzione multiplanare

PyDicom è una libreria per Python che permette di anonimizzare, gestire, analizzare, modificare e creare dataset contenuti in file in formato DICOM (versione utilizzata 1.3.0).

Numpy è il pacchetto fondamentale per il calcolo scientifico con Python: offre nello specifico la possibilità di utilizzare array n dimensionali e numerosi tool di computazione numerica (versione utilizzata 1.17.4).

Il loro utilizzo è risultato peculiare nella prima parte del lavoro, nello specifico, ha permesso di realizzare la procedura per ricreare i file dei tre differenti piani anatomici. L'algoritmo in questione ha come scopo quello di ricostruire i dataset dei piani assiale, sagittale e coronale, partendo dai dati di un solo piano, non importa quale esso sia.

Dal contenuto di particolari tag all'interno dei file (ottenuto grazie a *pydicom*), capisce il piano di appartenenza degli stessi, dopodiché salva i dati che risulteranno necessari per la ricostruzione prima volumetrica e poi planare, che successivamente andrà a mettere in atto. L'idea è quella di sistemare in una sequenza ordinata tutti i file, creando così un'immagine tridimensionale composta da un insieme di pixel (che grazie a *numpy* vengono salvati in un array 3D). Per ottenere poi le immagini dei singoli piani, si analizza il volume appena ottenuto, tenendo fisso un asse sul quale ci si sposta e "ri-sezionando" la mole di dati in possesso lungo quella direzione.

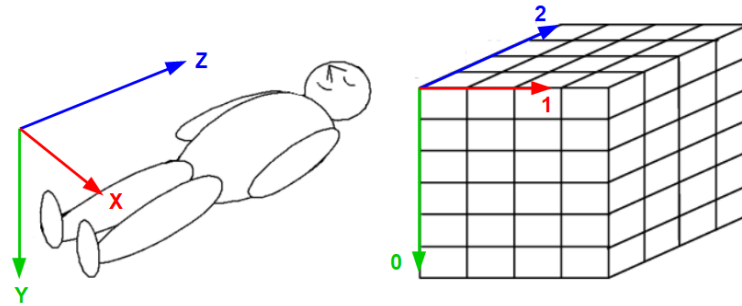
---

**Algorithm 1** Algoritmo di ricostruzione multiplanare

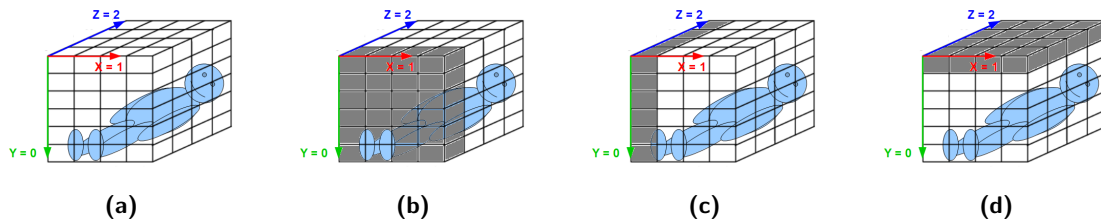
---

```
1: Capire il piano di appartenenza delle immagini
2: if piano sagittale then
3:   ordina gli elementi rispetto all'Asse X
4:   calcola i dati necessari per la ricostruzione
5: else if piano coronale then
6:   ordina gli elementi rispetto all'Asse Y
7:   calcola i dati necessari per la ricostruzione
8: else if piano assiale then
9:   ordina gli elementi rispetto all'Asse Z
10:  calcola i dati necessari per la ricostruzione
11: end if
12:
13: Copia i dati che saranno uguali per tutti i piani (datasetCopy)
14: Crea l'array 3D contenenti l'intero volume di dati (img3D)
15:
16: for  $i$  da 0 a numero scan assiali - 1 do
17:   img3D[:, :,  $i$ ] (Prendi i pixel della sezione)
18:   calcola le coordinate dello scan
19:   crea un file DICOM contenente datasetCopy, le coordinate e i dati della sezione
20: end for
21:
22: for  $i$  da 0 a numero scan sagittali - 1 do
23:   img3D[:,  $i$ , :] (Prendi i pixel della sezione)
24:   calcola le coordinate dello scan
25:   crea un file DICOM contenente datasetCopy, le coordinate e i dati della sezione
26: end for
27:
28: for  $i$  da 0 a numero scan coronali - 1 do
29:   img3D[ $i$ , :, :] (Prendi i pixel della sezione)
30:   calcola le coordinate dello scan
31:   crea un file DICOM contenente datasetCopy, le coordinate e i dati della sezione
32: end for
```

---



**Fig. 3.1:** Nell'immagine sono mostrati il sistema di coordinate relativo al paziente ed il relativo sistema di coordinate dell'array 3D, all'interno del quale dovranno essere inseriti tutti i pixel delle varie immagini dell'esame



**Fig. 3.2:** Nell'immagine sono mostrate le tre ricostruzioni dei tre differenti piani:

- a): mostra l'intero volume di pixel ottenuto dall'insieme di tutte le immagini dell'esame (l'array 3D)
- b): mostra la ricostruzione assiale; essa avviene tenendo come direzione di movimento l'asse Z (cioè il terzo campo dell'array) e, per ogni suo valore, vengono presi dall'array tutte le informazioni contenute negli altri due campi
- c): mostra la ricostruzione sagittale; essa avviene tenendo come direzione di movimento l'asse X (cioè il secondo campo dell'array) e, per ogni suo valore, vengono presi dall'array tutte le informazioni contenute negli altri due campi
- d): mostra la ricostruzione coronale; essa avviene tenendo come direzione di movimento l'asse Y (cioè il primo campo dell'array) e, per ogni suo valore, vengono presi dall'array tutte le informazioni contenute negli altri due campi

## 3.2 DicomToMesh e l'algoritmo Marching Cubes

DicomToMesh è uno strumento open source da riga di comando per Linux, OSX e Windows. Il suo scopo è quello di trasformare una serie di immagini DICOM 2D in una mesh di superficie 3D (in formato obj, stl o ply). Fornisce inoltre diverse routine di elaborazione della mesh; è possibile tra le altre cose ridurre la mesh, pulirla e levigarla. La mesh viene calcolata usando l'algoritmo *Marching Cubes*. Come input, richiede una

soglia che indica quale intervallo di valori voxel deve essere considerato. Questa soglia è anche nota come valore *iso*, che negli esami a TC, dipende dal tessuto (densità).

Le operazioni post-processing (fatte dopo l'effettiva creazione) che possono essere eseguite sulla mesh sono:

- **Riduzione:** Una mesh ottenuta da un'immagine DICOM può facilmente superare 1 GB di dimensione. È quindi utile ridurre il numero di poligoni, così da permettere un minore utilizzo di memoria.
- **Levigazione:** questa operazione permette di ridurre il "rumore" presente nei dati della mesh, permettendo così di avere una superficie più omogenea e priva di "spigolature".
- **Rimozione oggetti piccoli:** permette di rimuovere oggetti al di sotto di una determinata dimensione, che solitamente non sono di interesse (come ad esempio il tavolo su cui è stato effettuato l'esame). La soglia per la rimozione è relativa all'oggetto connesso con il maggior numero di vertici. Per esempio: se l'oggetto più grande connesso della mesh ha 1000 vertici, una soglia di 0,25 rimuove tutti gli oggetti collegati con meno di 250 vertici.

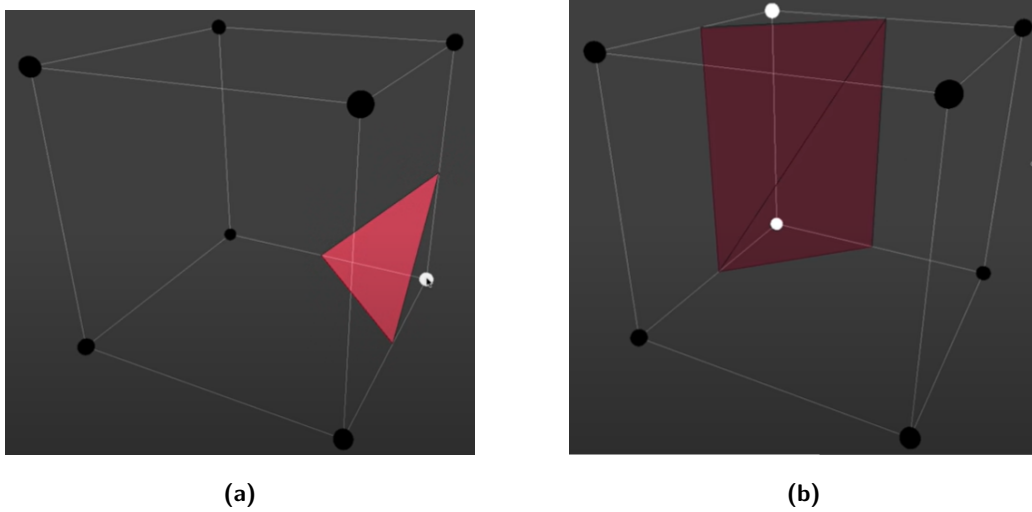
"*Marching Cubes*" (che tradotto significa letteralmente "cubi marcianti") è un algoritmo di computer grafica, pubblicato nel 1987, che permette di estrarre una mesh poligonale di una isosuperficie da un campo scalare 3D.

Per meglio comprendere questo procedimento, partiamo da una semplificazione del problema: abbiamo una superficie ed un cubo e vogliamo rappresentare, tramite triangoli, la parte di superficie che si trova all'interno del cubo. Per poterlo fare dobbiamo capire quali vertici del cubo si trovano dentro la superficie e quali no. Essendoci otto vertici da controllare, esistono  $2^8 = 256$  combinazioni possibili.

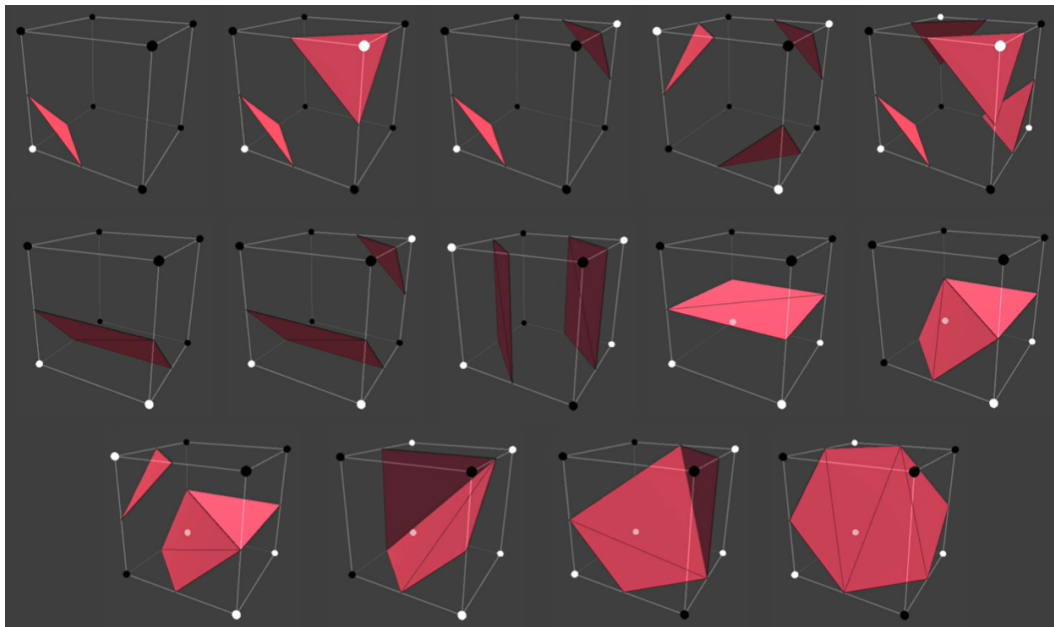
Se per esempio uno solo dei vertici si trova all'interno, allora la parte da rappresentare sarà composta da un singolo triangolo (Figura 3.3a). Se invece i vertici sono due, le situazioni possibili possono variare a seconda della posizione reciproca che i due hanno tra loro (in Figura 3.3b vediamo il caso in cui i vertici sono contigui e quindi anche i due triangoli sono contigui). Il funzionamento della rappresentazione è abbastanza semplice: per ogni vertice interno alla superficie, viene rappresentato un triangolo i cui estremi si trovano al centro degli spigoli che si incontrano nel punto

prima citato.

