

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA  
CAMPUS DI CESENA

---

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA  
Corso di Laurea in Ingegneria e Scienze Informatiche

STUDIO E SVILUPPO PROTOTIPALE DI  
UN SISTEMA PER LA VISUALIZZAZIONE  
E LA MANIPOLAZIONE DI CONTENUTI  
DICOM IN REALTÀ AUMENTATA

*Elaborato in*  
SISTEMI EMBEDDED E INTERNET OF THINGS

*Relatore*  
Prof. ALESSANDRO RICCI

*Presentata da*  
DANIELE GIULIANINI

*Corelatore*  
Dott. Ing. ANGELO CROATTI

Anno Accademico 2018 – 2019



*Alla mia famiglia*



# Indice

<b>Introduzione</b>	<b>ix</b>
<b>1 Realtà aumentata in ambito healthcare</b>	<b>1</b>
1.1 Realtà aumentata e realtà virtuale . . . . .	1
1.2 Realtà mista . . . . .	2
1.3 Differenze fra realtà aumentata, realtà virtuale e realtà mista . .	3
1.4 Benefici e potenzialità di AR . . . . .	3
1.4.1 Nuovo Paradigma . . . . .	4
1.4.2 Panoramica generale . . . . .	4
1.5 Applicazioni mediche . . . . .	4
1.6 AR e MR in sala operatoria . . . . .	6
1.6.1 AR e lo staff chirurgico . . . . .	7
1.6.2 AR come tecnica complementare alla stampa 3D . . . . .	9
1.6.3 AR come supporto a procedure mininvasive . . . . .	9
1.6.4 Feedback tattile . . . . .	10
1.6.5 Formazione del personale sanitario . . . . .	10
1.7 Maggiori difficoltà e prospettive future . . . . .	11
1.7.1 Aspetti non prettamente tecnici . . . . .	11
<b>2 Il progetto di tesi</b>	<b>13</b>
2.1 Descrizione del progetto . . . . .	13
2.2 Potenzialità . . . . .	13
2.3 Motivazioni . . . . .	14
2.4 Analisi dei requisiti . . . . .	15
2.4.1 Input del sistema . . . . .	15
2.4.2 Requisiti funzionali . . . . .	16
2.4.3 Requisiti non funzionali . . . . .	17
2.5 Problematiche principali . . . . .	17
<b>3 DICOM</b>	<b>19</b>
3.1 Introduzione e scopo . . . . .	19
3.2 Ambito di applicazione . . . . .	20

3.3	Breve storia . . . . .	20
3.4	Modello di informazione . . . . .	21
3.4.1	Information Object Definition . . . . .	21
3.4.2	SOP Class e Service Class . . . . .	22
3.4.3	Information Entity . . . . .	22
3.4.4	UID . . . . .	23
3.5	Formato dati . . . . .	24
3.5.1	Data structure . . . . .	24
3.5.2	Le innovazioni introdotte . . . . .	25
3.5.3	File structure . . . . .	26
3.5.4	Transfer syntax . . . . .	27
3.6	DICOM e l'immagine . . . . .	28
3.6.1	Rappresentazione dell'immagine . . . . .	28
3.6.2	Greyscale rendering pipeline . . . . .	30
3.6.3	Orientazione dell'immagine nello spazio . . . . .	33
<b>4</b>	<b>Progettazione e implementazione</b>	<b>39</b>
4.1	Dal DICOM alla AR . . . . .	39
4.1.1	Step operativi . . . . .	39
4.2	Pipeline operativa . . . . .	41
4.2.1	Tecnologie . . . . .	41
4.2.2	Elaborazione procedurale dei dati sulla mesh . . . . .	50
4.3	Modellazione e sviluppo degli elementi DICOM . . . . .	52
4.3.1	Design . . . . .	52
4.4	Progettazione e sviluppo con ARCore e Meta 2 . . . . .	57
4.4.1	Prototipazione . . . . .	57
4.4.2	Tecnologie utilizzate . . . . .	57
4.4.3	ARCore . . . . .	60
<b>5</b>	<b>Estensioni alle funzionalità di base</b>	<b>69</b>
5.1	Descrizione . . . . .	69
5.2	Analisi . . . . .	70
5.2.1	Nuove funzionalità . . . . .	70
5.2.2	Problemi principali . . . . .	71
5.3	Progettazione e Sviluppo . . . . .	74
<b>6</b>	<b>Validazione e sviluppi futuri</b>	<b>79</b>
6.1	Validazione . . . . .	79
6.2	Problematiche . . . . .	81
6.3	Lavori futuri . . . . .	81
	<b>Conclusioni</b>	<b>87</b>

*INDICE*

vii

**Ringraziamenti**

**89**





# Introduzione

Negli ultimi anni c'è stato un notevole incremento dell'interesse nei confronti della realtà aumentata, dovuto soprattutto ai recenti sviluppi tecnologici che hanno permesso la sua introduzione al grande pubblico. Sulle sue incredibili potenzialità a lungo termine, alcuni ricercatori [4] si sono espressi individuandola come la maggiore candidata a soppiantare la tecnologia degli smartphone e a divenire l'interfaccia dominante d'accesso all'informazione. L'estensione della percezione umana mediante la sovrapposizione di un layer informativo digitale, volto a potenziarla, preservando comunque l'interazione con l'ambiente circostante pone le basi per numerose applicazioni, in svariati ambiti e modalità. L'AR è in questo senso un esempio specifico di ciò che Fred Brooks definisce “amplificazione dell'intelligenza” (IA), utilizzare un dispositivo computazionale per rendere le attività umane più facili da eseguire (Brooks, 1996).

L'idea da cui si sviluppa il lavoro di tesi è quella di sfruttare le potenzialità della realtà aumentata nel contesto biomedicale, proponendo un'alternativa innovativa alle tradizionali modalità di visualizzazione dell'immagine. La scarsità di sistemi di questo tipo pubblicamente noti ha richiesto una fase preliminare di studio: dapprima sono state esplorate le specifiche di DICOM, ossia lo standard internazionale che definisce, oltre agli altri, i criteri di memorizzazione delle informazioni di tipo biomedico. In un secondo momento, sono state individuate ed approfondite le tecnologie attualmente disponibili nell'ambito di realtà aumentata e realtà mista, oltre che ai device attualmente con esse compatibili. Il percorso di tesi è poi proseguito con lo sviluppo prototipale di un sistema per la visualizzazione e la manipolazione in realtà aumentata di contenuti DICOM. Lo scopo di questo progetto è stato quello di implementare dapprima un prototipo che realizzasse l'idea iniziale, e in seguito di ricevere una valutazione sul reale impatto di realtà mista ed aumentata nel contesto medico, dal momento che, attualmente, le potenzialità in tale ambito applicativo risultano ancora prevalentemente sconosciute ai diretti interessati. Il prototipo sarà fruibile attraverso 2 diversi stack tecnologici, e il suo obiettivo sarà quello di ricevere la valutazione di esperti nella bioingegneria e di medici, che abbiamo coinvolto nel nostro progetto. La tesi si articola in 6 capitoli:

- nel primo capitolo vengono definiti i concetti di realtà aumentata, realtà mista e virtuale e vengono presentate le principali applicazioni in ambito medico presenti in letteratura. Partendo da un'introduzione generale, si passerà a descrivere un approfondimento sull'attività chirurgica, proponendo esempi di implementazioni già operative.
- il secondo capitolo introduce il progetto affrontato, definendone motivazioni, potenzialità e requisiti, in modo da orientare la trattazione successiva nella sua prospettiva.
- il terzo capitolo analizza il formato DICOM, ovvero lo standard internazionale de-facto impiegato nella codifica delle immagini medicali. Si tratta di uno standard molto affermato in ambito medico e la sua analisi è il punto di partenza per lo sviluppo del sistema.
- Il quarto capitolo descrive l'analisi, la progettazione e lo sviluppo del prototipo, in una sua prima versione di base. Oltre ad ottenere una rappresentazione in realtà mista del DICOM, si identifica e sviluppa un insieme di funzionalità di base volte a fornire un supporto efficace alla visualizzazione delle immagini estratte da DICOM.
- Il quinto capitolo illustra un esempio di funzionalità più specifiche rispetto al prototipo di base. Vengono definite le motivazioni di tale scelta e le strategie adottate per realizzarle.
- il sesto ed ultimo capitolo costituisce il capitolo di validazione. Vengono discussi i punti deboli del sistema e riportati i pareri di professionisti in ambito medico e biomedico che lo hanno testato. In questa sede vengono riassunte le indicazioni per gli sviluppi futuri, le quali sono il risultato del confronto con tali professionisti e dovranno essere le prime ad essere prese in considerazione per chi porterà avanti questo progetto.

# Capitolo 1

## Realtà aumentata in ambito healthcare

### 1.1 Realtà aumentata e realtà virtuale

Il termine realtà aumentata indica un'esperienza sensoriale in cui la percezione umana della realtà viene potenziata attraverso l'integrazione, dinamica e senza soluzione di continuità, di informazioni e contenuti digitali, altrimenti non percepibili attraverso i 5 sensi. Il significato del termine, coniato nel 1991, ha subito negli anni delle rielaborazioni e tuttora non gode di un'uniformità di interpretazioni. Dal punto di vista accademico, uno dei contributi più importanti alla definizione del termine è quello enunciato nel 1994 da Milgram, attraverso la rappresentazione del "virtuality continuum", dove la definizione di realtà aumentata assume significato in relazione a quella di realtà virtuale. Secondo Milgram e Kishin, la realtà virtuale è "l'esperienza nella quale l'osservatore è totalmente immerso in un mondo completamente sintetico, le cui proprietà possono imitare o meno quelle del mondo reale, dove le leggi della fisica che conosciamo possono non essere più vigenti". All'estremo opposto si trova, invece, la realtà come la conosciamo, ovvero regolata dalle leggi della gravità, del tempo e delle proprietà materiali. Realtà virtuale e mondo reale vengono identificati dunque come i due opposti in una scala del continuo che li unisce e che racchiude i concetti, fra loro simmetrici, di virtualità aumentata e realtà aumentata. La AR dunque si pone a metà strada tra un ambiente del tutto virtuale e un ambiente reale, in cui però gli elementi del secondo ambiente sono prevalenti rispetto a quelli del primo (Klopfer, 2008), ovvero "La realtà aumentata corrisponde alla situazione dove il mondo reale è aumentato da entità virtuali, mentre la virtualità aumentata è quella in cui il mondo virtuale è aumentato da entità reali" [8].

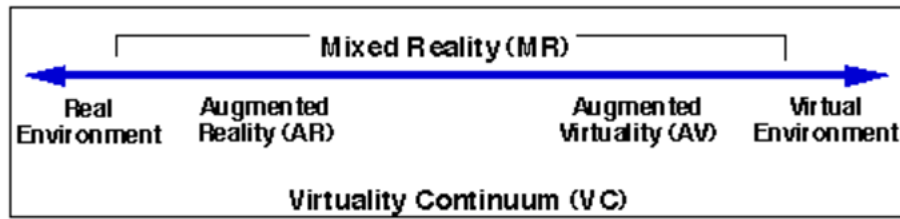


Figura 1.1: Rappresentazione schematica del reality-virtuality continuum, tratto da [15]

La definizione di AR più largamente accettata è, tuttavia, quella del ricercatore T. Ronald Azuma [3], che, pur rimanendo in continuità rispetto alla letteratura, introduce un maggiore specificità nel termine, esprimendolo non in dipendenza dal concetto di realtà virtuale, ma piuttosto vincolandolo al rispetto di 3 caratteristiche:

- Combinazione di reale e virtuale,
- Interattività in tempo reale,
- Registrazione in 3D.

Alla luce di queste precisazioni, le forme di sovrapposizioni di contenuti digitali in 2D e le tecniche di simulazione proprie dell'industria cinematografica sono escluse dalla sfera della realtà aumentata.

## 1.2 Realtà mista

Il termine realtà mista (MR) è in verità ancora più controverso dal punto di vista semantico. Recenti analisi [21] individuano in letteratura fino a 6 definizioni non del tutto compatibili. Come per la realtà aumentata, anche in questo caso la comprensione del concetto realtà mista è subordinata a quello di realtà virtuale. Il punto di vista attraverso il quale è stato tradizionalmente interpretato il termine è anche in questo caso quello enunciato nel 1994 da Milgram e Kishino [15], che, attraverso la rappresentazione del virtuality continuum, identificano la realtà mista come: “qualunque cosa fra gli estremi del continuum della virtualità”, ovvero qualsiasi esperienza o tecnologia posta fra la percezione del mondo reale e quella di un mondo totalmente virtuale (realtà virtuale)”. Oggi, viene largamente condivisa un'accezione più specifica del termine, che denota un'esperienza sensoriale dove l'attenzione è posta maggiormente sull'aspetto di interazione: sia essa fra le entità del mondo reale e quelle del mondo virtuale sia fra queste ultime e l'osservatore. In questo

testo si farà fede a questa interpretazione, anche se questo significato è incompatibile con la definizione di Milgram e Kishino [15], dato che non include in sé il concetto di realtà aumentata.

### 1.3 Differenze fra realtà aumentata, realtà virtuale e realtà mista

Nonostante i tre termini siano spesso utilizzati intercambiabilmente, nelle accezioni sopra descritte essi indicano concetti ben distinti. Mentre in AR e MR il substrato dell'esperienza è costituito dal mondo reale, la VR fa riferimento ad uno scenario, che, nonostante possa somigliare a quello reale, è di origine completamente artificiale. VR prevede dunque un'occlusione totale rispetto alla visione del mondo circostante, mentre AR e MR la integrano con informazioni aggiuntive, le quali costituiscono l'unica componente generata artificialmente. Per questo motivo, generalmente, la VR offre all'utente solamente 3 gradi di libertà di movimento, a differenza dei 6 di AR e MR. Sebbene condividano aspetti tecnologici, soprattutto relativi alla gestione grafica, i dispositivi di AR e MR sono estremamente più complessi, poiché devono farsi carico di sofisticati compiti, quali il costante tracciamento della posizione dell'osservatore e degli oggetti e, soprattutto, della coniugazione dei 2 flussi informativi (quello generato artificialmente e quello reale), che pongono serie questioni in materia di ottica e rappresentano ad oggi il fattore tecnico più limitante. Anche nell'AR gli oggetti digitali sovrapposti alla visione della realtà sono registrati spazialmente ed interattivi, ma, da questo punto di vista, la MR è ancora più esigente. Gli oggetti digitali non sono solamente sovrapposti al mondo reale, ma devono fondersi ed interagire con quelli reali come se fossero parte della stessa realtà. L'utente deve poterli manipolare senza che il loro comportamento diventi incoerente con l'ambiente che li circonda, nel contesto di un'esperienza in cui l'immersività e l'interazione sono portati ad un livello superiore di AR, che per questo richiede maggiori capacità computazionali ed un orizzonte temporale di piena realizzazione ancora più lontano.

### 1.4 Benefici e potenzialità di AR

La realtà aumentata non solo è stata individuata come possibile risorsa di innumerevoli applicazioni specifiche ai settori più disparati, ma potrà svolgere un ruolo fondamentale nel rivoluzionare l'accesso alle informazioni da parte dell'utente[2, 3...], inserendosi nel cambiamento in atto da anni in questa direzione.

### 1.4.1 Nuovo Paradigma

Come spiegato nell'introduzione, l'interesse nei confronti della realtà aumentata è in crescente aumento. Nonostante poco tempo fa fosse per lo più sconosciuta al grande pubblico, la letteratura da tempo l'ha individuata come futura tecnologia emergente. Le rosee aspettative che il mondo accademico vi ripone da anni derivano dalla semplice constatazione che l'informazione trasmessa dagli oggetti virtuali aiuta l'utente nell'esecuzione di compiti del mondo reale. Come evidenzia Schmalstieg [19], la promessa di creare un collegamento diretto, automatico ed azionabile fra il mondo fisico e l'informazione elettronica e la predisposizione ad un'interfaccia di utilizzo semplice ed immediato pongono le basi per un rivoluzionario cambio di paradigma nell'interfaccia fra utente ed informazione. Nel futuro, il cloud AR sarà il prossimo passo nell'evoluzione di internet e del web e il loro contenuto virtuale persistente sarà legato a specifiche posizioni reali, delineando una corrispondenza 1 a 1 fra il mondo reale e quello digitale. La realtà aumentata sarà l'interfaccia d'accesso a tale contenuto" [4].

### 1.4.2 Panoramica generale

La varietà delle possibili applicazioni d'influenza di realtà aumentata e realtà mista è stata sottolineata fin da alcuni decenni fa. Come evidenzia [3], almeno 6 aree di interesse erano state già riportate in letteratura a partire dal lontano 1997, fra le quali: manutenzione e riparazione, annotazione e visualizzazione di informazioni di interesse, pianificazione di movimenti in ambito robotico, navigazione aero-militare, configurazione dispositivi di puntamento, visualizzazione medica. Nonostante questo vasto arco temporale, paragonabile ad ere geologiche in un settore così dinamico ed in rapido sviluppo quale l'informatica, le implementazioni di tali idee risultano ancora in gran parte insoddisfacenti e tuttora limitano fortemente il loro impiego effettivo. Gli ambiti applicativi ad oggi documentati sono innumerevoli e spaziano dalla sfera videoludica a quella culturale, industriale, commerciale, clinica e medico-sanitaria. Come sottolineato da [16], questa lista pressoché infinita di applicazioni: "è una buona notizia in termini di opportunità di mercato e crescita, e una cattiva notizia se si vuole analizzare la tecnologia e il mercato", nella direzione delle implicazioni di sicurezza e privacy che pone dinnanzi.

## 1.5 Applicazioni mediche

In letteratura sono documentati casi di studio di AR applicata a numerose branche della medicina moderna. Il progressivo miglioramento tecnologico dei

dispositivi non solo sta consentendo di implementare progetti finora inattuabili, ma sta al contempo allargando ulteriormente la frontiera di possibile intervento: ad oggi la realtà aumentata viene per lo più sperimentata, ma ci si può aspettare che rientrerà nella prassi medica riconosciuta grazie ai benefici che può apportare ed alla sua accessibilità. I settori di maggior interesse comprendono la terapia contro deficit sensoriali, la riabilitazione motoria e cognitiva, la medicina diagnostica, il supporto alla disciplina chirurgica, l'apprendimento in un contesto di simulazione.

**Terapia contro deficit sensoriali** La capacità dell'AR di arricchire l'esperienza percettiva umana ha stimolato fin da subito la progettazione di sistemi di supporto ai pazienti con un deficit sensoriale finalizzati a potenziare o emulare la percezione coinvolta. Tali sistemi basano il loro funzionamento sulla sovrapposizione di uno strato informativo di diversa natura in base agli apparati coinvolti dal deficit, incaricato di sopperirvi, amplificando oppure trasformando l'informazione in una delle forme recepibili dal paziente. È il caso dell'applicazione per Android Voice (Google, 2016) o Smart Specs (Researcher Dr. Stephen Hicks). Nel caso di disfunzioni neurologiche o della parola, l'informazione sottoposta ad elaborazione può essere anche quella prodotta dal paziente (EyeSpeak).

**Terapia di contrasto contro disturbi mentali** Numerosi studi coinvolgono anche la sfera delle neuroscienze: sono stati progettati sistemi per migliorare le abitudini alimentari sulla base di una distorsione della visione dell'aspetto del cibo (University of Tokyo in 2012), esperienze di AR e VR per affrontare fobie di insetti e quadri clinici di schizofrenia, che hanno dimostrato in generale la loro efficacia, ma non sempre nella totalità dei soggetti sotto indagine [7].

**Settore farmacologico e terapia del dolore** Anche l'ambito farmacologico non è escluso da questa ondata d'innovazione: il dispositivo di AR può ricordare al paziente l'assunzione di medicinali facendo comparire un'annotazione all'interno del suo campo visivo ("Augmented Reality Revolutionizing Medicine"); è stato inoltre clinicamente riconosciuto l'effetto lenitivo del dolore che le esperienze immersive di VR a breve termine producono su pazienti reduci da interventi invasivi, traumi o ustioni (riduzione del 30%–50% negli indicatori soggettivi di dolore), senza effetti collaterali significativi riportati ([12], [20], [5]). "Questi studi propongono dati preliminari a supporto dell'uso clinico di tecniche analgesiche multimodali nella terapia del dolore (es. combinazione di terapie farmacologiche e non)", volte quantomeno a rivedere le dosi di oppioidi somministrate nelle fasi pre e post-operatorie, alla luce dei loro pesanti effetti collaterali.

**Terapia riabilitativa motoria e cognitiva** Nella medicina riabilitativa, AR, MR e VR sono già state impiegate sia in un contesto di recupero motorio e di rieducazione posturale, sia a livello psicologico-cognitivo, mediante programmi specifici, volti a recuperare o migliorare i deficit acquisiti e le aree di funzionalità compromesse. Le sessioni hanno l'obiettivo di riprodurre contesti della vita quotidiana, oppure gli scenari di interesse della malattia, per poter intervenire correggendo il deficit contestualmente alla sua manifestazione (es. revisione del gesto atletico nello sportivo, terapie psicologiche di esposizione graduale). La capacità computazionale dei dispositivi AR può essere inoltre sfruttata per elaborare i segnali provenienti da una rete di sensori collocati sul paziente e restituirli in tempo reale sotto forma di indicazioni variabili a seconda della risposta del paziente stesso [6]. Probabilmente il più importante ed approfondito ambito di ricerca di AR e MR nel settore medicale è tuttavia costituito dalla loro integrazione alle tecniche oggi a disposizione per la produzione di immagini mediche a scopo diagnostico, implantologico o chirurgico.

