



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



**DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE**

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

**CORSO DI LAUREA MAGISTRALE IN
BIOINGEGNERIA**

Tesi di Laurea

**SISTEMA DI REALTÀ VIRTUALE PER LA SIMULAZIONE
CHIRURGICA CAD-BASED**

Relatrice: Prof.ssa Maria Francesca Uccheddu

Laureanda: Giorgia Lagattolla

Matricola: 2022275

ANNO ACCADEMICO 2022-2023

Data di laurea 21.04.2023

*Ai miei genitori e a mio fratello,
Che ancora una volta hanno creduto in me.
Alle amicizie lontane,
Che mi hanno sempre accompagnato.
A Padova,
Per tutto ciò che di più bello mi ha regalato.*

Abstract

Una componente importante della pianificazione chirurgica è la progettazione di ausili chirurgici per la progettazione di impianti o in generale dispositivi medici personalizzati. Uno degli aspetti più cruciali di questa procedura è la comunicazione tra progettista e chirurgo. Le inefficienze derivanti da un confronto poco chiaro tra le due parti portano a risultati indesiderati e ad un sostanziale aumento del tempo necessario per completare la progettazione e la produzione, ma possono essere ridotti al minimo quando queste due figure sono la stessa persona. Dato che attualmente è improponibile pensare di escludere completamente il ruolo dell'esperto CAD da questo processo, si sono comunque perseguiti degli obiettivi come la standardizzazione della procedura di progettazione e il raggiungimento di un processo efficace; dal momento che la maggiore affidabilità di una prassi così standardizzata rappresenta uno dei fattori fondamentali al fine di ridurre anche i rischi per i pazienti. In letteratura è evidente come tra gli approcci presi in considerazione per tentare di ridurre queste divergenze c'è stata la realtà virtuale.

Si tratta di una tecnologia di tendenza, ampiamente accessibile e contemporanea di crescente utilità per le applicazioni biomediche e sanitarie. Tuttavia, la maggior parte delle implementazioni degli ambienti di realtà virtuale sono personalizzate per applicazioni specifiche.

Nel presente elaborato si descrive lo sviluppo completo di un nuovo ambiente di realtà virtuale, svolto attraverso la piattaforma Unity, il quale prevede l'importazione di modelli CAD di parti anatomiche che necessitano di intervento chirurgico ablativo di parte della struttura ossea. Ciò che l'applicazione consente di fare è visualizzare interattivamente i modelli CAD e di utilizzare strumenti che consentono la individuazione e selezione delle direzioni di taglio più ottimali rispetto alle quali sono successivamente progettate le dime chirurgiche, importando per ultimo queste informazioni sul software Geomagic Design X di 3D Systems.

L'approccio proposto è stato utile per verificare che il sistema di realtà virtuale permette di risparmiare step rispetto alla procedura convenzionalmente eseguita su Geomagic Design X e che pertanto offrirebbe al clinico uno strumento semplice ed intuitivo che lo renderebbe autonomo nelle prime fasi di simulazione chirurgica.

Indice

Capitolo 1 - Introduzione	3
Capitolo 2 – Pianificazione chirurgica virtuale	5
2.1 Ricostruzione di un modello anatomico 3D: metodi e strumenti	7
2.1.1 Pre-processing e Segmentazione	9
2.1.2 Post processing del modello 3D	11
2.2 Fase di design CAD della simulazione dell'intervento	12
Capitolo 3 – Sistemi di Realtà Virtuale	17
3.1 Visori di realtà virtuale	17
3.1.1 Storia	17
3.1.2 Classifica dei visori	22
3.1.2.1 Visori VR tethered	22
3.1.2.2 Visori VR stand-alone	24
3.1.2.3 Visori VR mobile	25
3.1.3 Realtà Virtuale in chirurgia	27
3.2 Il visore Meta Quest 2	32
3.2.1 Hardware	33
3.2.2 Sistema di tracciamento	34
3.2.3 Controller Oculus Touch	36
3.2.4 Audio posizionale	36
3.2.5 Vestibilità del Meta Quest 2	37
3.2.6 <i>Set-up</i> di gioco	37
3.2.7 Software	37
3.3 Unity (Game Engine)	38
3.3.1 Editor	38
3.3.2 Script	40
Capitolo 4 – System Design	41
4.1 L'applicazione: una visione generale	41
4.2 Game Design	41
4.3 Realizzazione dell'ambiente VR	44
4.3.1 Setting VR	44
4.3.2 Setting Controller Sinistro	45
4.3.3 Setting Controller Destro	46

4.3.4 Grabbing e Feedback tattile	47
4.3.5 UI Canvas	49
4.3.6 Esportazione file FBX	51
Capitolo 5 – Test e risultati	53
5.1 Procedura CAD standard	53
5.2 Procedura CAD con Realtà Virtuale	56
5.3 Risultati della procedura CAD con Realtà Virtuale	57
Capitolo 6 – Conclusioni	59
Bibliografia e sitografia	60

Capitolo 1

Introduzione

La rapida evoluzione dei trattamenti medici personalizzati ha evidenziato la difficoltà per i medici e il loro personale di personalizzare i trattamenti con la migliore risposta e la massima qualità delle cure erogate in un lasso di tempo adeguatamente breve. Ogni volta che questo tipo di approccio prevede l'uso di dispositivi medici personalizzati, come le guide di taglio chirurgiche, è necessario dedicare sforzi significativi nelle fasi di progettazione e fabbricazione per il quale oggi il mercato offre diversi software appositamente studiati per la simulazione di molti passaggi chirurgici.

Data tuttavia l'enorme potenzialità di questi programmi, gli utenti devono essere ben formati; per questo motivo il processo di pianificazione preoperatoria richiede il coinvolgimento degli ingegneri e/o progettisti CAD oltre a quello dei clinici; i primi si occupano degli aspetti tecnici della progettazione e della produzione. I chirurghi, d'altra parte, forniscono dettagli specifici sull'approccio chirurgico e pianificano l'intervento definendo piani di taglio, punti di ancoraggio e producendo ulteriori vincoli che possono influenzare il design complessivo dello strumento. Entrambi i contributi sono integrati mediante un approccio progettuale iterativo, con ingegneri che tipicamente propongono diverse soluzioni, ricorsivamente esaminate e corrette dai chirurghi, fino al raggiungimento di una validazione finale. Questo metodo è evidentemente dispendioso in termini di tempo. Inoltre, il progetto è tipicamente gestito da società terze specializzate nella stampa 3D di dispositivi medici che non sono in stretto contatto con i medici, il che rende la comunicazione inefficiente e lenta. Si tratta, pertanto, di una tecnologia che si è dimostrata essere la più intuitiva, rapida ed economica, in quanto elimina la necessità di una strumentazione sofisticata. Tuttavia, nonostante la sua efficacia, questa operazione può risultare onerosa anche per un progettista esperto a causa dei diversi limiti percepiti.

Spesso sono necessarie diverse iterazioni per sviluppare un progetto soddisfacente a causa di problemi di comunicazione tra ingegneri e medici sull'approccio chirurgico. La natura specifica di ogni caso clinico fa sì che gli ingegneri di solito non siano a conoscenza di come verrà eseguito in dettaglio l'intervento chirurgico. Il tempo è cruciale nella chirurgia oncologica perché i tumori possono evolvere rapidamente tra il momento in cui viene eseguita la diagnostica per immagini e l'intervento chirurgico.

Al fine di aumentare l'applicabilità dei trattamenti medici personalizzati, è stato speso un grande sforzo nella ricerca di soluzioni per rendere l'intero processo più semplice e veloce. Non solo, l'obiettivo principale essenzialmente è la standardizzazione della procedura di progettazione e il raggiungimento di un processo efficace stabilendo una serie di regole e principi fissi che possono essere utilizzati come linee guida. Sono state sviluppate soluzioni software per velocizzare la fase di progettazione che è la più lunga e macchinosa. La maggiore affidabilità di una procedura così standardizzata e la conseguente riduzione dei rischi per i pazienti sono fondamentali. La letteratura scientifica presenta esempi significativi di metodi che perseguono un approccio più efficiente per progettare dispositivi medici specifici per il paziente, al fine anche di fornire a ingegneri e medici una piattaforma di comunicazione per condividere informazioni ed evitare molte iterazioni.

Con il miglioramento della computer grafica e dei sensori, tecnologie come la realtà virtuale (VR) e la realtà aumentata (AR) hanno creato nuove possibilità per lo sviluppo di tecniche diagnostiche e chirurgiche nel campo della chirurgia. Sono le ultime modalità tecnologiche che sono state integrate nella pratica clinica e nell'educazione medica. Il primo utilizzo della realtà virtuale in campo medico è stato segnalato negli anni '90, che prevedeva la visualizzazione di dati medici complessi durante l'intervento chirurgico e la definizione di piani chirurgici preoperatori con l'assistenza della realtà virtuale.

Lo scopo di questa tesi è stato adottare e valutare un approccio alternativo a quello tradizionale basato sulla realtà virtuale, al fine di semplificare le vie di comunicazione tra ingegnere e clinico, in quanto questi ultimi di solito forniscono solo indicazioni grossolane che devono essere successivamente interpretate per essere utilizzate come input di progettazione. A tal proposito, in questo elaborato si discute di quale ruolo abbia la realtà virtuale nell'ambito chirurgico e di quali siano attualmente i protocolli previsti e adottati per la progettazione e fabbricazione di una dima chirurgica di taglio nel caso di osteotomie su pazienti oncologici. Verrà illustrato il processo di sviluppo e realizzazione di uno strumento di pianificazione preoperatoria tramite simulazione in realtà virtuale il cui uso è destinato esclusivamente alla figura del clinico durante la prima parte del processo di design della dima, fornendogli una vista sul modello 3D anatomico per una migliore comprensione spaziale e una gestione più naturale dei piani di taglio, la scelta della loro direzione e disposizione rispetto al modello in modo da trasmettere poi questa informazione agli ingegneri per progettare accuratamente le guide di taglio.

Capitolo 2

Pianificazione chirurgica virtuale

La **pianificazione chirurgica virtuale** è il meccanismo mediante il quale un intervento chirurgico viene pianificato e provato completamente all'interno di un'ambiente virtuale sul computer. Questa procedura combinata con la stampa 3D ha dimostrato di essere uno strumento valido di visualizzazione e studio, sia per i casi più basilari che per quelli più complessi, offrendo al chirurgo l'opportunità di prevedere e prevenire potenziali complicazioni chirurgiche. È ormai altamente dimostrato come la pianificazione in ambito virtuale aumenti la precisione chirurgica limitando il rischio di infezioni e di danneggiamento dei tessuti vicini [1] [2] [3], elimini la necessità di utilizzare strumenti complessi e procedure che richiedono tempo in sala operatoria, riducendo anche la pressione sui tecnici chirurgici e sugli infermieri o di ricorrere alla fluoroscopia riducendo l'esposizione alle radiazioni dei membri dell'equipe chirurgica e del paziente [4].

Questa tipologia di pianificazione consiste principalmente nello sviluppo di due fasi e questo capitolo ha lo scopo di illustrarle nello specifico: la prima è relativa alla creazione di una replica dell'anatomia di interesse (Paragrafo 2.1) e la seconda riguarda l'elaborazione di un modello CAD di una guida di taglio a partire dalla scelta dei piani target (Paragrafo 2.2).

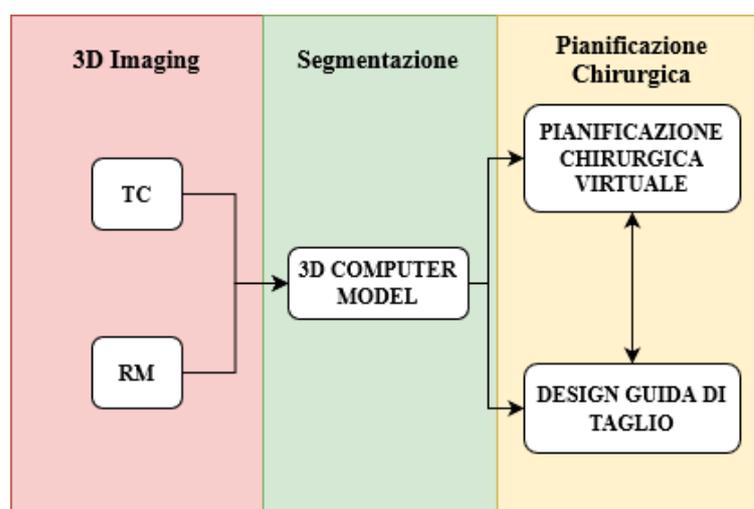


Figura 1: Panoramica del workflow dall'acquisizione dei dati alla pianificazione chirurgica

In questo lavoro di tesi ci si è concentrati principalmente sulla pianificazione di resezioni chirurgiche di tumori ossei in particolare tramite l'esecuzione di osteotomie. Con le modalità di *imaging* avanzate attualmente disponibili, i chirurghi possono identificare con precisione i

confini dei tumori ossei prima della resezione chirurgica. Il chirurgo può delineare un piano di resezione accurato e ideale prima dell'intervento per soddisfare i principi di resezione oncologica di base e massimizzare la conservazione del tessuto sano. Tuttavia, gli strumenti e le tecniche standard utilizzati dai chirurghi possono limitare la capacità di riprodurre in modo affidabile e coerente il piano preoperatorio ideale al momento della resezione con conseguenze potenzialmente gravi. Ad esempio, il chirurgo può inavvertitamente tagliare il tumore durante l'intervento chirurgico. Per soddisfare i principi oncologici di base, il clinico può scegliere di asportare più tessuto sano per evitare di toccare la zona affetta dal tumore, rischiando però di influenzare sostanzialmente l'esito funzionale [5].

In letteratura sono state segnalate diffuse tecniche che utilizzano **guide di taglio chirurgiche customizzate**. La guida di taglio si tratta di uno strumento che viene progettato e prodotto da un progettista CAD con competenze molto specifiche insieme al contributo dei chirurghi in interventi chirurgici complessi basandosi in partenza su scansioni TC (Tomografia Computerizzata) o RM (Risonanza Magnetica) preoperatorie e utilizzando tecniche di **progettazione assistita da computer/produzione assistita da computer (CAD/CAM)** e di **prototipazione rapida**. Realizzata in resina o in termoplastiche, la maschera si adatta alla superficie ossea in una sola configurazione possibile, orientando così il chirurgo verso la corretta posizione dell'osso per il corretto posizionamento della maschera. La maschera può essere progettata per guidare il percorso degli strumenti di taglio chirurgici, consentendo al chirurgo di riprodurre esattamente un piano preoperatorio ideale.

Il concetto di utilizzo di un modello specifico per il paziente è stato introdotto negli anni '90 da Radermacher et al. [6] per il posizionamento di viti peduncolari, l'artroplastica totale del ginocchio, la decompressione del rachide cervicale e la tripla osteotomia del bacino. Hanno eseguito la pianificazione preoperatoria basata su TC e hanno concepito un modello per adattarsi alla superficie ossea. Il modello è stato prodotto mediante fresatura perché la prototipazione rapida non era ancora ben sviluppata ed era molto più costosa. Anche nella chirurgia della ricostruzione maxillo-facciale, l'uso di un dispositivo paziente-specifico per guidare le osteotomie ha permesso al chirurgo di evitare strutture importanti, come i nervi dentali e componenti dell'anatomia vascolare [7].

C'è un progresso in diverse tecniche di prototipazione rapida come la stereolitografia (SLA), la modellazione a deposizione fusa (FDM) e la stampa tridimensionale (3DP), ma inizialmente le dime chirurgiche customizzate venivano prodotte in vari materiali plastici. Al giorno d'oggi, recenti tecniche avanzate come la sinterizzazione laser selettiva (SLS), la fusione laser selettiva (SLM), la sinterizzazione laser diretta del metallo (DMLS), la fusione con fascio di elettroni

(EBM) vengono utilizzate per produrre strumenti in forma metallica completamente densa. Il vantaggio di questo insieme di tecniche industriali sta nel riuscire a creare un prodotto finito superiore più rapidamente a un costo minimo rispetto alle tecniche convenzionali. Il design dipende in primo luogo dal tipo e dalle dimensioni dello strumento utilizzato dal chirurgo per realizzare un'osteotomia o un percorso di perforazione e dovrebbe consentire una tolleranza per lo strumento per evitare l'abrasione della guida e la formazione di detriti da usura [4].

2.1 Ricostruzione di un modello anatomico 3D: metodi e strumenti

La procedura parte dalla raccolta di immagini mediche, solitamente sotto forma di dati volumetrici che consistono in sezioni 2D sequenziali in cui sono rappresentate le strutture interne del corpo umano. Tali scansioni non saranno solo utili al fine di programmare l'intervento ma specialmente per l'ultima fase di questa procedura, ovvero di progettazione dello strumento chirurgico paziente-specifico, la cui accuratezza e adattabilità con l'osso di interesse dipenderà fortemente dalla qualità dello scanner utilizzato [4].

Le tecniche di *imaging* medica più comuni sono la TC, la RM seguite da ultrasuoni e PET.

Negli ultimi decenni la ricostruzione tomografica è stato il metodo più utilizzato per la ricostruzione di modelli tridimensionali. A gettare le basi matematiche per la tomografia 3D è stato Johann Radon, il quale definì la trasformata inversa di Radon utilizzata proprio per la ricostruzione di immagini tomografiche [8]. Le **immagini tomografiche** non sono altro che il risultato delle TC, RM e PET e si trattano di informazioni 3D digitali fornite sotto forma di *slice* trasversali dell'intero corpo o di parti di interesse. Trattandosi di contenuti digitali, ogni pixel delle *slice* rappresenta la posizione spaziale dell'elemento ripreso e il suo colore (valore di grigio o RGB), la reazione del tessuto alle radiazioni nel caso di TC e PET (raggi X o gamma – raggi) o al campo magnetico nel caso di RM.

Infatti la RM, a differenza della TC, non utilizza l'esposizione diretta del corpo umano ai raggi X, ma viene emesso un campo magnetico uniforme che applicato al corpo umano va ad agire direttamente sui protoni, presenti all'interno dei tessuti, riallineandoli. La successiva rimozione del campo magnetico fa sì che i protoni ritornino al loro stato originale a velocità variabile producendo onde radio (processo definito come precessione). Queste onde radio sono uniche per ciascun tessuto, in quanto strettamente dipendenti dalla velocità con cui i protoni perdono la loro energia magnetica, fornendo così immensi dettagli nelle intensità dell'immagine in scala

di grigi. Pertanto, la RM non produce radiazioni dannose rispetto alla TC ed è favorita rispetto a quest'ultima specie per produrre immagini ad alta risoluzione dei tessuti molli.