Dai 256 i possibili casi, è possibile estrapolarne 14 che possono essere considerati come casi primitivi; tutte le altre combinazioni infatti, saranno delle simmetrie di uno di questi casi.



**Fig. 3.3:** Due situazioni che si possono verificare analizzando il problema semplificato del marching cubes algorithm. a) solamente un vertice all'interno della superficie. b) due vertici all'interno della superficie

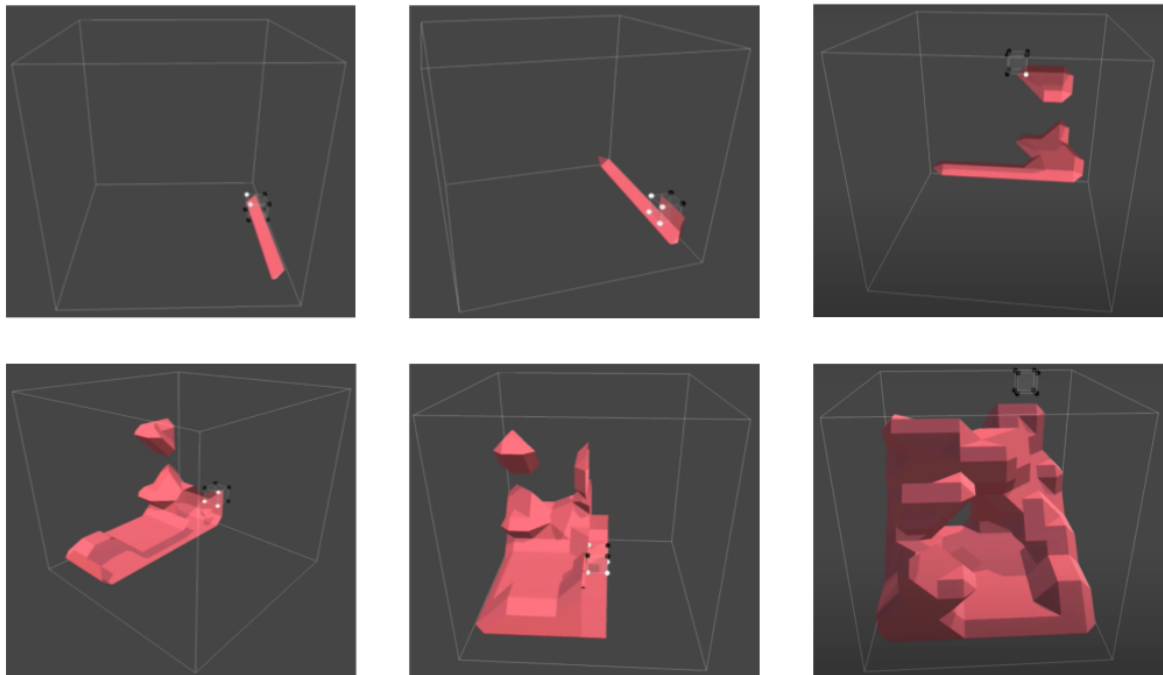


**Fig. 3.4:** I 14 casi base del Marching Cubes Algorithm



Il funzionamento generale di questo algoritmo non è altro che la ripetizione iterativa del caso base precedentemente descritto.

Si divide infatti la superficie in esame in tanti piccoli campioni di forma cubica e per ogni campione si rappresenta la parte di superficie interna al cubo secondo i criteri elencati nel caso base.



**Fig. 3.5:** *Alcuni step del Marching Cubes Algorithm*

### 3.3 Unity

Unity è un motore grafico multiplatforma, creato da Unity Technologies, che consente la realizzazione di videogiochi ed altri contenuti interattivi.

Con esso è possibile creare progetti legati all'ambito 2D, 3D e virtuale (RV, RA, RM, XR, ecc.).

Nel seguito saranno elencate e descritte tutte le features utilizzate durante lo sviluppo del progetto di tesi.

### 3.3.1 AssetBundle e Batchmode

Un AssetBundle è un archivio di file che contiene Asset privi di codice e platform-specifici (come modelli, textures, prefabs, clip audio ed anche intere scene) che Unity può caricare runtime [35].

Il loro utilizzo si è reso necessario per permettere al visualizzatore di lavorare su di un pool di file dinamico (infatti tale insieme si baserà sul contenuto di un server, che verrà aggiornato frequentemente).

Gli AB creati per ogni esame sono quattro: uno contenente il modello e tre contenenti ognuno i dati di uno specifico piano anatomico. Essi verranno scaricati separatamente sul visore quando si renderà necessaria la loro visualizzazione. Quando si creano gli AB ci sono diverse opzioni che possono essere aggiunte per dare all'archivio determinate caratteristiche, la scelta di una tipologia piuttosto che un'altra dipende dal contesto in cui gli AB verranno utilizzati.

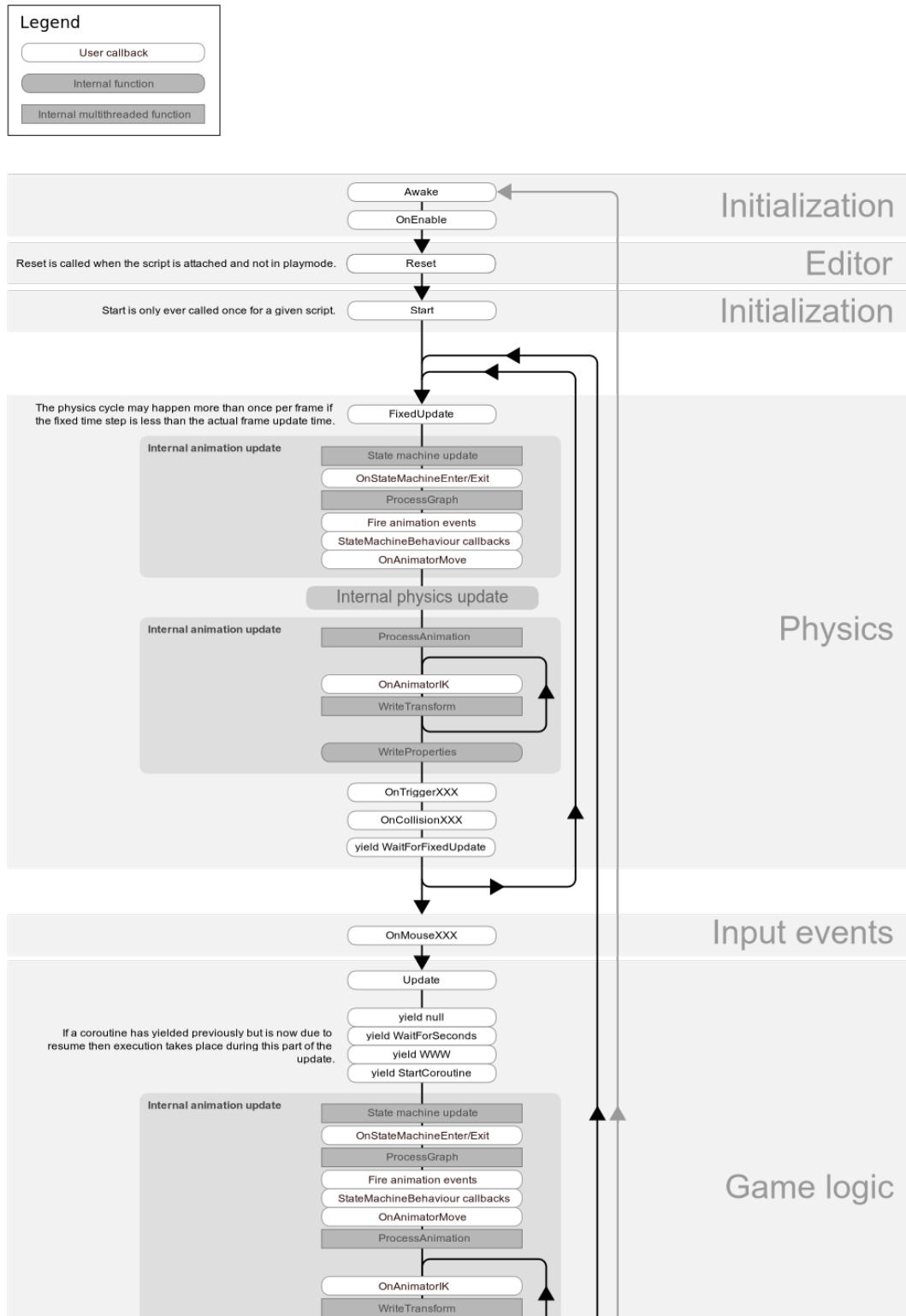
La scelta per questo progetto di tesi, è ricaduta sul non aggiungere ulteriori opzioni, poiché come spiegato in [36]: *BuildAssetBundleOptions.None: questa opzione utilizza la compressione del formato LZMA, che consiste in un singolo flusso compresso di file di dati serializzati. La compressione LZMA richiede che l'intero pacchetto sia decompresso prima di essere utilizzato. Ciò comporta dimensioni del file più piccole possibili, ma un tempo di caricamento leggermente più lungo a causa della decompressione.*

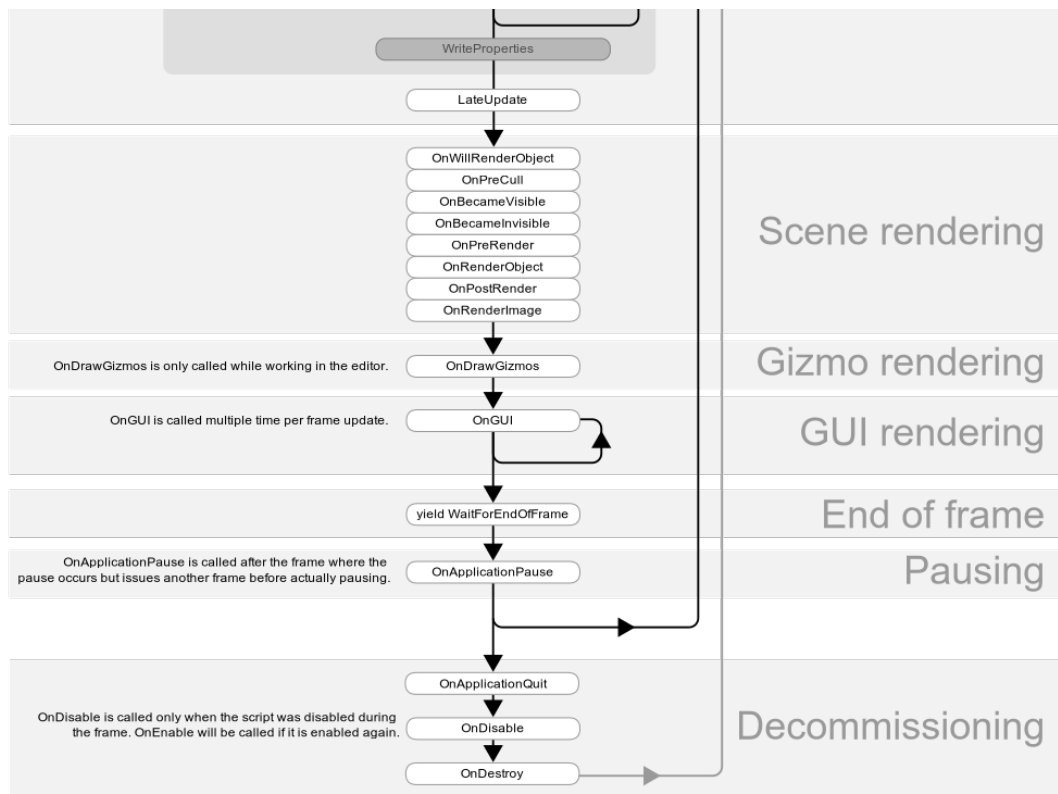
L'applicativo in python ha il compito di creare e caricare sul server gli AB. Per farlo, esso deve poter eseguire all'interno dell'ambiente di Unity gli script dedicati alla creazione di questi ultimi. La modalità batch di Unity, permette di eseguire alcuni suoi specifici script direttamente da terminale, senza dover aprire un'istanza grafica dell'applicazione. Le azioni che si possono effettuare in modalità batch, sono tutte quelle eseguibili quando nessuna scena è in esecuzione. Per creare un AB, bisogna quindi avere uno script di appartenente alla tipologia precedentemente descritta, che si occupi della realizzazione dell'archivio compresso.