## 1.6 AR e MR in sala operatoria

L'introduzione della tecnica a raggi X in ambito medico ha rivoluzionato la procedura di diagnosi di patologie e traumi i cui sintomi non sono manifestabili solo esternamente, consentendo la visione dell'interno del paziente senza alcun trattamento chirurgico. AR e MR rappresentano l'ultimo stadio, ancora non completato, in questa direzione, ovvero quella di integrare la visione interna ed esterna del paziente, rendendole contemporaneamente disponibili al chirurgo o al radiologo [[4], ...]. Sistemi di questo tipo sono presenti in letteratura già dagli anni '90; fra questi, un prototipo che, sulla base immagini ecografiche del grembo di una donna incinta registrate in tempo reale, fornisce una rappresentazione 3D del feto, visualizzabile mediante HMD (visore immersivo per esperienze di realtà aumentata e mista) [4].

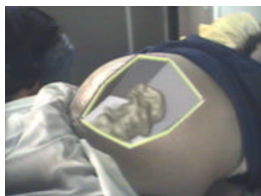


Figura 1.2: Rappresentazione virtuale del feto nel grembo di una donna in gravidanza.



### 1.6.1 AR e lo staff chirurgico

Nonostante negli ultimi decenni le tecniche di acquisizione di immagini diagnostiche siano progredite notevolmente, popolandosi di soluzioni specifiche per le varie situazioni, le modalità di visualizzazione di tali informazioni da parte dello staff medico risultano pressoché inalterate. Durante procedure chirurgiche invasive, il medico deve inferire l'area coinvolta dall'incisione, creandosi una mappa mentale della situazione di intervento osservando separatamente i monitor nei quali sono proiettate immagini (anche più di uno) e il corpo del paziente. L'esecuzione risulta non solo gravata dal reiterato cambio di messa a fuoco fra i 2 (o più) punti distinti da monitorare, ma anche dallo sforzo interpretativo del medico volto ad adattare la prospettiva del referto a quella che si trova di fronte a lui, osservando il paziente col proprio sguardo. Sistemi già operativi mirano a trasferire questo compito ai dispositivi di AR, permettendo al chirurgo di concentrarsi esclusivamente sull'incisione, con benefici in termini di precisione e riduzione dei rischi, di errori e dei tempi di intervento. Sul campo visivo del medico possono essere sovrapposti anche i parametri vitali del paziente, quali la pressione sanguigna e la frequenza cardiaca, in modo da ridurre la strumentazione intraoperatoria da monitorare costantemente. Avanzati sistemi di visualizzazione a supporto della chirurgia sono già disponibili in numerose sale operatorie da anni. Uno di questi è CAMC che, grazie alla combinazione di immagini a raggi x e di un live video stream registrato da una telecamera, ha permesso ai chirurghi della sezione traumatologica del Leiden University Medical Center di Monaco, di avere all'attivo, già nel 2013, più di 40 procedure operatorie per la risoluzione di traumi e procedure ortopediche. La disposizione coassiale della videocamera rispetto al fascio di raggi x permette di ottenere un'alta precisione nell'accostamento delle immagini [14]. Grazie alla sua facilità d'uso, lo strumento è facilmente entrato nelle dinamiche intraoperatorie, senza stravolgerle, richiedendo un addestramento minimale da parte dello staff medico. Nonostante uno dei suoi pregi sia la possibilità di variare l'asse di orientamento dello strumento per facilitare l'acquisizione dei dati, persistono tuttavia problemi di illuminazione. Sono inoltre possibili scenari di disallineamento fra le immagini ottenute staticamente mediante i mezzi radiografici e la reale collocazione della zona anatomica coinvolta durante l'intervento, causati dai leggeri movimenti del paziente rispetto alla posizione di riferimento, che possono causare un'errata interpretazione del flusso aumentato della telecamera. La staticità delle immagini diagnostiche pone limite superabili o mediante il passaggio ad una loro acquisizione dinamica in tempo reale o mediante l'utilizzo di un marker rigidamente fissato al paziente[14]. Il problema del disallineamento rimane ancora uno dei punti deboli su cui lavorare in futuro. Ad oggi, il metodo di risolu-

zione più comunemente accettato nella pratica clinica è quello di stabilizzare fisicamente il paziente mediante lettini appositamente predisposti.

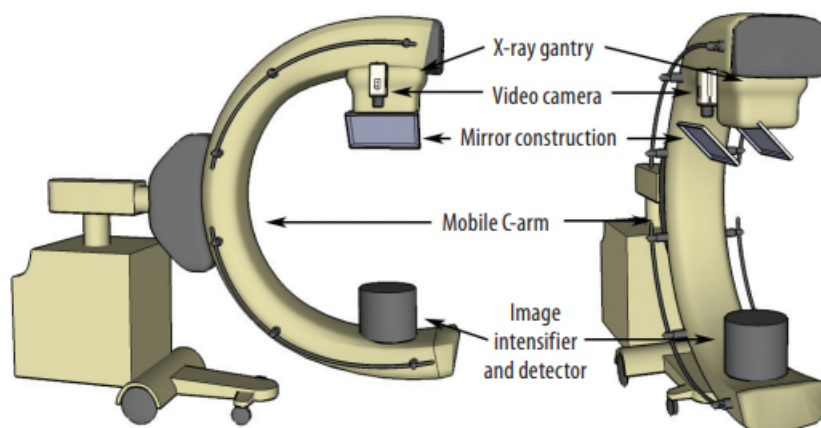


Figura 1.3: Setup di CAMC, tratto da [14].

Un sistema simile a CAMC, sviluppato recentemente da Philips e impiegato per la prima volta in sala operatoria nel 2017 come supporto alla navigazione chirurgica basata su AR, fonda il suo funzionamento sull'acquisizione in tempo reale delle immagini diagnostiche [14]. Fra i benefici segnalati dal gruppo di ricerca coinvolto, anche “la possibilità, grazie alla ricostruzione 3D mediante MR, di eseguire controlli in sala operatoria e, in caso di necessità di ulteriori interventi, di intervenire direttamente, risparmiando al paziente una seconda operazione”.

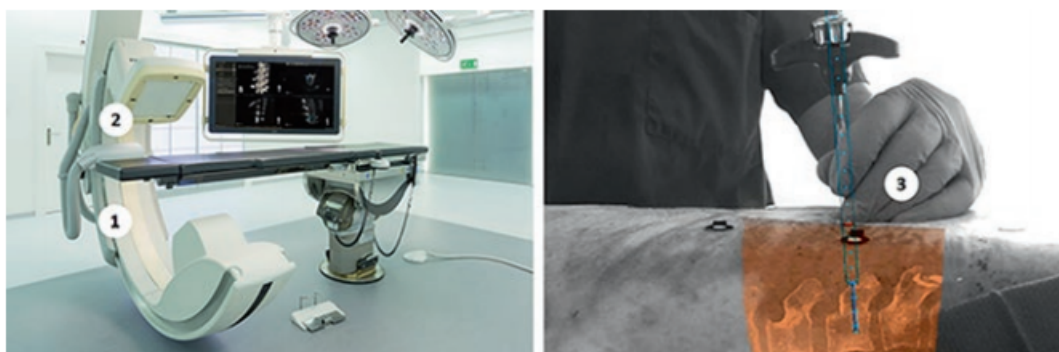


Figura 1.4: Il sistema fabbricato da Philips (1) costituito dalle telecamere ottiche poste sul pannello che rileva i raggi X (2). La figura a destra (3) mostra l'effetto risultante dalla combinazione del flusso 3D aumentato della visione interna del paziente con quella esterna. Tratto da [8]

### 1.6.2 AR come tecnica complementare alla stampa 3D

In ambito diagnostico e di pianificazione chirurgica, la MR si sta affermando sia come tecnica complementare che (in alcuni casi) alternativa alla stampa 3D di modelli di organi e strutture anatomiche, grazie ai più ridotti tempi di produzione (per il processo di stampa 3D sono richieste decine di ore per un organo umano). Studi recenti hanno dimostrato la migliore accoglienza dei modelli 3D (sia virtuali che tattili) rispetto alle tradizionali immagini 2D in scala di grigio nella diagnosi e decisione di intervento in casi di tumori alla prostata e ai reni [23]. Pur non potendo usufruire della percezione tattile offerta da un modello concreto, la possibilità di manipolare ed applicare le trasformazioni geometriche (scala, rotazione) ai modelli digitali sta contribuendo ad introdurre questa tecnologia anche nell'ambito dell'implantologia dentale, della chirurgia protesica e dell'ortopedia, soprattutto come mezzo prototipale nelle fasi intermedie fra la pianificazione e la stampa delle protesi, dove sono richieste grande specificità al soggetto e precisione, al fine di rilevare e correggere direttamente sul modello digitale le imperfezioni riscontrabili prima della stampa e dell'installazione sul corpo del paziente, senza riprodurre ripetutamente modelli errati, riducendo così tempi e costi. L'integrazione delle 2 tecnologie è facilitata dalla comunanza del percorso di produzione dei rispettivi modelli, come confermato dalla larga condivisione dei formati standard oggi utilizzati dai motori grafici impiegati dai dispositivi di AR e MR e dai software di 3D printing. A differenza della stampa 3D, AR può inoltre usufruire di un maggior numero di tecniche di visualizzazione (fra queste il rendering volumetrico), grazie alle quali i dati di partenza subiscono un processo di elaborazione meno invasivo rispetto a quelli impiegati dalla stampa 3D, i quali applicano generalmente tecniche di segmentazione che generano una parziale perdita di informazione e possono introdurre errori nel modello.

### 1.6.3 AR come supporto a procedure mininvasive

Sebbene la pratica della chirurgia correttiva a cielo aperto resti l'unica soluzione in casi specifici e complessi, per via della maggior libertà d'azione e della migliore visibilità delle strutture anatomiche, molte delle procedure chirurgiche precedentemente eseguite in maniera pesantemente invasiva prevedono ora interventi mininvasivi. L'importanza clinica della riduzione delle incisioni è data dalla migliore conservazione dei fasci neuro-vascolari, dal rispetto delle strutture legamentose ed articolari (con conseguente accorciamento dei tempi di recupero) e dalla rinuncia all'anestesia generale, possibile fonte di pesanti complicanze. La ricostruzione digitale e la proiezione sul paziente delle strutture coinvolte può inserirsi in questa tendenza di transizione verso l'adozione

di tecniche mininvasive, servendosi come supporto alla riduzione della visibilità [4].

#### **1.6.4 Feedback tattile**

L'AR non è la sola tecnica innovativa in corso di sperimentazione ed utilizzo in ambito chirurgico. A differenza della chirurgia robotica, che consiste nell'impiego in sala operatoria di mezzi robotici non completamente autonomi, ma capaci di eseguire manovre comandate a distanza dal chirurgo [1], le attuali più diffuse implementazioni di AR e MR mirano a migliorare il grado di precisione e il comfort del medico, ma non mediante la sostituzione della mano umana con un braccio robotico, permettendo in questo modo al chirurgo di conservare il cosiddetto "feedback tattile", la cui assenza è attualmente uno dei limiti maggiori nell'applicazione dei sistemi robotici alla chirurgia; l'integrazione di queste 2 tecnologie rappresenta in prospettiva uno scenario comunque incoraggiante [22].

#### **1.6.5 Formazione del personale sanitario**

Un altro importante filone applicativo in ambito medico è dato dalla formazione del personale medico – sanitario. Il raggiungimento degli standard professionali in termini di efficacia ed efficienza all'interno del complesso dominio medico necessita, secondo le metodiche tradizionali, di tempi di formazione molto lunghi e di costi ingenti [Augmented reality in medical education?]. La possibilità di interagire con un modello digitale consente di simulare procedure chirurgiche e metodiche sanitarie in maniera sicura ed efficace, grazie alla facilità di manipolazione e all'alto grado di fedeltà delle ricostruzioni ([10]). Le potenzialità e l'interesse in questo campo sono testimoniati dalla recente diffusione di sistemi educazionali, app e piattaforme e-learning (Proximie, Miracle). Recenti studi confermano come le tecnologie AR e MR migliorino le capacità ed aiutino a costruire una attitudine positiva nell'ambito delle attività di laboratorio. Le esperienze didattiche possono anche uscire dal contesto di pura simulazione e coinvolgere scenari di intervento veri e propri, consentendo la collaborazione remota di medici e contribuendo in questo modo alla causa della telemedicina (anch'essa in sviluppo negli ultimi anni grazie alla concomitanza di fattori medicali e di mercato [13]). Ne è un esempio uno dei primi interventi svolti col supporto di tecnologie AR e MR dal team chirurgico dell'University of Alabama di Birmingham nel 2013. Mediante una videocamera installata nel visore del chirurgo in sala operatoria, un collega situato in un altro luogo è stato in grado di ottenere e, a sua volta, fornire indicazioni visuali (dispiegate sopra un pad), poi proiettate sul visore da cui il flusso iniziale aveva

avuto origine. Ciò che era visibile al medico in sala era la sovrapposizione della mano del chirurgo remoto sul paziente, indicante le aree di incisione [[16], [18], [11]]. Sempre risalente al 2013 è l'impiego di Google glasses da parte di un'equipe di medici durante un intervento chirurgico; sebbene siano stati utilizzati esclusivamente per registrare le immagini a fini didattici, senza che nessuna informazione fosse sovrapposta sul campo visivo dei medici, questa circostanza ha dimostrato l'accettazione in ambienti operatori di tali dispositivi [16].

## 1.7 Maggiori difficoltà e prospettive future

### 1.7.1 Aspetti non prettamente tecnici

Nonostante gli aspetti prettamente tecnici siano ben lontani dal soddisfare appieno il loro pieno potenziale, le applicazioni nel settore medicale hanno dimostrato che la completa integrazione di AR, MR e VR all'interno delle complesse e consolidate dinamiche intraospedaliere necessita ancora di un radicale cambio culturale, che deve coinvolgere non solo la formazione del personale specializzato, ma anche la nostra vita di tutti i giorni [4]; si pensi in questo senso alla ribalta mediatica del 2016 scaturita dal videogioco Pokémon Go!, la quale ha contribuito in maniera rilevante alla rivelazione dell'AR al grande pubblico. “Se l'adozione di sistemi di formazione basati su AR all'interno dell'educazione e della preparazione medica diverrà consolidata, la prossima generazione di dottori sarà meglio preparata e più ben disposta ad accettarla ed integrarla con la disciplina medica all'interno delle sale operatorie” [14]. All'interno del contesto ospedaliero, altri ostacoli alla sua diffusione sono costituiti dal reperimento e dall'accesso dei dati clinici, oltre che dai costi necessari per avviare la transizione a tale tecnologia. Molto incoraggianti in previsione futura sono però l'interesse nei suoi confronti dei leader tecnologici, il progressivo consolidamento della loro collaborazione con le strutture ospedaliere di tutto il mondo [9] e i crescenti finanziamenti che le alimentano. Buone notizie giungono anche dalla recente approvazione della FDA (Food and Drug Administration, ovvero l'ente governativo statunitense che si occupa della regolamentazione dei prodotti alimentari e farmaceutici) di due sistemi (2018 e 2019) basati su AR in ambito chirurgico.



# Capitolo 2

## Il progetto di tesi

### 2.1 Descrizione del progetto

L'intenzione di questa tesi è quella di presentare un'applicazione che permetta di visualizzare e manipolare tramite realtà aumentata e mista le immagini generate da dispositivi biomedicali di diagnostica. L'obiettivo è quello di fornire uno strumento di supporto alla visualizzazione e alla manipolazione di tali immagini, che si affianchi alle tradizionali modalità di visualizzazione già impiegate nel contesto ospedaliero soprattutto a scopo diagnostico. La tesi va nella direzione di descrivere la pipeline tecnologica tramite la quale effettuare questa transizione e, contemporaneamente, fornire un prototipo che possa raccogliere feedback di valutazione da parte di esperti del settore medicale. Il progetto è nato dalla collaborazione con un team di bioingegneri dell'università di Bologna, sede di Cesena, ed è il risultato di un costante scambio di opinioni volto ad individuare dapprima le funzionalità che potessero essere al futuro utente dell'applicazione e, allo stesso tempo, le strategie di risoluzione da applicare per realizzarle. Il dip. di ingegneria biomedica ci ha anche fornito i dati sui quali testare il prototipo nelle fasi dello sviluppo.

### 2.2 Potenzialità

L'utente destinatario dell'applicazione è potenzialmente chiunque impieghi immagini biomedicali nel suo ambito professionale. Il crescente interesse verso l'adozione di procedure preventive e di tecniche mininvasive in ambito chirurgico è alla base della messa in discussione delle tecniche di visualizzazione attualmente disponibili sul mercato. Come evidenziato nel cap. 1, durante gli ultimi anni, negli ambulatori e nelle sale operatorie di tutto il mondo, l'importante evoluzione delle tecniche di acquisizione delle immagini non è stata

supportata da un analogo progresso nelle modalità di analisi e consultazione delle stesse. Ciò è attribuito prevalentemente a ragioni economiche e alle limitazioni dei dispositivi, che saranno presto superate. Questo progetto si colloca in questo scenario di transizione tecnologica e vuole costituire un collegamento che metta in comunicazione il mondo biomedicale con la MR, attraverso l'integrazione dell'immagine morfologica proveniente dal DICOM e della sua rappresentazione tridimensionale, o segmentata. Il contesto applicativo nel quale questo progetto si inserisce non è comunque limitato al solo utilizzo in un ambito ospedaliero. I professori di ingegneria biomedica hanno riconosciuto l'importanza associata all'integrazione delle 2 modalità di rappresentazione presentate, al fine di stabilire la qualità del processo di segmentazione a partire da dati di risonanza magnetica e CT, di cui sono stati essi stessi artefici. L'utilità deriva dalla possibilità di confrontare i modelli con il dato anatomico estratto dal DICOM, il quale, non essendo stato sottoposto ad alcuna elaborazione, fornisce una solida base di raffronto per individuare eventuali errori del processo. Questo apre scenari di applicazione alle scienze biomediche, soprattutto relative alla produzione e alla validazione di modelli tridimensionali ottenuti a partire da immagini diagnostiche (anatomia computazionale), che può inserirsi e trasformare le loro dinamiche di sviluppo, sostituendo o affiancando, almeno nelle fasi prototipali, l'analisi mediante AR alla stampa 3D, potenzialmente riducendo i costi di produzione.

## 2.3 Motivazioni

Sebbene la realtà aumentata stia divenendo sempre più popolare, soprattutto in ambito healthcare, allo stato attuale le effettive implementazioni di una pipeline di conversione dal DICOM all'AR/MR non sono numerose, e comunque circoscritte a software proprietari. Personalmente, non ho trovato in letteratura altri esempi in questa direzione a cui ispirarmi. L'esistenza di DICOM, che è l'attuale standard internazionale de-facto per l'archiviazione e la trasmissione di tali immagini, ha permesso di dare una connotazione generale al progetto svolto, che possa poggiare su robusti meccanismi di interscambio già predisposti e messi in campo da anni nel settore medicale. Nonostante esista da molto tempo, l'ampia diffusione raggiunta all'interno degli ambienti ospedalieri garantisce che DICOM continuerà ad esistere ancora a lungo. Sebbene il progetto di tesi sia specificatamente rivolto alle immagini medicali, l'ottica complessiva nella quale si inserisce è quella di costruire un workbench virtuale, tramite il quale il medico, o un team di medici, possa interagire con la controparte virtuale della propria strumentazione di lavoro, che si sostituisce



a quella tradizionale, eliminando la necessità di usufruire del supporto fisico (es. scrivania, documenti cartacei, monitor).

## 2.4 Analisi dei requisiti

Come evidenziato nel dettaglio nel cap. 2, generalmente il formato DICOM non codifica direttamente modelli tridimensionali, quanto piuttosto insiemi di frame, ovvero rappresentazioni bidimensionali (slice) di un volume anatomico, acquisite ad istanti di tempo diversi e raffiguranti sezioni diverse, la cui interpretazione permette di ricostruirlo. La possibilità di visualizzare il modello in 3 dimensioni in AR è data da un processo preliminare di elaborazione del dato DICOM, chiamato segmentazione.

### 2.4.1 Input del sistema

Nell'ambito di trattare il procedimento necessario alla conversione del dato DICOM originale in MR, il progetto si pone l'obiettivo di visualizzare e manipolare separatamente i modelli 3D ottenuti a partire dallo stesso dato DICOM mediante 2 diversi processi di segmentazione:

- Segmentazione eseguita dai nostri collaboratori di ingegneria biomedica utilizzando algoritmi da loro stessi sviluppati, nel contesto dell'automatizzazione del processo che ha portato alla sua produzione.
- Segmentazione semi-automatica ottenuta attraverso l'utilizzo di software open-source.