Le scansioni TC sono immagini a raggi X equi spaziate, distanziate in media di circa 5 mm l'una dall'altra, che riprendono una vista a 360 gradi dell'organo umano di interesse [9]. Lo scanner della TC è costituito da una sorgente di raggi X che emette una serie di fasci stretti di fotoni attraverso il corpo umano, dei quali alcuni sono assorbiti ed altri diffusi. I raggi X attenuati vengono raccolti dai rivelatori di raggi X posti sul lato opposto dello scanner. Questi dati vengono quindi trasferiti a un computer sotto forma di visualizzazioni bidimensionali in sezione trasversale sullo schermo, il computer effettua una loro elaborazione e produce in fine le immagini tomografiche. Il valore di ogni pixel in una TC rappresenta il valore di attenuazione o riduzione della trasmissione dei raggi X, che dipende dalla composizione atomica e dalla densità del tessuto attraversato. I valori di attenuazione sono espressi in unità Hounsfield (HU) che rappresenta la misura di attenuazione relativa all'acqua (Equazione 1). La scala Hounsfield è una scala lineare, che va da -1024 a +3071 HU. Il valore di ciascun elemento della scansione 2D viene calcolato come

$$\text{Hounsfield unit}(x, y) = 1000 \times \frac{\mu_{(x,y)} - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}} \quad (1)$$

dove μ_{water} e μ_{air} sono rispettivamente l'attenuazione lineare di acqua e aria. Il coefficiente di attenuazione dell'aria è quasi zero, quindi una variazione di un'unità Hounsfield (HU) può essere interpretata come una variazione dello 0,1% del coefficiente di attenuazione dell'acqua. Di conseguenza, è possibile distinguere facilmente varie densità che utilizzano diversi valori di attenuazione come aria, calcio, grasso e tessuti molli (fig. 2).

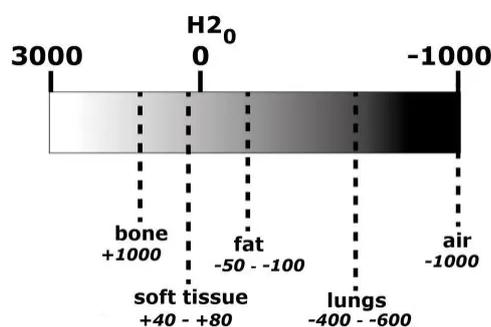


Figura 2: Hounsfield Unit

Tutti gli aspetti della pianificazione chirurgica assistita da computer richiedono l'uso di immagini ad alta risoluzione e la modalità di immagine selezionata. La risoluzione spaziale e la risoluzione del contrasto sono considerazioni importanti. La **risoluzione spaziale** è la

capacità di una modalità di *imaging* di distinguere tra due oggetti separati. Ad esempio, la distinzione tra il canale infraorbitario e la mascella circostante. Le immagini ad alta risoluzione spaziale sono più adatte per la segmentazione manuale e automatica delle immagini. La **risoluzione del contrasto** è la capacità di una modalità di *imaging* di distinguere tra intensità di immagini adiacenti. Ad esempio, la capacità di differenziare le aree di potenziamento del contrasto tumorale dai tessuti circostanti. La TC ha un'elevata risoluzione spaziale, inversamente proporzionale allo spessore delle *slice*, ma con una risoluzione di contrasto variabile e limitata che rende questa modalità l'*imaging* di scelta per interventi sui tessuti duri.

Le immagini ottenute vengono quindi archiviate in un formato standard DICOM (*Digital Imaging and Communications in Medicine*) che consente la condivisione universale tra sistemi hardware e software. I file DICOM non forniscono alcun layout dimensionale e sono costituiti da una serie di attributi che contengono: (i) informazioni relative al paziente (nome, sesso, età, ecc.), (ii) informazioni sulla modalità e sulla procedura di *imaging* (tipo di macchinario, parametri di acquisizione, dose di radiazioni, mezzo di contrasto, ecc.) e (iii) informazioni relative all'immagine (dimensione, risoluzione nel piano o dimensione dei pixel, spessore della *slice*, distanza della *slice*, ecc.).

2.1.1 Pre-processing e Segmentazione

Il successo della pianificazione chirurgica dipende fortemente dalla capacità del dataset di riprodurre accuratamente i dettagli anatomici e costruire i corrispondenti modelli 3D; pertanto a volte si rende prima necessario elaborare i dati grezzi con software di ripristino dell'immagine e dopo di miglioramento della stessa e solo alla fine ricorrere a software di modellazione medica che delineano le strutture anatomiche chiave e i loro punti di riferimento.

Spesso infatti le immagini mediche soffrono di diversi tipi di effetti di degradazione, come artefatti dovuti al movimento del paziente o interferenze, scarso contrasto, rumore e sfocatura. Gli **algoritmi di ripristino delle immagini** mirano principalmente a ridurre la sfocatura e il rumore nell'immagine elaborata, naturalmente correlati e introdotti dal processo di acquisizione dei dati. Le **tecniche di miglioramento dell'immagine** sono principalmente dedicate al miglioramento del contrasto al fine di estrarre o accentuare determinate caratteristiche. Lo scopo del processo di miglioramento è quello di migliorare la comprensione del contenuto informativo e ottenere un'immagine più adatta dell'originale per l'elaborazione automatizzata delle immagini, come nel caso di immagini mediche per evidenziare strutture (tessuti, organi, ecc.) di interesse [10].

A questo punto, si prosegue con l'importazione dei dati DICOM in un software di modellazione. La regione di interesse deve essere partizionata digitalmente rispetto al tessuto circostante tramite un processo noto come **segmentazione**, che consiste nel partizionamento di un'immagine in più parti, in cui ciascuna di queste parti è una raccolta di pixel (o voxel) corrispondenti a una particolare regione con proprietà simili come livello di grigio, colore, trama, luminosità e contrasto. Separare la regione di interesse in singoli componenti permette di identificarli e manipolarli digitalmente in maniera più semplice. Questa fase richiede tradizionalmente uno sforzo considerevole da parte di tecnici o medici qualificati: sia mediante l'identificazione manuale *slice per slice*, che richiede molto tempo, sia mediante modifiche sostanziali delle segmentazioni risultanti dagli strumenti semi-automatici. In letteratura è stata proposta un'ampia varietà di tecniche di segmentazione. Gli algoritmi di segmentazione tradizionali utilizzano tecniche come *thresholding*, *region growing*, *edge detection* e *clustering*.

La **segmentazione manuale** è l'approccio che richiede più tempo, e questo dipende in gran parte dal numero di strutture segmentate ma anche dalla risoluzione dell'immagine (un'alta risoluzione equivale a più *slice* da esaminare), dall'hardware di interfaccia (utilizzando il mouse di un computer rispetto all'utilizzo di un display interattivo) e dalla tecnica dell'utente [11]. Per quest'ultimo aspetto c'è anche da aggiungere che la segmentazione manuale è molto sensibile alla variabilità intra- e inter-operatore, necessitando quindi di operatori esperti dedicati. In conclusione, la segmentazione manuale è impraticabile per un utilizzo routinario. Nonostante ciò rimane uno degli approcci più flessibili rispetto alla **segmentazione automatica** e **semiautomatica** [12] che, in base all'ambito di applicazione, hanno dimostrato di essere veloci e più accurati dal punto di vista della riproducibilità con variabilità e inconsistenza ridotte rispetto al metodo manuale. Sfortunatamente, la variabilità anatomica, in particolare per quanto riguarda la vascolarizzazione, e i problemi di immagine intrinseca limitano l'affidabilità degli approcci completamente automatici. La segmentazione semiautomatica, invece, richiede l'interazione dell'utente per impostare parametri come *seed point* o soglie e per verificare ed eventualmente ripetere la procedura con un'impostazione diversa in funzione del risultato. Più metodi di segmentazione potrebbero essere utilizzati insieme su un singolo modello 3D per eliminare le carenze minori intrinseche a ogni algoritmo [13], infatti anche se l'approccio manuale è lento può comunque essere utile il successivo utilizzo di uno degli altri due metodi per poter rifinire i contorni delle strutture anatomiche [14].

Attualmente, l'elaborazione delle immagini tomografiche viene eseguita di routine con software commerciali e tra questi il più utilizzato e all'avanguardia è **Materialise Mimics** (*Materialise Interactive Medical Image Control System*) [15]. È un software di elaborazione,

progettazione e modellazione di immagini 3D, sviluppato da Materialise NV, una società belga specializzata in software e tecnologie di produzione additiva per i settori medico, dentale e della stampa 3D. Nella pratica, Materialise Mimics consente di convertire i file DICOM in layout 3D e di eseguire la segmentazione sfruttando gli strumenti proposti dal software [16].

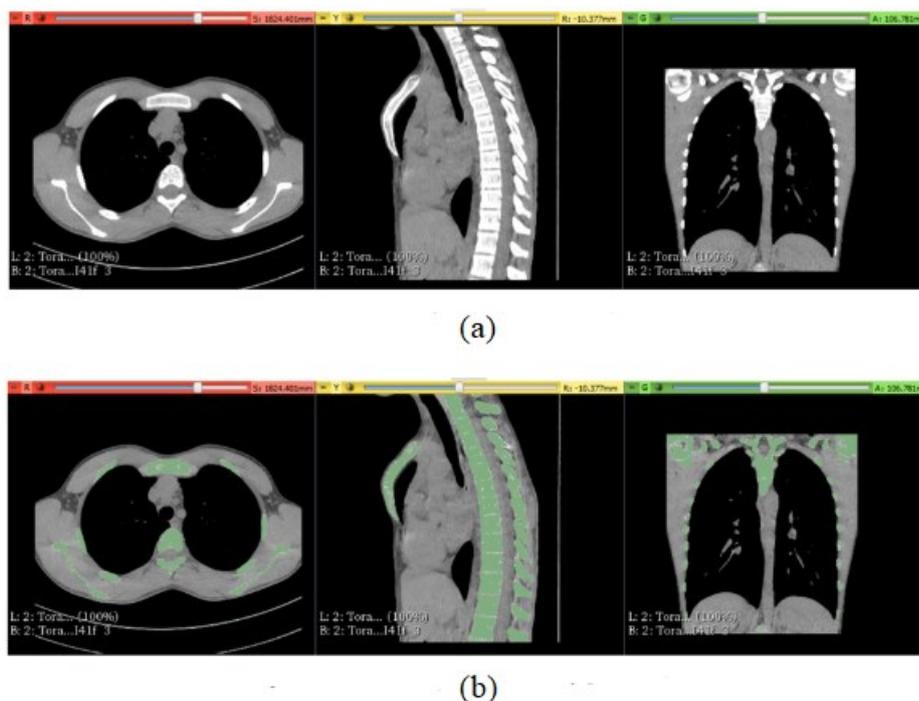


Figura 3: (a) scansione TC del paziente; (b) esempio segmentazione della scansione TC [17]

I confini del tessuto possono essere identificati su immagini 2D successive (Fig. 3) che vengono successivamente collegate per formare una superficie chiusa e memorizzate nel formato STL (*Standard Tessellation Language* o *Standard Triangle Language*).

2.1.2 Post processing del modello 3D

Una volta che l'anatomia è stata segmentata, tuttavia, non è ancora adatta per essere impiegata a causa dei limiti del processo di segmentazione. Questi includono (i) effetti di volume parziale, cioè artefatti che si verificano quando diversi tipi di tessuto si mescolano insieme in un singolo pixel e risultano in confini non netti, (ii) disomogeneità di intensità di una singola classe di tessuto che varia gradualmente nell'immagine, producendo un effetto di sfumatura, (iii) presenza di artefatti di movimento e segnale, che fanno sì che i risultati della segmentazione siano spesso caratterizzati da forme irregolari e imperfette, (iv) somiglianza dei valori di grigio per diversi tessuti. Pertanto, è necessario affrontare una fase il cui grado di difficoltà è proporzionale alla complessità della geometria coinvolta. La sfida sta nel rimuovere gli artefatti

mantenendo la fedeltà all'anatomia del paziente poiché gli errori in questa fase possono anche portare a guide/impianti per osteotomia inadeguati. Il file STL passa quindi alla fase di modellazione della superficie della mesh per rimuovere errori tramite ulteriori software. Tra i pacchetti software commerciali il più utilizzato è **Geomagic Studio** di 3D Systems [18]. In alternativa, risultati simili sono ottenibili utilizzando strumenti open source gratuiti come **MeshLab** [19].

Tra le diverse procedure di ottimizzazione si possono includere quelle relative alla riduzione del numero di facce, mantenendo allo stesso tempo il livello di dettaglio e risoluzione del modello originale, e algoritmi di *smoothing* che consentono di smussare una superficie provando al contempo di preservare il volume e i lineamenti del solido generato.

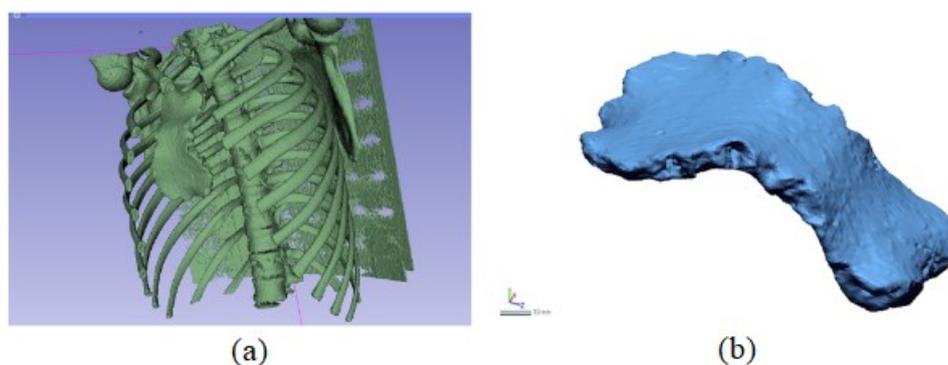


Figura 4: (a) modello 3D risultante dalla segmentazione; (b) modello STL isolato dello sterno [17]

2.2 Fase di design CAD della simulazione dell'intervento

Una volta disponibile il modello 3D dell'anatomia del paziente, gli esperti CAD progettano virtualmente la migliore posizione e l'inclinazione dei piani che definiscono il taglio seguendo le indicazioni del chirurgo. Tale scelta si basa principalmente sia sulla relazione tra lesione e strutture circostanti sia sull'idea di preservare la massima quantità di osso sano tenendo conto di un certo margine di sicurezza. Il chirurgo può quindi visualizzare le osteotomie (angoli di taglio, dimensioni di resezione), il riposizionamento osseo e il risultato finale atteso. La simulazione virtuale consente di esplorare infinite variazioni, confrontando varie tecniche, senza il tempo e le risorse necessarie per creare una stampa 3D per ogni nuova simulazione, anche se la combinazione della pianificazione virtuale con prove chirurgiche su modelli stampati in 3D offre i vantaggi di entrambi gli approcci per i casi più complessi [3] [20].

La simulazione virtuale, così come la progettazione della guida di taglio, può essere effettuata utilizzando uno dei pacchetti software di *Reverse Engineering* disponibili **Geomagic Design X**

[18]. Le valutazioni sulla forma e sulla dimensione della dima tengono conto sempre della grandezza del taglio che si vuole effettuare, dello spessore e deformità dell'osso che si vuole recidere, della posizione e della comodità del chirurgo. Generalmente, questo dispositivo medico è composto da due componenti: (i) la guida, cioè cilindri/fori per l'utensile di perforazione (punta da trapano) o fessura di taglio per utensile da taglio (sega, fresa) o entrambi (unione della guida di taglio e di foratura) e (ii) la base [4]. Il modello e la dimensione della lama influenzano sia l'altezza che lo spessore della fessura. L'altezza deve garantire una corsa sufficiente senza interferire con l'attrezzo, mentre lo spessore della fessura deve essere pari allo spessore della lama più il gioco per evitare fenomeni di usura. Al fine di bloccare la dima sono inclusi anche dei fori per l'alloggiamento delle viti autofilettanti i quali, per garantire stabilità, è importante che non siano paralleli tra loro. Nell'utilizzo di questi strumenti è associato uno svantaggio in quanto i fori lasciati per fissare la guida potrebbero potenzialmente aumentare il rischio di fratture e richiedere a loro volta restrizioni postoperatorie più severe. Infine, la base deve possedere uno spessore tale da garantire una buona resistenza e rigidità minimizzando massa e materiale [21].

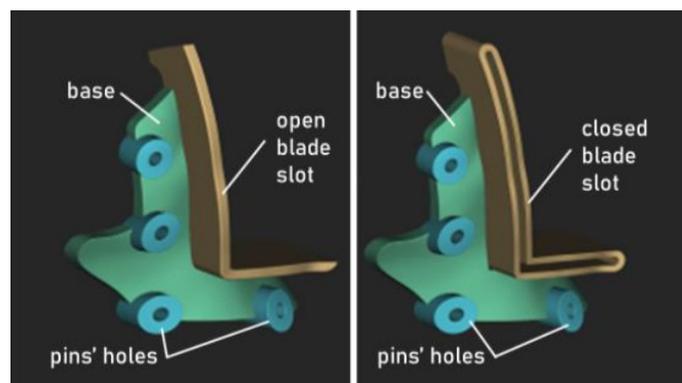


Figura 5: Componenti principali di una dima chirurgica di taglio: fessura di taglio, base e fori [21]

Durante il processo di progettazione (Figura 7), tutti gli schizzi vengono tracciati sul piano di base (Fig. 7b), la cui normale è la direzione di taglio fornita dal chirurgo. Se sono presenti più direzioni di taglio, è necessario considerare diversi piani di base, i quali possono essere creati separatamente e uniti alla fine. Per garantire un posizionamento rapido e univoco della guida chirurgica sulla parte anatomica di interesse e, allo stesso tempo, per evitare disallineamenti, è necessario che il primo aspetto da considerare sia che l'interfaccia tra la base e l'anatomia sia coincidente. Poiché molto spesso l'anatomia del corpo può risultare irregolare, il primo passaggio che viene svolto è quello di rendere più regolare e continua la mesh che è stata precedentemente selezionata ed isolata attraverso strumenti di *Reverse Engineering* (ad esempio riempimento di fori, pulizia, levigatura e correzione dei confini), accertandosi sempre

di conservare le principali conformazioni anatomiche della superficie di interesse. La fase successiva prevede la definizione della fessura della lama: dapprima ogni segmento tracciato sulla superficie ossea dal chirurgo viene rifinito per ottenere una linea (formata da uno o più segmenti rettilinei incidenti) che viene poi estrusa secondo la direzione di taglio. L'estrusione viene eseguita sia attraverso l'osso che in direzione opposta almeno fino all'altezza prescelta della guida. La superficie viene quindi ispessita, come spiegato sopra, del valore dello spessore della lama più il gioco desiderato. Dopodiché, deve essere abbozzato il profilo esterno della fessura. Questo passaggio è solitamente strettamente correlato all'esperienza dell'utente e il risultato può cambiare di conseguenza. La fessura della lama si ottiene eseguendo un'operazione booleana tra questa estrusione e la superficie ispessita che rappresenta la dima. Una volta che l'asola della lama è stata completamente progettata, è necessario definire la posizione dei fori. Solitamente, il progettista prima disegna tutti i cerchi centrati negli assi dei fori e successivamente esegue l'estrusione e il loro "foraggio" tramite operazioni di taglio; il diametro di ciascun cerchio è uguale alla somma del diametro dei perni scelti più lo spessore minimo sufficiente a garantire una fissazione stabile. In questo modo, includendo questi cerchi all'interno del disegno del profilo di base, è possibile garantire il corretto fissaggio senza successivi aggiustamenti.