### 3.3.2 Coroutines e Multithreading

Le Coroutines sono dei particolari metodi che Unity offre, che hanno una peculiarità: la loro esecuzione può essere interrotta e ripresa successivamente.

Per meglio comprendere la struttura e la funzione di una CRT, occorre prima osservare la normale esecuzione di una scena di Unity.





**Fig. 3.6:** Il ciclo di vita e l'ordine di esecuzione degli script in Unity, a partire dall'inizio dell'esecuzione di una scena

In Figura 3.6 possiamo notare come l'esecuzione di una CRT sia parte integrante del ciclo di vita di uno script Unity.

Come già detto in precedenza, la particolarità di una CRT sta nel fatto che la sua esecuzione può essere stoppata, in attesa che una specifica condizione venga verificata (una condizione booleana, il termine dell'esecuzione di un metodo, il termine del frame, ecc.). In questo modo possiamo spezzare e suddividere l'impatto che un certo metodo può avere sui vari frame. L'utilizzo delle CRT risulta molto utile, specialmente quando si ha a che fare con tecnologie che possono risultare limitanti, al fine di mantenere la fluidità di scorrimento dell'esperienza ed evitare, per quanto riguarda la RV, l'insorgenza delle problematiche psicofisiche descritte nel capitolo precedente.

Sebbene l'utilizzo delle CRT si avvicina in un qualche modo all'esecuzione multithreading, come possiamo notare dal ciclo di vita di uno script, esse sono comunque sequenziali e sincrone. Vengono eseguite nel blocco Game Logic, subito dopo che l'esecuzione del

metodo Update è terminata, a patto che una delle condizioni che la arrestavano venga verificata. L'esecuzione multithreading in Unity resta comunque possibile, ma in alcuni casi il suo utilizzo risulta molto limitato, in quanto buona parte delle routine offerte da unity possono essere chiamate solamente dal thread principale.

### 3.3.3 Lightweight Rendering Pipeline

La rendering pipeline stabilisce la sequenza di passaggi che deve essere eseguita dalla GPU e dalla scheda grafica durante il rendering di oggetti.

Viene avviata quando si esegue un'operazione di rendering. Le operazioni di rendering richiedono la presenza di due elementi: un array di vertici e un Program Pipeline Object, che fornisce gli shader per le fasi programmabili della pipeline.

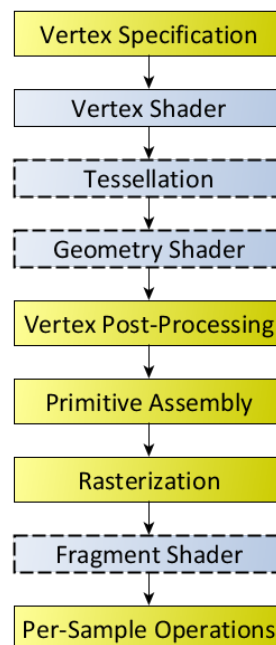
Occorre a questo punto spiegare che cos'è uno shader. Fondamentalmente uno shader è un piccolo programma, che viene eseguito su una scheda grafica o su una GPU, che manipola una scena 3D durante la pipeline di rendering, prima che l'immagine venga disegnata sullo schermo. Grazie ad essi è possibile ottenere diversi effetti sia nel campo dell'illuminazione e dell'ombreggiatura, che nel campo del movimento (ad esempio è possibile realizzare l'effetto di increspatura e movimento dell'acqua attraverso un Vertex Shader). Esistono diverse tipologie di shader (nel proseguo della sezione ne descriveremo velocemente alcuni).

La sequenza delle operazioni svolte dalla pipeline è la seguente:

#### 1. Vertex Specification

- (a) **Vertex Shader:** elaborazione dei vertici ricevuti come input, modifica della posizione, del colore e delle textures dei vertici.
- (b) **Tesselation:** divisione dei poligoni ricevuti tramite un set di vertici (che rappresentano gli oggetti in una scena), in strutture adatte per il rendering (generalmente triangoli). È una fase opzionale.
- (c) **Geometry Shader:** accetta come input un insieme di vertici che formano una singola primitiva, ad esempio un punto o un triangolo, e li trasforma come ritiene opportuno. È in grado di convertire la primitiva originale (set di vertici) in primitive completamente diverse, generando più vertici di quelli inizialmente dati. È una fase opzionale.

2. **Vertex Post-Processing:** qui i vertici ottenuti come output dalla fase precedente vengono sottoposti a varie operazioni, che risultano preparatorie per le fasi successive. In primis, vengono registrati i dati delle primitive ottenute in oggetti buffer, che poi verranno modificati tramite le azioni di clipping, perspective division and view transformation.
3. **Primitive Assembly:** qui i vertici ricevuti vengono assemblati nelle primitive specificate. Per esempio, se l'input è una sequenza di sei vertici e le informazioni sulla connettività specificano che stiamo usando triangoli, l'output sarà di due triangoli.
4. **Rasterization:** processo nel quale ogni singola primitiva viene suddivisa in elementi discreti chiamati frammenti (i pixel).
5. **Fragment Shader:** elaborazione dei frammenti ricevuti come input, modifica della posizione, del colore e delle textures dei pixel.
6. **Per-Sample Operations:** vengono elaborati i frammenti prodotti dal fragment shader e i dati risultanti vengono scritti in vari buffer.



**Fig. 3.7:** Diagramma della rendering pipeline. Le caselle blu sono stadi shader programmabili.

La LWRP, è una pipeline di rendering preimpostata creata da Unity.

Grazie ad essa è possibile ottenere un rendering rapido con un'alta qualità, senza bisogno della tecnologia degli shader di calcolo [34].

È supportata su diverse piattaforme:

- Windows e UWP
- Mac e iOS
- Android
- Xbox One
- Playstation 4
- Nintendo Switch
- Tutte le piattaforme di RV attuali

Questo tipo di pipeline utilizza una particolare tecnica: il Single-Pass Forward Rendering, che è la combinazione di due diverse procedure: Il Single-Pass Stereo Rendering e il Forward Rendering Path.

Unity in RV, renderizza la scena due volte, una per l'occhio destro e una per quello sinistro. Esegue due chiamate per ogni GameObject da mostrare a video, tuttavia, scorre il grafico della scena solo una volta.

Se si effettua il Single-Pass Stereo Rendering, Unity renderizzerà contemporaneamente le immagini dell'occhio sinistro e destro, creando una Render Texture compatta di dimensioni doppie rispetto a quelle di una texture per il singolo occhio. Entrambi gli occhi condividono il lavoro richiesto per il calcolo delle ombre e per il culling (operazione che calcola quali oggetti sono all'interno del campo visivo, che quindi sono da renderizzare, e quali no). Con questa metodologia sono necessarie anche meno opzioni relative al cambio di stato del comando grafico, poiché la GPU esegue il rendering di ogni GameObject in modo alternato tra i due occhi. In questo modo, basta solo iterare attraverso tutti gli oggetti nella scena una sola volta, per effettuare il culling, quindi si esegue il rendering di quelli sopravvissuti al processo di abbattimento.

Un percorso di rendering consiste in una serie di operazioni relative all'illuminazione

ed all'ombreggiatura. Diversi percorsi di rendering hanno differenti capacità e caratteristiche prestazionali. La scelta del percorso di rendering più adatto dipende dal tipo di progetto e dall'hardware di destinazione [37].

Il percorso che è stato utilizzato è il Forward Rendering Path.

Nel FRP si esegue il rendering di ogni triangolo sullo schermo, uno dopo l'altro, utilizzando gli shader. Ha il vantaggio di essere semplice, ma ha lo svantaggio del "disegno sovrapposto" (se si disegnano diversi triangoli uno sopra l'altro, si sta sprecando molto sforzo computazionale inutile).

L'illuminazione è la ragione principale decidere se usare il FRP o un'altra tipologia. In una rendering pipeline con FRP, i calcoli dell'illuminazione devono essere eseguiti su ogni vertice e su ogni frammento nella scena visibile, per ogni luce nella scena. Se si ha una scena con 100 geometrie e ogni geometria ha 1.000 vertici, si potrebbero avere circa 100.000 poligoni (una stima molto approssimativa). Le schede video possono gestirlo abbastanza facilmente, ma quando quei poligoni vengono inviati al fragment shader, è lì che avvengono i calcoli più costosi dell'illuminazione e può verificarsi il vero rallentamento.

Quindi, se il progetto non utilizza una grande quantità di luci in tempo reale o se la fedeltà dell'illuminazione non è importante, il FRP rappresenta una buona scelta.

La LWRP utilizza un approccio di ombreggiatura diverso rispetto alla rendering pipeline integrata di Unity. Perciò essa ha un diverso set di shader standard. Due di questi sono risultati utili durante lo sviluppo del visualizzatore DICOM e sono:

- **Unlit Shader:** utile per effetti o oggetti che non necessitano di illuminazione. Poiché non ci sono calcoli o ricerche di illuminazione che richiedono tempo, questo Shader è ottimale per hardware di fascia bassa. Unlit Shader utilizza il modello di ombreggiatura più semplice della LWRP.
- **Lit Shader:** consente di eseguire il rendering di superfici del mondo reale come pietra, legno, vetro, plastica e metalli con qualità foto-realistica. I livelli di luce e i riflessi sembrano realistici e reagiscono correttamente in varie condizioni di illuminazione, ad esempio una luce solare intensa o una grotta buia. Questo shader utilizza il modello di ombreggiatura più pesante, dal punto di vista computazionale, delle LWRP.



### 3.3.4 Baked Lights

Unity fornisce la possibilità di eseguire i calcoli per alcune luci direttamente nell'Editor (quindi quando la scena non è attiva) e salva i risultati su disco come dati di illuminazione. Questo processo si chiama "Baking". In fase di esecuzione, Unity carica i dati di illuminazione e li utilizza per illuminare la scena. Poiché i calcoli complessi vengono eseguiti in anticipo, le BL riducono i costi di ombreggiatura in fase di esecuzione.

Questa tipologia di luce è utile per illuminare elementi che non cambieranno in fase di run.

Unity trasforma sia l'illuminazione diretta che quella indiretta delle BL in lightmaps (textures contenenti i dati ottenuti dal processo di pre-calcolo della luminosità delle superfici di una scena).

Esistono però alcune limitazioni legate all'utilizzo delle BL, che possono rendere inutile il loro utilizzo:

- non è possibile modificarle in fase di esecuzione
- non contribuiscono all'illuminazione speculare
- oggetti dinamici non riceveranno luci e ombre dalle BL.

Per poter pre-calcolare l'effetto di una luce baked su di un oggetto di scena è necessario che esso sia impostato all'interno della scena come *oggetto statico*.

In Unity esistono due tipologie di oggetti di scena: statico e dinamico. La prima identifica oggetti che durante l'intera esecuzione della scena non cambieranno mai la loro posizione (e quindi sarà possibile effettuare su essi tutti i calcoli relativi all'illuminazione della scena, pre-esecuzione), mentre la seconda indica l'esatto opposto, cioè oggetti che si muoveranno durante lo svolgimento dell'esperienza (e quindi per i quali sarà possibile avere solo calcoli realtime per la luce).

### 3.3.5 Oculus Integration

Unity fornisce supporto per la RV integrato per i dispositivi Oculus.

Il pacchetto Oculus Integration è composto da script, prefabs, campioni e altre risorse che forniscono un supporto integrato per Unity [5]. Il pacchetto include, tra le altre cose:

- un'interfaccia per il controllo del comportamento della telecamera di RV
- un'API di input unificata per i controller

Il pacchetto Oculus Integration contiene un prefab (oggetto con caratteristiche e componenti preimpostate, ma in alcune parti customizzabile) chiamato `OVRCameraRig`, che fornisce l'elemento per rappresentare lo spazio di tracciamento di Oculus. All'interno di esso, è presente una videocamera per RV personalizzata, che sostituisce la videocamera convenzionale di Unity. `OVRCameraRig` fornisce anche l'accesso a `OVRManager`, che è l'interfaccia principale per l'hardware del visore.