Pertanto, nel primo caso l'input al sistema è costituito da due elementi:

- Informazione in formato DICOM (insieme di slices codificate secondo le convenzioni esposte nel capitolo 2). Rappresenta l'informazione originale senza alcuna elaborazione. Oltre ai dati dei pixel di ciascuna immagine, contiene un corposo elenco di metadati atti a ricostruire, fra gli altri, l'orientazione delle slice nello spazio e i pixel.
- File contenente mesh poligonale risultante dal processo di segmentazione applicato al medesimo dato DICOM. Contiene informazioni relative alle facce e alle coordinate dei vertici dai quali sono composte. Rappresenta una discretizzazione dell'informazione DICOM originale, che comporta necessariamente una perdita di informazione.

Nel secondo caso, invece:

- non si assume di disporre già del risultato ottenuto dalla precedente fase di segmentazione e dunque si esaminano tutti i passi necessari alla visualizzazione del modello a partire dal DICOM originale.

Il capitolo 3 proporrà una disamina del formato DICOM finalizzata a comprendere più nel dettaglio il punto di partenza del progetto.

## 2.4.2 Requisiti funzionali

Il prototipo rientra nella categoria di mixed reality per come essa è stata definita nel capitolo 1, dal momento che l'aspetto interattivo fra l'utente e gli oggetti virtuali è significativo. Nel seguito sono elencate le specifiche risultanti dal confronto con il team di ingegneri e professori biomedici con cui è stato portato avanti il progetto.

- L'utente deve essere in grado di visualizzare la ricostruzione 3D della struttura anatomica risultante dalla segmentazione, ed in grado di manipolarla, ovvero ruotarla, zoomarla e spostarla all'interno del mondo aumentato, così come poterci ruotare attorno per poterne evidenziare caratteristiche strutturali a scopi diagnostici e di valutazione della qualità del processo che ha portato alla sua produzione.
- Il prototipo deve poter estrarre le rappresentazioni anatomiche bidimensionali direttamente contenute nel dataset DICOM originale, ovvero quelle corrispondenti ai cosiddetti "piani acquisiti", interpretando i dati dei pixel in esso contenuti e le informazioni DICOM relative alle loro disposizioni nello spazio.
- È importante che l'utente possa scegliere dove posizionare gli oggetti virtuali, poiché l'utilità dell'applicazione deriva anche dalla eliminazione del vincolo fisico fra l'oggetto da visualizzare e il mezzo di visualizzazione (si pensi ai monitor che costringono il medico a stazionare di fronte ad esso). L'applicazione dovrà riconoscere dinamicamente il piano deputato ad ospitare gli oggetti virtuali.
- Relativamente ai distretti anatomici in cui l'intensità del valore di contrasto ottenuto ha una rilevanza diagnostica (ad es. il distretto cardiaco, nel quale esiste una correlazione fra esso e la presenza di fibrosi), il modello 3D ottenuto da segmentazione deve fornire in MR delle indicazioni visuali che vi associano un valore nel modello RGB.

### 2.4.3 Requisiti non funzionali

È stata riconosciuta una serie di requisiti non direttamente connessi alle funzionalità del sistema, ma comunque influente sulla qualità dell'esperienza di MR. Fra questi:

- Permettere un rapido riconoscimento dei piani sui quali disporre gli oggetti tridimensionali.
- Impedire la perdita di registrazione degli oggetti sul piano, in modo che, ritornando sul punto in cui sono stati istanziati dopo aver cambiato momentaneamente visuale, l'utente possa ritrovarli dove li aveva precedentemente posizionati.

Anche per motivi di tempo, una serie di aspetti funzionali non è stata considerata e sarà la base di partenza per sviluppi futuri, a tal proposito si consulti il capitolo 6.

## 2.5 Problematiche principali

Nonostante queste saranno meglio descritte nel capitolo 4, che presenterà le strategie per la loro risoluzione, in questa fase è già possibile riconoscere le problematiche principali nel processo di trasformazione che porterà ad ottenere una rappresentazione virtuale del dato ottenuto da segmentazione e del dato morfologico.

- Il dato risultante dalla segmentazione non è codificato in un formato standard di memorizzazione di modelli 3D, bensì in un formato binario proprietario, il `.mat`, che è predisposto all'archiviazione dati, funzioni ed altri costrutti del linguaggio, ma che non dispone del supporto nativo per vertici e facce, cioè le componenti che costituiscono un modello segmentato. Pertanto, non potrà essere utilizzato direttamente dal game engine responsabile del rendering dell'esperienza aumentata e dovrà passare per un processo di conversione verso uno dei formati compatibili con esso.
- Un file DICOM non è assimilabile ad un qualsiasi formato di memorizzazione di un'immagine, nel senso che non definisce uno specifico standard di codifica che permetta di uniformare la gestione dei pixel dell'immagine. Sarà necessario interpretare i campi contenuti nel file al fine di mettere in piedi strategie diverse di risoluzione per poter gestire diversi criteri di memorizzazione dell'immagine.



# Capitolo 3

## DICOM

### 3.1 Introduzione e scopo

DICOM (Digital Imaging and COmmunications in Medicine, immagini e comunicazione digitali in medicina) è uno standard internazionale per la gestione e comunicazione di immagini mediche e di informazioni ad esse correlate [2].

Rappresenta lo standard di riferimento in ambito di immagini medicali ed è stato sviluppato per agevolare l'interoperabilità tra apparecchiature e software di diversi produttori. Definisce i criteri di comunicazione, visualizzazione, stampa ed archiviazione di informazioni biomedicali. È uno standard aperto, largamente diffuso ed in continua evoluzione, sviluppato da numero organizzazioni internazionali quali l'American College of Radiology (ACR) e la National Electrical Manufacturers Association (NEMA), CEN TC251 in Europa e il JIRA in Giappone, IEEE, HL7 ed ANSI, ma che ha visto anche la partecipazione di numerose industrie del settore.

La sua introduzione ha dato forma al panorama della medicina contemporanea, fornendo contemporaneamente:

- Uno standard universale per la medicina digitale
- Il supporto ad immagini di maggiore qualità
- Pieno supporto per ampio spettro di parametri di acquisizione dell'immagine e di tipi di informazioni differenti
- Codifica completa dei dati medicali del paziente
- Chiarezza nella descrizione dei dispositivi di acquisizione digitale e del loro funzionamento [17].

Nella trattazione che segue sono elencati le caratteristiche del modello DICOM che hanno avuto un'influenza nella realizzazione del progetto di tesi.

## 3.2 Ambito di applicazione

Sebbene spesso sia stato associato esclusivamente ad un formato di memorizzazione di file o immagini, DICOM costituisce sia uno standard di comunicazione che di archiviazione di documenti, includendo le specifiche relative a:

- un insieme di protocolli di rete costruiti sopra a TCP/IP;
- la struttura del formato dei file e della directory dei documenti medicali oggetto della comunicazione.

Attualmente DICOM è il protocollo standard adottato dalla larga maggioranza di case costruttrici di apparecchiature diagnostiche di visualizzazione e di comunicazione di ultima generazione. La sua diffusione non coinvolge solamente le strutture ospedaliere, ma si sta affermando anche in realtà più piccole, quali studi dentistici e ambulatori privati.

L'importanza di DICOM è testimoniata dalle presenza di numerosi working group al suo interno e alla sua pervasività nei settori della medicina dove l'impiego di immagini è prevalente. Fra questi: la radiologia, la medicina nucleare, l'oncologia, la radioterapia, la cardiologia, neurologia, ortopedica, ostetricia, ginecologia oftalmologia, dermatologia, chirurgia, medicina veterinaria e tanti altri [1].

## 3.3 Breve storia

L'origine del progetto risale al 1983, quando, The American College of Radiology (ACR) e National Electrical Manufacturers Association (NEMA), decisero di riunirsi in un comitato in risposta all'esigenza di rendere indipendenti le immagini medicali dagli specifici dispositivi di acquisizione e di facilitare la digitalizzazione e la comunicazione delle immagini. Già nel 1985 venne prodotto un primo documento: ACR/NEMA 300; nonostante le molte incongruenze, fra i punti principali vi era già l'idea di scomporre l'informazione contenuta nei documenti in piccole componenti identificate mediante tag, in maniera simile all'approccio impiegato dai linguaggi di markup (HTML e XML) [17]. La seconda versione del 1988 ebbe migliore accoglienza fra i produttori, tanto che ancora oggi è possibile incontrare dispositivi aderenti a tale standard ancora

in funzione. Il documento conteneva la definizione del formato dei file contenenti le immagini, ma il protocollo di trasmissione punto a punto impiegato costituiva ancora un elemento limitante, soprattutto di fronte alla crescente digitalizzazione dei dispositivi in commercio, che richiedevano un più solido modello di informazione e scambio dei dati. L'ultima versione, rilasciata nel 1993 con il nome di "DICOM 3.0" prevedeva l'introduzione di servizi aggiuntivi e una sostanziale rivisitazione delle specifiche inerenti i protocolli di comunicazione e riuscì infine ad ottenere un livello di accettazione pressoché universale fra i produttori di dispositivi di acquisizione di immagini e le organizzazioni ITC, soprattutto grazie alla scelta di TCP/IP come protocollo a livello di rete. Il contributo del comitato ACR e NEMA fu determinante anche nello sviluppo e nell'espansione di Picture Archiving and Communication Systems (PACS), ovvero infrastrutture hardware e software dedicate all'archiviazione, trasmissione, visualizzazione e stampa delle immagini diagnostiche digitali. L'intento alla base di questa iniziativa era quello di agevolare l'interfacciamento fra diversi sistemi informativi dell'ospedale, nel contesto dei quali DICOM svolge una funzione di intermediario, definendone, fra gli altri, i criteri di comunicazione.

## 3.4 Modello di informazione

DICOM modella le informazioni che gestisce e i servizi che predispone mediante un'astrazione ispirata al paradigma object-oriented, spesso descritta nella documentazione dello standard mediante il formalismo ER. L'utilizzo di un modello di informazioni rappresenta una delle novità di DICOM rispetto ad altri standard di comunicazione di dati medici e informazioni correlate (es. HL7).

### 3.4.1 Information Object Definition

Una Information Object Definition (IOD) specifica informazioni relative ad oggetti del mondo reale. Una IOD non rappresenta una specifica istanza di un oggetto, quanto invece una classe di oggetti che condividono le stesse proprietà. Le IOD si suddividono in normalizzate e composte; mentre le prime contengono esclusivamente gli attributi strettamente inerenti all'entità che modellano (è il caso di uno studio su un paziente, del quale non si memorizza il nome in una classe normalizzata sullo studio, perché non attinente allo studio in sé), le classi composte, invece, possono contenere anche informazioni correlate (si pensi, ad esempio, ad un'immagine assieme alla quale viene conservato anche il nome del paziente), permettendo di accorpate concetti che necessitano di un più alto grado di associazione [2][ ps3.3]. Una IOD si compone di moduli,

solo alcuni dei quali obbligatori, ossia quelli corrispondenti alle 4 entità di base (Image Module, General Series Module, General Study Module, Patient Module). Pertanto, questi moduli sono presenti in numerose IOD. Altri, invece, appaiono solo in specifiche IOD: l'MR Image Module, ad esempio, è parte solo dell'MR Image IOD.

DICOM è uno standard contraddistinto dall'opzionalità nell'utilizzo di determinati componenti e da un buon grado di personalizzazione: è infatti consentita l'aggiunta di nuovi elementi (private element), anche se il loro utilizzo è ovviamente subordinato a quello dei tag già esistenti. Tali caratteristiche sono state spesso fonte di ambiguità nell'implementazione di sistemi basati su questo standard e rappresentano uno dei suoi punti deboli.

### 3.4.2 SOP Class e Service Class

Oltre alla definizione degli oggetti e dei loro attributi, DICOM definisce quali operazioni possono essere eseguite e su quali oggetti. Tali operazioni sono chiamate DIMSE service (DICOM Message Service). La combinazione di un oggetto e delle corrispettive operazioni (DIMSE) prende il nome di SOP (Service Object Pair). L'insieme delle SOP relative ad un unico oggetto forma una SOP Class, la cui definizione contiene i criteri che potrebbe limitare l'utilizzo dei servizi o degli attributi associati alla IOD.

Una Service Class definisce un gruppo di una o più SOP Class correlate ad una specifica operazione (es. l'invio di immagini) che viene eseguita tra applicazioni in fase di comunicazione. Due SOP Classes definite da una singola definizione di Service Class possono differire per la IOD. Due IOD differenti non potranno, comunque, far parte della stessa SOP Class.

### 3.4.3 Information Entity

Una specifica entità del mondo reale (information Entity) è costituita da un'IOD composita, ovvero da aggregazione di IOD. C'è una corrispondenza 1 a 1 fra Information Entities e le entità del DICOM Application Model. Nonostante siano numerose (fra le quali Visit, Equipment, Clinical Trial, Procedure), le entità base su cui si fonda il modello DICOM sono essenzialmente 4:

- PAZIENTE: include le informazioni anagrafiche della persona sottoposta all'indagine, utili ad identificarla;
- STUDIO: contiene le caratteristiche dell'indagine diagnostica e diverse modalità.



- SERIE: per ogni modalità diagnostica è descritta ciascuna collezione di immagini. Ogni serie è associata ad esattamente uno studio.
- IMMAGINE/i: attributi dei pixel che compongono l'immagine (dimensione matrice, profondità pixel, rappresentazione del pixel e interpretazione fotometrica etc.).

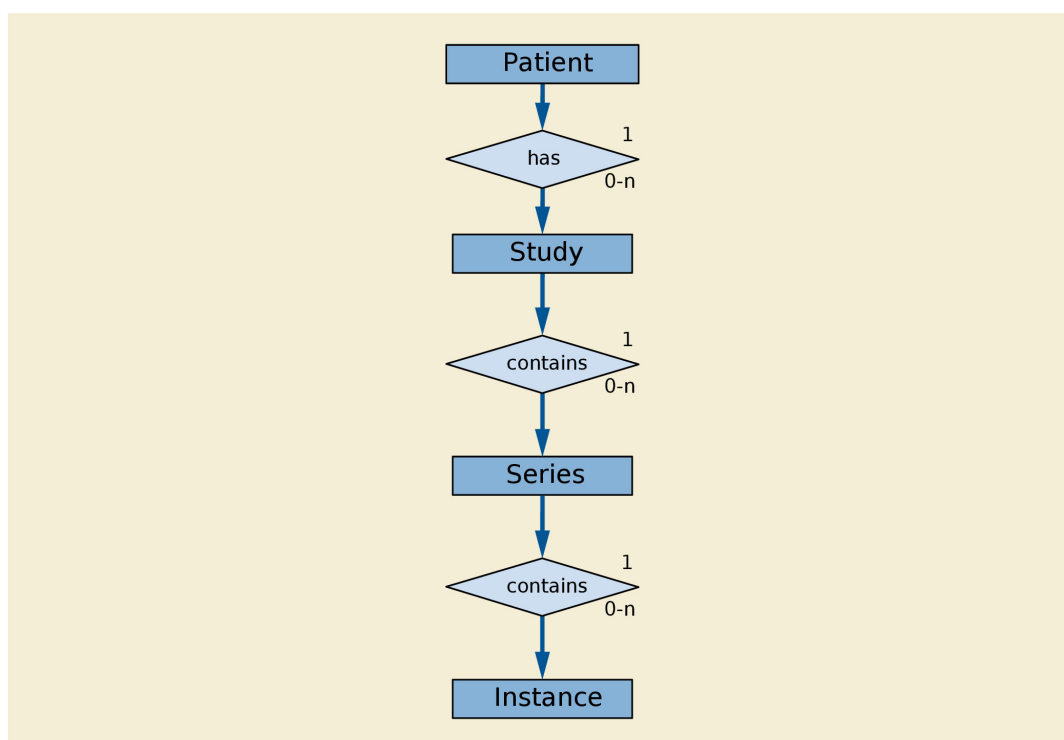


Figura 3.1: Rappresentazione mediante il formalismo ER delle 4 entità base per il modello di informazione DICOM

#### 3.4.4 UID

Per favorire il loro riconoscimento globale, ciascuna IOD viene identificata da un UID (Unique Identifier), ovvero un codice univoco, mediante il quale i sistemi di archiviazione possano riferirsi ad essa. Gli UID relativi alle IOD sono documentati nel capitolo 6 dello standard. Anche gli oggetti DICOM, ossia le istanze di SOP Class, contengono il proprio UID. Lo schema di identificazione è sottoposto a precise convenzioni, regolamentate a loro volta da uno standard distinto (ISO 8824 standard). Ogni UID si compone di 2 parti:

$$\text{UID} = \langle \text{org root} \rangle . \langle \text{suffix} \rangle$$

La porzione <org root> dell' UID identifica univocamente un'organizzazione, (ad esempio una organizzazione di ricerca, manifatturiera, NEMA, etc.), e si compone di una serie di cifre come definite dallo standard ISO 8824. La componente <suffix> è costituita a sua volta da una stringa numerica e dovrebbe essere unica all'interno dell'ambito della <org root>. Questo implica che l'organizzazione identificata dal <org root> sia responsabile di garantire l'unicità di <suffix> al suo interno attraverso l'adozione di politiche di registrazione. Diversamente da <root> che potrebbe essere condiviso dagli UID di un'organizzazione, il componente <suffix> deve differire fra UID che identificano oggetti diversi. Ogni entità del DICOM Data Model possiede una propria UID ad eccezione del paziente, che è identificato dalla combinazione del suo nome e ID. Gli UID della Series Instance e della SOP Instance sono sempre generati dal dispositivo di acquisizione. Gli archivi DICOM (PACS) dovrebbero fare uso degli UID per indicizzare gli oggetti DICOM, in modo che le applicazioni interessate a tali oggetti possano usufruire di una più rapida risposta alla loro richiesta (query).

## 3.5 Formato dati

Il formato è estremamente dettagliato, con un documento di specifiche costituito da più di 2500 pagine. I file vengono strutturati all'interno di una cartella, solitamente l'estensione impiegata è .dcm, ma anche altri sono possibili (.dc3, .dic, oppure nessuna).

### 3.5.1 Data structure

L'unità di informazione nel modello informativo di DICOM è costituita dai data element, che rappresentano gli attributi di una IOD. Ciascun data element si compone delle seguenti componenti principali:

- tag che identifica l'attributo, solitamente espresso in the format (XX-XX,XXXX) tramite notazione esadecimale, che può essere ulteriormente suddiviso in DICOM Group e DICOM Element Number
- DICOM Value Representation (VR) che descrive il tipo di dato e il valore dell'attributo. The VR è rappresentato da un codice composto da 2 caratteri. Fra gli esempi di Value representation si trovano: UI per "Unique identifier", US for "Unsigned Short", CS for "Coded String" and OB for "Other Byte".

- value length: è il numero di byte necessari a rappresentare il successivo campo valore. Dal momento che DICOM è un protocollo binario (diversamente da altri, quali HTML e XML, che sono testuali) gli elementi richiedono di specificare una lunghezza, che è sempre un valore pari, anche nel caso in cui il campo valore sia costituito da un singolo carattere. Il campo valore dei tipi stringa (come CS e UI) è colmato con uno spazio (un esempio è il data element Patient Sex (0010,0040), il cui valore sarebbe costituito da un singolo carattere), mentre per quelli binari, con il valore NULL.
- un campo valore: contiene il valore del data element.

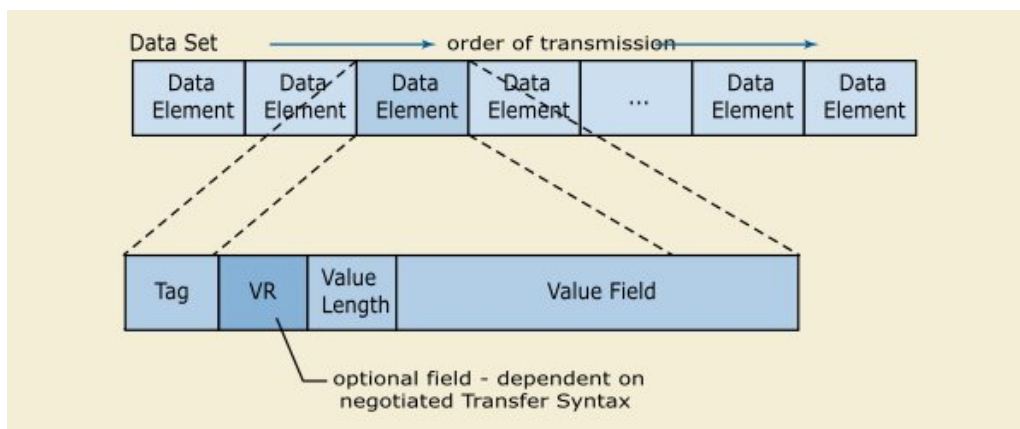


Figura 3.2: DICOM Data Set e Data Element Structures

Tag fra loro correlati condividono lo stesso Group number. L'elenco di tutti i tag definiti da DICOM costituisce il data Dictionary, ed è consultabile nella parte 6 (PS3.6) dello standard.

Un Data Set è una collezione di Data Elements memorizzati in un file DICOM. I data element appartenenti ad un determinato data set sono ordinati in base al group number; nell'ambito di un determinato gruppo, gli elementi hanno ordinamento crescente per element number e non possono avere più di un'occorrenza all'interno del data set. Il data set contenuto in ciascun file dovrebbe essere relativo ad una singola SOP Instance e riferito ad una singola SOP Class, anche se tale convenzione non è sempre rispettata.