In fase di design i valori utili per definire vincoli di progettazione riguardano i) il diametro dei perni; ii) lo spessore della lama; iii) l'altezza della fessura della lama; iv) lo spessore della base; v) il gioco lama-dima di taglio. Per minimizzare i parametri di input, l'altezza dei fori del perno è fissa e imposta pari a 10 mm. Gli input richiesti per la procedura proposta sono i) le superfici di taglio, ottenute estrudendo il percorso di taglio lungo la direzione di taglio fornita dal chirurgo; ii) il contorno di base, proiettato su un piano perpendicolare alla direzione di taglio; iii) gli assi dei perni chirurgici consegnati come una coppia di punti; iv) la maglia triangolare dell'osso bersaglio, estratta dalla diagnostica per immagini.

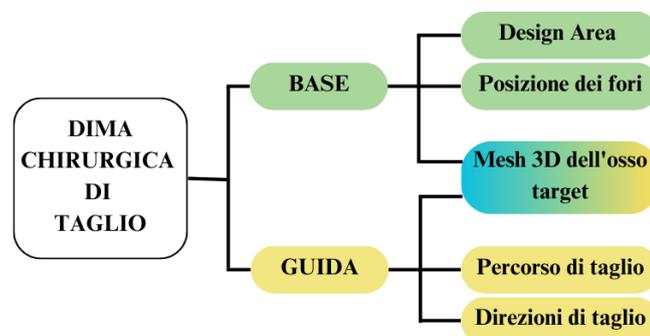


Figura 6: Workflow tipico per la progettazione della dima chirurgica di taglio basata su CAD

I progetti finali vengono dunque esportati come formato 3D STL di alta qualità pronti per la prototipazione rapida. Pertanto con la disponibilità di modelli CAD dell'organo del paziente, della guida chirurgica e dello strumento chirurgico (sega oscillante), è possibile simulare virtualmente l'intervento chirurgico insieme a diversi compiti come, ad esempio, l'accessibilità della lama nella guida e la sua corretta penetrazione [22]. Un esempio significativo di un metodo semi-automatico per progettare dime di taglio per sternotomie nel caso di petto arcuato è presentato in [23]. Gli autori propongono una procedura semi-automatica sviluppata con Rhinoceros®, un modellatore di superfici, insieme al suo plugin di modellazione parametrica Grasshopper®. La procedura si basa su una serie di rigorose regole geometriche basate sull'anatomia dello sterno per produrre una guida chirurgica di taglio su misura per correggere il difetto dell'osso originale, evidenziando la velocità del metodo rispetto ai processi tradizionali e il vantaggio di offrire un nuovo strumento per rendere i clinici più indipendenti.

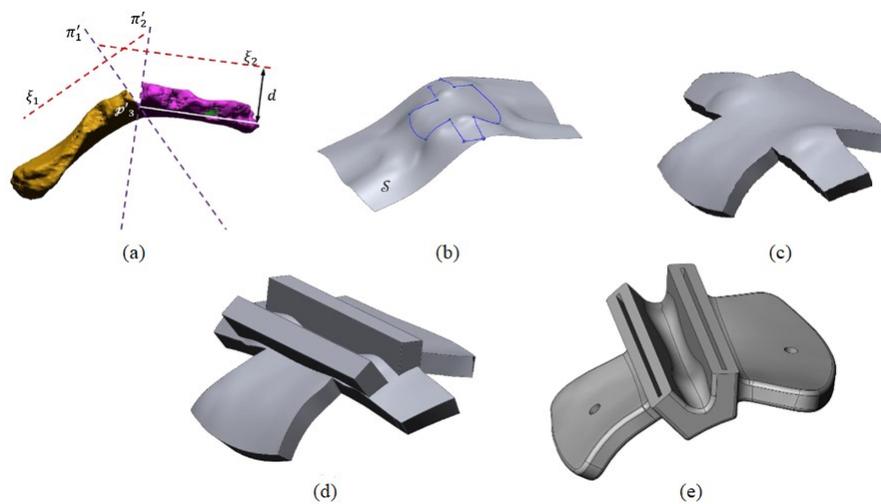


Figura 7: (a) definizione dei due piani; (b) contorno chiuso ottenuto proiettando in superficie lo schizzo 2D disegnato sui due piani; (c) primo tentativo della guida chirurgica; (d) primo tentativo della guida chirurgica con rettangoli estrusi; (e) modello 3d finale della guida chirurgica [17]

Capitolo 3

Sistemi di realtà virtuale

3.1 Visori di realtà virtuale

Molti termini sono usati per descrivere gli ambienti virtuali; questi includono la "realtà artificiale", "cyberspazio", "VR", "mondi virtuali" e "ambiente sintetico". Tutti questi termini si riferiscono a un'applicazione che consente al partecipante di vedere e interagire con oggetti distanti, costosi, pericolosi o ambienti tridimensionali altrimenti inaccessibili. Un traguardo importante nello sviluppo di questi sistemi virtuali è, per l'esperienza sensoriale e interattiva dell'utente, avvicinarsi a una simulazione credibile del reale.

In questo capitolo si è cercato di capire l'evoluzione e gli utilizzi della realtà virtuale, con l'obiettivo di evidenziare il potenziale di questa tecnologia nell'ambito della pianificazione chirurgica. Successivamente si è spiegato nel dettaglio il modello del visore che si è usato per questo lavoro di tesi, sia dal punto di vista hardware che software, insieme alla piattaforma grafica impiegata per lo sviluppo del simulatore.

3.1.1 Storia

Quando parliamo di Realtà Virtuale ci vengono subito in mente i dispositivi di ultima generazione che garantiscono un'esperienza totalmente immersiva. L'Oculus, l'HTC e il PlayStation VR sono in realtà figli di una tecnologia che ha visto l'origine del suo termine nel 1989 ma la cui nascita in realtà risale agli anni '60. Di fatti l'idea di volersi connettere a un mondo virtuale esisteva già da prima, ma solo in quel periodo sono cominciati i primi tentativi pratici. Si parla di dispositivi molto più ingombranti di quelli attuali, ma condividevano lo stesso desiderio di regalare all'uomo un'esperienza immersiva e coinvolgente.

Il primo vero dispositivo ottico fu quello ideato da Morton Heilig nel 1960: la *Telesphere Mask*. Si trattava di un dispositivo brevettato come "*Stereoscopic-Television Apparatus For Individual Use*" di forma portatile che poteva essere indossato sulla testa dell'utente. La *Telesphere Mask* comprendeva un paio di lenti stereoscopiche a cui dietro sono stati aggiunti due tubi di raggi catodici, assieme ad un paio di cuffie esterne e un paio di ugelli di scarico dell'aria, tutti cooperanti per consentire all'utente di osservare immagini, percepire gli effetti sonori e le correnti d'aria di velocità, temperatura e odori variabili [24].

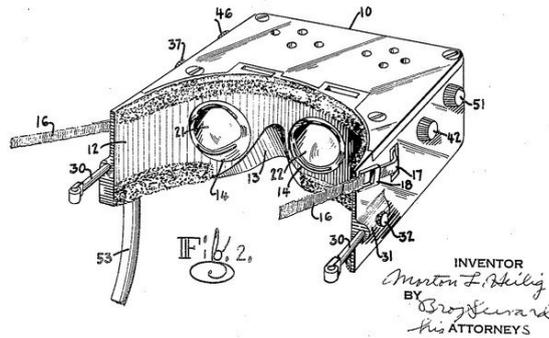


Figura 8: Telesphere Mask

Nel 1962 Heilig presentò come prototipo un secondo dispositivo chiamato *Sensorama*; una macchina complessa creata per il cosiddetto “cinema d’esperienza”. Accompagnava la visione di un film di breve durata grazie ad un sistema di lenti stereoscopiche che ricreavano l’effetto 3D delle riprese, con sedile mobile, vibrazioni, bocchette per la presa d’aria e diffusori di profumi, così che lo spettatore si sentisse come direttamente presente nella scena anche se incapace di interagire con esso. Un progetto estremamente innovativo, fin troppo avanti rispetto al suo tempo, poiché in grado di stimolare non solo i sensi della vista e dell’udito, ma anche dell’olfatto e del tatto. La macchina però aveva un costo eccessivo e non ricevette nessun finanziamento dalle case cinematografiche statunitensi, così il tutto si risolse con la realizzazione di una versione portatile che venne successivamente brevettata.



Figura 9: Sensorama

Anche se si tratta di realtà aumentata e non realtà virtuale, è doveroso citare quello che viene considerato il primo visore per la realtà aumentata, la “Spada di Damocle”, inventato tra il 1965 e il 1968 dall’informatico statunitense Ivan Sutherland, conosciuto come “*padre della computer grafica*”. Strutturalmente il prototipo era caratterizzato dai seguenti elementi: due tubi di raggi catodici che scendono verticalmente sopra la testa dell’utente ed attaccati al soffitto (caratteristica che ne definisce il nome), un computer usato come matrice principale, un visore che mostra l’oggetto tridimensionale e i sensori di posizionamento della testa. Questi ultimi rilevano e forniscono le coordinate del posizionamento dell’individuo all’interno della stanza, il quale è comunque condizionato dai tubi catodici che ne limitavano fortemente il movimento. Le coordinate vengono lette dal computer che le elabora assieme agli algoritmi dell’oggetto costruito tridimensionalmente grazie all’uso di una matrice moltiplicatrice che calcola e realizza l’immagine, passando dalle coordinate della stanza alle coordinate dell’occhio umano. Pertanto era in grado di mostrare un mondo costruito tridimensionalmente e nel quale l’utente che ne fa esperienza ha la possibilità di girare la testa, cambiare quindi la percezione e la prospettiva di ciò che sta osservando. Successivamente Sutherland superò il problema della poca maneggevolezza creando dei video-caschi, più leggeri e all’avanguardia rispetto al visore precedente, che usavano trasmettitori ultrasonici per tracciare il movimento.

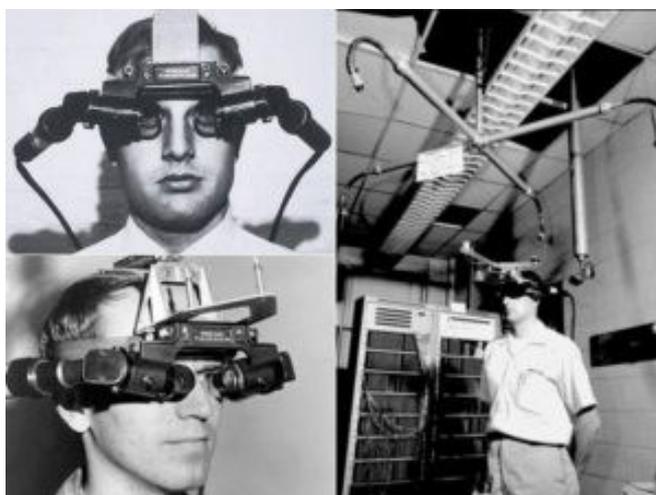


Figura 10: Spada di Damocle

L’espressione ossimorica *Virtual Reality* deve la sua divulgazione a Jaron Lanier, *Computer Scientist* e *Visual Artist*. L’utopia di Lanier e dei pionieri della realtà virtuale era quella di aumentare la percezione dell’utente di essere parte integrante di un mondo simulato, circondato da immagini e da suoni che contribuiscono alla sensazione di presenza e di immersività. D’altra parte, le ricerche di Lanier erano partite nei primi anni Ottanta alla Atari, la leggendaria azienda di videogiochi; nel suo periodo di massimo splendore fondò insieme a Thomas Zimmerman la

start up *VPL Research Inc.* dedicata alla ricerca sulle tecnologie di realtà virtuale. Ci si trova in un periodo storico nel quale il visore inizia ad assumere connotati tecnici e stilistici più simili ai modelli attualmente venduti sul mercato con monitor dotati di cristalli liquidi. Per di più fu qui che le ricerche dei due ricercatori avevano messo a punto le prime forme di controller esterni indossabili. Tra questi c'era il *DataGlove*. Si trattava di un guanto speciale che convertiva i movimenti della mano in segnali elettrici da trasmettere attraverso fibre ottiche [25].



Figura 11: Jaron Lanier indossa un guanto con sensori in fibra ottica per il trasferimento dei movimenti in realtà virtuale

Uno dei primi progetti in cui il *DataGlove* è stato incorporato fu il *Virtual Environment Workstation* risalente al 1985. Il software per il primo prototipo del sistema è stato sviluppato da Warren Robinett in collaborazione con la NASA. Includeva un elmetto in grado di mostrare sia ambienti virtuali che reali, rendendolo così un visore sia di realtà aumentata che virtuale. Il guanto in questo caso misurava gli angoli di flessione delle articolazioni delle dita e includeva anche un tracker per misurare la posizione e l'orientamento della mano. Pertanto, la mano quantata dell'utente poteva interagire con il mondo virtuale, afferrando oggetti virtuali o utilizzando i gesti delle mani come comandi. Possedeva due schermi da 2.7 pollici ciascuno, che fornivano un angolo di visione di 120°.



Figura 12: Virtual Environment Workstation

Nel 1991, il *Virtuality1000CS*, probabilmente il dispositivo VR più influente degli anni '90, arrivò nelle sale giochi. Questo è stato un notevole salto nel contenuto VR a livello di consumatore. Utilizzava un computer Amiga 3000 per gestire la gestione dei calcoli e possedeva un visore per video e audio. In dotazione c'era un *joystick* 3D con il quale gli utenti si muovevano ed interagivano con il mondo virtuale. Il costo pari a 60 mila dollari fu ciò che causò però vendite insufficienti e quindi il fallimento del progetto.



Figura 13: Virtual Reality 1000CS

Il fatto che il costo fosse troppo alto, e quindi causa di un fallimento in sé, non lo rende un caso isolato, anzi. Durante gli anni '90 si susseguirono una serie di visori dal design più pulito, lineare e simile a quello degli attuali visori, ma nessuno di questi fu un vero e proprio successo.

Questo periodo, infatti, vide un forte rallentamento della virtualità in seguito alla comprensione di ricercatori e investitori sull'arretratezza della tecnologia impiegata, legata al fatto che era necessario avere processori che mantenessero costante il render grafico del mondo virtuale 3D e seguissero contemporaneamente i movimenti della testa e del corpo con una velocità giusta al fine di non acuire il senso di nausea e vertigine che può crearsi con l'uso della realtà virtuale [25]. Dunque l'elettronica degli anni '90 imponeva grossi limiti di calcolo, troppo stringenti per prodotti soddisfacenti e allo stesso tempo accessibili al mercato di massa.

Il vero boom della realtà virtuale arriverà solamente nel 2012 dando il merito ad un teenager americano, Palmer Luckey, che nel garage di casa, mescolando e abbinando diverse parti della sua collezione di hardware per VR, ha perfezionato l'attrezzatura di tracciamento del movimento della testa in tempo reale e realizzando quindi un prototipo di visore rivoluzionario. Ed è stato quando il leggendario programmatore di videogiochi e fondatore di id Software John Carmack si è imbattuto nel suo lavoro online. Rimase colpito dal lavoro di Luckey al punto da portarlo all'E3, il carnevale commerciale annuale dell'industria dei giochi, in cui diede dimostrazione del suo visore. Poco dopo Luckey lanciò una campagna Kickstarter che ha sfruttato un finanziamento di 2.4 milioni di dollari, unendosi dopo due anni e mezzo all'impero di Facebook. Da quel momento in poi, i dispositivi VR migliorarono sempre di più e vennero presi in considerazione da altre compagnie legate al mondo del gaming, come ad esempio la Sony.

3.1.2 Classifica dei visori

È possibile suddividere i visori presenti in commercio principalmente in tre macro-categorie, la cui descrizione è riportata nei prossimi paragrafi.

3.1.2.1 Visori VR tethered

I visori VR cablati sono dispositivi fisicamente connessi ad un computer tramite cavi, come HDMI e/o USB, e il loro funzionamento è garantito solo tramite questo collegamento che alimenta il visore stesso ricevendo al contempo l'esperienza di realtà virtuale. I visori VR con *tethering* sono attualmente molto più coinvolgenti rispetto ad altri grazie all'esperienza di alta qualità che possono offrire. Questi visori richiedono una certa quantità di spazio di configurazione oltre a una connessione via cavo costante a un potente PC da gioco, per assicurare al contempo una migliore accuratezza del tracking e qualità grafica. Un esempio di visore cablato è l'HTC Vive. Questo modello richiede l'utilizzo di sensori da apporre attorno

alla stanza, in posizione degli angoli superiori, per consentire il tracciamento della posizione a 360° e la rilevazione del movimento nella realtà con la sua successiva traduzione in movimento nella realtà virtuale.



Figura 14: HTC Vive

Altri visori cablati includono *Oculus Rift* (tracciamento a 180° se non si acquistano sensori aggiuntivi) e *PlayStation VR*, entrambi solitamente utilizzati sedendosi su una sedia girevole e limitando quindi lo spazio di gioco.