L'interfaccia per il controllo della telecamera, contenuta nel prefab sopra elencato, è divisa in due script contenenti diverse variabili da gestire (di seguito verranno elencate e brevemente descritte solo quelle principali):

- **OVR Camera Rig:**
  - **Use Per Eye Camera:** selezionando questa opzione è possibile utilizzare telecamere separate per gli occhi sinistro e destro
  - **Use Fixed Update For Tracking:** selezionando questa opzione è possibile aggiornare tutti gli ancoraggi tracciati nel metodo `FixedUpdate()` anziché nel metodo `Update()` per favorire la fedeltà fisica. Tuttavia, se la frequenza di aggiornamento fissa non corrisponde alla frequenza dei fotogrammi di rendering, perde di efficacia.
  - **Disable Eye Anchor Cameras:** selezionando questa opzione si possono disabilitare le telecamere sugli ancoraggi. In questo caso, la fotocamera principale dell'app verrà utilizzata per fornire il rendering in RV e gli ancoraggi dello spazio di tracciamento vengono aggiornati per fornire le posizioni di riferimento.
- **OVR Manager:** questo script permette la gestione di diverse aree:
  - **Target Devices:** permette di selezionare per quali dispositivi è destinata l'app (in base a questa scelta potrebbero risultare necessaria la gestione di alcune voci opzionali)
  - **Performance and Quality:**

- \* **Use Recommended MSAA Level:** permette la gestione automatica del livello di Multisample Anti-Aliasing (che una tecnica dedicata a migliorare la qualità di un'immagine) in base al dispositivo sul quale si sta eseguendo l'applicazione. Questa voce non è selezionabile se si utilizza una LWRP.
  - \* **Monoscopic:** se attivo, entrambi gli occhi vedono la stessa immagine, renderizzata partendo dalla posizione dell'occhio centrale, risparmiando prestazioni su dispositivi di fascia bassa. Non è consigliabile utilizzare questa impostazione, in quanto non fornisce una corretta esperienza in RV
  - \* **Enable Adaptive Resolution:** se attiva permette di diminuire la risoluzione dell'app quando la GPU supera l'85% di utilizzo, per poi farla tornare ai livelli iniziali quando l'uso scende al di sotto della soglia.
  - \* **Min Render Scale:** permette di impostare il limite minimo per l'adaptive resolution (il valore predefinito è 0.7)
  - \* **Max Render Scale (solo per Rift):** permette di impostare il limite massimo per l'adaptive resolution (il valore predefinito è 1.0)
  - \* **Head Pose Relative Offset Rotation:** imposta l'offset di rotazione della camera relativamente alla posizione della testa
  - \* **Head Pose Relative Offset Translation:** imposta l'offset di traslazione della camera relativamente alla posizione della testa
  - \* **Profiler TCP Port:** imposta la porta TCP nel quale Oculus Profiler Service rimane in ascolto (utile solamente ai fini di debug o development build)
- **Tracking:**
- \* **Tracking Origin Type:** imposta il punto di origine del tracciamento (se è *Eye Level*, il tracciamento della posizione e della rotazione avviene rispetto alla posizione del dispositivo; se è *Floor Level* avviene rispetto all'altezza tra dispositivo e pavimento)
  - \* **Use Positional Tracking:** se abilitato, la posizione della testa influisce sulla posizione delle telecamere virtuali

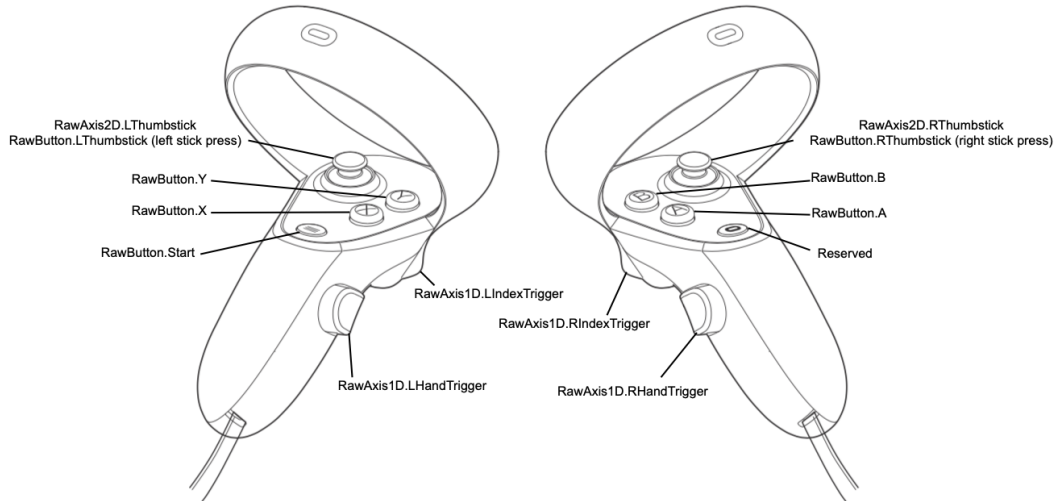
- \* **Use IPD in Positional Tracking:** se abilitato, la distanza tra gli occhi influenza la posizione di ciascuna telecamera
- \* **Reset Tracker on Load:** se abilitato, ogni volta che una scena viene caricata, la posizione della testa viene resettata
- \* **Allow Recenter:** selezionando questa opzione, è possibile ripristinare la posizione della testa quando l'utente fa clic sull'opzione *Reset View* dal menu universale. È necessario selezionare questa opzione per le app con una posizione fissa nel mondo virtuale e consentire l'opzione *Reset View* per riportare l'utente in una posizione predefinita (come un posto di guida). Non bisogna selezionare questa opzione se si dispone di un sistema di locomozione, poiché il ripristino della vista potrebbe teletrasportare l'utente in posizioni potenzialmente non valide.
- \* **Reorient HMD On Controller Recenter:** specifica il comportamento del dispositivo quando viene eseguito il recentering del controller. Se abilitato, aggiorna anche il dispositivo (cioè aggiorna anche la posizione della testa), altrimenti non aggiorna l'HMD.
- \* **Late Controller Update:** selezionando questa opzione, si può aggiornare la posizione dei controller immediatamente prima del rendering dovuto ad una piccola latenza tra il movimento del controller nel mondo reale e in quello virtuale.

Per quanto riguarda l'interazione del controller all'interno della scena, sono risultati utili diversi script presenti nel pacchetto:

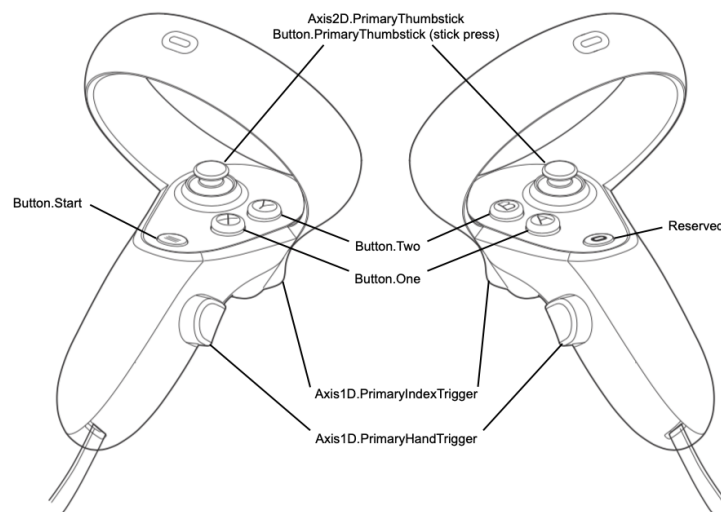
- **OVRInput:** questo script contiene una mappatura dei singoli bottoni ed assi presenti nei controller. Ci sono due tipi di mappature: una grezza (Raw Mapping) ed una Virtuale (Virtual Mapping).

La prima possiamo definirla una mappatura sia "controller-dipendente" che "device-dipendente", poiché va a mappare ogni singolo bottone del controller e fa differenza tra controller destro e sinistro. La seconda invece è più articolata: permette di eliminare la dipendenza da device, andando a creare una mappatura universale tra i vari controller. In questa opzione viene lasciata la possibilità di accedere ai controller in modo combinato (ossia facendo differenza tra i tasti del

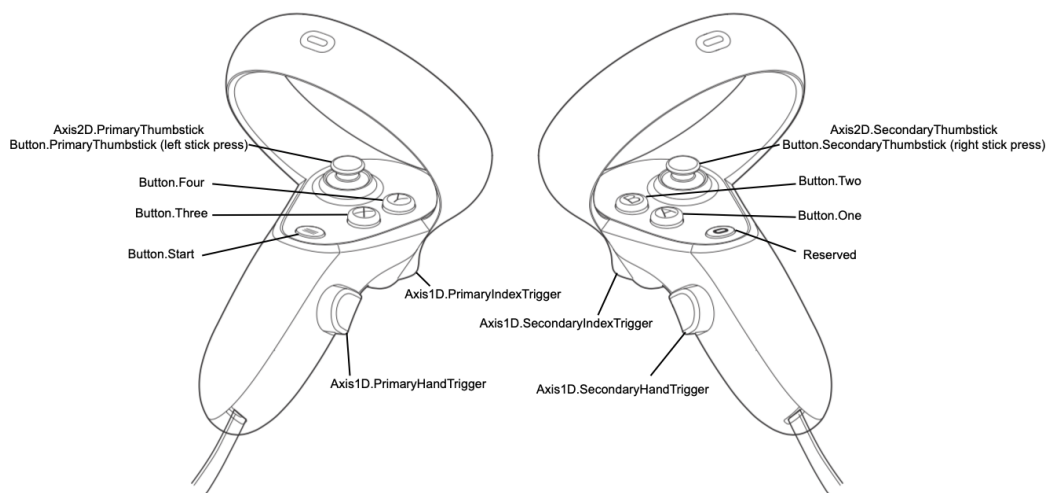
controller destro e di quello sinistro), oppure in modo individuale (non importa quale controller stia usando)



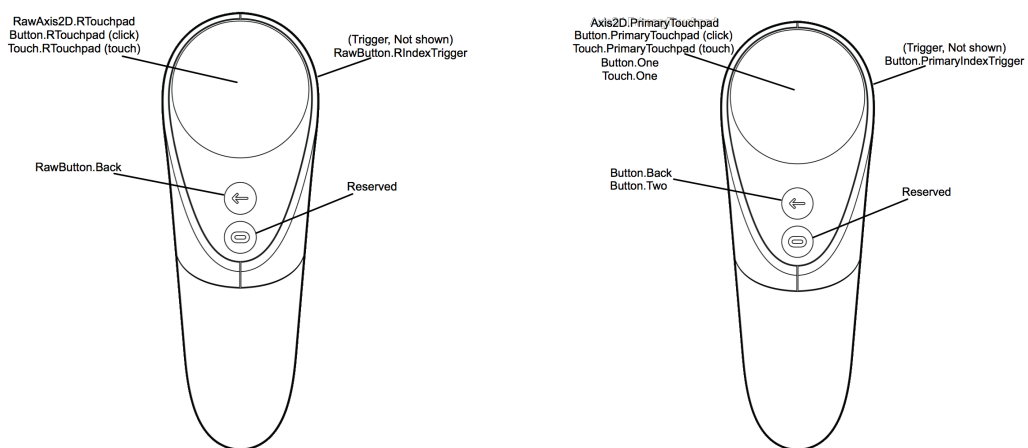
(a) Oculus Quest Raw Mapping



(b) Oculus Quest Virtual Mapping (Individual)



(c) Oculus Quest Virtual Mapping (Combined)



(d) Oculus GO Raw Mapping (Individual)

(e) Oculus GO Virtual Mapping

**Fig. 3.8:** Il mapping dei controller Oculus. Si può notare come nel mapping virtuale Individuale non venga fatta differenza tra controller destro e sinistro (Figura b) e tra controller Oculus GO e Oculus Quest (Figure b ed e)

- **OVR Input Module:** è lo script che ha il compito di captare e gestire gli eventi di pressione e rilascio dei tasti dei controller.