### 3.5.2 Le innovazioni introdotte

Fra le maggiori novità introdotte da DICOM vi è la non divisibilità tra le immagini e i dati descrittivi il procedimento attraverso il quale sono state

ottenute. Questa caratteristica è alla base della dell'interoperabilità che lo standard garantisce. In questo senso, il modello di informazione di DICOM è predisposto alla memorizzazione di immagini prodotte da una grande varietà di dispositivi di acquisizione [17], fra le quali: CT (computed tomography), MRI (magnetic resonance imaging), ultrasound, X-ray, fluoroscopy, angiography, mammography, breast tomosynthesis, PET (positron emission tomography), SPECT (single photon emission computed tomography), Endoscopy, microscopy, nd whole slide imaging, OCT (optical coherence tomography). Da questo punto di vista il file DICOM dunque non può essere considerato alla stregua di un qualsiasi altro formato di memorizzazione (PNG, JIFF etc.), sia perché contiene informazioni non solo relative alla immagine rappresentata sia perché lo standard non contiene la definizione di un nuovo criterio di compressione. Spesso le immagini sono memorizzate in forma non compressa, o utilizzano altri meccanismi di codifica già noti.

### 3.5.3 File structure

Il dispositivo per l'acquisizione di immagini diagnostiche produce una serie di rappresentazioni bidimensionali, corrispondenti alle sezioni trasversali dell'area anatomica interessata dalla scansione. Ciascun file DICOM si compone non solo di una o più di queste immagini, ma anche di una serie di informazioni (dati anagrafici del paziente e del medico, il tipo di scansione, posizione e dimensione dell'immagine etc.) che sono strutturati attraverso l'astrazione descritta nelle sezioni precedenti. Prima dei dati veri e propri, il formato prevede la presenza di un'intestazione, costituita da un preambolo da 128 byte seguito da un prefisso da 4 byte, il cui scopo è quello di facilitare l'accesso alle immagini e agli altri dati contenuti del file DICOM anche ad applicazioni multimediali, agevolando la compatibilità con un gran numero di formati immagine utilizzati comunemente al giorno d'oggi.

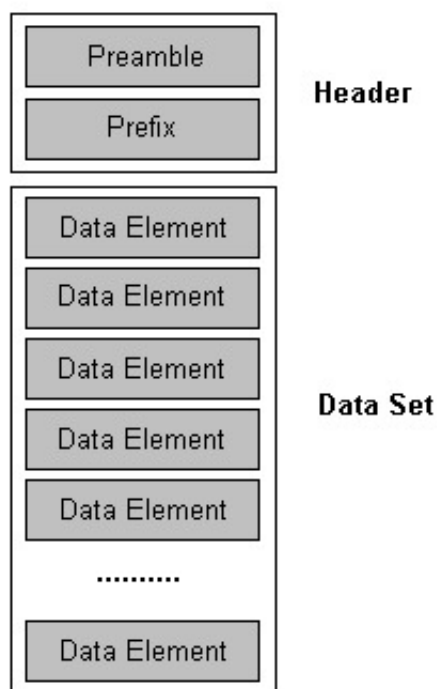


Figura 3.3: Rappresentazione semplificativa della struttura del formato file DICOM

### 3.5.4 Transfer syntax

La Transfer Syntax è l'insieme di criteri di codifica volti a rappresentare in maniera non ambigua i concetti definiti tramite un'astrazione di più alto livello, ovvero quella del DICOM Application Model. Essa definisce le modalità in cui gli oggetti DICOM vengono serializzati nel formato file e coinvolge il processo di incapsulamento dei dati necessario al loro trasferimento fra applicazione e contesti di utilizzo diversi. La transfer syntax impiegata in un file DICOM viene codificata anch'essa al suo interno, possiede un proprio UID, ed è reperibile dal file mediante le stesse convenzioni degli altri elementi. La transfer syntax specifica 3 criteri che influenzano la maniera in cui interpretare gli oggetti DICOM:

- Value representation: in riferimento alla VR esposta precedentemente, viene specificato se la lunghezza (value length) debba essere espressamente indicata (explicit VR) o meno (implicit VR).

- Byte endiannes: specifica l'ordine di trasferimento dei byte nel caso di tipi di dato multi-byte.
- Native / encapsulated pixel data: contiene l'indicazione sulla eventuale compressione utilizzata nei pixel dei dati delle immagini e l'algoritmo di codifica impiegato

Se non specificata, la transfer syntax utilizzata di default da DICOM è la DICOM Implicit VR Little Endian Transfer Syntax.

## 3.6 DICOM e l'immagine

### 3.6.1 Rappresentazione dell'immagine

Per ottenere una corretta interpretazione delle immagini medicali contenute nel file DICOM è necessario esaminare l'insieme degli attributi delle Image Information Entities. I tag in questione ricadono, per la maggior parte, all'interno dell'Image module, e sono quasi tutti obbligatori, ovvero la loro mancanza determina la non validità del file DICOM. Di seguito sono elencati i data element, associati ai rispettivi tag.

- Rows (0028,0010) e Colume (0028, 0011): rappresentano rispettivamente il numero di pixel per riga e per colonna di ciascun frame. Per frame si intende una singola immagine.
- Photometric interpretation (0028,0004): Il valore di questo attributo specifica la interpretazione dei dati relativi ai pixel dell'immagine, identificando lo spazio di colore utilizzato. Esso è influenzato dalla transfer syntax adottata, nello specifico dalla presenza o meno di una codifica di compressione dei dati. Inoltre, il contenuto del tag deve essere compatibile con quello di Samples per pixel(0028,0002). I valori più comuni sono:
  - MONOCHROME1: indica che a valori di pixel crescenti corrispondono valori di grigio più scuri.
  - MONOCHROME2: indica che a valori di pixel crescenti corrispondono valori di grigio più chiari.
  - RGB: in questo caso i dati contenuti nel tag pixel data rappresentano i colori dell'immagine attraverso i 3 canali rosso, verde e blu. Questo valore può essere scelto solo se l'attributo Planar Configuration vale 0 oppure 1 e Sample Per Pixel è 3. Può essere utilizzato in concomitanza con pixel data codificati in maniera nativa o compressa.

Immagini derivanti da MRI o CT solitamente sono codificate in scala di grigio, mentre la codifica RGB è tipica di immagini ottenute da acquisizioni ad ultrasuoni.

- **Samples per pixel (0028,0002):** definisce il numero di canali per ciascun colore nello spazio di colori adottato. Può avere un valore da 1 a 3. Per immagini in scala i grigi (monochrome) e immagini da palette di colori, questo attributo contiene il valore 1, per RGB e altri modelli di colore: 3.
- **Pixel data (7FE0, 0010):** contiene lo stream di byte raw dell'immagine, codificato secondo le disposizioni specificate dalla transfer syntax. L'ordine dei pixel memorizzati per ogni piano (o canale) dell'immagine procede da sinistra verso destra, dall'alto al basso, cioè il pixel in alto a sinistra (identificato 1, 1) è codificato per primo seguito rispettivamente dai rimanenti pixel della prima riga, dal primo pixel della 2a riga(2, 1) e i rimanenti.
- **Multi-frame:** Nonostante ciascun oggetto DICOM possa contenere un solo attributo pixel data, ciò non impedisce che il numero di slice ospitate nello stesso file sia maggiore di 1. Questo attributo indica quanti sono i frame effettivamente memorizzati nello stream di byte contenuto in pixel data. Un'immagine multi-frame è definita come un'immagine i cui pixel consistono di una insieme sequenziale di singoli image frame. In questo modo, DICOM può ospitare al proprio interno anche immagini single-frame multiple relative ad uno stesso studio, oppure cine-loops (sequenze dinamiche acquisite in istanti temporali diversi spesso utilizzate in esami ad ultrasuoni). Nel primo caso, questo accorgimento permette di evitare di ripetere le stesse informazioni più volte (quali informazioni sul paziente, sulle caratteristiche delle immagini etc.), dato che esse vengono incluse nel file DICOM anziché comparire in tanti file diversi. Ciò comporta anche delle restrizioni sulle caratteristiche delle immagini (ad esempio dimensioni e orientazione del frame nello spazio), dal momento che esse devono essere necessariamente le stesse fra i diversi frame, dato che i rispettivi tag sono condivisi. Il supporto a questo tipo di archiviazione è stato inferiore rispetto alla memorizzazione single-frame, anche se negli ultimi anni la situazione è migliorata in tal senso.
- **Planar configuration:** contiene informazioni relative alla disposizione dei canali (o piani) rispettivamente al buffer dei pixel, ovvero se essi sono raggruppati a seconda della loro posizione o del canale. In realtà, la seconda circostanza è infrequente ed associata a specifici algoritmi di

compressione (RTE). La sua presenza è indispensabile solamente quando Sample Per Pixel <sup>1</sup> contiene un valore maggiore di 1.

- Bits Allocated, Bits stored e High Bit: rappresentano rispettivamente:
  - Il numero di bit allocati per ogni pixel sample. Ogni pixel sample dovrebbe contenere lo stesso valore. Il valore può essere 1 oppure un multiplo di 8.
  - Il numero di bits allocati (bits allocated) effettivamente utilizzati.
  - Il most-significant bit (bit più significativo), ovvero la posizione della cifra binaria, all'interno del pixel sample, che ha il valore più grande. High bit non può dunque essere superiore al valore di bits allocated -1.

### 3.6.2 Greyscale rendering pipeline

Fatta eccezione per quelli opzionali, sono stati presi in considerazione tutti gli elementi dell'Image module indispensabili alla corretta scansione dei dati sulle immagini, che le specifiche DICOM definiscono in modo dettagliato. Tuttavia, lo standard non contiene precisi criteri su come interpretare i parametri contenuti in un'immagine DICOM relativamente alla sua presentazione. A questo proposito, DICOM prevede la possibilità di sottoporre i dati sui pixel estratti dal file ad un ulteriore processo di elaborazione, costituito da 3 step, i cui parametri e l'effettiva applicazione risultano specificati da appositi elementi. Questa "rendering pipeline" prevede diversi step di elaborazione, costituiti dall'applicazione di funzioni lineari e non lineari mediante look up table, ed è destinata alle immagini che presentino un valore di Photometric interpretation corrispondente a MONOCHROME1 o MONOCHROME2. A tal proposito è necessario estendere la trattazione includendo parte dei data element del Modality LUT module, VOI LUT Module e Presentation LUT Module. Affinché il processo vada a buon fine, l'applicazione di tali trasformazioni deve rispettare l'ordine attraverso il quale saranno presentate. Ciascuna operazione è eseguita sui valori dei pixel ottenuti dal pixel data element in forma non compressa, ovvero ottenuti in seguito all'eventuale processo di decodifica. Non è necessario che un set di immagini sia sottoposto alla pipeline nella sua interezza: gli step da svolgere sono determinati in base, fra le altre cose, alla tipologia di acquisizione dell'immagine.

---

<sup>1</sup>In computer graphics, un "pixel sample è costituito dall'intersezione fra un canale e un pixel, ovvero dai bit del pixel associati alla codifica di un canale



### Mask transform

Questa trasformazione può essere applicata nel caso di immagini multi-frame e coinvolge la sottrazione di alcuni specifici frame a quello corrente. La trasformazione può comprendere preliminari operazioni di averaging e di subpixel-shifting relativamente ai frame coinvolti. La trasformazione impiega attributi contenuti nel Mask Module e può essere applicata su qualsiasi Storage SOP Instance che contenga immagini multi-frame, a patto che siano codificate in valori di grigio.

### Modality transform, rescale slop e rescale intercept

Questa trasformazione ha il compito di convertire i pixel da una rappresentazione dipendente dal dispositivo con la quale è stata acquisita l'immagine ad una che sia significativa per la modalità stessa e sia al contempo indipendente dal produttore del device. I pixel vengono mappati verso uno spazio di valori che ha un significato dal punto di vista fisico (quali ad esempio la densità ottica per CR, le unità di Hounsfield per CT, Becquerels/milliliter per PET). Queste unità di misurazione non sono necessariamente le più convenienti o efficienti dal punto di vista dell'archiviazione perché non è questo il loro scopo. La Modality transform è identificata dalla presenza dell'attributo Modality LUT Sequence element (0028,3000) contenente direttamente la/e LUT. Nel caso questo tag sia assente, il file deve contenere i 2 parametri che servono ad ottenerla, specificati dai tag RescaleSlope (0028,1053) e RescaleIntercept(0028,1052). Essi esprimono la dipendenza lineare che intercorre fra il valore memorizzato in ciascun pixel contenuto nel pixel data tag e quello di output, secondo la relazione:

$$y = (m * x) + b$$

dove:

- x è il valore del pixel
- m è definito dall'attributo Rescale Slope (0028,1053)
- b corrisponde all'attributo Rescale Intercept (0028,1052)

### Value of interest (VOI) transform

La Value of Interest LUT (VOI LUT) specifica quale range di intensità di pixel dovrebbe essere mostrato a video, convertendo i valori dei pixel in un intervallo di interesse per l'applicazione o per l'utente. Le sue specifiche sono descritte nel VOI LUT Module. Fra gli attributi del modulo, il tag VOI LUT

Function (0028,1056) permette di specificare una relazione potenzialmente non lineare da applicare al risultato della Modality Transform e come input alla Presentation transform. I possibili valori che può assumere sono: LINEAR, LINEAR\_EXACT e SIGMOID. Se l'elemento è assente o ha valore di LINEAR, la conversione da applicare ai pixel è lineare ed è specificata dai 2 tag Window Center (0028,1050) e Window Width (0028,1051), i quali contengono rispettivamente il valore di input assunto come centro della finestra e la larghezza della stessa: tali attributi non dovrebbero essere presenti nel caso la LUT sia specificata dall'elemento VOI LUT sequence (0028,3010). L'incidenza svolta da Window width e center sulla conversione dei pixel è espressa attraverso il seguente pseudo-codice, dove  $x$  corrisponde al valore di input,  $y$  a quello di output (compreso fra  $y_{min}$  e  $y_{max}$ ),  $c$  è il Window Center (0028,1050) e  $w$  indica Window Width (0028,1051):

---

```

if  $x \leq c - 0.5 - (w - 1)/2$  then
   $y = y_{min}$ 
else
  if  $x > c - 0.5 + (w - 1)/2$  then
     $y = y_{max}$ 
  else
     $y = ((x - (c - 0.5))/(w - 1) + 0.5) * (y_{max} - y_{min}) + y_{min}$ 
  end if
end if

```

---

A valori di input al di sotto della finestra corrisponde il valore minimo di output e, a valori superiori, quello massimo. La relazione non indica se il valore massimo debba essere visualizzato con il colore bianco oppure nero dal momento che questa informazione è memorizzata dall'attributo Photometric Interpretation (0028,0004) e può subire variazioni in seguito allo step successivo della pipeline: Presentation Transform. Diversi valori del tag VOI LUT Function determinano unicamente la funzione di conversione, che nel caso di SIGMOID non è più lineare e prende la forma:

$$f(x) = \frac{(y_{max} - y_{min})}{1 + \exp(-4 \frac{x-c}{w})} + y_{min}$$

dove:

- $x$  è il valore di input della LUT
- $c$  è il Window Centre definito in maniera interattiva dall'utente o utilizzando il valore memorizzato in (0028,1050).

- $W$  è la window width definita anch'essa interattivamente o contenuta in (0028,1051).
- $Y$  è il valore di output ottenuto
- $Y_{min}$  è il minimo valore di output
- $Y_{max}$  è il massimo valore di output

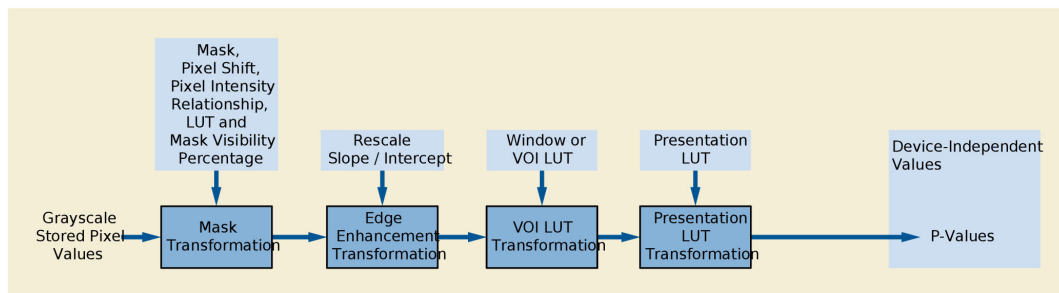


Figura 3.4: Schema illustrativo del processo di elaborazione a cui pixel sono sottoposti

### Presentation transform

La Presentation LUT transformation converte i valori dei pixel in “P-Values”, ossia uno spazio indipendente dal dispositivo. Rappresenta l'ultimo step di applicazione della pipeline di immagini greyscale. Nel caso di una trasformazione lineare, the Presentation LUT è contenuta nel data element Presentation LUT Shape (2050,0020). Nel caso non lineare, invece, la Presentation LUT è descritta dal Presentation LUT Sequence.

### 3.6.3 Orientazione dell'immagine nello spazio

#### Sistema di riferimento anatomico

Relativamente alle tecniche di acquisizione delle immagini medicali, il più importante sistema di riferimento è quello dello spazio anatomico. Questo spazio si serve di 3 piani per descrivere la posizione anatomica umana:

- piano assiale (o trasversale): è parallelo al terreno e separa la testa (parte superiore) dai piedi (parte inferiore), è il piano delle sezioni tradizionali in TC;
- piano coronale (o frontale): è perpendicolare al terreno e separa la porzione anteriore da quella posteriore;

- piano sagittale, che decorre in posizione antero-posteriore, verticalmente, dividendo il corpo in due parti simmetriche, destra e sinistra.

Da questa definizione segue quella dei 3 assi di riferimento, ognuno dei quali è perpendicolare a ciascuno dei piani anatomici. Gli assi si distinguono in:

- Asse longitudinale (a): perpendicolare al piano trasversale;
- Asse trasversale (b): perpendicolare al piano sagittale;
- Asse antero-posteriore (c): perpendicolare al piano frontale.

La descrizione dei punti nello spazio avviene mediante l'utilizzo di coordinate, serve quindi scegliere la direzione in cui i valori di tali coordinate aumentano e, di conseguenza, quella in cui decrescono. Siccome è necessario fare una scelta discriminante, non c'è uniformità nella definizione del sistema di riferimento, e differenti applicazioni mediche impiegano convenzioni differenti. Le più diffuse sono:

- LPS (Left, Posterior, Superior):
  - da destra verso sinistra
  - dalla zona anteriore a quella posteriore
  - dalla zona inferiore a quella superiore
- RAS (Right, Anterior, Superior):
  - da sinistra verso destra
  - dalla zona posteriore a quella anteriore
  - dalla zona inferiore a quella superiore

### **Sistema di riferimento DICOM**

Sebbene possa essere specificato diversamente tramite l'apposito attributo (Anatomical Orientation Type, tag: (0010,2210)), il sistema di riferimento adottato da DICOM nella quasi totalità delle acquisizioni è del tipo LPH (LPS), ovvero:

- l'asse x aumenta verso il lato sinistro del paziente,
- l'asse y aumenta nella verso il lato posteriore
- l'asse z aumenta verso la testa del paziente.

Una notazione frequentemente impiegata per definire tali convenzioni è la regola della mano destra. Nelle sue versioni precedenti, lo standard DICOM impiegava un sistema di riferimento ormai deprecato relativo allo specifico equipaggiamento hardware di acquisizione, non dettagliatamente definito e fonte di ambiguità. Ciò ha portato all'adozione di un nuovo sistema, basato direttamente sul paziente, la cui origine nello spazio tridimensionale non ha valore universale, ma ha significato relativamente alla singola serie.

### Sistema di riferimento dell'immagine

Come descritto nel paragrafo “Rappresentazione dell'immagine”, in seguito all'acquisizione, gli scanner medicali creano una matrice rettangolare di pixel ordinata a partire dal margine sinistro dell'immagine. Questa informazione è codificata nel formato file mediante un array monodimensionale. La collocazione spaziale dei pixel all'interno dell'immagine va quindi ricostruita a partire dagli attributi Rows e Columns. A tal riguardo, è utile identificare la posizione di ciascun pixel mediante 2 indici,  $i$  e  $j$ , con il seguente significato:

- $i$ : indice della colonna del piano dell'immagine, la prima colonna è zero.
- $j$ : indice di riga, la prima riga ha indice  $j = 0$ .

Il valore di  $i$  aumenta verso destra, quello di  $j$  verso il basso. La matrice di pixel così ottenuta può essere quindi suddivisa in un insieme di vettori riga e vettori colonna.

### Image plane module

Definite queste convenzioni, i campi DICOM che contengono le informazioni necessarie ad orientare ciascuna slice nello spazio si trovano all'interno dell'Image Plane Module.

**Table C.7-10**  
**IMAGE PLANE MODULE ATTRIBUTES**

Attribute Name	Tag	Type	Attribute Description
Pixel Spacing	(0028,0030)	1	Physical distance in the patient between the center of each pixel, specified by a numeric pair - adjacent row spacing (delimiter) adjacent column spacing in mm. See 10.7.1.3 for further explanation.
Image Orientation (Patient)	(0020,0037)	1	The direction cosines of the first row and the first column with respect to the patient. See C.7.6.2.1.1 for further explanation.
Image Position (Patient)	(0020,0032)	1	The x, y, and z coordinates of the upper left hand corner (center of the first voxel transmitted) of the image, in mm. See C.7.6.2.1.1 for further explanation.
Slice Thickness	(0018,0050)	2	Nominal slice thickness, in mm.
Slice Location	(0020,1041)	3	Relative position of the image plane expressed in mm. C.7.6.2.1.2 for further explanation.