Guardando ai vantaggi che questi visori offrono vediamo:

- La comodità nel condividere la vista del visore su uno schermo più grande;
- Una maggiore potenza rispetto ad un visore wireless;
- Una grafica di fascia alta con illuminazione ed effetti in tempo reale.

Gli svantaggi dei visori VR *tethered* includono:

- Limitatezza nei movimenti e nella distanza dal computer;
- Costo più elevato;
- Necessità di computer predisposti e con specifici requisiti e/o apparecchiature aggiuntive come *base stations* e *lightbox* installate all'interno della stanza.

Tabella 1: Comparazione tra i principali visori VR tethered e le loro specifiche [26]

					
	Oculus Rift S	HTC Vive Pro 2	Valve Index VR Kit	Playstation VR	HP Reverb G2
Risoluzione / occhio	1280 x 1440	2448 x 2448	1440 x 1600	960 x 1080	2160 X 2160

Frequenza di aggiornamento (Hz)	80	120	144	90 / 120	90
Tipologia schermo	LCD	LCD	LCD	OLED	LCD
Campo visivo	110°	120°	130°	100°	114°
Peso	561 g	850 g	570 g	610 g	550 g
Room-scale tracking	Inside - out	Outside - in	Outside - in	Inside - out	Inside - out
Sistema richiesto	PC	PC	PC	Playstation 4 or Playstation 5	PC
Prezzo	\$399	\$1.399	\$999	\$349	\$399

3.1.2.2 Visori VR stand-alone

Al contrario dei precedenti, questa categoria di visori funziona completamente senza utilizzo di un PC o di cavi, di fatto includono al loro interno processori, sensori e telecamere per il *tracking*, batterie, memoria di archiviazione e display integrati [27]. Facoltativamente, per una maggiore potenza di elaborazione, i visori wireless possono essere forniti con un cavo per la connessione a un PC; ad esempio l'*Oculus Link* o *SteamVR* consentono di configurare e avviare la propria attività direttamente dal proprio computer sfruttando una maggior potenza di elaborazione del computer per frame rate più elevati ed una grafica migliore.

I vantaggi che questi offrono sono:

- Maggiore comfort e mobilità per l'utente;
- Configurazione più semplice rispetto ai dispositivi collegati al computer;
- Costo inferiore.

Gli svantaggi che questi visori offrono sono:

- Meno potenza dei visori cablati;
- Grafica ed effetti di qualità inferiore;
- Frequenze di aggiornamento inferiori.

Tuttavia, varie importanti aziende tecnologiche come Google, Facebook e HTC sembrano concentrare sempre più sforzi su questa categoria VR rispetto a quelli cablati.

Tabella 2: Comparazione tra i principali visori VR stand-alone e le loro specifiche [26]

				
	Oculus Quest 2	Oculus Quest Pro	Pico Neo 3 Pro	HTC Vive Focus 3
Risoluzione / occhio	1832 x 1920	1800 x 1920	1832 x 1920	2448 x 2448
Frequenza di aggiornamento (Hz)	60, 72, 90, 120	72, 90	90	90
Tipologia schermo	LCD	LCD	LCD	LCD
Campo visivo	89°	106°	98°	120°
Peso	503 g	722 g	620g	785
Room-scale tracking	Inside - out	Inside - out	Inside - out	Inside - out
Sistema richiesto	Nessuno (PC opzionalmente compatibile)	Nessuno (PC opzionalmente compatibile)	Nessuno (PC opzionalmente compatibile)	Nessuno (PC opzionalmente compatibile)
Prezzo	\$449	\$1.499	\$449	\$1.300

3.1.2.3 Visori VR mobile

I visori VR per smartphone, come indica il nome, utilizzano gli smartphone per fornire un'esperienza di realtà virtuale. Gli utenti devono semplicemente inserire il loro smartphone all'interno del visore; lo schermo sarà proprio davanti agli occhi dell'utente, con una serie di lenti che creano un senso di profondità.

La qualità dell'esperienza VR è strettamente dipendente dallo smartphone utilizzato e di fatti gli smartphone rilasciati di recente sono anche i più adatti perché hanno maggior potenza. Altri fattori variabili includono il tipo di schermo e la sua risoluzione.

È ragionevole pensare che questa categoria di visori non è in grado di fornire lo stesso tipo di esperienza e interazione virtuale in termini di senso di immersività ed illusione. Dato che si

trattano di dispositivi senza cavo, la posizione del cellulare sarà dedotta dall'accelerometro integrato al cellulare e i movimenti compiuti nel mondo reale non saranno di conseguenza tradotti in movimenti nella realtà virtuale.

Tabella 3: Comparazione tra i principali visori VR mobile e le loro specifiche [26]

			
	Google Daydream View	Google Cardboard	Samsung GearVR
Risoluzione per occhio	Dipende dal dispositivo	Dipende dal dispositivo	1280 x 1440
Frequenza di aggiornamento (Hz)	Dipende dal dispositivo	60 o più	90
Tipologia schermo	Dipende dal dispositivo	Dipende dal dispositivo	AMOLED
Campo visivo	100°	80°	96°
Peso	261 g	96 g	318 g
Room-scale tracking	Nessuno	Nessuno	Nessuno
Sistema richiesto	Smartphone Android	Smartphone Android e Apple	Smartphone Samsung
Prezzo	\$79	\$15	\$100

Le caratteristiche che permettono di distinguere i visori sulla base della fascia di prezzo in visori economici, fascia media e fascia alta in genere sono legati a caratteristiche tecniche quali:

- *Risoluzione dello schermo*: generalmente sono menzionati o il numero totale di pixel o il numero di pixel per grado. Migliore è la risoluzione e più realistica apparirà l'immagine.
- *Frequenza di aggiornamento (o refresh rate)*: numero di volte in cui un display è in grado di aggiornarsi in un secondo ed è misurata in Hertz (Hz). Maggiore è la frequenza

di aggiornamento, migliore è la qualità dell'esperienza visiva. Il valore minimo consigliato per questo parametro è di 60 Hz, questo perché aiuta a ridurre il *motion sickness*, ovvero la sensazione di nausea e disorientamento che può colpire il soggetto durante l'utilizzo del dispositivo [28]. I display con oltre 120 Hz invece possono offrire un'esperienza visiva più fluida e confortevole.

- *Tipologia di display*: il più diffuso è il display LCD, ma si stanno sempre più adottando i display OLED.
- *Campo visivo*: definisce quanta parte della visuale dell'utente è coperta dal display e dall'ottica di un determinato visore. Gli esseri umani hanno un campo visivo di circa 200 gradi. Il campo visivo medio tra i migliori visori VR è di circa 100 gradi, con variazioni di circa 10 gradi in entrambe le direzioni. Maggiore è il campo visivo, più si è in grado di vedere intorno a sé e migliore sarà il visore che ricreerà la visione umana naturale [29].
- *Room-scale tracking*: prevede due possibili tipi di tracciamento *outside-in* e *inside-out*. Con *outside-in*, il visore VR e i controller vengono monitorati da un dispositivo esterno, a volte chiamato *base stations* o "fari". Ciò consente un *room-scale tracking* più accurato con precisione millimetrica. Con il tracciamento *inside-out*, il visore VR e i controller vengono tracciati tramite sensori o telecamere integrati nel visore stesso. Ciò offre la comodità di non aver bisogno di hardware aggiuntivo per la configurazione, comportando però un tracciamento leggermente meno accurato [30].

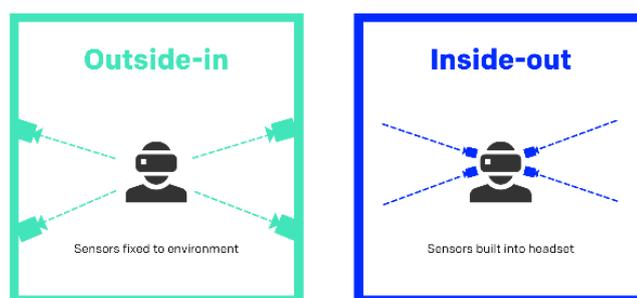


Figura 15: Sistema di tracciamento *outside-in* e *inside-out*

3.1.3 Realtà Virtuale in chirurgia

L'esercizio della medicina è un processo decisionale complesso che necessita conoscenza, capacità manuali e esperienza; un percorso lento che richiede diversi anni e metodi di formazione che agevolano gli studenti nel migliorare le loro prestazioni [31].

Queste abilità devono essere apprese e, nel migliore dei casi, allenate, eseguendo ripetutamente la stessa procedura. A causa della riduzione dell'orario di lavoro, dell'aumento dei costi e dei problemi di sicurezza del paziente, i tirocinanti spesso non possono raggiungere l'esperienza richiesta quando eseguono queste procedure solo in sala operatoria. Pertanto, la simulazione delle procedure chirurgiche è diventata molto importante e sostenuta come un problema di sicurezza del paziente nell'educazione chirurgica.

Il rapido sviluppo tecnologico ha portato a nuove forme di simulazione chirurgica in contesti simili a quelli reali, consentendo ai chirurghi di eseguire procedure in chirurgia aperta su simulatori fisici come modelli sintetici, cadaveri umani e animali vivi o su simulatori basati sulla VR e AR. I vantaggi dell'uso delle simulazioni virtuali sono numerosi e se ne evincono maggiormente nell'ambito dell'apprendimento delle abilità procedurali, le cosiddette "simulazioni cognitive", che vertono all'insegnamento delle procedure, di come utilizzare gli strumenti corretti, secondo quale ordine, con quale mano e in quale momento, senza tenere conto dei requisiti di abilità tecnica [32]. Il tirocinante può apprendere le procedure e le tecniche chirurgiche passo dopo passo in un ambiente confortante, può comprendere e confermare concetti, fare diagnosi più precise e nel minor tempo e pianificare procedure a basso rischio prima di intervenire direttamente sul paziente in sala operatoria [33] [34]. Il tutto è favorito dal fatto che gli utenti sono in grado di percepire l'ambiente virtuale circostante come se si trattasse di un ambiente reale che invia segnali intuitivi come la collisione con ostacoli, l'attrito, l'inerzia, il rendering 3D e il suono. È importante quindi sviluppare un'esperienza virtuale multisensoriale accurata [34]. A volte però è stato dimostrato che è necessario sacrificare il livello di realismo per dare priorità all'elaborazione dei dati e alla risposta temporale del sistema.

Ciò che consente all'allievo di eseguire le procedure con o senza supervisione di esperti è che la maggior parte dei simulatori VR ha un programma di formazione già implementato. Ciò riduce i costi personali nell'utilizzo di questi simulatori fintanto che i tirocinanti hanno accesso a questi sistemi. Hanno inoltre affermato che l'addestramento VR, sotto supervisione con istruzioni, feedback tempestivi e l'uso del feedback tattile, si è dimostrato il metodo di apprendimento più efficace [34] [35] o quanto meno efficace tanto quanto l'uso dei cadaveri e dei *box trainers* [36]. Il feedback tattile media il senso del tatto tra l'utente e l'ambiente virtuale pertanto gli algoritmi informatici devono essere in grado di rilevare quando gli oggetti entrano in contatto tra loro e con quale forza lo fanno.

Uno dei vantaggi della simulazione tramite VR è la capacità di misurare la competenza attraverso la valutazione delle metriche delle prestazioni come il tempo impiegato per

completare un'attività, il tipo e il numero di errori commessi. Tali metriche consentono di definire soglie di performance, permettendo di svolgere test e acquisire certificazioni. Per quanto riguarda quest'ultime, esistono già strumenti di simulazione validati per la certificazione delle competenze tecniche per il posizionamento di *stent* carotideo e per i fondamenti della chirurgia laparoscopica.

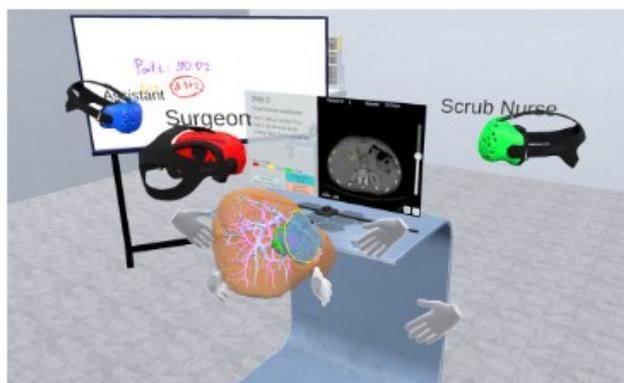
Un aspetto cruciale per un esito positivo della chirurgia è la comprensione reciproca e la collaborazione tra i membri del team; aspetto che una decina di anni fa non era affatto scontato e che oggi in realtà sembra essere superato grazie allo sviluppo di questa tecnologia. Il lato negativo era il fatto che la simulazione VR si concentrasse esclusivamente sulle competenze tecniche e che non fossero affrontate le competenze non tecniche, come la capacità di saper lavorare in gruppo, in quanto i simulatori VR erano progettati per l'apprendimento individuale e non erano collegati in rete per soddisfare la formazione del team. È stato bene quindi evidenziare che la competenza in chirurgia andasse oltre la semplice destrezza manuale ed è stato importante apprezzare i molti fattori che determinano buoni risultati, tra cui l'ambiente della sala operatoria e la capacità dei membri dell'equipe chirurgica di lavorare efficacemente insieme. Gli infermieri, il personale chirurgico, gli anestesisti e i tecnici anestesisti ricevevano tutti i propri programmi di formazione. Venivano quindi riuniti in una situazione ad alto rischio, spesso non avendo mai lavorato insieme come una squadra e ci si aspettava che svolgessero insieme compiti complessi in sicurezza [32].



Figura 16: Esempio di scenario virtuale

Attualmente, lo scambio di informazioni tra chirurghi avviene principalmente tramite riunioni faccia a faccia del team e videotelefonate. Le riunioni faccia a faccia offrono vantaggi rispetto alle riunioni online, inclusa una minore possibilità di problemi di comunicazione. Tuttavia, è difficile che gli incontri faccia a faccia si svolgano nell'ambito di una collaborazione a distanza, ad esempio durante una pandemia di Covid-19.

Ecco che le applicazioni VR collaborative consentono a più utenti di sperimentare e manipolare simultaneamente oggetti all'interno di ambienti virtuali per una collaborazione in loco o a distanza tra chirurghi, ma soprattutto supportano la comunicazione, la comprensione e il lavoro di squadra per la formazione e l'istruzione medica [37]. Questo ambiente favorisce uno scambio di informazioni per costruire un modello mentale condiviso per evitare errori [38] per visualizzare ed esaminare i dati di imaging e/o volumetrici [39].



(a)



(b)

Figura 17: Utenti multipli che eseguono la pianificazione chirurgica (a) gli utenti visti all'interno dell'ambiente virtuale e (b) utenti visti nel mondo reale.

Un esempio di realtà aziendale che si dedica a fornire formazione e valutazione chirurgica utilizzando la tecnologia VR è la società Osso VR. Osso VR [40] ha sviluppato una piattaforma in cui i chirurghi e il personale ospedaliero possono allenarsi insieme nella stessa sala operatoria virtuale consultando una libreria di formazione chirurgica virtuale. I medici possono unirsi a chirurghi esperti di tutto il mondo in sessioni di formazione virtuale per apprendere le migliori pratiche e tecniche. Inoltre, sempre ai fini educativi, questi sistemi possono tenere traccia dei progressi misurando la precisione e l'accuratezza delle prestazioni, nonché dei tassi di errore, che possono essere calcolati facilmente.

Attualmente l'uso della VR nell'educazione chirurgica è stato descritto principalmente per la chirurgia laparoscopica e artroscopica [41], ma anche neurologica e cardiaca [42]. Un esempio

di progettazione di scenari virtuali per la pianificazione chirurgica è stato presentato da Reinschluessel et al. [43]. L'obiettivo del suo lavoro è stato quello di sviluppare uno strumento di pianificazione chirurgica VR per supportare la comprensione dell'anatomia individuale e la disposizione spaziale della stessa. L'analisi viene svolta a partire dai modelli di organi 3D paziente-specifici generati da dati di RM o TC e la visualizzazione di questi avviene attraverso la vista tridimensionale stereoscopica proposta dalla VR, che concede la corretta percezione della profondità.

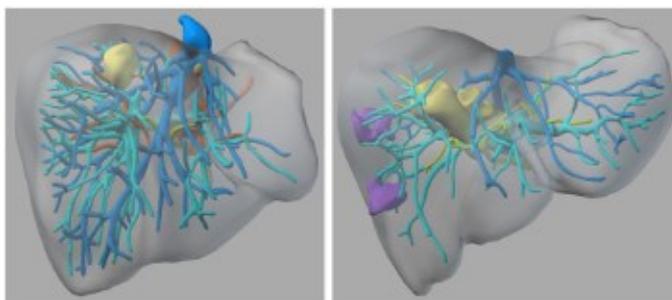


Figura 18: modelli 3D di due casi di studio sul fegato

Mentre la AR offre vantaggi simili, la VR è particolarmente utile perché consente agli utenti di immergersi in un ambiente di pianificazione dedicato senza distrazioni dall'ambiente circostante. Al contrario, l'AR è particolarmente utile in contesti intraoperatori, poiché l'utente, cioè il chirurgo, può ancora sperimentare direttamente il mondo reale [44].

Adottare la VR risulta essenziale e ricco di benefici perché anche se tutti gli interventi chirurgici seguono una procedura operativa standard, è comunque necessario creare un piano individuale per il paziente in base alla procedura che si vuole eseguire in quanto le strutture anatomiche di ogni persona variano in posizione e misura. Prendendo come soggetto le masse tumorali, le relazioni spaziali tra queste e i vasi sanguigni stabiliscono anche il modo di come andare ad affrontare l'operazione e, cosa più importante, se il tumore può essere rimosso del tutto. Pertanto, la pianificazione dell'intervento è una fase critica nel processo decisionale su come trattare il paziente. Questa forma di interazione può permettere agli utenti di orientarsi ed esprimersi meglio nello spazio 3D sfruttando gli strumenti che vengono messi a disposizione, come ad esempio lo strumento di disegno per contrassegnare specifiche zone di interesse o per indicare tagli, che sono attività intrinsecamente spaziali.