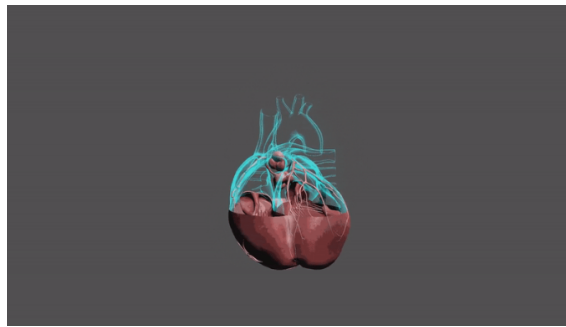
È possibile tra le altre cose: regolare alcuni valori relativi sensori dei controller (come ad esempio, la deadzone del Thumbstick o del Touchpad, in base al dispositivo che stiamo considerando), oppure decidere se eseguire lo SphereCast (al posto che inviare un raggio per captare ciò che mi circonda invio una sfera) e con che dimensioni.

- **OVR Raycaster e OVR Physics Raycaster:** sono gli script che permettono

ad un elemento appartenente alla GUI o fisico, di essere interattivo con i controller (è possibile creare una maschera composta dai soli elementi con i quali l'utente può interagire)

### 3.3.6 Cross Section Shader Graph

Cross Section Shader Graph è un progetto open source, che consiste in un package Unity che permette di effettuare la sezione trasversale di un modello, differenziando visivamente la porzione di modello già superata dal piano sezione e quella che ancora deve essere attraversata.

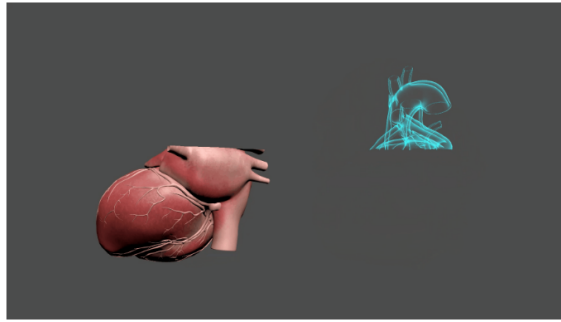


**Fig. 3.9:** *Il Cross Section Shader Graph in azione: la parte trasparente rappresenta la porzione di modello già attraversata dal piano sezionante, mentre quella normale indica il tratto che ancora deve essere superato*

Per realizzare questo effetto sono necessari due Shader, uno chiamato *Fresnel Shader* (quello simile ad un ologramma) e l'altro detto *Textured Cross Section Shader* (quello relativo alla parte normale) e pure due modelli, rispettivamente costituiti da un materiale basato sul primo e sul secondo shader.

La necessità di questo sdoppiamento, è legata al fatto che quando dobbiamo tagliare un modello, i pixel della parte sezionata vengono settati come trasparenti e quindi non vengono renderizzati (più precisamente non verranno inseriti nel buffer di profondità e quindi verranno successivamente scartati prima del rendering). Con un solo shader ed un solo modello, si può ottenere solamente uno dei due effetti, con la parte mancante completamente assente.

È perciò fondamentale la presenza di due modelli identici, aventi uguale rotazione e posizione, e di due shader, i quali, rispetto al modello a cui sono assegnati, mostreranno



**Fig. 3.10:** In figura sono mostrati i due shader che agiscono separatamente

rispettivamente l'uno la parte opposta dell'altro (Figura 3.10).

L'idea risulta quindi abbastanza chiara: dovremo avere un piano di sezione, definito dalla sua posizione e dalle sue normali, e qualsiasi pixel sopra il piano dovrà essere scartato, mentre qualsiasi pixel sotto il piano dovrà essere renderizzato.

Per determinare la posizione di un punto rispetto ad un piano ci si riconduce alla geometria lineare. Se abbiamo un piano  $\alpha$ , la sua normale  $\vec{n}$  ed un suo punto  $P_0$ , per determinare se un secondo punto  $P$  si trova sopra o sotto  $\alpha$ , l'unica cosa che dobbiamo fare è calcolare il prodotto scalare tra il vettore  $\vec{n}$  ed il vettore  $\overrightarrow{P_0P}$ : se il risultato è positivo significa che l'angolo tra  $\vec{n}$  e  $\overrightarrow{P_0P}$  è compreso tra  $0^\circ$  e  $90^\circ$  e quindi il punto  $P$  è sopra il piano, viceversa se è negativo significa che l'angolo va da  $90^\circ$  a  $180^\circ$  ed il punto si trova sotto il piano.

### 3.4 Fellow Oak Dicom

Fellow Oak Dicom (più brevemente conosciuto come FO-Dicom), è un toolkit open source che permette la gestione e la visualizzazione di file DICOM su diverse piattaforme e framework:

- .NET 4.5 e superiori
- .NET Core (.NET Standard 1.3 e superiori)
- Universal Windows Platform
- Xamarin iOS



- Xamarin Android
- Mono
- Unity

La versione fo-dicom per Unity utilizza l'ultima versione del dizionario DICOM. Supporta la lettura e la scrittura di file DICOM con formati di immagine non compressi. I formati compressi come per esempio JPEG2000, non sono attualmente supportati. Fornisce anche la possibilità, mediante poche e semplici funzioni, di estrarre e ricreare runtime una texture, partendo dai Pixel Data contenuti nel dataset del file DICOM.

## Capitolo 4

# Scenario Applicativo

L'iniziale organizzazione del lavoro di tesi, prevedeva una fase nella quale si sarebbe dovuto far sperimentare il prototipo a del personale specializzato nel campo della visualizzazione medica, così da ottenere dei feedback volti al miglioramento e ad un più preciso utilizzo del software.

Purtroppo a causa delle misure restrittive dovute al COVID-19, questa fase non si è potuta mettere in atto.

### 4.1 Preparazione dei dati

La preparazione dei dati avviene tramite uno script python, che si occupa di svolgere tutte le operazioni pre-visualizzazione.

Nello specifico questo programma ha il compito, partendo dalla sequenza di file DICOM di un piano, di:

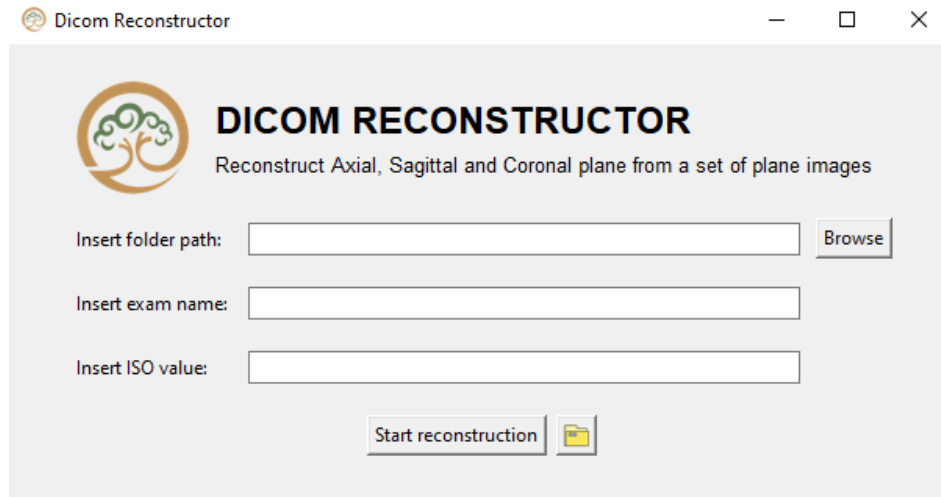
- Creare nuovi DICOM contenenti i dati delle immagini dei tre piani anatomici (assiale, sagittale e coronale)
- Creare la ricostruzione tridimensionale al livello *iso* desiderato
- Caricare i nuovi dati creati (piani e modello 3D) sul server

L'applicativo per poter funzionare si deve trovare sulla stessa macchina in cui è presente il server nel quale salvare i dati.

Lo scopo di questo applicativo è quello di ridurre il carico di lavoro che impatterà sul visore, migliorando quindi le prestazioni del visualizzatore DICOM.

Nel proseguo della sezione verranno presentate le varie schermate presenti nell'applicativo: inserimento dati, impostazioni, schermate di lavoro, schermata conclusiva.

### Inserimento dati



**Fig. 4.1:** Nell'immagine sono mostrati il sistema di coordinate relativo al paziente ed il relativo sistema di coordinate dell'array 3D, all'interno del quale dovranno essere inseriti tutti i pixel delle varie immagini dell'esame

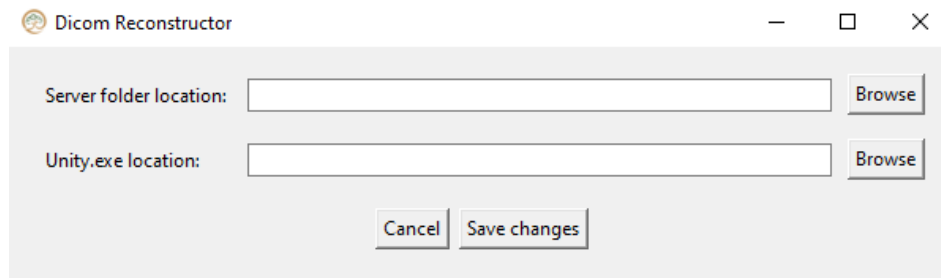
Questa è la schermata iniziale, che viene mostrata all'avvio dell'applicazione.

Contiene diversi campi di inserimento, tutti necessari per procedere al passo successivo:

1. **insert folder path:** in questo campo va inserito il percorso che porta alla cartella contenente la serie di file DICOM
2. **insert exam name:** qui bisogna inserire il nome che verrà dato alla cartella contenente tutti i dati relativi all'esame che stiamo ricostruendo (questo sarà il nome che visualizzeremo nel visore quando dovremo scegliere quale dati caricare)
3. **insert ISO value:** è richiesto il valore di radiodensità (in HU) al quale vogliamo ricostruire il modello 3D (così facendo verrà ricostruito un modello composto dai tessuti di nostro interesse)

Cliccando *Start Reconstruction* si passa direttamente alla fase di ricostruzione, mentre scegliendo il bottone con disegnata una cartella si va nella schermata delle impostazioni.