Figura 3.5: Gli attributi dell'Image Plane Module

Per ciascun frame ottenuto dal file DICOM corrispondente, l'attributo Image Position (Patient) (0020,0032) fornisce la collocazione spaziale dell'origine dell'immagine rispetto al sistema di coordinate basato sul paziente precedentemente descritto (RCS), intesa come centro del primo voxel trasmesso, specificata in mm. Il valore di Image Orientation (Patient) (0020,0037) specifica inoltre l'orientazione della prima riga e della prima colonna del frame mediante due vettori, che contengono il coseno dei loro angoli di inclinazione rispetto al RCS. È espresso nella forma:  $(X_x \backslash X_y \backslash X_z \backslash Y_x \backslash Y_y \backslash Y_z)$ . La corrispondenza fra pixel e coordinate spaziali necessita di un parametro di conversione, definito dall'attributo Pixel Spacing. Esso esprime, mediante un vettore composto da due componenti, la distanza fisica in mm fra il centro di 2 pixel contigui rispettivamente nella direzione colonna e nella direzione riga.

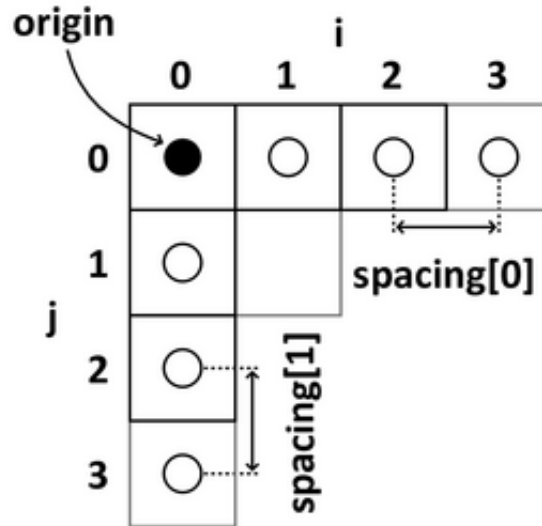


Figura 3.6: Sistema di riferimento dell'immagine

Un esempio dell'interpretazione di tale attributo è il seguente. Se si assume che image orientation del file DICOM contenga il vettore direzione riga  $(1,0,0)$  e un vettore direzione colonna  $(0,1,0)$ , la stessa informazione può essere espressa in questo modo:

- vettore riga:  $(\cos 0^\circ, \cos 90^\circ, \cos 90^\circ)$
- vettore colonna:  $(\cos 90^\circ, \cos 0^\circ, \cos 90^\circ)$

Questo indica che l'orientazione delle slice corrisponde esattamente alle direzioni X e Y ortogonali dell'RCS. Al contrario, se l'attributo fosse stato ad esempio  $0.5 \backslash 0 \backslash -0.8660254 \backslash 0 \backslash 1 \backslash 0$ , i vettori sarebbero stati così definiti:

- vettore riga:  $(\cos 60^\circ, \cos 90^\circ, \cos 150^\circ)$
- vettore colonna:  $(\cos 90^\circ, \cos 0^\circ, \cos 90^\circ)$

Dagli angoli ottenuti si desume che le righe del frame sono orientate di  $60^\circ$  rispetto alla direzione X dell'RCS e  $150^\circ$  dalla direzione Z, mentre la direzione delle colonne coincide con quella dell'RCS, ovvero è perpendicolare all'asse X e all'asse Y.

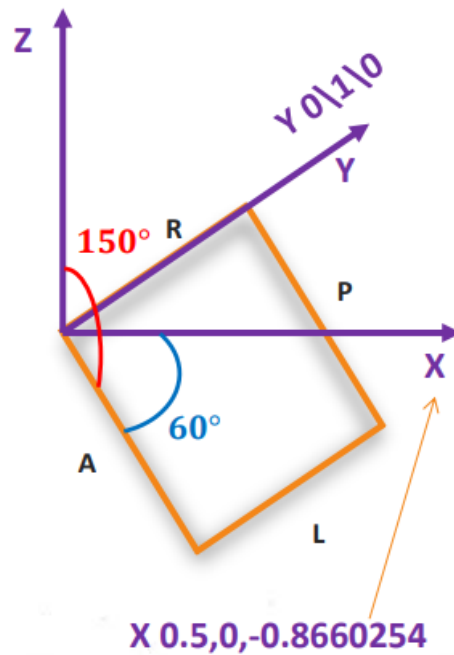


Figura 3.7: Risultato dell'interpretazione dell'attributo Image Orientation (Patient), tratto da *Fundamental of Medical Imaging*, Department of Computer Science, University of Houston

Lo standard DICOM prevede ulteriori attributi che descrivono la disposizione spaziale di ciascuna immagine all'interno dell'RCS, in parte collocati in altri moduli. Fra questi:

- Slice Location (0020,1041). Rappresenta la posizione relativa del piano dell'immagine espressa in mm.
- Position patient: come descritto precedentemente, specifica la posizione del paziente rispetto al dispositivo di acquisizione.

Queste informazioni risultano ridondanti oppure dipendenti dallo specifico device di acquisizione e la documentazione è abbastanza chiara nel limitarne l'utilizzo al solo scopo di annotazione.

Queste considerazioni sono il punto di partenza per la ricostruzione del volume 3D. La descrizione delle specifiche modalità attraverso le quali è realizzarla verrà esposta all'interno del capitolo successivo.



# Capitolo 4

## Progettazione e implementazione

### 4.1 Dal DICOM alla AR

Questo capitolo illustra il procedimento da impiegare per realizzare i requisiti definiti nel capitolo 2, cioè per :

- disporre di una rappresentazione virtuale in realtà aumentata a partire da file in formato DICOM.

#### 4.1.1 Step operativi

Come già elencato, si vuole disporre in AR, in maniera separata, sia del dato bidimensionale sia di quello tridimensionale, è necessario quindi distinguere i due procedimenti.

Per quanto riguarda la visualizzazione del modello segmentato nell'universo aumentato, la sequenza dei passi operativi si compone, ad alto livello, di:

- Processo di segmentazione: lo step preliminare riguarda l'estrazione del modello tridimensionale dal volume ottenuto dalla sovrapposizione dei frame DICOM.
- Riconoscimento di un formato 3D compatibile con il game engine utilizzato: quest'ultimo non è predisposto per l'importazione diretta dei dati risultanti dalla segmentazione, che definiscono il modello tridimensionale in termini di vertici e di facce, senza cioè i metadati necessari alla loro interpretazione. È pertanto necessario individuare fra i formati di memorizzazione di modelli 3D digitali, uno che risponda alle esigenze delle operative e con il game engine.

- Conversione dei dati sul modello segmentato (mesh triangolare) al formato precedentemente scelto. Nel nostro caso specifico, il dato proveniente dal processo di segmentazione è stato prodotto mediante le tecnologie utilizzate dai nostri collaboratori di ing. biomedica (matlab), che sono incompatibili con quelle impiegate dal motore grafico responsabile del rendering in MR. Essendo quelle informazioni memorizzate in un formato dati proprietario (.mat) e non attraverso una codifica interoperabile (quale ad esempio json), sarà necessario fare ricorso alle loro medesime tecnologie per estrarre le informazioni di interesse.
- Importazione nel game engine responsabile del rendering aumentato. Una volta convertito, il modello può essere importato nel game engine. Nel mio caso, è stato utilizzato il più popolare game engine a disposizione degli sviluppatori di videogiochi: Unity. Unity predispone una serie di plugin per l'utilizzo dei principali SDK per lo sviluppo di applicazioni di realtà aumentata, fra i quali Vuforia, ARCore e ARKit. Unity ha proattivamente abbracciato l'integrazione al suo interno dello sviluppo di applicazioni AR fornendo gli strumenti per una rapida prototipazione e applicazioni solide. Naturalmente, a seconda della specifica piattaforma utilizzata, gli SDK compatibili variano.
- Individuazione di un SDK specifico per il dispositivo di destinazione. In base al tipo di device e alle esigenze dell'applicazione, la scelta di quale sia la migliore soluzione cambia. In questo senso, il recente progresso tecnologico dei dispositivi per AR e MR sta influenzando pesantemente le scelte delle case produttrici, che stanno volgendo una sempre maggiore attenzione a queste tecnologie e incrementando le scelte disponibili agli sviluppatori. Le funzionalità messe a disposizione dai vari tool sono eterogenee e numerose: spaziano dal riconoscimento di specifici oggetti nel mondo fisico, alla stima dell'illuminazione dell'ambiente, così come al supporto per smartglasses o all'acquisizione delle immagini mediante videocamere singole o doppie, o all'integrazione con il GPS per la creazione di app location-based. Per lo specifico caso di rappresentare l'informazione visualizzata da DICOM saranno essenzialmente necessarie:
  - la ricostruzione della conformazione dell'ambiente, specificatamente alle superfici piane, al fine di scegliere i piani sui quali posizionare gli oggetti virtuali.
  - Il tracking del dispositivo, per non vincolare l'utente a stare in una posizione fissa.

Grande importanza avrà la gestione dell'input, al fine di agevolare l'interazione fra l'utente e gli oggetti virtuali, aspetto cruciale per una esperienza di mixed reality.

- Sviluppo del prototipo mediante le API fornite dall'SDK. A questo punto si dispone degli strumenti per gestire l'oggetto virtuale nell'astrazione propria dello specifico SDK utilizzato, il quale permette di sviluppare l'esperienza aumentata di visualizzazione e manipolazione.

Relativamente al dato DICOM:

- Estrazione ed interpretazione dei pixel dal dataset DICOM (ovvero dall'insieme di slice).
- Modellazione del dato DICOM sull'universo AR/MR: DICOM utilizza un proprio modello di informazione, il quale prevede un numero corposo di elementi, non tutti necessari alla rappresentazione in AR/MR. Contemporaneamente è necessario convertire alcuni dei campi nell'astrazione disponibile nella tecnologia utilizzata per la costruzione dell'esperienza di MR. Ovviamente, questo compito è fortemente dipendente dalla scelta di quest'ultima.
- La scelta dell'SDK e lo sviluppo sono in comune con il modello tridimensionale.

La procedura descritta in questo capitolo è stata applicata a 2 dataset DICOM offerti dal dip. di ingegneria biomedica, costituiti da 2 stack di immagini DICOM, relativamente al distretto cardiaco e renale, rispettivamente ottenuti da acquisizioni MRI e CT.

## 4.2 Pipeline operativa

Questa sezione si concentra su un sottoinsieme degli step elencati precedentemente, focalizzandosi sul procedimento necessario ad ottenere un modello tridimensionale segmentato a partire da un set di file DICOM.

### 4.2.1 Tecnologie

Si procede in questa sede ad una breve trattazione degli strumenti software impiegati per completare la pipeline descritta, al fine di comprendere meglio gli step di cui è composta.

**Slicer** Slicer è una piattaforma open-source per la gestione, visualizzazione, analisi scientifica ed elaborazione delle immagini medicali. Slicer permette di creare "stack di immagini", partendo da immagini DICOM, trasformandole in blocchi di dati tri, quadri e pentadimensionali (ad esempio mostrando la contemporanea variazione di due parametri, in un reticolo tridimensionale nel tempo), come in filmati 3D riguardanti la fMRI, la PET, etc. Slicer permette anche di confrontare due o più risonanze magnetiche di uno stesso soggetto nel tempo, per determinare se è apparsa una lesione (nel individuo sano), se sono apparse nuove lesioni, oppure se queste si sono ingrandite o ridotte. Viene utilizzato in molte applicazioni medicali, tra queste: lo studio dell'autismo, della sclerosi multipla, del lupus eritematoso sistemico, del carcinoma prostatico, della schizofrenia, della biomeccanica ortopedica, della BPCO, malattie cardiovascolari, neurochirurgia, radioterapia etc.[1]

**Blender** Blender è un tool-set opensource per la grafica 3D utilizzato per modellazione, rigging, animazione, montaggio video, composizione e rendering di immagini tridimensionali e bidimensionali. Fra le sue funzionalità ci sono la modellazione 3D, la mappatura UV, simulazioni di fluidi, rivestimenti, particelle, la creazione di applicazioni/giochi 3D. È disponibile su numerosi sistemi operativi ed è dotato di un robusto insieme di funzionalità paragonabili, per caratteristiche e complessità, ad altri noti programmi per la modellazione 3D come Softimage XSI, Cinema 4D, 3D Studio Max, LightWave 3D e Maya [1].

**Segmentazione** Il primo passo nella direzione di ottenere una visualizzazione tridimensionale del dato DICOM consiste nella segmentazione, ovvero nell'estrazione del modello della zona di interesse dal volume derivante dalla sovrapposizione dei frame DICOM. Come già discusso precedentemente, si tratta di una tecnica complessa che richiede l'analisi di numerosi parametri per produrre risultati soddisfacenti. Inoltre, ha la caratteristica di essere difficilmente generalizzabile, ovvero gli accorgimenti da attuare risultano dipendenti dalla natura del tessuto della specifica struttura da individuare, così come dalla modalità e dal dispositivo di acquisizione. Nonostante disponessimo già in partenza dei dati risultanti dal processo di segmentazione offerti dai nostri collaboratori di ingegneria biomedica, è stato ottenuto un nuovo modello mediante un procedimento indipendente, per poter valutare il processo di conversione dal DICOM alla MR in maniera completa, cioè a partire dal dataset originale. La segmentazione è stata realizzata tramite il software di analisi ed elaborazione di immagini biomedicali Slicer, attraverso un procedimento semiautomatico, mediante l'impiego di una sogliatura globale e la successiva rimozione dei segmenti non di interesse precedentemente estratti. L'alta specificità di questa operazione rappresenta l'ostacolo principale alla realizzazione

di un processo di conversione automatizzato che possa astrarre contemporaneamente dalle tecniche di acquisizione e dalle zone anatomiche sotto esame. Al contrario, la possibilità di includere l'intervento dell'utente all'interno della pipeline di conversione, al fine di selezionare autonomamente o registrare alcuni dei parametri, può rappresentare una soluzione potenzialmente più generale, ma non è stata contemplata nello sviluppo di questo prototipo.

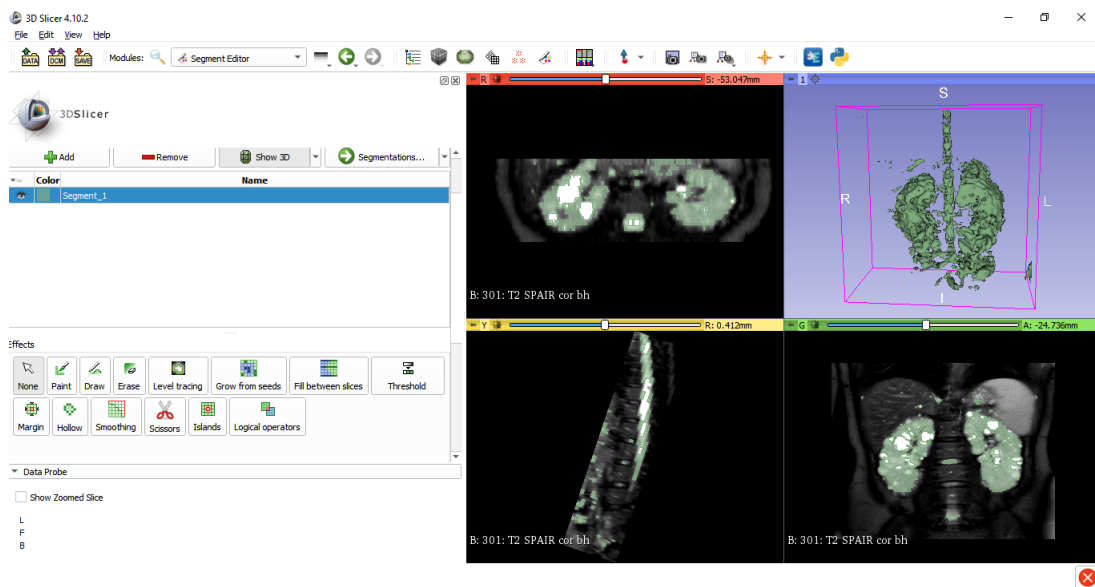


Figura 4.1: L'interfaccia utente di Slicer.

Per poter essere visualizzati all'interno dell'editor, i dataset DICOM devono essere precedentemente importati. A seguito di questa operazione, l'ambiente consente la visualizzazione delle slices corrispondenti non solo ai piani acquisiti, bensì è in grado di ricostruirsi anche le rimanenti, consentendo all'utente la visualizzazione dei 3 piani ortogonali. Slicer predispose un apposito modulo per la segmentazione dei modelli estratti da DICOM, chiamato segment editor, che fornisce il supporto alla sovrapposizione di segmenti, visualizzazione in 2D e 3D, modifiche direttamente sul volume segmentato e tanto altro. L'operazione di segmentazione è agevolata da una serie di parametri che permettono di elaborare l'immagine attraverso filtri e algoritmi appositi (smoothing, operatori logici, level tracing).

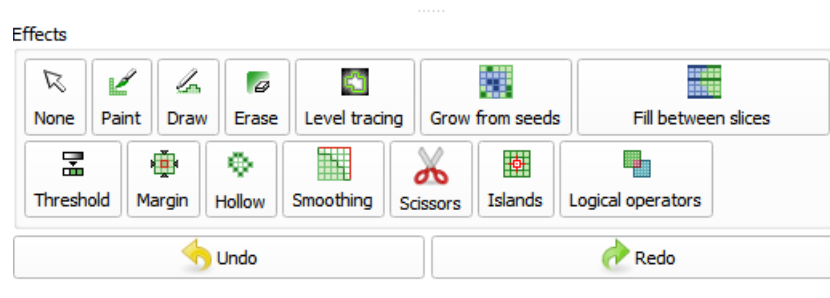


Figura 4.2: Dettaglio degli strumenti di segmentazione predisposti dal segment editor.

Una volta completata, è sufficiente esportare il segmento ottenuto; un modulo apposito consente di specificare il formato, fra `.stl` e `.obj`, di destinazione. Fra le funzionalità offerte prima di esportare, c'è anche il ridimensionamento del modello.

**Conversione del modello** Per quanto riguarda i dati provenienti da ing. biomedica, l'output prodotto dalla segmentazione è costituito da dati raw, che codificano cioè l'informazione su facce e vertici secondo convenzioni non standard. In questa forma, non sono supportati direttamente dalla piattaforma Unity e devono quindi passare attraverso una preliminare conversione. Fra i formati attualmente compatibili con Unity (`.fbx`, `.dae` (Collada), `.3ds`, `.dxf`, `.obj`, and `.skp`) è stato utilizzato il `.fbx`. Anche se non presente sui 2 dataset DICOM descritti come input al progetto, su tale scelta ha influito la necessità di rappresentare visivamente (mediante colori) le informazioni di contrasto ottenute da acquisizioni con mezzo di contrasto, ed è stata testata su un set a parte di dati, ottenuti da immagini biomedicali di risonanza magnetica (gadolinio) memorizzate tramite codifica diversa dal DICOM (`nrrd`). Queste riflessioni si estendono naturalmente anche al caso DICOM. Alcuni dei formati compatibili con Unity, come ad esempio `fbx` e `obj`, si differenziano sostanzialmente fra loro per: scopi, complessità e licenza di utilizzo. L'approccio per la traduzione dei dati è stato il seguente: convertire (in modo autonomo) dapprima il modello ad un formato universale e piuttosto semplice da codificare come l'`obj` e, in seguito, incaricare un potente software di grafica 3D come Blender della traduzione da uno con esso compatibile a un formato proprietario come `fbx`; realizzare da capo un exporter `fbx` avrebbe richiesto molto più tempo. Nonostante si disponesse già dell'implementazione di un exporter fornita da matlab verso un formato importabile in Blender(`stl`), si è scelto di svilupparne uno nuovo, ad `obj`. Infatti, il formato `.stl`, a differenza di `.obj`, non fornisce un supporto standardizzato per il colore, perché definisce la geometria del modello.

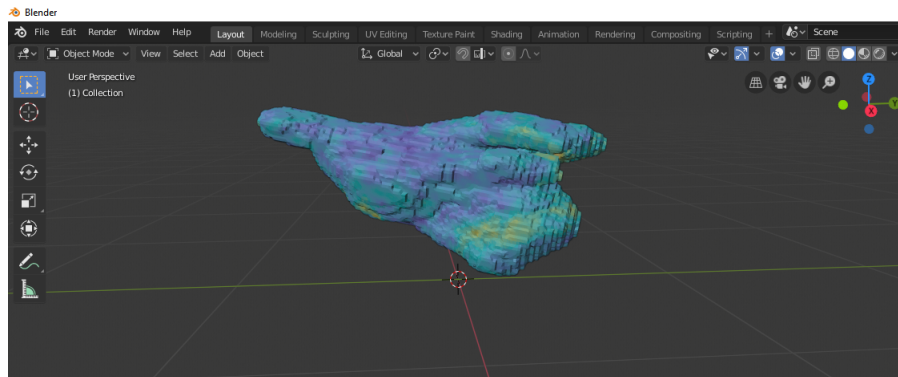


Figura 4.3: Il modello dell'atrio derivante dall'associazione ai valori di contrasto di un'intensità nello spazio RGB, visualizzato in Blender.