Sempre Reinschluessel et al. [43] ha riportato che, per i partecipanti che hanno partecipato al suo studio, utilizzare la VR è stato di aiuto per diversi aspetti ed i principali sono stati che:

- Il modello 3D era ben chiaro e fisso nella loro memoria anche durante l'esecuzione dell'intervento e questo ha permesso loro di essere più focalizzati e più sicuri di sé stessi

senza perdere di vista il fatto che tra il modello virtuale e la situazione reale ci possono essere delle incongruenze;

- Visualizzare i dati attraverso lo scenario virtuale ha potuto mettere in evidenza anche nuovi aspetti che non sarebbero stati presi in considerazione se si fosse utilizzata la procedura tradizionale;
- Il senso di immersione ha notevolmente aumentato le loro capacità di apprendimento.

Tutto ciò dimostra come le applicazioni VR e i relativi concetti di sensi immersione (“*embodiment*”) e interazione naturale possono apportare sviluppi significativi all’interno del contesto medico professionale.

3.2 Il visore Meta Quest 2

Il visore VR Meta Quest 2 è stato presentato il 16 settembre 2020 e rilasciato successivamente il 13 ottobre 2020 inizialmente con la denominazione Oculus Quest 2, ma assumendo poi il nome che possiede attualmente nel novembre 2021, e sviluppato da Meta Platforms (formalmente chiamato Facebook, Inc.).

Quest 2, come il modello che lo ha preceduto, Oculus Quest, si tratta di un dispositivo *stand-alone*, pertanto può essere utilizzato in modalità wireless senza bisogno di supporti fisici come un PC con requisiti specifici o ulteriori sensori di tracciamento esterni al dispositivo. Offre quindi la possibilità di potersi muovere liberamente all’interno di una zona circoscritta senza essere vincolato dalla presenza di cavi.

Tuttavia, il collegamento con il PC è del tutto facoltativo e può essere effettuato con il cavo Meta Link, cavo ufficiale della casa produttrice realizzato in fibra ottica, o in alternativa si può ricorrere a un qualsiasi cavo USB-C ad alta velocità, tale da fornire una potenza maggiore e permettere contestualmente all’utente di vivere una esperienza immersiva più intensa e realistica [45].

Come il suo predecessore, lo scopo del Meta Quest 2 è di rendere la VR alla portata di tutti, sia dal punto di vista dell’installazione che dal punto di vista dell’interazione.

Quest 2 presenta un design simile al Oculus Quest, ma con un peso più leggero pari a 503 grammi rispetto ai 571 grammi iniziali [46], specifiche interne aggiornate, un display LCD con una frequenza di aggiornamento e una risoluzione per occhio più elevate e controller Oculus Touch con una durata della batteria più lunga.

Il funzionamento di questa tecnologia funziona tramite la combinazione di strumenti hardware e software che, insieme, permettono all'utente di calarsi all'interno di una realtà simulata, talvolta estremamente realistica.



Figura 19: Meta Quest 2

3.2.1 Hardware

Quest 2 è dotato di 6 GB di RAM e di un processore Qualcomm Snapdragon XR2, consentendo di riprodurre giochi e applicazioni in maniera più efficace anche in assenza di PC rispetto a quanto avveniva con l'Oculus Quest che al contrario possedeva il processore Qualcomm Snapdragon 835 [47].

I doppi display OLED del Quest di prima generazione sono stati sostituiti da un singolo pannello LCD di tipo *Fast-switch*, a rapida commutazione, con una risoluzione per occhio di 1832×1920 e una frequenza di aggiornamento fino a 120 Hz. Il display utilizza subpixel a strisce anziché una matrice PenTile migliorando la nitidezza dell'immagine e riducendo l'effetto *screen-door*.

Una delle novità che ha più incuriosito il pubblico prima ancora dell'uscita ufficiale dell'Oculus Quest 2, e a farlo reputare uno tra i dispositivi più coinvolgenti e innovativi, è stata la notizia che avrebbe potuto offrire, oltre agli strumenti di tracciamento della mano (*hand tracking*) e della testa (*head tracking*), anche un possibile supporto per il tracciamento del corpo. Questa in realtà poi si è rivelata essere una cosa non possibile da effettuare, o meglio non in maniera automatica. L'abilitazione richiede elementi aggiuntivi come tracker esterni e cinturini per indossarli, e stazioni base utili per rilevare i tracker stessi [48]. Si tratta, dunque, di un sistema di tracciamento di tipo *Inside-out* giustificando i costi contenuti.

3.2.2 Sistema di tracciamento

Il visore Meta Quest 2 sfrutta un sistema di tracciamento di tipo Oculus Insight. Si tratta di un'architettura hardware personalizzata e algoritmi avanzati di visione artificiale tra cui la mappatura visuale-inerziale, il riconoscimento del luogo e la ricostruzione della geometria per stabilire la posizione degli oggetti in relazione ad altri oggetti all'interno di un dato spazio. Il metodo di visione utilizzato è chiamato SLAM, che sta per *Simultaneous Localization and Mapping*.

Questo nuovo *stack* di algoritmi consente a un dispositivo VR di individuare la sua posizione, identificare aspetti della geometria della stanza (come la posizione del pavimento) e tracciare le posizioni del visore e dei controller rispetto a una mappa tridimensionale generata e costantemente aggiornata da Insight.

Il sistema sfrutta tre flussi di dati provenienti da:

- unità di misura inerziali (IMUs) presenti sul visore e sui controller che continuamente valutano l'accelerazione lineare e la velocità di rotazione, rispettivamente tramite accelerometri e giroscopi, per ottenere la posizione con bassa latenza;
- quattro telecamere poste sui bordi del visore, tutti obiettivi *fisheye* ad ampio campo visivo, e cioè che coprono gran parte della sfera visiva; creano una mappa tridimensionale della stanza individuando dei punti di riferimento, ad esempio angoli degli arredi, i quali vengono osservati ripetutamente per compensare la deriva degli IMU. Servono anche per il tracciamento del corpo dell'utente, delle sue mani (*hand tracking*) e dei controller Oculus Touch.
- LED a infrarossi e sistema IMU per il costante tracciamento della posizione dei controller, favorito dall'ampio campo visivo del visore [49].



Figura 20: In azzurro sono evidenziate le videocamere presenti nel visore e in arancione sono evidenziate le direzioni degli occhi dell'utente

Oculus Insight rileva e traccia nel tempo pixel dell'immagine ad alto contrasto, triangola la posizione tridimensionale di ogni punto nell'ambiente circostante, creando una nuvola di punti, rispetto al quale calcola poi la posizione del visore generando una mappa tridimensionale dell'ambiente.



Figura 21: I puntini bianchi evidenziano gli angoli degli oggetti appartenenti all'ambiente e coincidono con i punti di riferimento o punti 3D che costituiscono infine la mappa tridimensionale [50]

Un'altra caratteristica innovativa di questa tecnologia è l'**hand tracking**, una funzione che si può abilitare nelle impostazioni Oculus e che consente di interagire con l'ambiente virtuale unicamente attraverso gesti delle mani [51] solo su siti web e app supportati.

I gesti supportati da Meta Quest 2 sono i seguenti:

Tabella 4 [52]:

Gesti	Funzione
Puntare e avvicinare il pollice verso l'indice	Selezione
Avvicinare il pollice verso l'indice e scorrere con la mano	Scorrimento
Con il palmo della mano orientato verso l'alto, avvicinare il pollice verso l'indice	Accesso alle azioni più comuni del menu Azioni Rapide 

L'**hand tracking** funziona grazie all'uso di algoritmi di visione artificiale che sfruttano l'informazione proveniente dalle fotocamere *inside-out* che svolgono il compito di rilevamento della posizione e dell'orientamento delle mani, inclusa la disposizione delle dita.

In particolare, il software esegue un compito di raccolta delle immagini delle mani provenienti dai sensori, stima la posizione di determinati punti chiave come le nocche o la punta delle dita e successivamente adatta un modello generico di mano ai punti stimati ottenendo informazioni sulla posa della mano e le dimensioni stimate rappresentate da valori numerici standardizzati. Per garantire un'esperienza coerente, queste due informazioni vengono memorizzate non appena sono generate sul dispositivo per poi essere eliminate una volta completata l'elaborazione [53].

3.2.3 Controller Oculus Touch



Figura 22: Controller Oculus Touch

I controller Oculus Touch sono stati completamente riprogettati con una migliore ergonomia, ispirata al design dei controller Rift. Si trattano di veri e propri joystick, dotati di levette e grilletti analogici, che trasferiscono in modo facile e preciso i gesti e le azioni dell'utente nel mondo virtuale tramite l'insieme di sensori che si trovano al loro interno, permettendo di essere riprodotti poligonalmente e quindi essere facilmente individuati dal visore. Inoltre, a differenza del passato, basta impugnarli perché si attivino automaticamente, senza dover premere alcun pulsante. Quando non vengono utilizzati, inoltre, si spengono automaticamente, prolungando così l'autonomia della batteria.

3.2.4 Audio posizionale

Gli altoparlanti sono rimasti invariati rispetto al modello precedente, collocati in due braccetti di plastica che si trovano vicino al pannello. Si tratta di un sistema che, a fronte di una leggera perdita in termini di qualità del suono, adatta l'emissione dei suoni in funzione dei movimenti della testa, garantendo un'immersione uditiva assolutamente credibile.

L'algoritmo che gestisce l'audio posiziona la fonte dei suoni all'interno di una sfera virtuale a 360 gradi. Il movimento della testa nella realtà non interagisce con questa sfera, in modo da poter continuare a sentire il suono proveniente dall'ambiente reale circostante e a percepirne la sua posizione d'origine in maniera fedele, indipendentemente dai movimenti compiuti [54].

3.2.5 Vestibilità del Meta Quest 2

Altra novità è il meccanismo di regolazione della **distanza interpupillare (IPD)**. Si riferisce alla distanza tra le pupille degli occhi sinistro e destro, la quale può essere regolata fisicamente portandola in tre possibili posizioni (58 mm, 63 mm e 68 mm) a seconda del livello di nitidezza con cui si percepiscono le immagini in VR. Per produrre contenuti stereoscopici immersivi su scala umana, le produzioni dovrebbero utilizzare attrezzature per telecamere con obiettivi il più vicino possibile all'IPD umano medio. Un IPD più grande del normale può esagerare il 3D e far apparire gli oggetti miniaturizzati ("*iperstereo*"). Un IPD più piccolo del tipico ridurrà il 3D stereoscopico e farà apparire i soggetti più grandi del normale ("*hypostereo*") [55].

3.2.6 Set-up di gioco

A seconda di come è concepita l'app VR, questa può richiedere un tipo di interazione di tipo **Roomscale**, **Standing** e **Setting**. Quando si acquista una app nello *store*, questo dettaglio è espresso in maniera molto chiara. Nel dettaglio:

- **Roomscale**: queste app devono essere utilizzate in un'area 2x2 metri con un perimetro *roomscale* definito;
- **Standing**: si può interagire con tutti gli elementi del gioco stando in piedi, accovacciandosi, girandosi, raggiungendo oggetti virtuali sia con confini stazionari che con *roomscale*;
- **Sitting**: l'intera esperienza è giocabile da seduti. Questo tipo di app può essere utilizzato sia con confini stazionari che con *roomscale*. [54]

3.2.7 Software

Quest 2 esegue un sistema operativo basato su Android. Per eseguire la configurazione iniziale, è necessario utilizzare uno smartphone con l'app Meta Quest.

3.3 Unity (Game Engine)

Unity è un ambiente di sviluppo creato da Unity Technologies e utilizzato per sviluppare simulazioni interattive e applicazioni di gioco 2D e 3D per Android e iOS e le principali console in commercio. Tra le funzionalità che offre vediamo infatti: rendering grafico, effetti di luce, creazione di terreni, simulazioni fisiche, implementazione dell'audio, funzionalità di rete, supporto al *multiplayer* ed anche un sistema di *scripting* che permette agli sviluppatori di creare componenti *custom*.

Quando si tratta di codificare, Unity passerà a un *editor* alternativo di propria scelta che formalmente prendono il nome di *Integrated Development Environment* (IDE). L'opzione più comune è Visual Studio di Microsoft, un'interfaccia che dà accesso ad una moltitudine di strumenti necessari per lo sviluppo in un unico posto [56]. Il linguaggio principalmente supportato del sistema di *scripting* è C#.

Per Unity è a disposizione **Unity Asset Store** [57], una crescente libreria di risorse gratuite e commerciali create sia da *Unity Technologies* che dai membri della *community*. È disponibile un'ampia varietà di risorse che includono trame, modelli e animazioni o interi esempi di progetti, tutorial ed estensioni dell'*editor*. Lo scopo è quello di semplificare notevolmente il lavoro necessario per lo sviluppo degli ambienti virtuali interattivi e il loro rendering in *real-time*. Di seguito vengono introdotti alcuni concetti base fondamentali per capire alcuni aspetti del lavoro svolto. Allo stesso modo di cui sopra, Unity stesso offre gratuitamente una vasta lista di pacchetti e risorse sviluppati internamente che estendono le funzionalità del motore in modi utili.

3.3.1 Editor

L'*editor* di Unity si suddivide principalmente in sei sezioni come mostrato in Fig.23:

- *Scene*: si tratta della finestra di visualizzazione principale del software. All'interno di questa sezione è fornita la possibilità di posizionare tutti i componenti della scena, di scalarli e ruotarli. È possibile inoltre cambiare vista tra quelle predefinite, come vista dall'alto, dal basso, laterale o anche cambiare l'orientamento della camera posizionandola in prospettiva o in posizione isometrica; ruotare la scena, fare lo zoom o spostarsi semplicemente utilizzando il mouse.
- *Game*: questa sezione mostra il *render* della scena percepito dalla telecamera impostata come principale. Cliccando il tasto *play* l'applicazione viene eseguita direttamente all'interno

dell'*editor*, il che permette un controllo in tempo reale di quello che sarà l'aspetto e il comportamento finale dell'applicazione.

- *Hierarchy*: in questa sezione sono riportati tutti gli elementi inclusi nella scena, i cosiddetti *Game Object* [58], suddivisi in ordine di parentela tramite una rappresentazione a *directory*. Ogni componente può avere al suo interno altri componenti figli, i quali sono più o meno vincolati al padre.
- *Project*: qui è possibile esplorare tutti i file e cartelle che fanno parte del progetto. Tramite il tasto Create possiamo aggiungere cartelle, scene, *script*, materiali, *prefab* [59] e molto altro. Il sistema dei *prefab* permette di creare, configurare e archiviare un *Game Object* completo di tutti i suoi componenti fungendo da modello che può essere riutilizzato un numero indefinito di volte nella scena e sul quale possono essere apportate delle modifiche che si riflettono nelle istanze di tale *prefab*, consentendo allo stesso tempo di non dover apportare ripetutamente la stessa modifica ad ogni copia del *prefab*. Questo non vuol dire che tutte le istanze dei *prefab* saranno identici. È possibile ignorare le impostazioni su singole istanze di un *prefab* se si desidera che queste siano diverse da altre.
- *Inspector*: mostra tutti i dati relativi alle caratteristiche dell'oggetto della scena correntemente selezionato, nonché tutti i componenti dell'oggetto, che possono essere modificati. Ciò potrebbe comportare la modifica delle dimensioni (*Scale*) o della posizione (*Transform*) oppure l'aggiunta di "componenti" come script C# o *collider*.
- *Console*: durante la fase di test riporta tutti gli errori o avvisi legati al codice, o se ci sono problemi che devono essere risolti con la configurazione del software stesso.

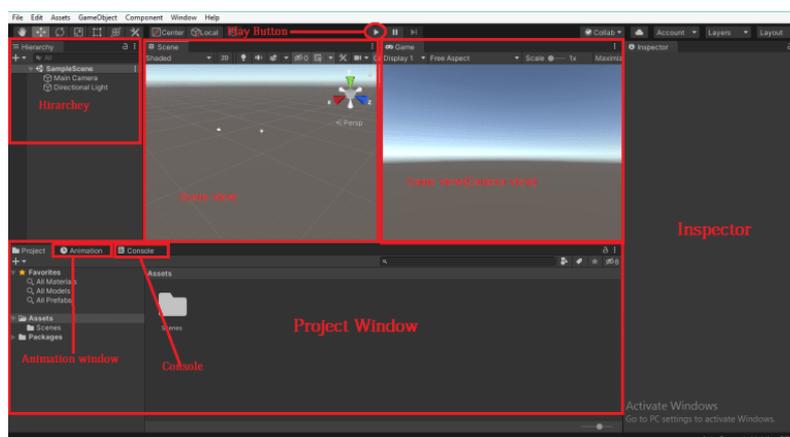


Figura 23: Unity Editor

3.3.2 Script

Ad ogni *Game Object* è possibile assegnare uno o più script, i quali a loro volta, possono essere assegnati a più *Game Object*. Nella sezione *Project*, è possibile creare un file con estensione *.cs* (nel caso di C#), editato successivamente attraverso l'*editor* di propria scelta.

Come riportato in Fig. 24, uno *script* effettua il suo collegamento con il funzionamento interno di Unity implementando una classe derivante dalla classe integrata *MonoBehaviour* [60] e/o creando classi normali (non derivante da *MonoBehaviour*). Alla creazione della classe, in automatico vengono ereditate le funzioni *callback* di *MonoBehaviour*, utili per eseguire codice in momenti specifici della vita dell'applicazione.

Le funzioni più rilevanti definite all'interno della classe sono:

- *Start()*: chiamata da Unity prima all'esecuzione del programma e chiamata una sola volta. È utile principalmente per inizializzare variabili o per compiere altre operazioni di *set up*.
- *Update()*: chiamata regolarmente all'update di ogni *frame*. Riguarda tutto ciò che deve essere gestito durante l'esecuzione dell'applicazione, per esempio l'attivazione di azioni o la risposta all'input dell'utente.

```
using UnityEngine;
using System.Collections;

public class NewBehaviourScript : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

}
```

Figura 1: Script Unity

Capitolo 4

Sviluppo dell'applicazione VR

In questo capitolo viene illustrato il processo di creazione e sviluppo dell'applicazione, dall'idea iniziale fino al prodotto finito, passando per tutte le fasi intermedie.