## Impostazioni

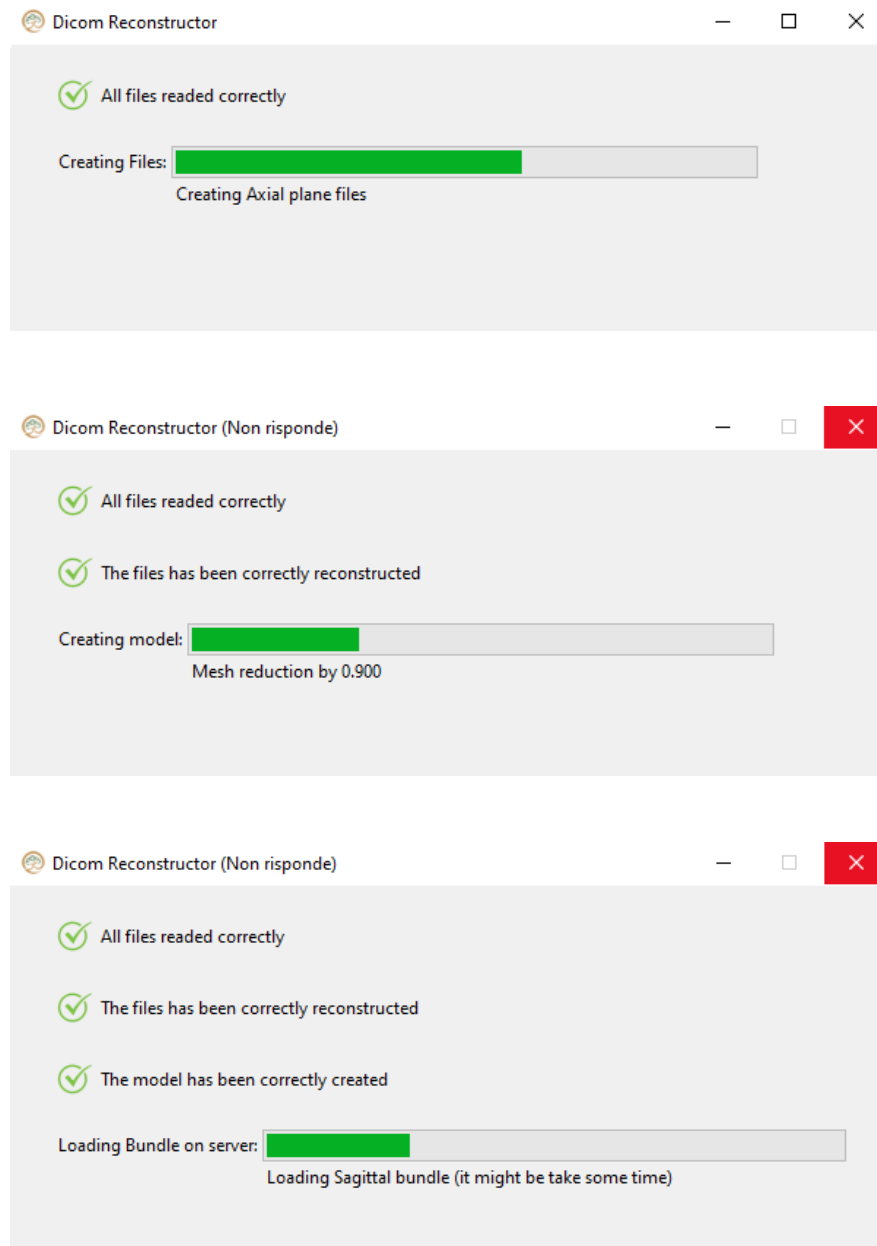


**Fig. 4.2:** *Nell'immagine sono mostrate le due impostazioni da dettare per consentire il corretto funzionamento dell'applicativo*

Le impostazioni necessarie per il corretto funzionamento del ricostruttore sono due:

1. **server folder location:** il percorso della cartella del server nella quale dovranno essere salvati gli esami
2. **Unity.exe location:** il percorso del file Unity.exe (questo percorso è necessario per poter eseguire Unity in modalità batch)

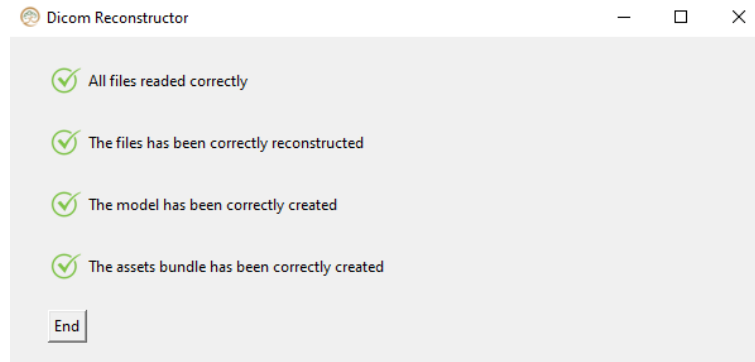
## Schermate di lavoro



**Fig. 4.3:** Nell'immagine sono mostrate le schermate attive durante il processo di ricostruzione

Le schermate presentate qui sopra, mostrano la sequenza e lo stato delle azioni che sono effettuate durante il processo di ricostruttore multiplanare e volumetrica.

### Schermata finale



**Fig. 4.4:** *Nell'immagine è mostrata la schermata finale, che si presenta una volta che la ricostruzione è terminata*

Una volta terminato il processo di ricostruzione, che si conclude con il caricamento dei file sulla cartella precedentemente definita nelle impostazioni, ci si ritrova davanti a questa schermata. Cliccando su *End* si potrà tornare alla schermata iniziale e cominciare una nuova ricostruzione.

## 4.2 Il visualizzatore

Il prototipo di visualizzatore DICOM in RV, ha rappresentato il cuore di questo progetto di tesi.

Esso permette all'utente di effettuare diverse azioni:

- caricare i dati di uno specifico esame dal server
- gestire separatamente le serie di immagini dei tre diversi piani anatomici
- visualizzare e gestire la ricostruzione 3D

Tutto questo avviene all'interno di una ricostruzione virtuale di una stanza circolare, nella quale sono presenti tre monitor (per le immagini bidimensionali) ed un proiettore olografico, che permette la renderizzazione del modello tridimensionale.

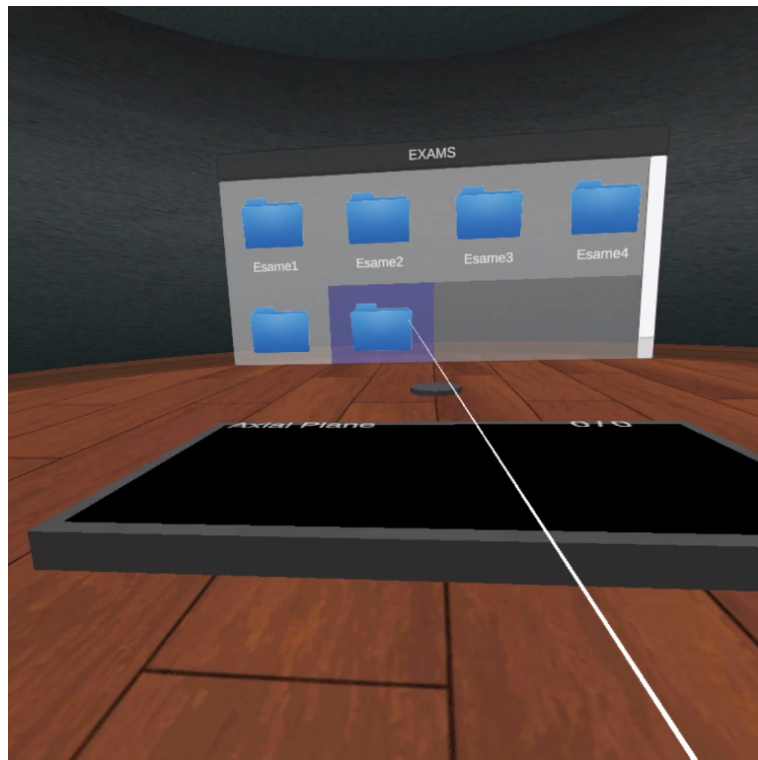
L'esperienza è strutturata per una singola persona, che, per ridurre ulteriormente l'insorgenza di problemi di sickness o di altra natura, prima e per tutta la durata dell'immersione, deve preferibilmente rimanere seduta.

Di seguito verranno mostrate e brevemente descritte alcune immagini catturate durante l'utilizzo del visualizzatore su Oculus GO.

Nello specifico:

1. selezione esame
2. caricamento dati
3. modalità immagini (visualizzazione e gestione delle slice dei piani)
4. modalità modello (visualizzazione e gestione del modello 3D)

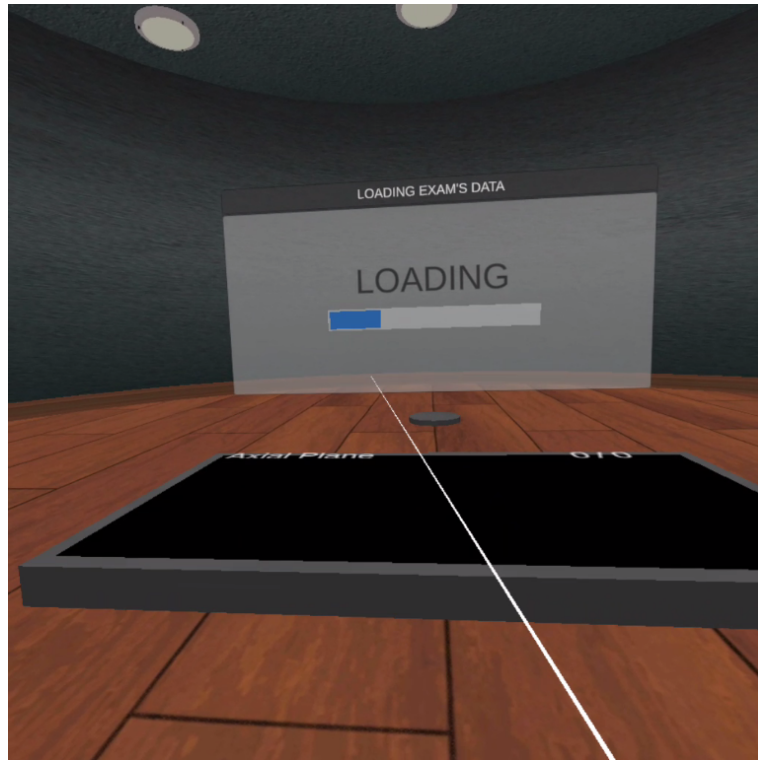
## Selezione esame



**Fig. 4.5:** *L'UI che permette all'utente la scelta dell'esame da visualizzare*

In questa immagine è contenuta l'UI relativa alla scelta dell'esame da caricare. In essa sono mostrati i nomi degli esami presenti sul server, che sono pronti per essere caricati. L'utente dovrà semplicemente puntare il cursore sulla sua scelta e selezionarla.



**Caricamento esame**

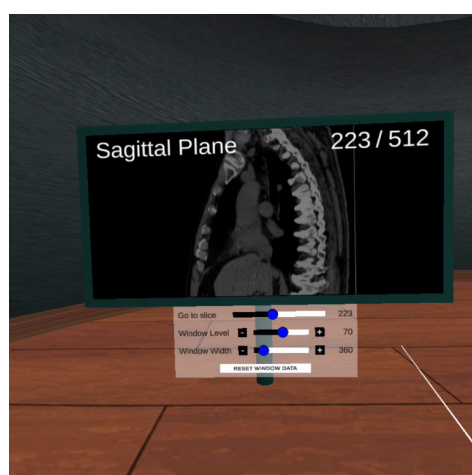
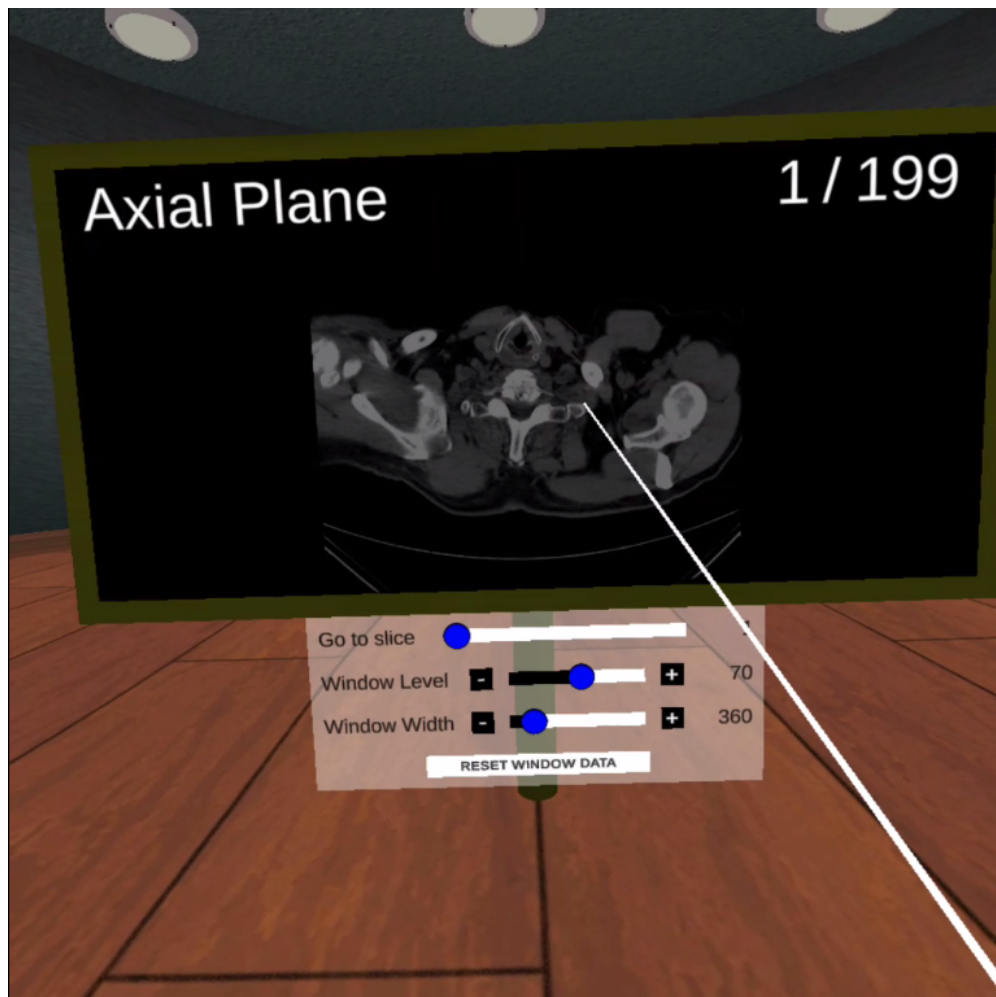
**Fig. 4.6:** *L'UI che permette all'utente la scelta dell'esame da visualizzare*

Questa UI viene mostrata durante la fase più lunga e dispendiosa a livello computazionale di tutta l'applicazione, lo scaricamento dei dati dal server ed il conseguente caricamento degli stessi nella scena.