**STL file format** STL (STereo Lithography interface format o acronimo di "Standard Triangulation Language") è un formato di file, binario o ASCII, rilasciato nel 1987 per i software di stereolitografia CAD. È utilizzato nella prototipazione rapida attraverso software CAD e nella stampa 3D.

Un file .stl rappresenta un modello la cui superficie è stata discretizzata in facce triangolari. Impiega il sistema cartesiano attraverso le coordinate X, Y e Z ripetute per ciascuno dei tre vertici di ciascun triangolo, ordinati secondo la regola della mano destra, con un vettore unitario per descrivere l'orientazione della normale alla superficie. Un file stl non contiene informazioni sulla scala e l'unità di misura è arbitraria.

Il formato STL presenta dei vantaggi quali la semplicità, in quanto risulta molto facile da generare e da processare, mentre a suo sfavore presenta una geometria approssimata e la sua struttura dati, che pur risultando semplice, può presentare la ripetizione dello stesso vertice più volte [1].

**Obj file format** OBJ (o .OBJ) è un formato file per definire geometrie 3D sviluppato dalla Wavefront Technologies per il software Advanced Visualizer. È un formato aperto che gode di una vastissima popolarità fra gli applicativi per la grafica 3D per l'interscambio di dati con altri programmi. È un formato semplice, con un data-format che rappresenta solamente la geometria 3D, ossia la posizione di ogni vertice, la posizione di ogni coordinata UV per le texture, le normali e le facce che compongono il modello. I vertici sono memorizzati di default in un ordine antiorario rendendo la dichiarazione esplicita delle normali non necessaria. Le coordinate di un OBJ non hanno unità di misura, ma informazioni sulla scala del modello possono essere contenute in una linea di codice commentata.

**FBX file format** FBX (Filmbox) è un formato file proprietario (.fbx) sviluppato da Kaydara e di proprietà di Autodesk dal 2006. Viene utilizzato per fornire l'interoperabilità tra le applicazioni di creazione di contenuti digitali. FBX è anche parte di Autodesk Gameware, una serie di video game middleware. Può essere salvato su disco sia in forma binaria sia come ASCII dato che il suo SDK supporta la lettura e la scrittura di entrambi. Per il formato di file binario FBX, Blender Foundation ha pubblicato una specifica non ufficiale, così come una specifica di livello superiore non ufficiale per come i dati reali sono disposti in FBX (indipendentemente dal formato ASCII o binario)[1].

**Ridefinizione del pivot point del modello** Il punto di perno (pivot point) è il punto nello spazio che definisce il centro matematico dell'oggetto per la rotazione e il ridimensionamento. L'utilizzo di Blender è stato fondamentale non solo per ottenere una codifica .fbx della mesh, ma anche per ri-collocare il pivot point nel baricentro del modello tridimensionale ottenuto. Se fosse stata direttamente importata in Unity senza questo accorgimento, la manipolazione sarebbe stata controintuitiva e problematica, poiché la rotazione e il posizionamento sarebbero stati definiti rispetto all'origine del sistema di riferimento impiegato nella segmentazione e codificato nei dati sui vertici ottenuti. Bisogna sottolineare come questo step non sia necessario nel caso la segmentazione sia stata effettuata tramite software di visualizzazione 3D come Invesalius o Slicer.



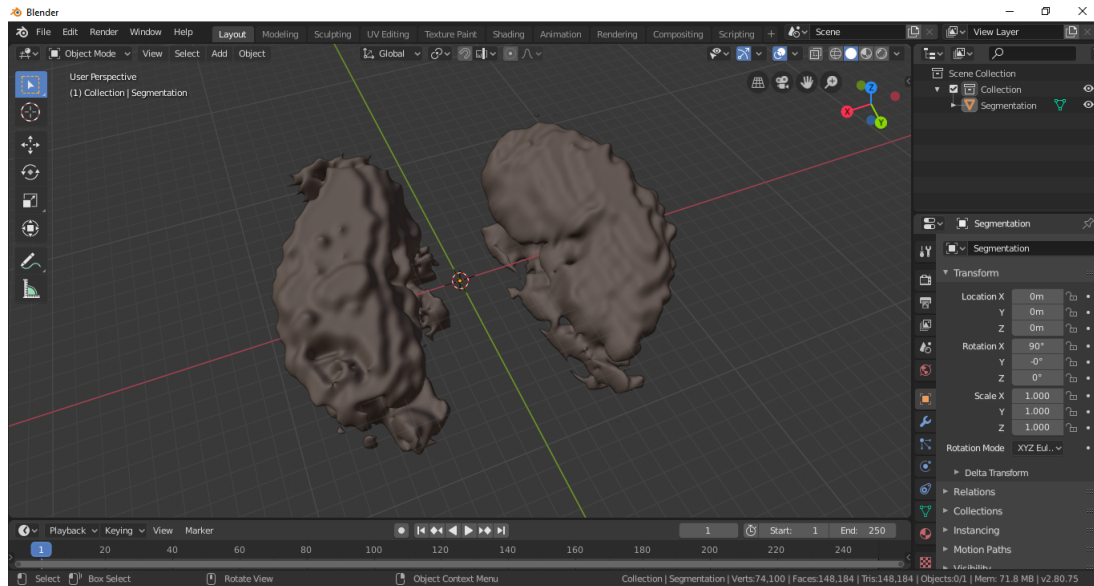


Figura 4.4: Il risultato dell’importazione in Blender della mesh dell’anatomia renale. Al centro si può notare il pivot point ridefinito.

**Interpretazione dei riferimenti spaziali** Prima di capire in quale sistema di riferimento sia stata acquisita la mesh, è necessario comprendere quello adottato da DICOM. Una volta ottenuto, basterà analizzare visivamente la mesh per capire se è necessario procedere o meno con il cambio di riferimento. Gli elementi tag necessari in questo compito sono quelli descritti alla sezione “Image plane module” del capitolo 3 e qui riassunti:

- Image Position (Patient)
- Image Orientation (Patient)
- Pixel Spacing
- Spacing Between Slices

Un file DICOM si può comporre di più frame, ovvero immagini. In questo caso si assume che l’orientazione di ciascun frame sia la stessa.

La trasformazione spaziale che si ottiene da DICOM è una trasformazione affine, ovvero una composizione di una trasformazione lineare con una traslazione. Secondo DICOM, ciascun punto di un frame viene collocato nello spazio attraverso la seguente relazione

$$\vec{x} = A(ijk)' + t$$

, dove  $ij$  rappresentano le coordinate di un pixel nel sistema di riferimento dell'immagine (descritto nel cap. 2), mentre  $k$  rappresenta l'indice della slice ed è crescente verso la zona superiore del paziente, queste considerazioni derivano dal fatto che DICOM è un sistema di riferimento LPS. La rappresentazione matriciale di tale trasformazione è la seguente:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

, che è una trasformazione affine poiché combina una traslazione con trasformazione lineare. Quest'ultima si compone di rotazione e scala, le quali sono ottenute dal DICOM rispettivamente dai campi Image Orientation (Patient) e pixel spacing, mentre il vettore di traslazione è costituito direttamente da Image Position Patient. Risulta conveniente dal punto di vista computazionale esprimere la composizione di matrice e della traslazione attraverso una singola matrice, poiché permette di ridurre la trasformazione geometrica ad un singolo prodotto. Questo si realizza aggiungendo una riga in coda a quella della matrice ed esprimendo i 2 vettori tramite coordinate omogenee. Pertanto, l'equazione risultante è la seguente:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & t_1 \\ A_{21} & A_{22} & A_{23} & t_2 \\ A_{31} & A_{32} & A_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_i \\ A_j \\ A_k \\ 1 \end{bmatrix}$$

Si può verificare come il risultato della nuova matrice in 4 dimensioni esprima correttamente la combinazione di traslazione e trasformazione lineare in 3 dimensioni. Considerando infatti una rappresentazione semplificata:

$$\begin{pmatrix} P & v \\ 0 & 1 \end{pmatrix}$$

, a seguito della trasformazione si ottiene:

$$\begin{pmatrix} P & v \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} Pu + v \\ 1 \end{pmatrix}$$

, che è facilmente riconducibile alla trasformazione richiesta.

Seguendo le considerazioni descritte, la matrice completa di trasformazione estratta dal DICOM è la seguente:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x \Delta i & Y_x \Delta j & 0 & S_x \\ X_y \Delta i & Y_y \Delta j & 0 & S_y \\ X_z \Delta i & Y_z \Delta j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ j \\ 0 \\ 1 \end{bmatrix} = M \begin{bmatrix} 1 \\ j \\ 0 \\ 1 \end{bmatrix}$$

, dove:

- $P_{xyz}$  rappresenta, in mm, le coordinate del voxel  $ij$  dell'immagine.
- $S_{xyz}$  sono i 3 vettori contenuti in Image Position Patient.
- $Y_{xyz}$  è la direzione del vettore della prima riga dell'immagine, espressa in forma di coseno dell'angolo rispetto al sistema di riferimento del paziente (RCS).
- $Z_{xyz}$  è la direzione del vettore colonna, espressa in forma di coseno dell'angolo rispetto all'RCS.
- $i$  è l'indice riga nel sistema di riferimento dell'immagine
- $\Delta i$  è l'attributo pixel spacing  $x$ , che esprime la distanza fisica in mm fra il centro di due pixel contigui della stessa riga.
- $j$  è l'indice della colonna nel sistema di riferimento dell'immagine
- $\Delta j$  è l'attributo pixel spacing  $y$ , che esprime la distanza fisica in mm fra il centro di due pixel contigui della stessa colonna.

Bisogna evidenziare che non è necessario prendere in considerazione l'attributo `sliceLocation`, ovvero la posizione relativa di ciascun piano rispetto ad un punto imprecisato, e che questa fornisce pertanto un'informazione ridondante, oltre che non affidabile, perché spesso assente, data la sua opzionalità.

Analizzando visivamente attraverso un software apposito, è risultato evidente come il sistema di riferimento tramite il quale erano espressi i vertici delle sue facce triangolari non era quello del paziente. Nello specifico è servito traslarla e ruotarla correttamente. Per quanto riguarda la mesh prodotta autonomamente, Slicer consente di specificare il sistema di riferimento all'atto della esportazione, e quindi questo passaggio non è stato in questo caso necessario.

### 4.2.2 Elaborazione procedurale dei dati sulla mesh

Per quanto riguarda la catena di passaggi necessari a visualizzare una rappresentazione bidimensionale del DICOM, come sarà spiegato più avanti, la sua implementazione è completamente automatizzata. Sebbene alcune funzionalità non siano ancora supportate (si veda capitolo 6), essa rappresenta un'implementazione in C# delle logiche descritte nel capitolo 3. Riguardo al modello tridimensionale ottenuto dalla mesh, si è già spiegata la necessità di elaborare il dato iniziale attraverso trasformazioni geometriche. Per questo motivo, per implementare questa parte della catena, si è deciso di utilizzare un approccio procedurale, sfruttando la potenza e la espressività con cui matlab permette di gestire matrici e vettori, e quindi non impiegando l'astrazione ad oggetti.

**Estrazione delle informazioni sul colore** Come già spiegato, per convertire le informazioni su vertici e facce al formato obj si è sfruttato un toolbox predisposto da matlab, a cui si è integrato un componente implementato autonomamente. Oltre a codificare la geometria del modello, lo script estrae anche le informazioni sul contrasto dandovi un valore corrispondente sullo spazio RGB, sulla base di una palette di colori di matlab (cmap). Le informazioni sulle intensità sono state elaborate solo per i dati provenienti da acquisizioni predisposte (in questo caso la mesh offertaci per testare questa funzionalità non era fra quelle già citate).

```
for i = 1:size(cmap, 1)
    material(i).type='newmtl';
    material(i).data=strcat('mat-', num2str(i));
    i=i+1;
    material(i).type='Ka';
    material(i).data=[cmap(i, 1) cmap(i,2) cmap(i, 3)]; % rgb
        normalized to 1
    i=i+1;
    material(i).type='Kd';
    material(i).data=[cmap(i, 1) cmap(i,2) cmap(i, 3)]; % rgb
        normalized to 1
    i=i+1;
    material(i).type='Ks';
    material(i).data=[1 1 1];
    numEntry=i+1;
    material(i).type='illum';
    material(i).data=2;
    numEntry=numEntry+1;
    material(i).type='Ns';
```

```

    material(i).data=27;
    i+1;
end

```

Listato 4.1: Generazione dei materiali corrispondenti ai valori di contrasto ottenuti

**Estrazione della matrice affine** In virtù della facilità di indicizzazione di array e matrici, della predisposizione nativa alla loro manipolazione e delle funzioni già implementate per la trigonometria e la geometria, lo step della pipeline relativo all'estrazione della matrice affine dal file DICOM, secondo la metodologia descritta precedentemente, è stato realizzato in matlab.

```

function [M,R] = TransMatrix(info)
    ipp=info.ImagePositionPatient;
    iop=info.ImageOrientationPatient;
    ps=info.PixelSpacing;
    Tipp=[1 0 0 ipp(1); 0 1 0 ipp(2); 0 0 1 ipp(3); 0 0 0 1];
    r=iop(1:3); c=iop(4:6); s=cross(r',c');
    R = [r(1) c(1) s(1) 0; r(2) c(2) s(2) 0; r(3) c(3) s(3) 0; 0 0 0
        1];
    S = [ps(2) 0 0 0; 0 ps(1) 0 0; 0 0 info.SpacingBetweenSlices 0; 0 0
        0 1];
    T0 = [ 1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
    M = Tipp * R * T0; %M = Tipp * R * S * T0;
end

```

Listato 4.2: Codice matlab per l'estrazione di matrice affine e matrice di rotazione dal file DICOM, (da Yaman)

Oltre a queste funzionalità, la potenza e la flessibilità di matlab sono state sfruttate anche per visualizzare i risultati delle trasformazioni geometriche sui dati DICOM al fine di verificarne la correttezza, e persino per visualizzare, a fini di debug e studio, il valore dei singoli campi di DICOM, in quanto anche questa funzionalità è disponibile mediante l'apposita funzione dicominfo. Riguardo agli sviluppi futuri, che probabilmente richiederanno l'utilizzo di tecniche di interpolazione, sono stati condotti esperimenti sfruttando l'approccio intuitivo offerto da questo linguaggio, che permette di cimentarsi direttamente sull'algoritmo di risoluzione senza dover perdere troppo tempo nella comprensione della sintassi. Questo pregio può essere alla base per la comprensione delle tecniche di elaborazione necessarie in futuro, salvo poi convertire il codice ad un altro linguaggio.

## 4.3 Modellazione e sviluppo degli elementi DICOM

### 4.3.1 Design

Definita nel dettaglio la catena tecnologica di conversione relativa al modello tridimensionale, è necessario affrontare il procedimento di traduzione del dato bidimensionale. Di seguito viene presentato il modo in cui sono stati progettate le entità DICOM di interesse per l'applicazione. Il modello di informazione DICOM comprende numerose informazioni, solo alcune delle quali sono necessarie al fine di raggiungere i requisiti stabiliti nell'analisi; è pertanto stato necessario selezionare solo alcune di esse e tradurne altre. Questa sezione è stata distinta dalle successive poiché queste classi costituiscono un modulo potenzialmente utilizzabile da altre applicazioni (e così sarà per lo sviluppo di una seconda versione del prototipo).

**DICOMFrame** La classe DICOMFrame modella un frame DICOM nel mondo Unity. Il nome, che intenzionalmente non richiama il termine “slice”, sebbene frequentemente citato finora in queste pagine, è dovuto al fatto che, secondo le specifiche DICOM, ciascuna immagine è composta da un frame piuttosto che da una slice, dal momento che essa può contenerne molteplici, accomunati dalle stesse caratteristiche di acquisizione e associate alla stessa serie. DICOMFrame racchiude le sole informazioni di interesse per la sua rappresentazione all'interno del game engine, modellate attraverso le astrazioni offerte da Unity, cioè:

- un componente bidimensionale, che funge da contenitore della texture vera e propria costituita dai pixel ottenuti dal file
- le informazioni spaziali necessarie alla sua collocazione all'interno del mondo aumentato. I dati contenuti sono derivanti dall'interpretazione dei tag DICOM reperiti dal dataset estratto dal rispettivo file, secondo le convenzioni esposte nel capitolo 3.

**PixelProcessor** Il compito di estrarre i pixel dallo stream di byte contenuto nell'elemento pixel data, nel rispetto della specifica codifica utilizzata, è affidato alla classe PixelProcessor, che incapsula al suo interno la logica di decodifica e restituisce la immagine sottoforma di Texture2D. Rappresenta l'unica classe che accede direttamente ai pixel e quindi sarà l'unica coinvolta da futuri cambiamenti volti a rendere il sistema più solido e in grado di supportare un maggior numero di formati di compressione rispetto a quelli attualmente disponibili.

**La gerarchia di IImageTransformation** Dopo aver estratto i pixel, DICOM prevede una serie di parametri volti a definire l'insieme delle trasformazioni da applicare all'immagine, al fine di adattarla alla presentazione, definita nel capitolo 2 come "greyscale pipeline". Ciascuna delle trasformazioni in questione deve aderire a questa interfaccia, che espone il singolo metodo Apply(). Anche la greyscale pipeline nel suo complesso, che si compone di VOITransformation e di ModalityTransformation, implementa IImageTransformation. Come specificato dallo standard, ognuna delle trasformazioni impiega una LookupTable per rendere più efficiente l'insieme delle operazioni ad essa connesse, e per questo VOITransformation e ModalityTransformation si servono rispettivamente di LUTUshortTrasformation e LUTByteTransformation. PixelProcessor demanda l'esecuzione della pipeline a DICOMGreyscalePipeline, che affida la logica dell'elaborazione dei pixel per delegazione a VOITransformation e ModalityTransformation, le quali, a loro volta, affidano il compito dell'elaborazione a LUTUshortTrasformation e LUTByteTransformation per delegazione, anziché estenderle. Dunque, l'approccio generale è stato quello di rendere modulare la logica di elaborazione dell'immagine, in accordo con il Single responsibility principle.

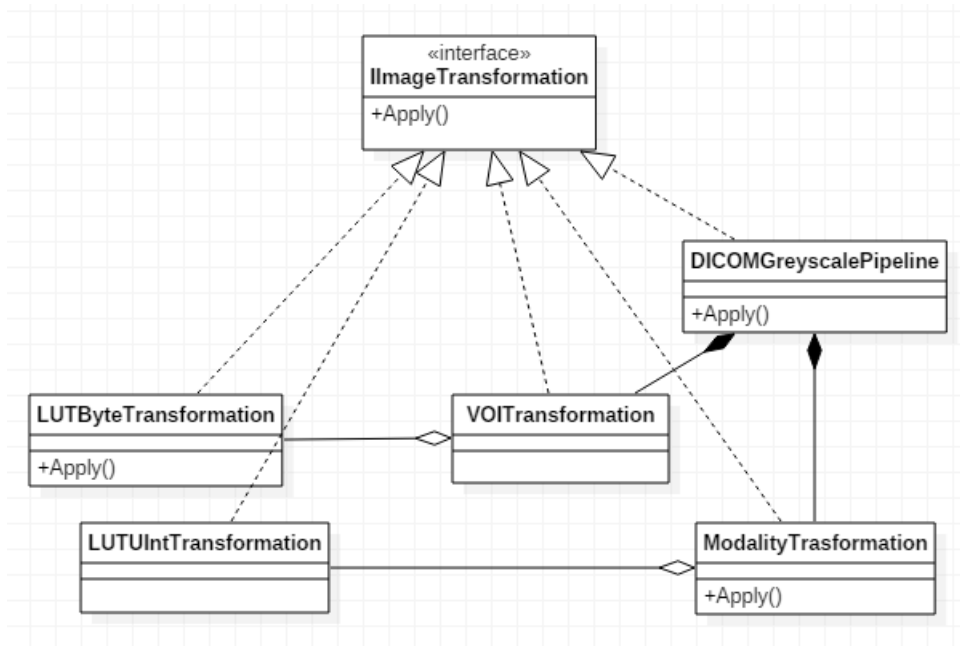


Figura 4.5: La gerarchia delle classi che modellano la trasformazione dell'immagine in DICOM.

Future aggiunte da apportare al sistema, al fine di supportare ulteriori step di elaborazione, dovranno implementare questa interfaccia e potranno servirsi

delle trasformazioni tramite look-up già predisposte.