4.1 L'applicazione: una visione generale

Lo scopo principale di questa applicazione è quello di essere utilizzata in alternativa ai metodi tradizionali di cui attualmente i clinici usufruiscono al momento della pianificazione di un intervento. Come già si è anticipato, questo processo si basa sulla collaborazione tra ingegnere e chirurgo dove in particolare il chirurgo guida l'ingegnere nella selezione delle direzioni di taglio rispetto al quale saranno poi progettate le guide chirurgiche. Ecco che quindi questo prototipo ha come primo aspetto caratterizzante il fatto di voler rendere il chirurgo completamente autonomo nell'esecuzione di questa prima parte del processo di design, dal momento che attualmente sarebbe improponibile pensare che possa imparare ad utilizzare software di design come Geomagic Design X. Non solo, un altro vantaggio è fargli, al contempo, risparmiare dei passaggi rispetto a quanto è previsto utilizzando Geomagic Design X.

4.2 Game Design

Tenendo conto del processo di progettazione per dispositivi medici oggi adottato (fig. 25), questo prototipo sarebbe pensato per essere inserito tra la fase di elaborazione dei modelli CAD e la progettazione della dima chirurgica.

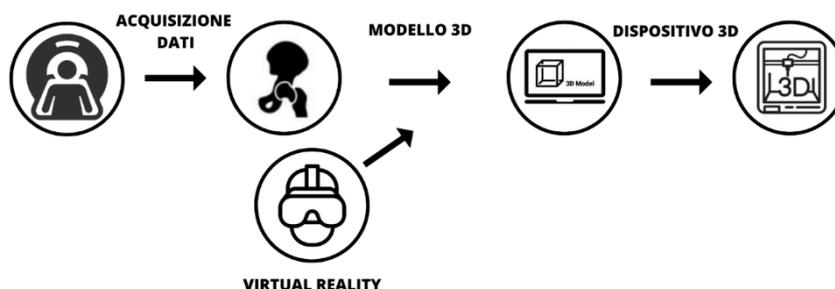


Figura 25: Workflow generale per la progettazione di dispositivi medici paziente-specifici

La procedura difatti prevede che la fase di sviluppo del file CAD del modello anatomico sia seguita dalla fase di pianificazione preoperatoria con conseguente progettazione della guida chirurgica customizzata. I modelli CAD in formato STL sono pronti per essere utilizzati all'interno di Unity solo una volta essere stati convertiti in formato OBJ. È importante evidenziare che è possibile importare non solo un unico file all'interno di Unity, rappresentativo di un unico modello, ma anche un gruppo di file, rappresentativi di più modelli CAD, ciascuno dei quali conserva l'informazione relativa alla loro posizione, orientamento e dimensione rispetto al sistema di riferimento globale.

Prima di immergersi nella scena virtuale, e quindi dare avvio alla modalità di gioco, bisogna procedere con una serie di considerazioni relative al sistema di riferimento locale del modello importato. Tra software non esiste un sistema di riferimento standard, pertanto Unity e gli altri software, sui quali sono stati elaborati i modelli, avranno un sistema di assi non coincidenti e questo lo si osserva nel momento in cui si importano i modelli all'interno della scena in quanto con molta probabilità sarà necessario correggere il loro orientamento rispetto al sistema di riferimento di Unity. Dato che il controller sinistro è stato settato in modo da controllare unicamente la rotazione del modello attorno all'asse verticale y e all'asse orizzontale x è necessario prima generare un *Empty Object* i cui *child objects* saranno i modelli generati dall'elaborazione. In questo modo, i modelli saranno gestiti come se ne fossero uno solo con un unico sistema di riferimento locale coincidente con il sistema di riferimento globale di Unity.

I modelli testati in questo prototipo erano per lo più rappresentativi di una situazione clinica che prevedeva la presenza di uno o più tumori. Se insieme ai modelli delle parti anatomiche affette da patologia non sono state importate anche le texture del tumore stesso, è possibile farlo attraverso l'assegnazione dei *Material Color*. Si tratta di una classe materiale che espone le proprietà di un materiale, in questo caso il colore principale del *Material*. Completata quindi la creazione e l'assegnazione dei *Material Color* ai relativi tumori, l'utilizzatore ha tutto il necessario per poter procedere ed entrare all'interno dell'ambiente virtuale.

L'esperienza ha inizio in un ambiente all'interno del quale l'utente visualizza davanti a sé il modello CAD del caso clinico preso in esame ed un Canvas che propone degli strumenti aggiuntivi. L'utente non è rappresentato graficamente da un proprio avatar ma è in grado di visualizzare solamente i due controller dai quali, in coincidenza di ciascun tasto, saranno proiettati le funzioni associate ad essi. In questo modo se l'utente dovesse spostarsi dalla sua posizione iniziale, ha la possibilità di regolare la rotazione del modello da qualunque punto dell'ambiente virtuale in cui esso si trova. L'esperienza non prevede che l'utente si muova fisicamente nell'ambiente reale ma propone due modalità di fruizione: seduta e in piedi. La

locomozione è eseguibile solo attraverso la levetta analogica del controller sinistro. Dunque, il controller sinistro è utile per la sola gestione della rotazione del modello attorno ai due assi, mentre il controller destro è stato impostato per la creazione e gestione dei piani di taglio.

L'utente crea il piano di taglio, il quale comparirà in coincidenza di uno specifico punto posto alla medesima altezza del controller e ad una distanza dal controller pari ad un offset regolabile dall'utente nelle impostazioni. In seguito, rilascia il piano in coincidenza di un punto che secondo la sua intuizione è il più giusto ed ha la possibilità di riprendere lo stesso e di riposizionarlo dove meglio crede. Diversamente, ha la possibilità di eliminare l'ultimo piano generato o di reimpostare la scena di partenza, quindi ritornare alla condizione in cui non sono stati creati piani di taglio.

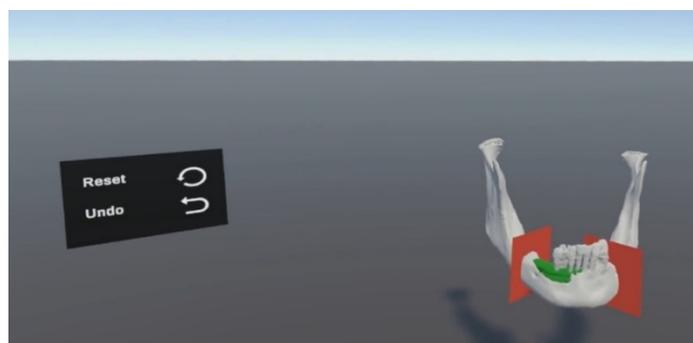


Figura 26: Scena della realtà virtuale

Una volta completata questa fase, l'utente, prestando sempre attenzione al fatto di non uscire dalla fase di play mode (in quanto si tratta di una anteprima in cui tutto ciò che viene svolto non viene memorizzato), deve passare al computer sul quale sta lavorando ed effettuare l'esportazione del modello, o parti di esso, in formato FBX.

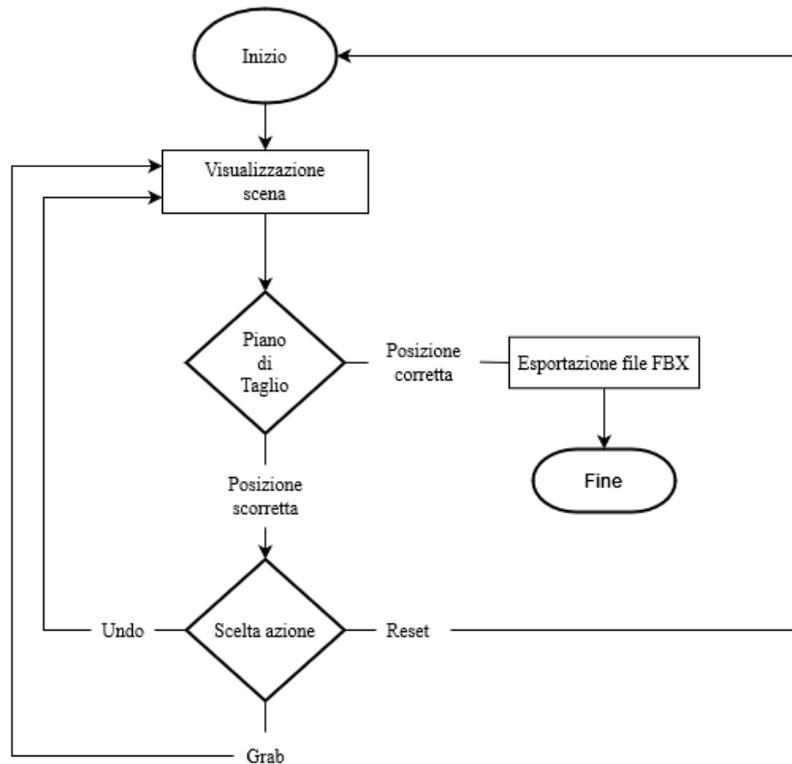


Figura 27: Diagramma di flusso che riassume il game design

4.3 Realizzazione dell'ambiente VR

4.3.1 Setting VR

In questo sottoparagrafo si discute di come è stato sviluppato il progetto per essere fruito virtualmente. Per prima cosa si è scaricato l'**Oculus Integration Package** dal Asset Store di Unity. Si tratta di un pacchetto che offre il supporto avanzato per lo sviluppo di applicazioni per i dispositivi Oculus VR e alcuni dispositivi supportati da Open VR, contenente script e *prefab* già pronti all'utilizzo. Per gestire l'interazione tra l'utente e l'ambiente virtuale, si è importato all'interno dello *Hierarchy* di Unity il *prefab* **OVRPlayerController** che include componenti e *Child Object* necessari per il controllo 3D ed ulteriori *prefab* come l'**OVRCameraRig** e l'**OVRControllerPrefab**.

OVRCameraRig

Il *prefab* **OVRCameraRig** è essenzialmente gli occhi dell'utente. Pertanto la fotocamera principale predefinita di Unity (Main Camera) si è sostituita, eliminandola, in quanto altrimenti sarebbero entrati in conflitto.

A questo punto ci si è occupati di settare l'OVRCameraRig. Si è passati a cercare all'interno della sua scheda *Inspector* la proprietà Tracking Origin Type nel componente OVR Manager (script) e si è selezionato Floor Level. In questo modo sono stati fissati la posizione e l'orientamento dell'utente rispetto al pavimento. La posizione in VR viene calcolata in base all'altezza dell'utente che è stata indicata durante la configurazione del proprio dispositivo Oculus.

OVRController Prefab

Ora che si è impostata la nostra fotocamera, si sono impostati i controller per rappresentare le mani del giocatore. OVRController Prefab è uno dei diversi *prefab* presenti nell'integrazione e si è impostato come *child object* per entrambe le mani (LeftControllerAnchor e RightControllerAnchor).

4.3.2 Setting Controller Sinistro

Si riporta lo script *Rotazione.cs* implementato al fine di mettere in rotazione il modello attorno agli assi verticale y e orizzontale x attraverso il semplice utilizzo dei pulsanti:

- *Pulsante X*: rotazione verso destra;
- *Pulsante Y*: rotazione verso sinistra;
- *Pulsante IndexTrigger*: rotazione verso l'alto;
- *Pulsante HandTrigger*: rotazione verso il basso;
- *Pulsante Thumbstick*: locomozione.

Per implementarlo è stato creato un *Empty Object* all'interno dello *Hierarchy* definito "Rotazione", il cui unico scopo è quello di aggiungergli un componente di script. All'interno dello script sono definite delle variabili pubbliche, ovvero proprietà edificabili all'interno dell'*Inspector* oltre che nello script stesso. In questo caso, queste variabili sono Velocità Rotazione, la velocità con cui si intende far ruotare il modello attorno agli assi, e Modello, oggetto tridimensionale importato.



Figura 28: script *Rotazione.cs*

4.3.3 Setting Controller Destro

Il controller destro è stato impostato in modo da gestire principalmente i piani di taglio. Anche in questo caso è stato creato un *Empty Object*, chiamato “Gestione Piani di Taglio” a cui è stato aggiunto un componente di script denominato *CreaPiani.cs* (fig. 29) dal quale sono richiamate le funzioni che governano le azioni associate ai piani di taglio. Inoltre, “Gestione Piani di Taglio” tiene anche memoria dei piani di taglio che vengono originati riportandoli sotto forma di lista in coincidenza della voce *Piani Creati*. Questo risulterà essenziale per l’esecuzione dell’operazione di Undo (sottoparagrafo 4.3.5).

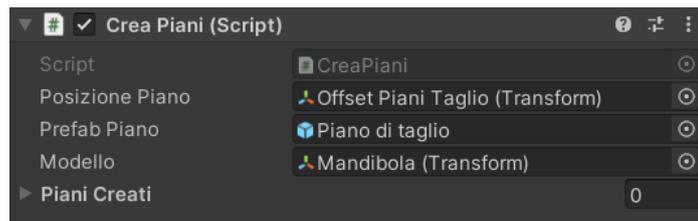


Figura 29: script *CreaPiani.cs*

I pulsanti del controller destro sono stati implementati nei seguenti modi:

- *Pulsante A*: implementa la funzione *LasciaPiano()* definita in *CreaPiani.cs*. Tale funzione permette all’utente di lasciare il piano selezionato nel punto reputato più adatto ai fini dell’intervento. Dato che un intento successivo può essere quello di poter osservare il modello da più punti di vista anche con il piano di taglio ad esso intersecato, si è fatto in modo che il *Parent* del piano generato diventasse il modello stesso contestualmente al momento in cui esso viene rilasciato in modo da poterli trattare come un oggetto unico.
- *Pulsante B*: implementa la funzione *CreaPiano()* definita in *CreaPiani.cs*. Tale funzione permette di generare il piano di taglio in coincidenza dell’offset scelto. L’offset è stato predisposto creando un *Empty Object* come *child object* di *RightHandAnchor* (*prefab* del controller destro). Al fine di conservare le dimensioni del piano di taglio, così come sono state definite nella componente *Transform.scale* del *prefab*, ed evitare che queste subiscano variazioni, si sono azzerate le componenti della posizione e dell’orientamento della trasformazione del piano generato rispetto alla trasformazione del *Parent*.
- *Pulsante HandTrigger*: implementa la funzione di grabbing (spiegato nel sottoparagrafo 4.3.4) dei piani di taglio nel caso in cui li si voglia riposizionare;
- *Pulsante IndexTrigger*: non è implementato nulla;
- *Pulsante Thumbstick*: implementa la rotazione della camera.

4.3.4 Grabbing e Feedback tattile

Per poter eseguire il grabbing del piano di taglio si è optato per la creazione di una sfera, la cui mesh non è visibile all'interno della scena, posta in coincidenza del controller destro, sulla quale è implementato lo script utile al raggiungimento di questo scopo.

Prima di questo è stato però necessario aggiungere una componente utile per il **rilevamento delle collisioni**: nel *Inspector* del OVR Player Controller si è aggiunta una componente definita Character Camera Constraint e nel campo Camera Rig, della stessa componente, si è importato l'OVR Camera Rig.

Successivamente, ci si è occupati di rendere il piano di taglio afferrabile con il controller. Le tre caratteristiche che il piano di taglio deve possedere sono: la componente Box Collider, la componente Rigidbody e la componente **OVR Grabbable (script)** che fa parte del Oculus Integration Package importato inizialmente.



Figura 30: componenti del piano di taglio

Infine, si è generato un *Empty Object* come *child object* di RightHandAnchor (prefab del controller destro) per poter abilitare il controller destro ad afferrare i piani di taglio. Per fare ciò, si è seguito un criterio analogo a quanto fatto precedentemente con i piani di taglio. Pertanto, generato l'*Empty Object* (denominato "Grabber") si sono impostate tre caratteristiche: uno *Sphere Collider* (un collisore primitivo a forma di sfera), una componente Rigidbody e la

componente **OVR Grabber (script)**, specificando nel campo GripTransform di quest'ultima, di fare riferimento alle proprietà di posizione, rotazione e scala del controller destro [61].



Figura 31: componenti di "Grabber"

Tra le componenti di OVR Grabber (script) è presente una voce definita Haptic Audio Clip in corrispondenza del quale è stato caricato un file audio. L'utilità di questo campo è quella di generare un **feedback tattile** ovvero trasmettere all'utente l'impressione di star interagendo con gli oggetti presenti nel mondo virtuale tramite l'attivazione di un motore di vibrazione tattile integrato nel controller. Pertanto il modello di vibrazione tattile viene dettato da un file audio. Per poter implementare questa proprietà sono state aggiunte delle righe di codice allo script OVR Grabber (fig. 32):

- è stata inizializzata una variabile pubblica per poter selezionare il file audio (riga 32);
- è stato modificato il metodo OnTriggerEnter() che generalmente si attiva ogniqualvolta la componente di collisione di un oggetto interagisce con altre componenti omologhe presenti nella scena. Pertanto, ciò che si è voluto fare dalla riga 206 alla riga 211 è stato voler aggiungere un impulso tattile che si attiva nel momento si verifica l'interazione

tra collider e che contestualmente riproduce il clip audio scelto. Si è creato un OVR Haptics Clip chiamato hapticsClip tramite il quale si è estratto il file audio che dopo sarà azionato sul motore del controller destro.

```
31 // audio clip for haprtics -- haptic system plays back via vibration motor
32 public AudioClip hapticAudioClip;

204 // Add a haptic pulse
205 // Get Audio clip used for output pattern
206 OVRHapticsClip hapticsClip = new OVRHapticsClip(hapticAudioClip);
207 // ..and play it via Preempt() on the appropriate controller
208 if (m_controller == OVRInput.Controller.LTouch)
209     OVRHaptics.LeftChannel.Preempt(hapticsClip);
210 else
211     OVRHaptics.RightChannel.Preempt(hapticsClip);
```

Figura 32: script per feedback tattile controller destro

4.3.5 UI Canvas

L'utente ha la possibilità di interagire con un Canvas attraverso il quale può eseguire due azioni:

- **resettare la scena;**
- **eseguire l'Undo**, ovvero l'annullamento dell'ultimo piano generato.

Il **Canvas** è la base di tutte le interfacce utente in Unity: ogni componente della UI (immagine, testo, bottone) deve obbligatoriamente trovarsi all'interno di esso, altrimenti non verrà renderizzato. La modalità di rendering predefinita per i Canvas è una sovrapposizione dello schermo 2D. Affinché un Canvas appaia nello spazio 3D, come richiesto dalla VR, bisognerà cambiare il suo Render Mode in World Space il quale renderizzerà il Canvas come se si trattasse di un elemento 3D nell'ambiente virtuale.