La barra di scorrimento denota lo stato del processo che in ordine sequenziale consiste nello:

1. scaricamento del modello
2. caricamento del modello nella scena
3. scaricamento e lettura uno alla volta dei file assiali
4. scaricamento e lettura uno alla volta dei file sagittali
5. scaricamento e lettura uno alla volta dei file coronali

## Modalità immagini



**Fig. 4.7:** Il visualizzatore in modalità immagini che mostra i tre monitor contenenti le slices dei piani anatomici

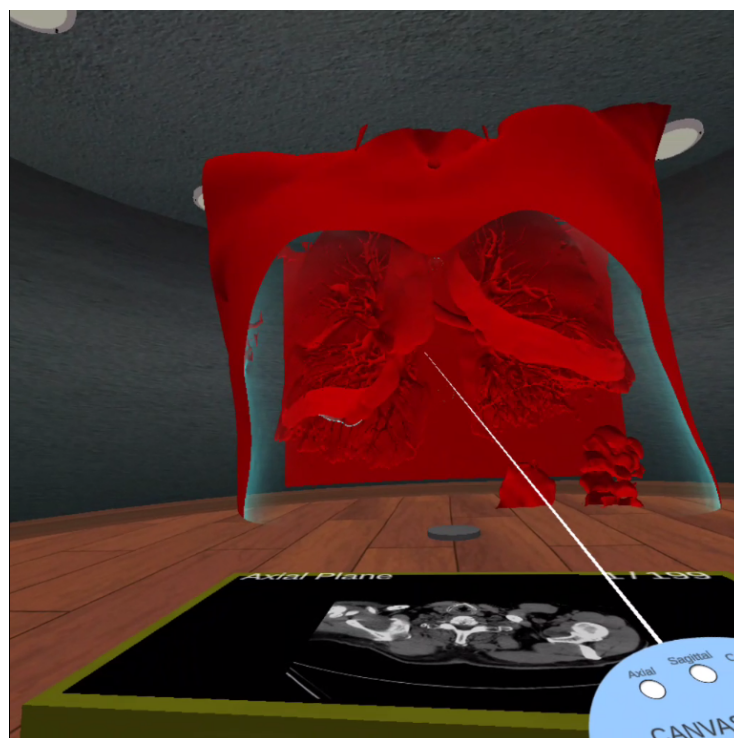
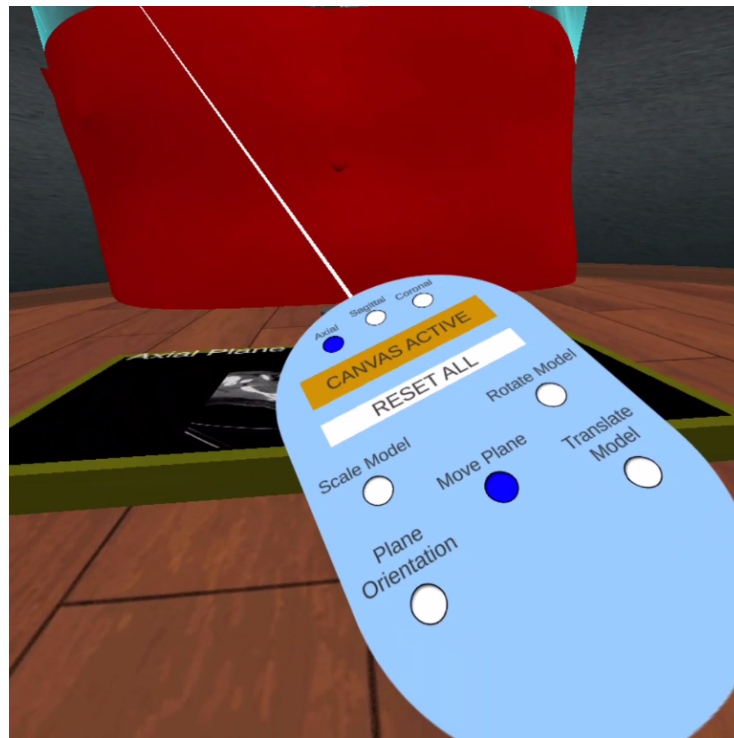
In questa modalità è possibile visualizzare le immagini dei piani anatomici in tre differenti monitor. Gli schermi sono situati rispetto all'osservatore:

- di fronte (quello contenente le immagini assiali)
- a sinistra (quello contenente le immagini sagittali)
- a destra (quello contenente le immagini coronali)

Tramite alcuni tasti del controller e grazie all'interazione con le UIs presenti sotto gli schermi per ogni schermo è possibile:

- attivare lo scorrimento continuo delle immagini
- scorrere le immagini una alla volta manualmente (sia in avanti che indietro)
- andare direttamente ad un immagine precisa
- effettuare il windowing sull'immagine andando a cambiare i valori della window tramite l'interfaccia.

## Modalità modello



**Fig. 4.8:** Il visualizzatore in modalità modello che mostra come viene rappresentato il modello 3D nella stanza e l'UI per gestirlo

La modalità modello permette di visualizzare la ricostruzione tridimensionale, ottenuta dall'insieme dei dati presi dai file DICOM.

Esso sarà attraversato da un piano sezionante, che permetterà all'utente di osservare i tessuti interni del corpo. Con tale volume è possibile:

- muovere il piano sezionante lungo una specifica direzione
- cambiare l'orientamento (e quindi anche la direzione di movimento) del piano sezionante (il piano potrà essere orientato in modo tale che la sua rotazione corrisponda a quella di uno dei tre piani anatomici: assiale, sagittale e coronale)
- ruotare il modello
- traslare il modello (orizzontalmente e verticalmente rispetto all'osservatore)
- scalare il modello (aumentando o diminuendo le sue dimensioni, fino al raggiungimento di valori estremi stabiliti)

### 4.3 Discussione

In questo capitolo sono riassunte le considerazioni riguardo i risultati ottenuti a seguito dei test effettuati sull'applicazione di visualizzazione. Particolare attenzione è stata dedicata al controllo della fluidità dell'esperienza e all'eventuale insorgenza di alcuni dei problemi psicofisici descritti nella sezione 2.2.3.

Poiché uno degli obiettivi di questo progetto è stato quello di voler rendere il viewer accessibile a visori all-in-one della famiglia Oculus, i test, anche per la disponibilità fisica dei visori, sono stati fatti esclusivamente su dispositivi GO, che sono i prodotti appartenenti alla fascia più bassa dal punto di vista prestazionale, della famiglia Oculus. I test di utilizzo, nello specifico, erano mirati a controllare i seguenti fattori:

- fluidità dell'esperienza
- risposta ad esami provenienti da fonti differenti
- insorgenza di problematiche psicofisiche
- intuitività dell'interazione

### **Fluidità dell'esperienza**

Durante le varie esecuzioni dell'applicazione, non sono stati riscontrati particolari deficit grafici o rallentamenti.

I movimenti del corpo (testa e braccio) sono risultati fluidi e sincroni con quelli effettuati nella realtà. In alcuni casi si è riscontrato un piccolo episodio di calo delle prestazioni: nel momento esatto del caricamento della prima immagine sui monitor il rendering della scena ha subito un piccolo rallentamento; questo fenomeno si è verificato solamente nei casi in cui l'utente stava fissando lo schermo nell'istante in cui l'immagine veniva caricata. Essendo relegato solo a quella frazione di secondo, non ha influito in maniera preponderante sull'esperienza.

Ci sono stati alcuni episodi di flickering relativamente alle UIs situate sotto gli schermi; essi però non hanno inficiato sulla leggibilità e sull'interazione con le interfacce.

È sorta però una limitazione, che può rappresentare un limite massimo sull'uso del prototipo su dispositivi di fascia bassa. Dopo circa 5 minuti di attività assidua e continuativa nella stanza, iniziano ad insorgere i primi fenomeni di surriscaldamento. Da quel punto in avanti si è notato che l'esecuzione continua regolarmente, ma dopo un certo intervallo di tempo l'esperienza arriva ad un blocco. L'ampiezza di questo arco di tempo si è rivelata essere dipendente dalla dimensione dell'esame che si sta analizzando: con studi di grosse dimensioni (scan relativi all'intero corpo, comprendenti più di 500 immagini per ogni piano) è di pochi minuti, mentre se consideriamo immagini relative ad una ristretta parte dell'organismo, si estende anche a 10/15 minuti. In generale quindi, ci muoviamo su un utilizzo che va dai 5/6 ai 15/20 minuti.

Considerando che maggiore è la durata dell'immersione, maggiori sono le probabilità che possano comparire complicazioni per l'utente, questa barriera può considerarsi comunque accettabile e visto che l'hardware del visore GO ha capacità inferiori rispetto a Quest si può asserire, con una buona approssimazione, che questo vincolo di tempo possa essere ampliato e traslato in avanti, oppure possa essere del tutto assente in strumenti sempre all-in-one ma con hardware migliori, come il prima citato Quest.

### **Risposta ad esami provenienti da fonti differenti**

Sono stati recuperati e testati diversi esami, alcuni di CT e alcuni di PET/CT:

- scan assiali dell'intero corpo
- scan assiali da collo a petto
- scan sagittali da collo a petto
- scan coronali da collo a petto
- scan assiali da testa a cosce

In tutti i casi l'applicativo python è riuscito a ricostruire correttamente i piani anatomici e i modelli 3D e su visore non ci sono stati problemi per quanto riguarda l'ordinamento, l'orientamento e la visualizzazione.

Possiamo quindi concludere che, poiché i tag utilizzati nel processo di creazione dei file sono tutti obbligatori relativamente allo standard DICOM, il software di ricostruzione, a patto di avere dati conformi allo standard, è indipendente dal piano al quale la serie di immagini appartiene.

#### **Insorgenza di problematiche psicofisiche**

Il tempo di esperienza medio per i tester è stato di circa 10 minuti. In tale lasso di tempo non è stata riscontrata nessuna delle problematiche psicofisiche descritte nel secondo capitolo.

#### **Intuitività dell'interazione**

Relativamente alle interazioni per l'utente non ci sono state particolari problematiche. Tutti gli utenti che si sono sottoposti ai test hanno facilmente compreso ed interagito sia con le immagini che con il modello, senza la necessità di ricevere informazioni aggiuntive. Gli unici accorgimenti sono stati forniti prima dell'utilizzo del prototipo riguardavano semplicemente le funzionalità principali dei tasti del controller. Nello specifico ad ogni utente è stata spiegata:

- la funzionalità del trigger, che permette di selezionare o deselezionare oggetti fisici o parti di UI
- la funzionalità del back button, che serve per passare da modalità immagini a modalità modello e viceversa

- la funzionalità del touchpad, che rappresenta gli assi di movimento con il quale spostare le slide o il modello

Tutte le altre tipologie di interazione sono state dedotte dagli utenti senza ulteriori aiuti durante la permanenza nell'ambiente virtuale. Possiamo quindi considerare buono il livello di comprensione delle UIs presenti nella scena.