**La gerarchia Loader: IFramesLoader, AcquisitionFramesLoader, OrientedFramesLoader, NonOrientedFramesLoader** Questa gerarchia modella la logica del caricamento dei frame e della loro messa a disposizione dell'applicazione, o di chiunque ne abbia necessità in futuro. Il contratto stabilito dall'interfaccia espone il metodo `LoadFrames`, che restituisce l'elenco dei `DICOMFrame` estratti dal file comunicato. Come evidenziato nella sezione d'analisi, i requisiti stabiliscono di visualizzare esclusivamente i frame "acquisiti", cioè quelli direttamente prodotti dal dispositivo di acquisizione. Solo in futuro l'attenzione sarà rivolta ai frame non acquisiti (vedi lavori futuri, cap. 6). Questa distinzione si riflette architetturealmente nella distinzione fra le classi astratte `AcquisitionFrames` e `NonAcquisitionFrames`. Quest'ultima non è stata implementata, ma il suo compito sarà quello di incapsulare le tecniche di interpolazione e re-sampling volte ad estrarre i frame non acquisiti. Questa diversificazione comportamentale non è l'unica, dal momento che esistono diverse necessità in relazione alle slice, alle loro caratteristiche e all'utilizzo che se ne voglia fare. In questa prima versione del prototipo, l'applicazione predisporrà all'utente i `DICOMFrame` senza le relazioni spaziali estratte dal DICOM, tuttavia in un secondo momento (vedi cap. 5) questa necessità sarà presente. Il bisogno di ricorrere a questa differenziazione nelle logiche di caricamento dei frame DICOM, che conservano comunque una parte invariante, per quanto riguarda la messa a disposizione dell'utente, è stata modellata attraverso l'impiego del pattern `Template Method`. Il metodo `LoadFrames` definisce un algoritmo utilizzando il metodo astratto `LoadFrame`, il quale è implementato solamente dalle classi che derivano da `AcquisitionSlicesFrame` e che costituiscono le uniche effettive implementazioni all'interno della gerarchia. In questo modo, viene catturato il comportamento comune ad un livello superiore, mentre la logica specifica è a carico delle sottoclassi.



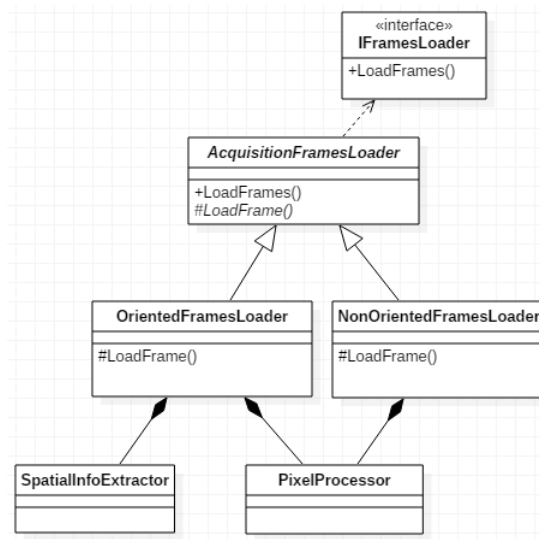


Figura 4.6: Gerarchia dei loader rappresentata mediante il diagramma delle classi di UML.

**DICOMDataset, DICOMDataElement e DICOMTagHelper** Lo scopo di queste 3 classi è quello incapsulare al proprio interno la libreria effettivamente impiegata per la gestione degli aspetti di basso livello relativi a DICOM, oltre che distinguere logicamente i soli elementi DICOM di interesse per questo progetto, cioè quelli relativi all'immagine e alle relazioni spaziali di ogni singolo frame all'interno dell'ambiente aumentato. Ciò consente di eliminare da tutti gli altri componenti del sistema le dipendenze dalla specifica libreria, sostituendovi queste classi. Nel caso in cui in futuro sia necessario cambiare la libreria impiegata, sarà necessario modificare unicamente queste 3 classi, mantenendo inalterato il funzionamento del prototipo.

**Le immagini: AImage, UnityImage, GreyscaleImage** Dato il ruolo centrale dell'elaborazione è stato necessario predisporre un set di classi per la modellazione degli aspetti più specifici. AImage è la classe astratta e generica che costituisce la radice della gerarchia delle immagini. Rappresenta l'immagine prima e durante i diversi step di elaborazione della greyscale pipeline e la sua genericità deriva dal fatto che le diverse trasformazioni determinano output di tipo fra loro diverso (ushort e byte). Astrae da Unity e potenzialmente potrebbe essere impiegata in qualsiasi altro sistema di elaborazione per le immagini. La classe astratta UnityImage, invece, modella un'immagine nell'universo Unity e per questo presenta il metodo ConvertToTexture. Ogni classe che la estende contiene la logica di generazione della Texture a partire da AImage, quindi conosce le convenzioni attraverso il quale Unity

crea le texture2D a partire da array di pixel. DICOMGreyscaleImage costituisce l'implementazione di tale astrazione, applicata alle immagini greyscale estratte da file DICOM con PhotometricRepresentation tag corrispondente a MONOCHROME1 o MONOCHROME2, ovvero le più diffuse negli archivi di immagini medicali.

**Elaborazione dell'immagine in C#** Le entità progettate in precedenza sono state implementate in C#, come classi di supporto agli script associati a gameobject. Il design dell'architettura è su più livelli, modulare.

A seguito di una richiesta espressa al IFramesLoader, quest'ultimo delega il compito di produrre una texture al pixelProcessor, che ha il compito di avviare la pipeline. Il flusso di byte viene decodificato dalla libreria EvilDicom, la quale è incapsulata da classi appositamente definite. Dopo essere stati decodificati, i pixel vengono sottoposti alla Greyscale pipeline, che si compone di Modality Transformation e da VOI Transformation, riunite sotto la stessa astrazione di ImageTransformation.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DICOMGreyscalePipeline
{
    private ModalityTransformation mt;
    private VOITransformation vt;
    AImage<ushort> input;

    public DICOMGreyscalePipeline(GreyscaleParameters gp,
        AImage<ushort> input)
    {
        mt = new ModalityTransformation(gp, input);
        vt = new VOITransformation(gp, mt.Apply());
    }