Affinché OVRCameraRig rilevi i componenti dell'interfaccia utente di Unity, è necessario proiettare un raggio fisico nella scena. È stato fondamentale dover configurare OVRCameraRig con la componente di script OVRPhysicsRaycaster. Allo stesso modo, il Canvas necessita del componente di script OVRRaycaster per essere rilevato da Oculus.

Per ultimo è stato importato il *prefab UIHelpers*, il quale funziona come un cursore del mouse per la VR che visualizza la direzione in cui l'utente sta puntando e si appoggia su qualsiasi oggetto interattivo, inclusa l'interfaccia utente.

All'interno di UIHelpers è presente un oggetto Event System che rileva gli eventi dell'interfaccia utente (clic, passaggio del mouse, ecc.). Si è impostato il campo Ray Transform nello script

OVR Input Module di Event System con un *child* di OVRCameraRig, in questo caso RightHandAnchor (fig. 33).

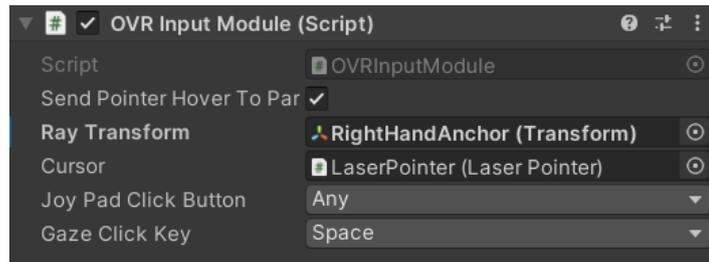


Figura 33: OVRInputModule di Event System

L'Event System del Canvas è stato disabilitato e sarà utilizzato solo l'Event System dell'UIHelpers. Come già anticipato, i pulsanti hanno la capacità di richiamare gli Unity Events, ovvero funzioni, a partire da script di codice, nel momento in cui si preme uno specifico pulsante. Dunque il pulsante Reset richiama la funzione Reset() in *ResetScena.cs* e il pulsante Undo richiama la funzione CancellaUltimoPianoCreato() presente in *CreaPiani.cs*.

Si riportano di seguito gli script eseguiti:

```
6 public class ResetScena : MonoBehaviour
7 {
8     public void Reset()
9     {
10         int scenaAttuale = SceneManager.GetActiveScene().buildIndex;
11         SceneManager.LoadScene(scenaAttuale);
12     }
13 }
```

Figura 34: script ResetScena.cs

Per riproporre la scena iniziale, ci si è avvalsi del metodo LoadScene() che ricarica la prima scena della lista richiamando il nome della scena.

```

59     public void CancellaUltimoPianoCreato()
60     {
61         if (PianiCreati == null || PianiCreati.Count == 0)
62             return;
63
64         // elimina l'ultimo piano della lista
65         int indiceUltimoPiano = PianiCreati.Count - 1;
66         Destroy(PianiCreati[indiceUltimoPiano]);
67
68         // aggiorna la lista dei piani creati
69         PianiCreati.RemoveAt(indiceUltimoPiano);
70
71         // aggiorna l'oggetto ultimoPianoCreato
72         ultimoPianoCreato = null;
73
74         if(indiceUltimoPiano > 0)
75         {
76             ultimoPianoCreato = PianiCreati[indiceUltimoPiano - 1];
77         }
78     }

```

Figura 35: script *CancellaUltimoPianoCreato.cs*

Per simulare l'operazione di Undo si è ricorsi all'utilizzo delle liste. In questo caso, essendo l'oggetto in considerazione un Game Object, si è inizializzata una lista di *Game Object* denominata *PianiCreati*.

CancellaUltimoPianoCreato() lavora nel seguente ordine:

- verifica inizialmente se la lista è vuota, ovvero se non è stato creato nessun piano di taglio, e a questo punto la funzione non ritorna niente;
- se, al contrario, la lista possiede degli elementi, svolge principalmente tre compiti:
 1. inizializza il numero di elementi di cui la lista è composta ed elimina l'ultimo elemento dalla lista, ovvero l'ultimo piano di taglio generato;
 2. elimina, oltre all'oggetto, anche il riferimento di quell'oggetto e cioè l'elemento a cui la lista sta puntando;
 3. fatta la cancellazione, aggiorna l'ultimo piano generato impostandolo inizialmente come nullo e successivamente ponendolo pari all'ultimo elemento della lista attuale.

4.3.6 Esportazione file FBX

Scaricato direttamente dal Package Manager di Unity, **FBX Exporter** [62] è stato il pacchetto che si è scelto al fine di selezionare gli oggetti della scena ed esportarli manualmente sotto forma di file FBX.

FBX Exporter esporterà le gerarchie dei *Game Object* insieme all'informazione contenuta in Transforms (posizione, rotazione e scala), la mesh, la texture, la telecamera, le luci e i vincoli imposti.

La scelta di esportarlo sotto forma di file FBX è dovuta alle difficoltà che si sono riscontrate nell'esportare il file in formato OBJ o STL utilizzando ProBuilder Mesh o file script.

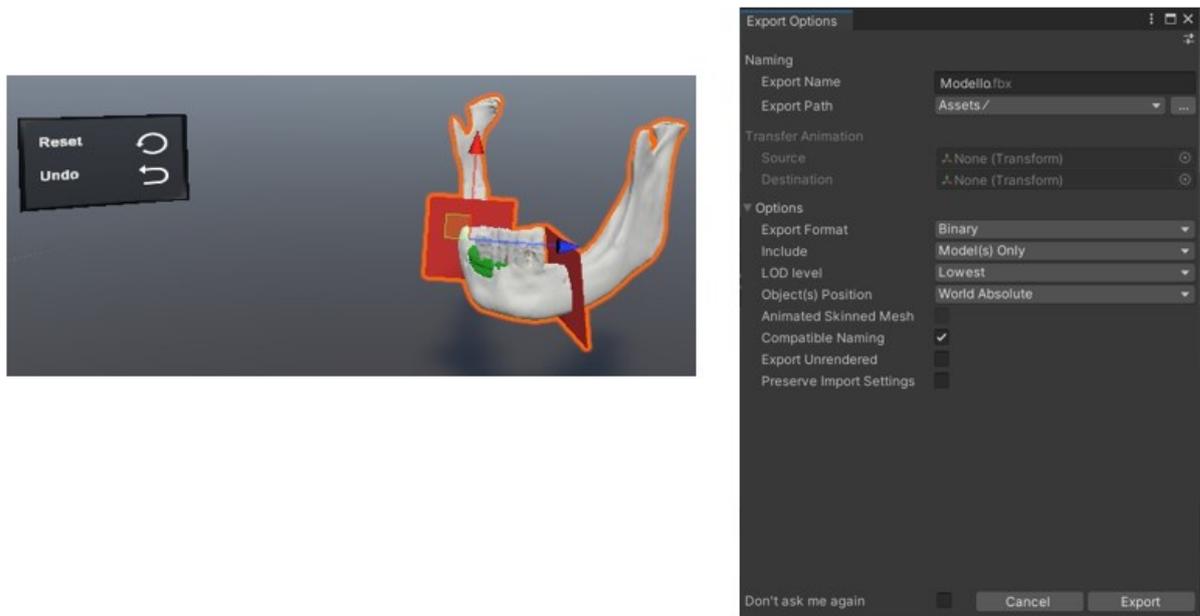


Figura 36: FBX Exporter

Capitolo 5

Test e risultati

Il caso clinico preso in esame in questo lavoro di tesi si tratta di tumore all'osso mandibolare. L'obiettivo è stato voler mettere a confronto la procedura di pianificazione chirurgica adottata da un esperto CAD con l'approccio proposto che prevede l'esportazione dei piani di taglio da Unity e la loro successiva importazione su Geomagic Design X.

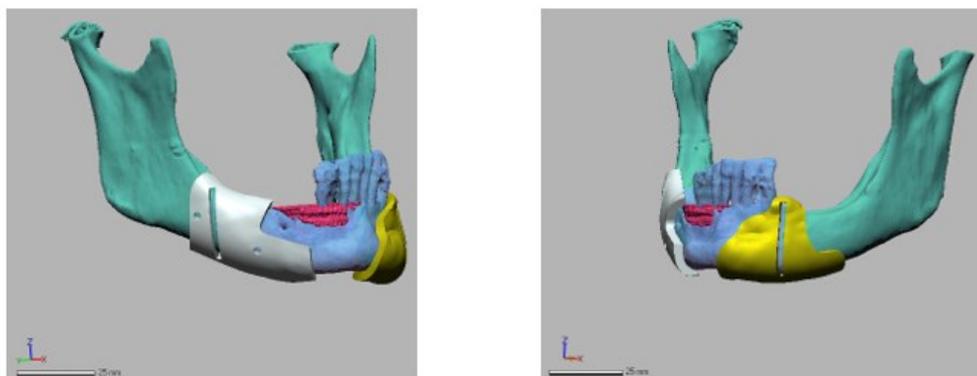


Figura 37: Dime di taglio rispettivamente per lato destro e lato sinistro dell'osso mandibolare progettate da esperto CAD

5.1 Procedura CAD standard

La procedura su Geomagic Design X prevede che le componenti della dima chirurgica siano costruite nel seguente ordine:

- Base della dima;
- Fessura per l'inserimento della lama;
- Fori per i perni.

Per dare inizio alla fase di design, il software necessita di elementi che siano descrivibili tramite funzioni matematiche come le superfici; in questo modo, è dunque possibile svolgere operazioni Booleane.

Aspetto fondamentale di questa procedura è quello di rispettare la distanza minima di 1,5 cm dal tumore (distanza che equivale al margine di sicurezza). Ecco che quindi, si sfrutta questo input al fine di stabilire quale sia la migliore collocazione dei piani lungo il quale andare ad effettuare la resezione. In realtà, questa selezione non è quella definitiva e pertanto non è detto che i piani di taglio creati in questa prima fase siano coincidenti con quelli finali. È da

considerare infatti che la direzione lungo il quale si recide l'osso, deve essere quantomeno perpendicolare rispetto all'asse dell'osso. Questo è evidente dalla fig. 38, sotto riportata: i piani Plane1 e Plane2 sono i primi due piani tracciati all'inizio della procedura; il piano CUT_DX deriva dalla traslazione del piano Plane1 e sarà la direzione definitiva per il taglio sul lato destro della mandibola; allo stesso modo, il piano CUT_SX deriva dalla traslazione del piano Plane2; il piano TaglioSX_ultimo rappresenta la direzione definitiva per la resezione sul lato sinistro della mandibola ma non è stato individuato in maniera così immediata.

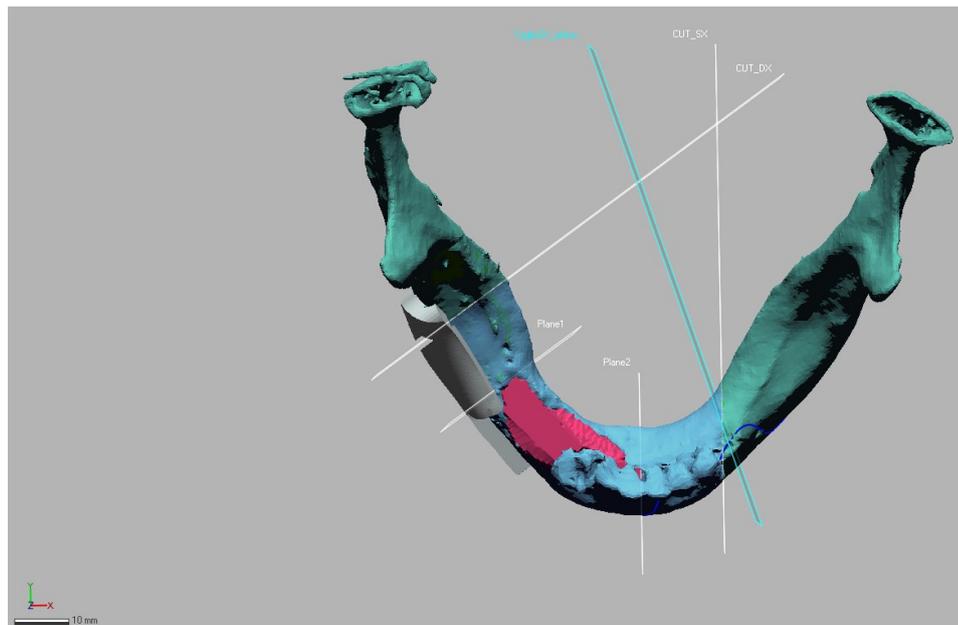


Figura 38: Anteprima dei piani di taglio su Geomagic Design X

Base della dima

Per creare la base della dima si sono seguite due procedure distinte sui lati destro e sinistro dell'osso mandibolare.

Sul lato destro della mandibola, al fine di assicurarsi che la superficie inferiore della dima sia coincidente con la superficie superiore della mesh dell'osso, è stato utilizzato un metodo che prevede la creazione di una superficie perfettamente inclusa tra le due nominate precedentemente. Di seguito, è stata ispessita di un valore pari a 3mm.

Sul lato sinistro della mandibola, la dima è stata progettata a partire da uno schizzo 3D libero basato sul tracciamento di una *spline* rispetto al quale sarà poi effettuato l'ispessimento in modo da generare la dima (fig. 39).

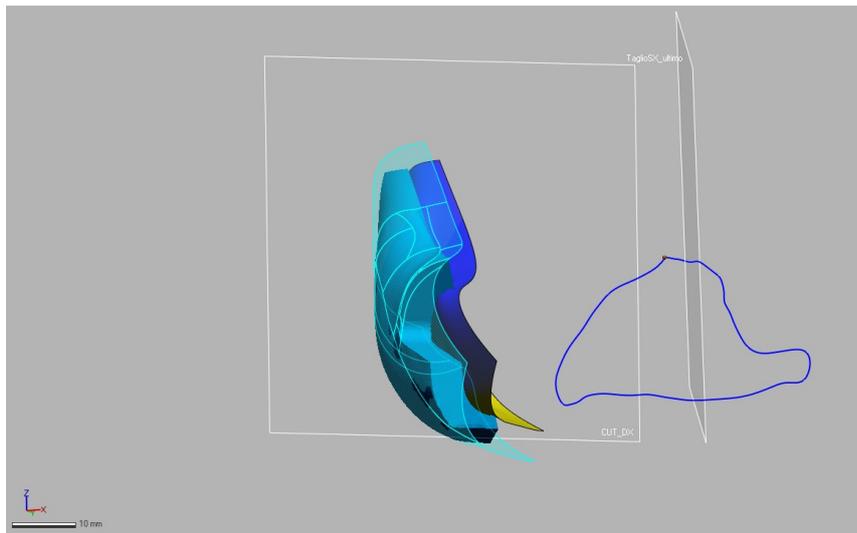


Figura 39: Metodologia di partenza per la progettazione delle dime chirurgiche

Fessure di taglio e fori dei perni

Per entrambe le componenti, i passaggi svolti sono equivalenti. Pertanto sono previste la:

- Creazione di uno schizzo;
- Estrusione dello schizzo lungo le due direzioni;
- Operazione di taglio tra il parallelepipedo e il cilindro generati dall'ultimo passaggio e la superficie della dima di taglio generata nei passi precedenti, rispettivamente per la fessura di taglio e i fori.

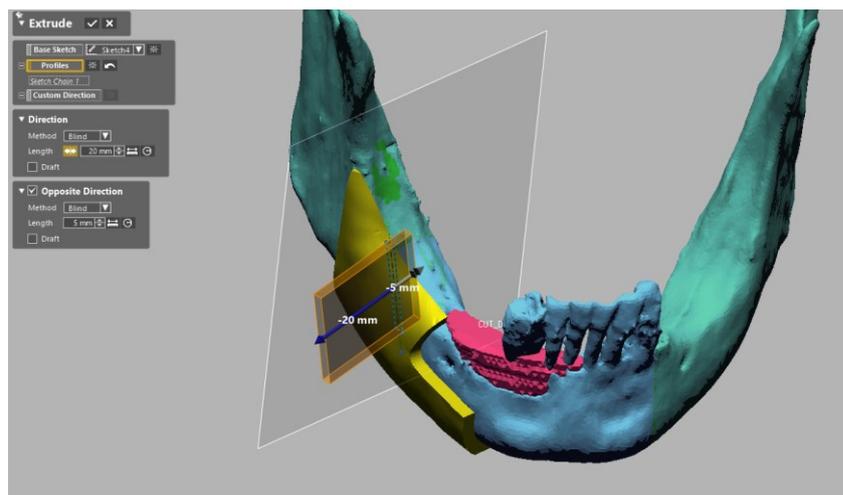


Figura 40: Creazione della fessura di taglio

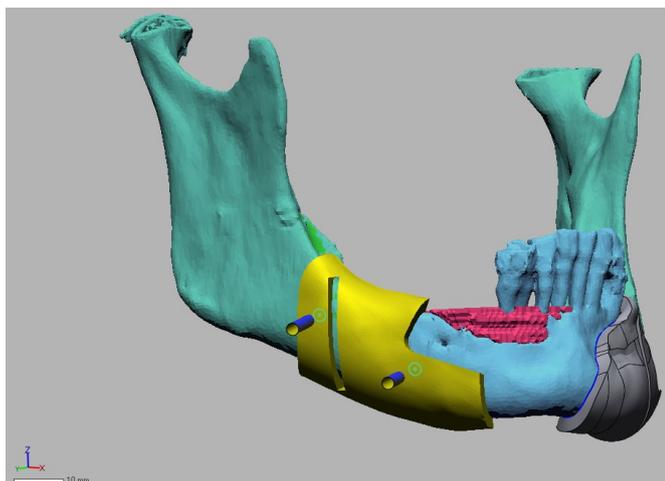


Figura 41: Creazione dei fori dei perni

5.2 Procedura CAD con Realtà Virtuale

Come già anticipato nel paragrafo 4.2, il risultato della procedura eseguita su Unity equivale all'esportazione di un insieme di file in formato FBX, successivamente convertiti in formato STL, al fine di poter essere importati all'interno della piattaforma di Geomagic Design X.

Giunti a questo punto della procedura proposta, il clinico trasmette questi dati all'ingegnere, inclusi i piani di taglio; quest'ultimo, a partire dalle mesh dei piani di taglio importati su Geomagic, è in grado di estrarre i piani veri e propri rispetto al quale progetterà la dima chirurgica seguendo la procedura convenzionale in maniera invariata.