Le uniche note che sono state fatte in questo ambito sono due: una legata alla dimensione e l'altra al metodo di interazione con le interfacce. Nel primo caso le UI sono state giudicate un po' massive ed occludenti elementi nella stanza. Nel secondo, è stato fatto notare da alcuni tester come potesse essere sfavorevole per un utente il dover continuare a spostare la testa verso il basso per interagire con l'UI.

Tali affermazioni hanno portato ad una leggera modifica, che ha consistito nella diminuzione delle dimensioni e nello spostamento dell'interfaccia per la visualizzazione del modello 3D sul controller. In questo modo, si è ovviato al fattore coprente nella scena e si è data la possibilità agli utilizzatori, di poter osservare l'UI semplicemente spostando il controller nel campo visivo, così da non essere costretti a spostare il capo.



## Capitolo 5

# Conclusioni e Sviluppi futuri

Lo sviluppo ha portato alla realizzazione di un prototipo, disponibile anche per tecnologie di fascia medio/bassa, che permette la visualizzazione delle ricostruzioni multiplanari e volumetriche, ottenute partendo da una serie di immagini DICOM.

Dai risultati ottenuti nei test possiamo concludere che l'applicazione è performante anche in dispositivi economici come Oculus GO, riuscendo a girare su di esso, in tempi di utilizzo non troppo lunghi, fluidamente e con pochi deficit grafici, anche a fronte di diverse tipologie di esame caricate. Come già detto in precedenza, la discriminante che determina l'intervallo di tempo dopo il quale iniziano ad manifestarsi surriscaldamento e calo delle prestazioni si è rivelata essere la dimensione dell'esame stesso, maggiore è il numero delle immagini iniziali, minore è il tempo di utilizzo.

Rimane ancora da testare su dispositivi all-in-one di fascia media come Oculus Quest, ma visti i risultati precedenti ottenuti su visori meno potenti è ragionevole pensare che le limitazioni precedentemente descritte o non si verifichino, oppure che aumenti considerevolmente l'intervallo di utilizzo.

Per via delle misure restrittive legate al COVID-19, non è stato possibile organizzare delle sessioni nelle quali far sperimentare il prototipo, come ci si era prefissati inizialmente, ad alcuni medici, per potere testare con i principali destinatari di questa tecnologia, la relativa efficacia ed utilità del DICOM viewer. Questo software non è stato ideato per un uso clinico, ma rimane comunque uno strumento che può risultare d'aiuto nel campo della ricerca.

Come sviluppi futuri, potrebbe essere interessante far confrontare alcuni medici esperti nel settore con il visualizzatore, così che essi forniscano diversi pareri ed idee volte

all'inserimento di diverse caratteristiche che possono risultare di interesse per un più efficace uso dell'applicazione.

Poiché il formato DICOM è utilizzato in diverse tipologie di esame (non solo CT e PET/CT, ma anche MRI ad esempio) questo software si potrebbe ampliare alla visualizzazione di DICOM di qualsiasi natura.

Un'ulteriore possibilità di sviluppo, che potrebbe aprire un branch parallelo a quello concernente l'uso di questo software per la ricerca, è rappresentata dall'utilizzo della visualizzazione in RV, nell'ambito di "educazione" del paziente sulla sua condizione. Con un'attenta e precisa valutazione a livello clinico del software di ricostruzione utilizzato per la creazione del modello 3D (valutando se è sufficiente o se si necessita di un tool migliore), il paziente, sotto le spiegazioni effettuate da un terapeuta preparato, potrebbe avere una più chiara panoramica della sua situazione, avendo la possibilità di collegare le spiegazioni del dottore all'osservazione in un ambiente virtuale della parte del suo corpo che ha reso necessario lo studio medico.

# Bibliografia

- [1] A. H. Alkurdi. Vr mobile acrophobia treatment. *Academic Journal of Nawroz University*, 6(3):124–128, 2017. ISSN 2520-789X. doi: 10.25007/ajnu.v6n3a95. URL <http://journals.nawroz.edu.krd/index.php/ajnu/article/view/95>.
- [2] M. Cai, R. Sato, K. Yoshida, and K. Takayasu. Development and application of vr support system for medical students. In *2019 Nicograph International (NicoInt)*, pages 78–81, 2019.
- [3] P. Carnahan, J. Moore, D. B. M.D., G. Wheeler, S. Deng, K. Pushparajah, E. C. S. Chen, J. M. Simpson, and T. M. Peters. Applications of VR medical image visualization to chordal length measurements for cardiac procedures. In B. Fei and C. A. Linte, editors, *Medical Imaging 2020: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 11315, pages 578–584. International Society for Optics and Photonics, SPIE, 2020. doi: 10.1117/12.2549597. URL <https://doi.org/10.1117/12.2549597>.
- [4] S. J. Colin Blakemore. *The Oxford Companion to the Body*. Oxford University Press, 2001. ISBN 9780198524038.
- [5] O. Developers. Unity integration, 2019. URL <https://developer.oculus.com/downloads/package/unity-integration/12.0/>.
- [6] I. S. di Sanità. Cinetosi, 2019. URL <https://www.issalute.it/index.php/la-salute-dalla-a-alla-z-menu/c/cinetosi#:~:text=Cause,dovuto%20il%20sistema%20nervoso%20centrale>.
- [7] dicom.innolitics.com. 10.7.1.3 pixel spacing value order and valid values, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/image-plane/00280030>.

- [8] dicom.innolitics.com. C.11.1.1.2 modality lut and rescale type, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/ct-image/00281054>.
- [9] dicom.innolitics.com. C.11.2.1.2 window center and window width, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/voi-lut/00281050>.
- [10] dicom.innolitics.com. C.11.2.1.2 window center and window width, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/voi-lut/00281051>.
- [11] dicom.innolitics.com. C.12.1.1.2 specific character set, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/sop-common/00080005>.
- [12] dicom.innolitics.com. C.7.5.1.1.2 pixel padding value and pixel padding range limit, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/general-equipment/00280120>.
- [13] dicom.innolitics.com. C.7.6.2.1.1 image position and image orientation, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/image-plane/00200032>.
- [14] dicom.innolitics.com. C.7.6.2.1.2 slice location, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/image-plane/00201041>.
- [15] dicom.innolitics.com. C.7.6.3.1.1 samples per pixel, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/general-image/00880200/00280002>.
- [16] dicom.innolitics.com. C.7.6.3.1.2 photometric interpretation, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/general-image/00880200/00280004>.
- [17] dicom.innolitics.com. C.7.6.3.1.4 pixel data, 2020. URL <https://dicom.innolitics.com/ciods/ct-image/image-pixel/7fe00010>.
- [18] dicomstandard.org. Overview: Dicom standard, 2020. URL <https://www.dicomstandard.org/about/>.
- [19] dicomstandard.org. 7.3 big endian versus little endian byte ordering, 2020. URL [http://dicom.nema.org/dicom/2013/output/chtml/part05/sect\\_7.3.html](http://dicom.nema.org/dicom/2013/output/chtml/part05/sect_7.3.html).

- [20] dicomstandard.org. Dicom ps3.5 2020b - data structures and encoding, 2020. URL <http://dicom.nema.org/medical/dicom/current/output/html/part05.html#PS3.5>.
- [21] D. C. Emanuele Neri, Paolo Marcheschi. *Produrre ed elaborare immagini diagnostiche*, chapter 5, page 51. Springer, 1 edition, 2008.
- [22] Humanitas. Pet/ct globale corporea, 2020. URL <https://www.humanitas.it/visite-esami/pet-ct-globale-corporea>.
- [23] J. Jerald. *The VR Book: Human-Centered Design for Virtual Reality*, chapter 1.1, page 9. Morgan, Claypool, 2015. ISBN 978-1970001129,1970001127.
- [24] J. Jerald. *The VR Book: Human-Centered Design for Virtual Reality*, chapter 2.2, page 21. Morgan, Claypool, 2015. ISBN 978-1970001129,1970001127.
- [25] E. Makled, A. Yassien, P. Elagroudy, M. Magdy, S. Abdennadher, and N. Hamdi. Pathogenius vr: Vr medical training. In *Proceedings of the 8th ACM International Symposium on Pervasive Displays*, PerDis '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367516. doi: 10.1145/3321335.3329694. URL <https://doi.org/10.1145/3321335.3329694>.
- [26] merriam webster.com. Virtual reality — definition of virtual reality by merriam-webster, 2015. URL <https://www.merriam-webster.com/dictionary/virtual%20reality>.
- [27] C. E. Rash. Awareness of causes and symptoms of flicker vertigo can limit ill effects., 2004. URL [https://flightsafety.org/hf/hf\\_mar-apr04.pdf](https://flightsafety.org/hf/hf_mar-apr04.pdf).
- [28] R. K. K.-P. H.-R. S. P. e. Rüdiger Kramme, Heike Kramme (auth.). *Springer Handbook of Medical Technology*, chapter 16, page 311. Springer Handbooks. Springer-Verlag Berlin Heidelberg, 1 edition, 2011.
- [29] R. K. K.-P. H.-R. S. P. e. Rüdiger Kramme, Heike Kramme (auth.). *Springer Handbook of Medical Technology*, chapter 16, pages 311–312. Springer Handbooks. Springer-Verlag Berlin Heidelberg, 1 edition, 2011.
- [30] W. B. K. B. R. M. Serrano Berenice, Botella Cristina. *Virtual Reality and Anxiety Disorders Treatment: Evolution and Future Perspectives*, pages 47–84.

- Springer New York, New York, NY, 2019. ISBN 978-1-4939-9482-3. doi: 10.1007/978-1-4939-9482-3\_3. URL [https://doi.org/10.1007/978-1-4939-9482-3\\_3](https://doi.org/10.1007/978-1-4939-9482-3_3).
- [31] M. A. B. C.-N. H. S. Stanica Iulia-Cristina, Dascalu Maria-Iuliana. A survey of virtual reality applications as psychotherapeutic tools to treat phobias. In *The International Scientific Conference eLearning and Software for Education*, volume 1, pages 392–399, 2016. doi: 10.12753/2066-026X-16-056.
- [32] L. S.-T. Stelian Nicola, Ioan Virag. Vr medical gamification for training and education, 2017.
- [33] H. K. Tang, Z. Q. Feng, T. Xu, and X. H. Yang. Vr system for active hand rehabilitation training. In *2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS)*, pages 316–320, 2017.
- [34] U. Technologies. Lightweight rendering pipeline, 2015-2018. URL <https://docs.unity3d.com/Packages/com.unity.render-pipelines.lightweight@5.10/manual/index.html>.
- [35] U. Technologies. Assetbundles, 2019. URL <https://docs.unity3d.com/Manual/AssetBundlesIntro.html>.
- [36] U. Technologies. Building assetbundles, 2019. URL <https://docs.unity3d.com/2019.4/Documentation/Manual/AssetBundles-Building.html>.
- [37] U. Technologies. Unity rendering paths, 2019. URL <https://docs.unity3d.com/Manual/RenderingPaths.html>.
- [38] J. E. Venson, J. Berni, C. S. Maia, A. M. da Silva, M. d’Ornelas, and A. Maciel. Medical imaging vr: Can immersive 3d aid in diagnosis? In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, VRST ’16*, page 349–350, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450344913. doi: 10.1145/2993369.2996333. URL <https://doi.org/10.1145/2993369.2996333>.
- [39] S. G. Weinbaum. *Pygmalion’s spectacles*, page 5. <http://gutenberg.net.au>, 1935.

- 
- [40] A. C. William R. Sherman. *Understanding Virtual Reality: Interface, Application, and Design (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann, 1st edition, 2002. ISBN 1558603530,9781558603530.
- [41] F. Y.-Y. M. K. Y. I. G. Z. Yu Jia, Jiang Rong. Mobile vr game design for stroke rehabilitation. In P.-L. P. Rau, editor, *Cross-Cultural Design. Applications in Cultural Heritage, Creativity and Social Development*, pages 95–116, Cham, 2018. Springer International Publishing. ISBN 978-3-319-92252-2.