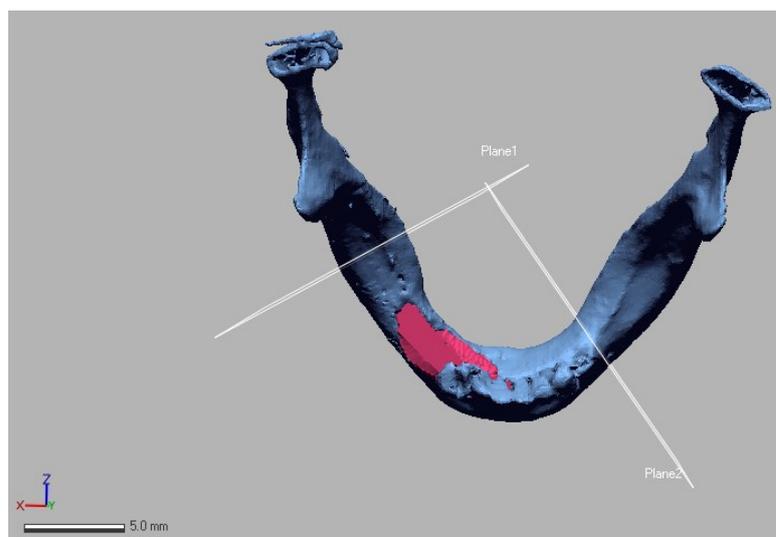


Figura 42: Anteprima dei piani di taglio su Geomagic Design X esportati da Unity

5.3 Risultati della procedura CAD con Realtà Virtuale

Nel testare l'applicazione di realtà virtuale si è potuto osservare, anche se non tramite direttamente un chirurgo, che è stato possibile sfruttare uno strumento alternativo per poter avvantaggiare la comunicazione e collaborazione tra il clinico e l'ingegnere.

Il vantaggio principale della procedura proposta rispetto a quella standard è che il clinico in maniera del tutto autonoma stabilisce in partenza quale sia la posizione ottimale per i piani di taglio e rispetto a questi saranno successivamente generati, tramite l'aiuto dell'esperto CAD, i profili delle fessure di resezione. In questo modo, si risparmiano i primi passaggi iniziali per stabilire i piani che delimitano i tumori e che saranno traslati di uno specifico offset al fine di individuare le direzioni di taglio definitive. La novità più importante qui sarebbe che il clinico prende parte attiva all'interno del processo di design, cosa che quasi certamente non può verificarsi attraverso Geomagic Design X data la sua complessità di utilizzo.

Si riconosce tuttavia che tale applicazione presenta degli svantaggi tra cui il fatto che durante l'immersione nella scena virtuale il chirurgo non ha la possibilità di poter stabilire quanto misura la distanza dal tumore. Sarebbe stato inoltre più intuitivo poter ruotare e manipolare il modello implementando l'hand tracking piuttosto che farlo attraverso i pulsanti del joystick.

Capitolo 6

Conclusioni

In conclusione, questo lavoro di tesi è consistito nello sviluppo di una applicazione VR in ambito di pianificazione chirurgica. La tecnologia basata sulla realtà virtuale e sulla realtà aumentata è un'area in rapida crescita della medicina, che è destinata a rivoluzionare le applicazioni mediche nel prossimo futuro grazie alle scoperte metodologiche, tecniche e produttive degli ultimi anni.

In questo elaborato si è prima posta attenzione al workflow adottato oggi nell'ambito della pianificazione chirurgica virtuale e dunque nello sviluppo di modelli CAD di strumenti chirurgici paziente-specifici. Si è poi entrati nel dettaglio del mondo della realtà virtuale, partendo dai cenni storici, continuando poi con le tipologie di visori di realtà virtuale attualmente presenti in commercio e come questi sono adottati nell'ambito chirurgico. Ci si è focalizzati maggiormente sul visore Oculus Quest 2 e il motore di gioco Unity utilizzati per questo progetto di tesi, andando ad investigare sulle loro caratteristiche innovative e cosa li rende facili da utilizzare ed accessibili a qualsiasi tipo di pubblico. Infine, si è descritta la fase di progettazione dell'applicazione e come questa è stata pensata per poter essere integrata all'interno della procedura preesistente di sviluppo di modelli CAD di dispositivi medici customizzati.

L'ambito medico sul quale ci si è più focalizzati è stato quello di oncologia muscolo-scheletrica nei casi in cui è possibile andare ad intervenire tramite un intervento di osteotomia. L'idea di partenza per questa applicazione era quella di poter sviluppare uno strumento che, oltre a poter fornire supporto nella visualizzazione del modello anatomico 3D e nella scelta della giusta collocazione per il piano di taglio, potesse simulare la resezione del modello al fine di fornire una migliore prospettiva della zona di intervento ma data la difficoltà nel reperire i giusti algoritmi per lo specifico caso di applicazione si è preferito sorvolare su questo punto e concentrarsi in ogni caso sull'idea che questa applicazione VR potesse comunque essere di interesse nell'ambito delle progettazioni di dime di taglio.

Per raggiungere questo scopo è stato fondamentale studiare la procedura convenzionalmente adottata dagli esperti CAD sulla piattaforma di Geomagic Design X e comprendere come poter integrare l'applicazione all'interno di questo metodo.

Bibliografia e sitografia

- [1] W. P. T. A. H. S. M. S. P. Benjamin D Foley, «Mandibular reconstruction using computer-aided desing and computer-aided manufacturing: an analysis of surgical results,» *Journal of Oral and Maxillofacial Surgeons*, pp. 111-9, 2013.
- [2] C. M. R. S. R. C. R. S. L. C. Simona Mazzoni, «Prosthetically guided maxillofacial surgery: evaluation of the accuracy of a surgical guide and custom-made bone plate in oncology patients after mandibular reconstruction,» *Plastic and Reconstructive Surgery*, pp. 1376-1385, 2013.
- [3] B. M. H. T. J. O. A. E. A. J. M. M. E. J. m. J. L. K. J. R. J. K. M. V. A. H. J. D. D. L. P. Matthew M May, «Short and long-term outcomes of three-dimensional printed surgical guides and virtual surgical planning versus conventional methods for fibula free flap reconstruction of the mandible: Decreased nonunion and complication rates,» *Head and Neck*, pp. 2342-2352, 2021.
- [4] A. M. K. M. B. M. A. D. B. Sandeep W Dahake, «Applications of medical rapid prototyping assisted customized surgical guides in complex surgeries,» *Rapid Prototyping Journal*, pp. 934-946, 2016.
- [5] J. D. L. A. D. P. P. J. B. J. H. H. Fazel A Khan, «Surgical technique: Computer-generated custom jigs improve accuracy of wide resection of bone tumors,» *Clinical Orthopaedics and Related Research*, pp. 2007-2016, 2013.
- [6] F. P. M. A. A. Z. G. K. G. R. H. W. S. K Radermacher, «Computer assisted orthopaedic surgery with image based individual templates,» *Clinical Orthopaedics and Related Research*, pp. 28-38, 1998.
- [7] L. P. P.-L. D. Laura Bellanova, «Surgical Guides (Patient-Specific Instruments) for Pediatric Tibial Bone Sarcoma Resection and Allograft Reconstruction,» *Sarcoma*, 2013.
- [8] Wolfram MathWorld, «Radon Transform,» [Online]. Available: <https://mathworld.wolfram.com/RadonTransform.html>. [Consultato il giorno Febbraio 2023].
- [9] A. Y. M. A. I. S. S. A. K. Usman Khan, «A Methodological Review of 3D Reconstruction Techniques in Tomographic Imaging,» *Journal od Medical System*, 2018.

- [10] F. R. F. U. M. G. Erica Nocerino, «3D Modelling and Rapid Prototyping for Cardiovascular Surgical Planning - Two Case Studies,» *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 887-893, 2016.
- [11] M. B. J. P. K. K. A. P. B. H. P. M. G. J. W. Steven Arild Wuyts Andersen, «Segmentation of Temporal Bone Anatomy for Patient-Specific Virtual Reality Simulation,» *The Annals of Otolaryngology, Rhinology and Laryngology*, pp. 724-730, 2021.
- [12] T. C. H. M. M. G. H. K. H. Lei Wang, «Principles and methods for automatic and semi-automatic tissue segmentation in MRI data,» *MAGMA*, pp. 95-110, 2016.
- [13] J. K. Zuzana Kresanova, «Comparison of Software for Medical Segmentation,» *Central European Researchers Journal*, pp. 66-80, 2018.
- [14] M. C. C. C. L. B. F. M. M. F. F. M. A. P. Vincenzo Ferrari, «Value of multidetector computed tomography image segmentation,» *Surgical Endoscopy*, pp. 616-626, 2012.
- [15] Materialise Mimics, «Materialise Mimics,» [Online]. Available: <https://www.materialise.com/en/healthcare/mimics-innovation-suite/mimics>. [Consultato il giorno Febbraio 2023].
- [16] A. M. A.-A. H. A. A. A. M. A. Emad A Nasr, «Developing a methodology for analysis and manufacturing of proximal interphalangeal (PIP) joint using rapid prototyping technique,» *Rapid Prototyping Journal*, pp. 449-460, 2015.
- [17] F. F. R. F. M. G. L. G. A. M. F. U. Y. V. Monica Carfagni, «Towards a CAD-based automatic procedure for patient specific cutting guides to assist sternal osteotomies in pectus arcuatum surgical correction,» *Journal of Computational Design and Engineering*, pp. 118-127, 2019.
- [18] 3D Systems, «3D Systems,» [Online]. Available: <https://it.3dsystems.com/software>. [Consultato il giorno Febbraio 2023].
- [19] MeshLab, «MeshLab,» [Online]. Available: <https://www.meshlab.net/>. [Consultato il giorno Febbraio 2023].
- [20] C. M. R. S. R. C. R. S. L. C. S. Mazzone, «Prosthetically guided maxillofacial surgery: evaluation of the accuracy of a surgical guide and custom-made bone plate in oncology patients after mandibular reconstruction,» *Plastic and Reconstructive Surgery*, pp. 1376-1385, 2013.
- [21] F. B. A. M. R. F. Lorenzo Guariento, «A Semi-Automatic CAD Procedure to Design Custom-made Surgical Cutting Guides,» *Computer-Aided Design and Applications*, pp. 733-740, 2022.

- [22] J. D. A. L. M. S. S. G. S. G. S. G. Caiti, «Implementation of a semiautomatic method to design patient-specific instruments for corrective osteotomy of the radius,» *International Journal of Computer Assisted Radiology and Surgery*, pp. 829-840, 2019.
- [23] F. B. M. C. Y. V. F. F. M. G. A. M. Michaela Servi, «CAD - based automatic modelling of customized cutting templates for Pectus Arcuatum surgical correction,» pp. 6044-6048, 2020.
- [24] M. L. Heiling, «Stereoscopic-television apparatus for individual use,» *New York United States Patent Office*, 1960.
- [25] A. Faisal, «Computer science: Visionary of virtual reality,» *Nature*, pp. 298-299, 2017.
- [26] D. Hershkovich, «Medium,» 2016. [Online]. Available: <https://medium.com/byondvr/a-beginners-guide-to-tethered-untethered-vr-headsets-43fa17df3902>. [Consultato il giorno Gennaio 2023].
- [27] Aniwaa Team, «Aniwaa,» 6 Agosto 2021. [Online]. Available: <https://www.aniwaa.com/guide/vr-ar/types-of-vr-headsets/>. [Consultato il giorno Gennaio 2023].
- [28] S. Aranzulla, «Salvatore Aranzulla,» [Online]. Available: <https://www.aranzulla.it/realta-virtuale-come-funziona-1157551.html>. [Consultato il giorno Gennaio 2023].
- [29] D. Robertson, «gamesradar,» 23 Gennaio 2023. [Online]. Available: <https://www.gamesradar.com/the-best-vr-headset/>. [Consultato il giorno Gennaio 2023].
- [30] A. Baker, «Thewild,» 31 Gennaio 2022. [Online]. Available: <https://thewild.com/blog/architect-getting-started-with-vr>. [Consultato il giorno Gennaio 2023].
- [31] G. Vaquez-Mata, «Virtual reality and simulation in the training of medical students,» *Educacion Medica*, 2008.
- [32] J. A. Windsor, «Role of simulation in surgical education and training,» *Journal of Surgery*, pp. 127-132, 2009.
- [33] L. Hanna, «Simulated surgery: the virtual reality of surgical training,» *Surgery*, pp. 463-468, 2010.
- [34] E. G.-V. C. P.-G. J. G.-V. L. T. M. R. J. H.I. Medel in-Castillo, «The evaluation of a novel haptic-enabled virtual reality approach for computer-aided cephalometry,» *Computer Methods and Programs in Biomedicine*, 2016.

- [35] H. I. M.-C. J. B. M. A. R.-F. Eder H. Govea-Valladares, «On the development of virtual reality scenarios for computer-assisted biomedical applications,» *Journal of Healthcare Engineering*, 2018.
- [36] M. P. Thomas, «The role of simulation in the development of technical competence during surgical training: a literature review,» *International Journal of Medical Education*, pp. 48-58, 2013.
- [37] F. V. B. H. H. T. H. F. K. W. P. B. S. P. H. C. Chheang V, «Toward interprofessional team training for surgeons and anesthesiologists using virtual reality,» *International Journal of Computer Assisted Radiology and Surgery*, pp. 2109-2118, 2020.
- [38] G. D. K. I. B.-Y. O. L. M. L. N. L. M. D. Y. Einav Y, «Preoperative briefing in the operating room: shared cognition, teamwork, and patient safety,» *Chest*, pp. 443-449, 2010.
- [39] S. P. J. F. B. C. H. T. H. F. L. H. P. B. H. C. Chheang V, «A collaborative virtual reality environment for liver surgery planning,» *Computer & Graphics*, pp. 234-246, 2021.
- [40] OssoVR, «OssoVR,» [Online]. Available: <https://www.ossovr.com/>.
- [41] N. S. B. J. Tobias Fritz, «Evidence in surgical training - a review,» *Innovative Surgical Sciences*, 2019.
- [42] J. V. J. Hitendra R.H. Patel, «Simulation training in laparoscopy and robotic surgery,» 2012.
- [43] T. M. D. S. T. D. R. M. D. W. Anke V. Reinschluessel, «Virtual reality for surgical planning - evaluation based on two liver tumor resections,» *Frontiers in surgery*, 2022.
- [44] E. P. R. B. H. B. R. P. D. L. A. A. A. F. B. E. O. J. E. Rahul Prasanna Kumar, «Use of mixed reality for surgery planning: assessment and development workflow,» *Journal of Biomedical Informatics*, 2020.
- [45] A. Robertson, «theverge.com,» 16 Settembre 2020. [Online]. Available: <https://www.theverge.com/21433030/oculus-quest-2-vr-headset-specs-comparison-htc-valve-microsoft>. [Consultato il giorno Gennaio 2023].
- [46] B. Kuchera, «Polygon,» 16 Settembre 2020. [Online]. Available: <https://www.polygon.com/reviews/2020/9/16/21437762/oculus-quest-2-review-virtual-reality-vr-facebook-oculus-power-resolution-tracking>. [Consultato il giorno Gennaio 2023].

- [47] «VRX,» 13 Aprile 2022. [Online]. Available: <https://vr-expert.com/the-most-important-differences-between-the-oculus-quest-1-and-2/>. [Consultato il giorno Gennaio 2023].
- [48] S. Shaw, «Stealthoptional,» 19 Agosto 2022. [Online]. Available: https://stealthoptional.com/gaming/how-to-get-full-body-tracking-with-oculus-quest-2/?_x_tr_sl=en&_x_tr_tl=it&_x_tr_hl=it&_x_tr_pto=wapp. [Consultato il giorno Gennaio 2023].
- [49] MetaAI, 22 Agosto 2019. [Online]. Available: <https://ai.facebook.com/blog/powered-by-ai-oculus-insight/>. [Consultato il giorno Gennaio 2023].
- [50] A. Ranganathan, «AI Accelerator Institute,» 27 Giugno 2022. [Online]. Available: <https://www.aiacceleratorinstitute.com/the-oculus-insight-positional-tracking-system-2/>. [Consultato il giorno Gennaio 2023].
- [51] «DigitalMosaik,» 15 Ottobre 2020. [Online]. Available: <https://www.digitalmosaik.com/blog/oculus-quest-2>. [Consultato il giorno Gennaio 2023].
- [52] Meta, [Online]. Available: <https://www.meta.com/it-it/help/quest/articles/headsets-and-accessories/controllers-and-hand-tracking/hand-tracking-quest-2/>. [Consultato il giorno Gennaio 2023].
- [53] «Meta,» [Online]. Available: <https://www.meta.com/it-it/help/quest/articles/accounts/privacy-information-and-settings/hand-tracking-privacy-notice/>. [Consultato il giorno Gennaio 2023].
- [54] R. Grasso, «Hardware Upgrade,» 16 Ottobre 2020. [Online]. Available: https://www.hwupgrade.it/articoli/periferiche/5854/recensione-oculus-quest-2-e-il-visore-vr-definitivo_index.html. [Consultato il giorno Gennaio 2023].
- [55] «Oculus,» 11 Novembre 2022. [Online]. Available: <https://creator.oculus.com/getting-started/immersive-media-glossary/>. [Consultato il giorno Gennaio 2023].
- [56] A. Sinicki, «Android Authority,» 20 Marzo 2021. [Online]. Available: <https://www.androidauthority.com/what-is-unity-1131558/>. [Consultato il giorno Gennaio 2023].
- [57] «Unity,» [Online]. Available: <https://assetstore.unity.com/>. [Consultato il giorno Gennaio 2023].
- [58] «Unity3d,» [Online]. Available: <https://docs.unity3d.com/Manual/GameObjects.html>. [Consultato il giorno Gennaio 2023].

[59] «Unity3d,» [Online]. Available: <https://docs.unity3d.com/Manual/Prefabs.html>. [Consultato il giorno Gennaio 2023].

[60] «Unity3d,» [Online]. Available:

<https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>. [Consultato il giorno Gennaio 2023].

[61] S. Nagvani, «Medium,» 4 Aprile 2020. [Online]. Available:

<https://medium.com/@shailendra2011991/grabbing-an-object-in-vr-oculus-quest-with-unity-1cc20510bde6>. [Consultato il giorno Gennaio 2023].

[62] Unity, «Unity,» [Online]. Available:

<https://docs.unity3d.com/Packages/com.unity.formats.fbx@2.0/manual/exporting.html>.
[Consultato il giorno Gennaio 2023].