    public AImage<byte> Apply()
    {
        return vt.Apply();
    }
}
```

Listato 4.3: Implementazione in C# della DICOMGreyscale Pipeline

La classe che modella un'immagine è generica, questo perché i vari step di elaborazione hanno tipi di input e output diversi fra loro. Al termine, il tipo dell'immagine deve essere byte, quindi sia la VOI Transformation che la GreyscalePipeline restituiscono pixel in forma di byte.

Alla conclusione del processo, viene generata una GreyScaleImage, dalla quale è possibile estrarre un texture, a partire dalla quale viene generato un frame. L'IFrameLoader, a prescindere da quale delle 2 implementazioni fornite sia scelta, accumula i frame uno alla volta ed infine li rende disponibili al chiamante.

## 4.4 Progettazione e sviluppo con ARCore e Meta 2

### 4.4.1 Prototipazione

Dopo la fase di progettazione, necessaria a modellare il sistema a fronte dei requisiti dell'analisi, è stata eseguita la prototipazione. Si è deciso di rendere disponibile il sistema su 2 diverse piattaforme hardware, sia su un tablet Samsung Galaxy S4 di ultima generazione e sia su un headset, Meta 2, ai vertici fra i dispositivi in ambito mixed reality, anche se di qualche anno. Fra le motivazioni di questa scelta, l'intenzione di mettere a confronto le reali potenzialità della realtà mista, in termini di resa e di comfort dell'esperienza aumentata rispetto ad un dispositivo attualmente di largo consumo. Come più volte scritto nel primo capitolo della tesi, dispositivi simili Meta 2 prima o poi faranno la loro comparsa nelle sale operatorie degli ospedali di tutto il mondo. Le prime apparizioni di un head mounted display molto simile a questo in sala operatoria sono già avvenute (si veda Hololens, che nel 2018 ha ricevuto l'approvazione della FDA), e Meta 2 si avvicina molto, in termini di resa, ai dispositivi già operativi. D'altra parte, l'utilizzo di un tablet in ambito ospedaliero non costituisce una strada veramente percorribile nella realtà: nel nostro caso il Samsung Galaxy Tab S4 è stato utilizzato fin dalle prime fasi della sperimentazione, per disporre più rapidamente di un riscontro visivo e poter verificare la correttezza degli step della pipeline descritta. Il lavoro di sperimentazione è poi servito per effettuare il deploy su Meta.

### 4.4.2 Tecnologie utilizzate

#### Tecnologie hardware

**Samsung Galaxy S4** Samsung Galaxy Tab S4 è un tablet di Samsung di fascia alta. Fornisce il supporto ad ARCore.



Figura 4.7: L'headset Meta 2, tratto da [1].

### Specifiche riassuntive

- Display 10,5" WQXGA - 1600 x 2560 px.
- CPU octa 2.35 GHz.
- RAM: 4 GB.
- Memoria interna: 64 / 256 GB.
- Batteria 7306 mAh.
- Android 8.1 Oreo.

**Meta 2** Meta 2 è un headset sviluppato dall'azienda omonima specializzata in dispositivi per realtà mista ed aumentata. Fra le caratteristiche più rilevanti di Meta 2 c'è l'ampio FOV (ovvero l'ampiezza di visione), pari a 90 gradi. L'insieme dei sensori a corredo del dispositivo, rende possibile il tracking della posizione dell'utente che lo indossa, oltre a fornirgli la possibilità di afferrare gli ologrammi come se fossero oggetti reali e tanto altro. La risoluzione offerta è 2560 x 1440 dpi e l'headset non troppo pesante rispetto alla concorrenza.



Figura 4.8: L'headset Meta 2, tratto da [1]

### Specifiche riassuntive

- risoluzione: 2550x1440 dpi (frequenza 60Hz)
- largo campo visivo (90 gradi)
- un sistema audio composto da 4 speaker posti vicino all'orecchio
- un camera frontale da 720p
- array di sensori per il tracking della posizione dell'utente.

**Tecnologie software** La scelta di effettuare il deploy su piattaforme hardware condiziona anche l'insieme delle tecnologie software mediante le quali implementare il sistema. Per quanto riguarda la sperimentazione su tablet Android, fra le numerose piattaforme di sviluppo di applicazioni in realtà aumentata (Vuforia, DeepAR, Wikitude e ArkToolkit), era necessaria una che fornisse le seguenti funzionalità:

- riconoscimento automatico di piani su cui posizionare gli oggetti virtuali
- capacità di stimare la posizione del dispositivo, registrando i movimenti dell'utente in real-time
- supporto all'integrazione di oggetti virtuali e reali.

Per via del supporto a tali caratteristiche e la compatibilità con Unity, è stata impiegata la piattaforma Google ARCore.

**ARCoreSDK** ARCore è l'SDK sviluppato da Google per lo sviluppo di applicazioni di realtà aumentata. Parte delle sue API sono compatibili con Android e IOS. Attraverso il riconoscimento, mediante la fotocamera del dispositivo, di specifici punti (chiamati "feature points") e grazie ai suoi sensori, ARCore è in grado di ricostruire dinamicamente l'ambiente. È supportato da una varietà di dispositivi Android, a partire da Nougat (Android 7.0).

**MetaSDK** Meta SDK è la piattaforma di sviluppo messa a disposizione dall'azienda per lo sviluppo con Unity sui propri dispositivi. Sebbene recentemente abbia visto scomparire il suo sito web e la sua documentazione non sia più accessibile, l'SDK include numerose funzionalità, fra le quali SLAM, il riconoscimento e la registrazione dell'interazione delle mani, aspetti di occlusione, riconoscimento dello sguardo etc.

**Lean Touch** Lean Touch è una libreria per la gestione dell'input dell'utente disponibile nell'Asset store di Unity. La sua architettura è modulare e ben si concilia con l'editor di Unity. Oltre al supporto ad un grande numero di gestures, è anche una libreria multi-piattaforma e multi-dispositivo, dato il supporto per Android e iOS e la capacità di calibrare autonomamente la sensibilità al tocco in base al device.

**EvilDICOM** Per gli aspetti di basso livello relativi alla codifica e all'interpretazioni dei campi DICOM è stata utilizzata la libreria evilDICOM. Si tratta di una libreria C# di facile utilizzo per la manipolazione e la lettura dei file DICOM. Il progetto è tuttora attivo e il codice è disponibile con licenza MIT. È stata scelta per la sua semplicità.

### 4.4.3 ARCore

Unity distingue gli ambienti aumentati in cui eseguire l'applicazione in scene. Date le due diverse implementazioni con target device differenti, sono state predisposte scene diverse. I componenti architettonici dell'applicazione saranno modellati sulle astrazioni disponibili in Unity, cioè quelle di gameobject e di script. Per quanto riguarda ARCore, sono state sviluppate 2 diverse scene che, rispettivamente, permettono all'utente di visualizzare il modello 3D e un frame selezionabile dallo stack. L'integrazione delle 2 rappresentazioni sarà invece l'oggetto di un lavoro ulteriore, descritto nel capitolo successivo.

Prima dell'avvio, le scene di ARCore sono costituite dai componenti di base per il funzionamento dell'applicazione, ovvero:

- Arcore App Controller
- Arcore Device
- Plane generator
- Directional light
- Point Cloud

- Event system
- Plane Discovery

, mentre gli oggetti virtuali corrispondenti al modello tridimensionale e alle immagini DICOM verranno istanziati dinamicamente. La maggior parte di questi gameobject ha lo scopo di gestire le funzionalità connesse ad ARCore, come ad esempio il riconoscimento dinamico dei piani o le condizioni di illuminazione. Molti di questi componenti sono dei prefab, ossia dei template composti dal game object, dagli script che definiscono il suo comportamento e dai riferimenti agli altri componenti della scena con cui interagiscono. Questa caratteristica offerta da Unity consente di riutilizzare direttamente parte dei componenti in scene diverse.

**Arcore Device** Contiene al proprio interno il game object first person camera, il quale utilizza la camera del posteriore del dispositivo per catturare le immagini del mondo reale e combinare il flusso degli oggetti virtuali con quello reale.

**Plane Discovery** Questo componente risponde alla specifica necessità, evidenziata nell'analisi dei requisiti, di riconoscere i piani sui quali disporre gli oggetti virtuali, al fine di costruire un piano di lavoro aumentato. Fornisce un supporto visuale all'utente, guidandolo nel caso in cui il dispositivo abbia difficoltà a rilevare i piani per problemi di luminosità o assenza di superficie piate. La scoperta di piani avviene dinamicamente e può richiedere tempo. In questo caso, l'unico script di questo componente (Plane Discovery Guide) mostra a video una mano che istruisce su come agevolare il riconoscimento, oltre ad una serie di indicazioni testuali.

**Environmental light** Si fa carico di rilevare ed correggere l'illuminazione dell'ambiente a partire dall'analisi del valore dei pixel del flusso catturato dalla fotocamera del dispositivo. Gli oggetti virtuali appariranno più luminosi in ambienti con maggiore illuminazione.

**ARAppController** Questo game object contiene gli script relativi a:

- gestione del ciclo di vita dell'applicazione Android. Questo comportamento è eseguito dallo script AppController.
- logica di istanziamento dell'oggetto virtuale, gestita dallo script PawningLogic. I requisiti sono chiari nel definire come l'utente debba poter scegliere il punto in cui veder comparire il modello, a patto che questo si

trovi su un piano. Nel caso del deploy su tablet, l'intenzione dell'utente è rilevata mediante il suo tocco sullo schermo. Per questo motivo, questo script conserva i riferimenti alla first camera contenuta dentro al gameobject `ArcoreDevice`. `PawningLogic` è quindi lo script responsabile di far apparire gli oggetti virtuali sul piano visualizzato da Arcore scelto dall'utente. Oltre a posizionare l'oggetto, lo script lo vincola ad un'ancora, ovvero uno strumento offerto da Arcore per stabilizzare la sua posizione ed orientazione ed impedire che esso si muova rispetto al punto di posizionamento iniziale. Senza questo accorgimento, tale inconveniente potrebbe quasi sicuramente capitare, perché il riconoscimento dinamico dei piani può indurre dei cambiamenti e correggere la maniera in cui Arcore aveva inizialmente interpretato la conformazione dell'ambiente. Altrimenti, l'utente potrebbe ritornare su un oggetto dopo aver cambiato visuale e non ritrovarlo più. Questo script è l'unico che interagisce con l'entry point del set di classi descritto per l'estrazione e la elaborazione delle immagini DICOM, ossia il loader `IFramesLoader`. All'avvio dell'applicazione, quando viene invocato il metodo `Start` di `PawningLogic`, questo script istanzia il loader ed invoca il metodo `LoadFrames()` che ritorna la lista di `DICOMFrames` estratti dai dataset. In questo caso l'applicazione non necessita che le slices rispettino l'orientazione e la posizione memorizzate nel file e quindi il loader impiegato è il `NonOrientedFramesLoader`.

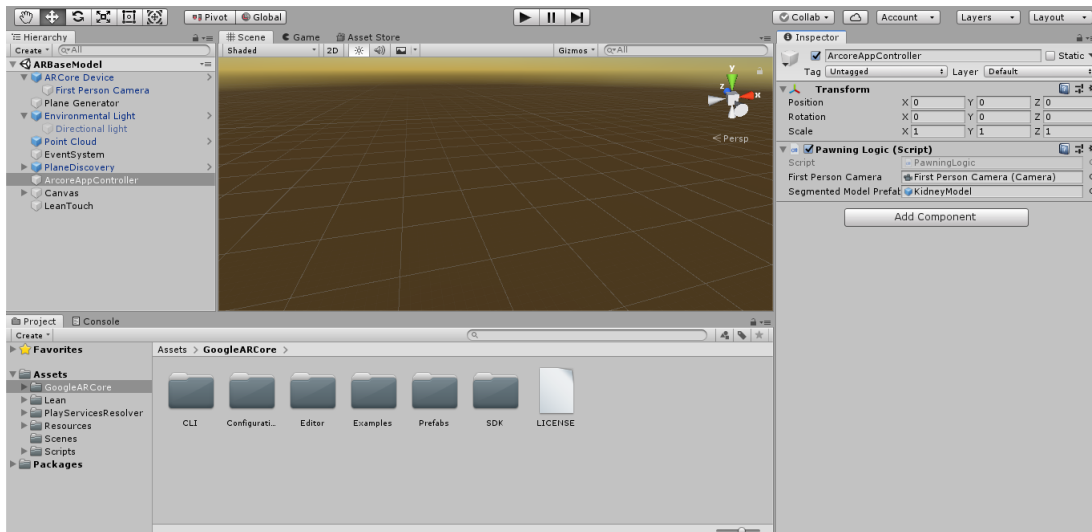


Figura 4.9: L'editor della scena unity relativa ad Arcore. A sinistra, nella "Hierarchy window" sono elencati i componenti della scena descritti. A destra, nell'"inspector", si trovano gli script, nello specifico sono visualizzati quelli del gameobject `ArcoreAppController` e i rispettivi campi.



**FramesContainer e lo script FramesController** La seconda delle scene di Arcore contiene il modello ottenuto da segmentazione, il quale non possiede una sua controparte fra gli script. È costituito da un prefab che viene istanziato nel momento in cui l'utente fa tap sullo schermo. Il prefab incorpora la mesh importata attraverso i procedimenti descritti e contiene le informazioni sul colore nei materiali delle facce. Al contrario, la gestione delle singole slice è ad opera dello script nel gameobject FramesContainer, FramesController. L'analisi dei requisiti ha messo in evidenza la importanza per l'utente di poter visualizzare l'immagine di interesse. Questo script si occupa di inferire quale immagine è stata scelta dall'utente e della sua visualizzazione. Il compito di caricarle è affidato per delegazione all'IDICOMLoader di cui si compone. La tipologia dei frame restituiti è a discrezione del FramesController, ma per come è stata definita la gerarchia dei Loader, per passare da piani acquisiti a piani non acquisiti basterà in futuro sostituire unicamente il loader scelto (vedi cap. 5). Con questa divisione dei compiti, FramesController ha l'unico compito di rispondere alla notifica della sorgente Slider e selezionare il frame giusto nell'elenco caricato dal loader.

Lo script è il componente di un prefab (FramesContainer), il quale viene istanziato al momento del tap sullo schermo. Questo meccanismo di istanziazione dinamica dei componenti (in forma di template, cioè prefab) fornisce la possibilità all'utente di selezionare il punto esatto in cui posare gli oggetti e sarà impiegato anche per le implementazioni descritte nel capitolo successivo.

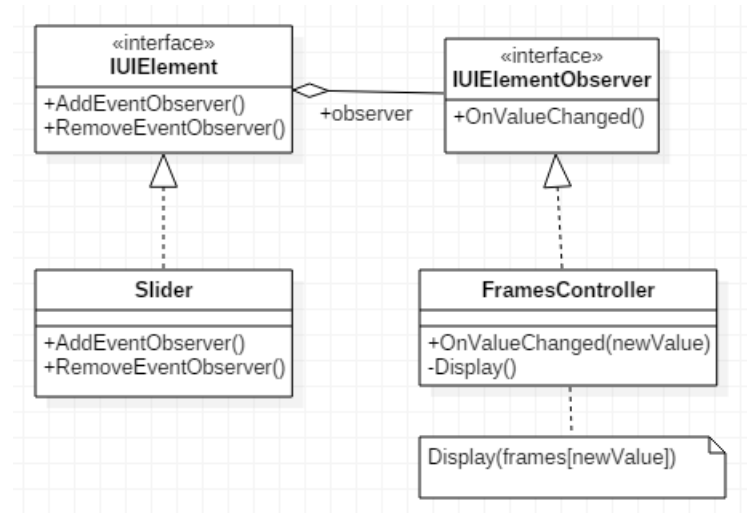


Figura 4.10: Il pattern observer applicato per rilevare l'input dell'utente

**Canvas e Slider** Nella scena Arcore per la visualizzazione delle immagini, l'input dell'utente, che secondo i requisiti deve poter scegliere un frame arbi-

trario da visualizzare, è catturato tramite uno slider. Uno slider è un controllo che consente all'utente di selezionare un valore numero all'interno di un intervallo predefinito. In Unity, lo slider, come gli altri componenti UI, deve essere figlio di un componente chiamato Canvas, che costituisce una finestra invisibile nella scena. A livello concettuale (e in Unity è così), uno slider è modellabile come una sorgente di eventi. Nel caso specifico, la sua collaborazione con la logica di controllo dei frame (FramesController) è descritta attraverso un'interazione di tipo publish-subscribe, mediante il pattern Observer. Dato che il comportamento dell'applicazione è influenzato dal valore dello slider, che memorizza l'indice del frame da visualizzare all'interno del set di immagini ordinato estratto dai dataset DICOM, FramesController costituirà l'osservatore di tali cambiamenti e reagirà di conseguenza. Allo slider non serve conoscere il destinatario della sua notifica di cambio di stato, ma vale il viceversa. Nel sistema allo stato attuale l'osservatore sarà solo uno, ma ciò non vieta che in futuro possano essere di più.

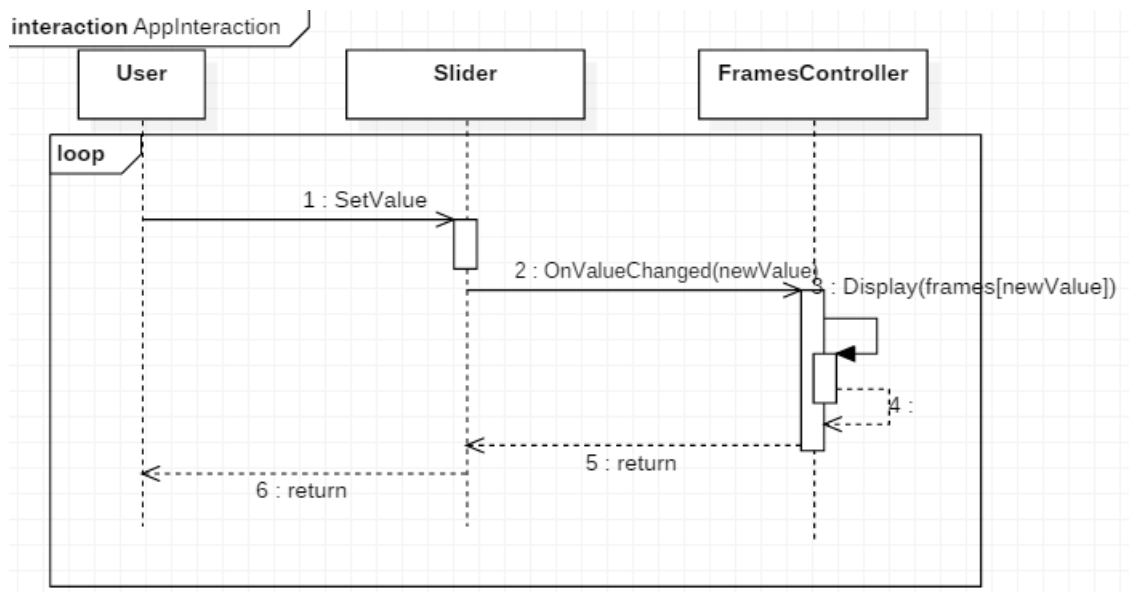


Figura 4.11: La gestione dell'input sul componente slider rappresentato mediante uno schema UML di sequenza.

### Gestione dell'input

**LeanTouch** Lean touch è il componente della scena responsabile, tramite il suo script, di convertire automaticamente ed elaborare l'insieme degli input generati dal tocco sul dispositivo.

La gestione dell'input è affidata ai seguenti script:

- `LeanDragTranslate`: permette di traslare l'oggetto selezionato rispetto alla telecamera mediante la drag gesture.
  
- `LeanTwistRotate`: consente di ruotare l'oggetto rispetto a qualunque degli assi attraverso twist gesture.
  
- `LeanPinchScale`:

Questi script devono essere assegnati ai game object che si intende spostare, ruotare o zoomare.

La loro combinazione fornisce il supporto per manipolare gli oggetti virtuali attraverso le più comuni gestures. In aggiunta, è richiesta la presenza nella scena di un altro gameobject che contenga lo script `LeanTouch`.

**Meta scene** Il nucleo di gestione ed estrazione delle informazioni dal file DICOM, relative alla decodifica dei pixel e alla pipeline di elaborazione, è il medesimo della scena precedente, tuttavia l'SDK impiegato è diverso, non più Arcore, ma Meta 2, poiché cambia il dispositivo di destinazione e quindi la modalità di interazione fra l'utente e i componenti della scena. All'utilizzo dell'input su schermo si sostituisce quello delle sole mani e il compito di rilevare l'intenzione dell'utente deriva dall'interpretazione dei suoi gesti.

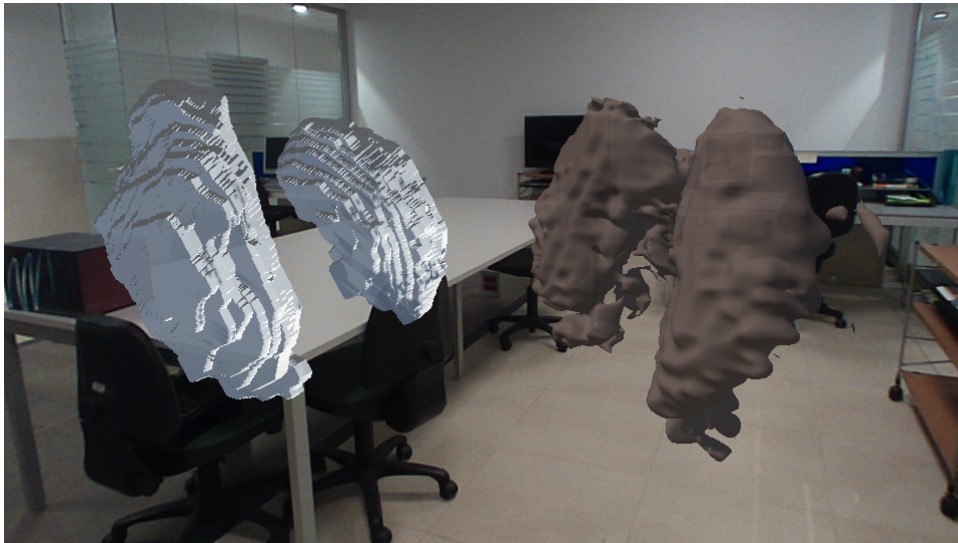


Figura 4.12: Confronto fra i 2 oggetti tridimensionali estratti dal dato DICOM relativo al distretto renale, per come visualizzati attraverso l’headset Meta 2. A sinistra, il modello fornitoci dai nostri collaboratori di ing. biomedica, a destra quello ottenuto autonomamente con il software Slicer. Anche se quest’ultimo presenta maggiore rumore, che può essere fortemente ridotto attraverso una rifinitura maggiore all’atto della segmentazione, come descritto nella sezione apposita di questo capitolo, la sua risoluzione è molto migliore.

**Prefab per la manipolazione** A differenza delle scene di Arcore, per la gestione dell’input dell’utente è stato utilizzato direttamente l’SDK di Meta 2. Per questo compito, i componenti fondamentali sono i prefab `MetaCameraRig` e `MetaHands`. `MetaCameraRig` contiene la logica di inizializzazione dell’ambiente, ed impiega i sensori del dispositivo per localizzare l’utente nello spazio. `MetaHands` gestisce l’interazione fra il dispositivo e le mani dell’utente.

Grazie ad essi, per manipolare gli oggetti (game object) di interesse è necessario aggiungervi i seguenti script:

- `GrabInteraction`: consente all’utente di afferrare un oggetto virtuale e spostarlo.
- `TwoHandGrabRotateInteraction`: mediante entrambe le mani, permette di ruotare l’oggetto.
- `TwoHandGrabScaleInteraction`: permette di zoomare o rimpicciolire l’oggetto con 2 mani.

Tutti e 3 gli script derivano dalla classe `Interaction`, e memorizzano le informazioni sulle mani che toccano l’oggetto sul quale sono applicati gli script.

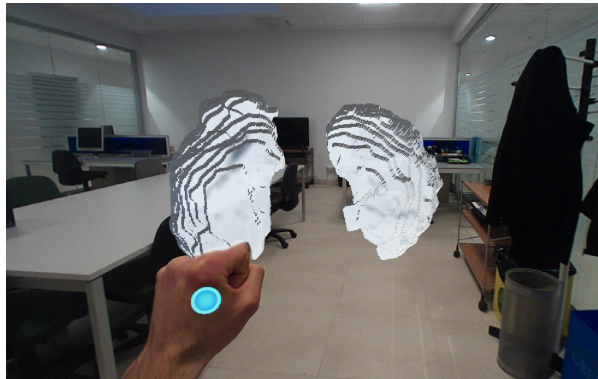


Figura 4.13: Visualizzazione del modello virtuale con headset Meta 2. Il marker sulla mano mostra la rilevazione da parte dello script GrabInteraction dell'intenzione dell'utente di afferrare l'oggetto.



# Capitolo 5

## Estensioni alle funzionalità di base

Come descritto nel capitolo precedente, il contributo principale della tesi è stato quello di realizzare un sistema per la visualizzazione delle immagini DICOM, implementando un primo prototipo di base che ha lo scopo di produrre una rappresentazione efficace per l'utente che lo utilizza. Il prototipo predispone una modalità di visualizzazione alternativa rispetto a quelle tradizionali e permette di osservare e studiare la ricostruzione virtuale attraverso la nuova prospettiva della MR. Dopo aver individuato un insieme di funzionalità di base, generalizzabili il più possibile rispetto alla particolare specializzazione dell'utente medico, lo step successivo è stato quello di individuare un insieme di estensioni in grado di supportare attività più specifiche e più vicine alle necessità di una particolare specializzazione medica. Un esempio di tali funzionalità, la cui importanza è stata testimoniata anche dal medico che abbiamo coinvolto (vedi cap. 6), è costituito dall'integrazione della rappresentazione morfologica bidimensionale estratta da un frame DICOM a quella tridimensionale finora discussa.

### 5.1 Descrizione

L'obiettivo di questa parte di tesi è quello di coniugare in MR il dato morfologico originale DICOM bidimensionale con il dato ottenuto da segmentazione, e sviluppare una seconda versione del prototipo per la loro manipolazione. Il progetto non mira semplicemente ad ottenere due rappresentazioni separate dei due dati all'interno dell'universo aumentato, bensì a coniugarle, dal momento che l'utilità dell'applicazione deriva dalla loro sovrapposizione nello stesso sistema di riferimento spaziale. L'idea, inizialmente proposta dai collaboratori di ing. biomedica, estende l'ambito di applicazione del progetto anche ad un

contesto extraospedaliero, abbracciando l'ambito delle scienze biomediche e biotecnologiche, costituendo un supporto di validazione al processo di produzione dei modelli tridimensionali ottenuti a partire da immagini diagnostiche (anatomia computazionale) e, nello specifico, alle tecniche di segmentazione impiegate nel processo. L'utilità di questa modalità di visualizzazione deriva dalla possibilità di confrontare direttamente i modelli con il dato anatomico estratto dal DICOM, il quale, non essendo stato sottoposto ad alcuna elaborazione, fornisce una solida base di raffronto per individuare eventuali errori del processo.

## 5.2 Analisi

### 5.2.1 Nuove funzionalità

Rispetto al prototipo base, sono stati riscontrati nuovi requisiti, che non costituiscono un super-set di quelli trattati precedentemente. L'obiettivo di lavori futuri sarà quello di estendere il grado di manipolazione disponibile nel sistema di base anche a questa rappresentazione.

- L'ambiente aumentato deve predisporre all'utente la possibilità di visualizzare contestualmente un frame DICOM fra quelli "acquisiti" da lui scelto e l'oggetto tridimensionale estratto da essi tramite segmentazione.
- I due oggetti virtuali sovrapposti saranno registrati spazialmente fra di loro all'interno del mondo aumentato, ovvero ciascuna slice DICOM deve essere sovrapposta esattamente alla superficie (interna) del volume che rappresenta, rispettandone l'inclinazione e la dimensione. A tal proposito è richiesto un grado di precisione significativo in quanto eventuali leggere approssimazioni comprometterebbero inevitabilmente il risultato.
- Deve essere fornita all'utente la possibilità di selezionare dinamicamente una slice DICOM all'interno del dataset e di verificare visivamente la sua collocazione nello spazio rispetto al modello tridimensionale.
- È importante che l'utente possa scegliere dove posizionare gli oggetti virtuali, poiché l'utilità dell'applicazione deriva anche dalla eliminazione del vincolo fisico fra l'oggetto da visualizzare e il mezzo di visualizzazione (si pensi ai monitor che costringono il medico a stazionare di fronte ad esso). L'applicazione dovrà riconoscere dinamicamente il piano deputato ad ospitare gli oggetti virtuali.
- Durante gli incontri è avanzata anche l'importanza di estendere in futuro la proiezione anche alle slice "non acquisite", ovvero alle ricostruzioni



(generate a posteriori) di piani anatomici non corrispondenti al piano di acquisizione. Sviluppi futuri (vedi cap.6) andranno nella direzione di ottenere e rappresentare dapprima i 3 piani ortogonali e, in un secondo momento, di fornire la possibilità all'utente di orientare dinamicamente il piano di intersezione rispetto al modello segmentato nello spazio. Nonostante queste funzionalità non siano richieste nel prototipo, è necessario predisporre l'architettura software verso questa direzione sin da subito, poiché saranno probabilmente oggetto di future implementazioni.

### 5.2.2 Problemi principali

Rispetto alla versione precedente del sistema, la realizzazione di queste specifiche risulta più problematica, poiché in più implica non solo la necessità di interpretare i riferimenti spaziali per ciascun frame, ma anche e soprattutto la questione dell'allineamento delle due rappresentazioni nello spazio. Infatti:

- Dal momento che i 2 oggetti virtuali devono coesistere è necessario che essi siano posizionati ed orientati nello stesso sistema di riferimento. Sarà quindi indispensabile dapprima comprendere le convenzioni attraverso le quali DICOM attribuisce i riferimenti spaziali a ciascun frame, poi acquisire quelle relative al modello segmentato ed infine convertire uno spazio di coordinate nell'altro mediante l'applicazione di trasformazione geometriche. Queste richiede la predisposizione a strutturare i dati secondo specifiche convenzioni, elencate nella sezione successiva.

Queste considerazioni pongono i seguenti, principali, problemi, la cui risoluzione è affrontata nella sezione sviluppo.

- Collocazione di ciascuna slice nello spazio
- Conversione fra diversi formalismi di rotazione

**Collocazione di ciascun frame nello spazio** L'interpretazione dei requisiti di questo secondo prototipo evidenzia come sia necessario collocare ciascuna slice nella sua posizione rispetto all'RCS. Nel caso precedente, si è proceduto ad individuare la matrice di trasformazione necessaria a registrare la mesh sul sistema di riferimento del paziente a partire unicamente dal primo frame, poiché, dato un vertice della mesh, bastava recuperare la sua posizione nell'RCS ed ottenere le informazioni per tutti gli altri. Nel caso in cui la mesh fosse già orientata correttamente (ad es. quella prodotta da Slicer lo era) questa operazione di conversione non era neppure necessaria. In questo specifico scenario, invece, è necessario leggere e decodificare tutti i file DICOM ed applicare la trasformazione alle coordinate definite al volume costituito dalla loro sovrapposizione.

**Conversione fra diversi formalismi di rotazione** In matematica, e in particolare in geometria, una rotazione è una trasformazione del piano o dello spazio euclideo che sposta gli oggetti in modo rigido e che lascia fisso almeno un punto, nel caso del piano, o una retta, nel caso dello spazio. I punti che restano fissi nella trasformazione formano più in generale un sottospazio: quando questo insieme è un punto o una retta, si chiama rispettivamente il centro e l'asse della rotazione [1]. Più precisamente, una rotazione è una isometria di uno spazio euclideo che ne preserva l'orientazione. Qualunque sia il numero delle dimensioni dello spazio di rotazione, gli elementi della rotazione sono:

- il verso (orario-antiorario);
- l'ampiezza dell'angolo di rotazione;
- il centro di rotazione (il punto attorno a cui avviene il movimento rotatorio).

Come spiegato nel capitolo precedente, le trasformazioni spaziali necessarie a collocare ogni frame nel sistema di riferimento del paziente, sono ricavabili dal file DICOM nella forma di una matrice affine. Nello specifico la rotazione si ottiene sulla base di 2 vettori che specificano la direzione della prima righe e delle colonne dell'immagine. Da queste 2 informazioni si ottiene una matrice di rotazione. Tuttavia, esiste più di una convenzione per esprimere la rotazione in uno spazio tridimensionale. Unity permette di specificarla mediante la notazione di Eulero e i quaternioni, per questo è necessaria una conversione.

**Angoli di Eulero** Gli angoli di Eulero descrivono la posizione di un sistema di riferimento XYZ solidale con un corpo rigido attraverso una serie di rotazioni a partire da un sistema di riferimento fisso xyz. I due sistemi di riferimento coincidono nell'origine. Nel caso in cui i piani xy e XY siano distinti, si intersecano in una retta (passante per l'origine), detta linea dei nodi (N). Se i piani coincidono, si definisce la linea dei nodi N come l'asse X. Gli angoli di Eulero sono i seguenti:

- $\alpha$  : è l'angolo tra l'asse x e la linea dei nodi (è definito "angolo di precessione");
- $\beta$ : è l'angolo tra gli assi z e Z, è detto "angolo di nutazione";
- $\gamma$ : è l'angolo tra l'asse X e la linea dei nodi ("angolo di rotazione propria").

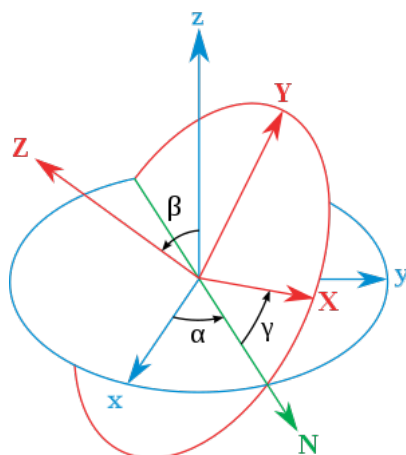


Figura 5.1: Gli angoli di Eulero. Il sistema fisso ( $xyz$ ) è rappresentato in blu, mentre quello ruotato ( $XYZ$ ) in rosso. La linea dei nodi, indicata con  $N$ , è in verde, tratto da [1]

Le rotazioni descritte dalla notazione euleriana possono distinguersi in :

- intrinseche: sono definite rispetto ad un sistema di riferimento  $XYZ$  solidale con un corpo in movimento. Per questo variano la loro orientazione ad ogni rotazione elementare
- estrinseche: sono definite rispetto ad un sistema di riferimento  $XYZ$  fisso.

Prima di una qualsiasi trasformazione elementare, le 2 rappresentazione sono comunque equivalenti. La stessa rotazione che mediante la notazione matriciale è definita da più entrate, con gli angoli di Eulero è definita da soli 3 valori. La maniera in cui gli angoli di rotazione si riferiscono agli assi determina una terna. Il numero di terne è dunque dato dalla combinazione degli assi.

**Conversione ad angoli di Eulero** Rispetto al problema inverso, la conversione da matrice di rotazione ad angoli di Eulero si rivela complesso, poiché non generalizzabile. Il metodo utilizzato di solito prevede l'analisi preliminare della specifica matrice di rotazione da convertire, la quale si costituisce di prodotti seni e coseni, e la ricerca delle relazioni fra gli angoli. L'approccio risolutivo dipende dalla scelta della terna di destinazione. Ad esempio, nel caso di una terna in notazione intrinseca  $z$ - $y$ - $z$ , a partire da una matrice di rotazione del tipo:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

Le relazioni che esprimono gli angoli di Eulero corrispondenti sono:

---

```

 $\beta = \text{atan2}(\sqrt{R_{31}^2 + R_{32}^2}, R_{33})$ 
if  $\sin \beta \neq 0$  then
     $\gamma = \text{atan2}(R_{32}, -R_{31})$ 
     $\alpha = \text{atan2}(R_{23}, R_{13})$ 
else
     $\gamma = 0$ 
     $\alpha = \text{atan2}(R_{21}, R_{11})$ 
end if

```

---

**Quaternioni** Gli angoli di Eulero sono affetti dal blocco cardanico, ovvero un problema che causa la perdita di uno degli assi di libertà nello spazio tridimensionale, dovuto all'allineamento di due assi rotanti verso la stessa direzione; ciò vincola la rotazione a uno spazio solamente bidimensionale. A questo proposito sarebbe opportuno impiegare in futuro i quaternioni, una notazione conveniente che non presenta il blocco cardanico.

### 5.3 Progettazione e Sviluppo

L'architettura software sulla quale si basa la nuova versione del prototipo è in gran parte condivisa con quella presentata in precedenza, poiché la fase di progettazione del primo prototipo aveva previsto cambiamenti in tale direzione. Come descritto nell'analisi, la novità sostanziale nei requisiti rispetto alla versione base è costituita dal supporto all'orientazione dei frame nello spazio. Ora ciascun frame deve essere consapevole della sua esatta collocazione nel sistema di riferimento del paziente, mentre nel prototipo base la sua collocazione era arbitraria e a scelta dell'utente. Questo cambiamento si riflette nell'introduzione nell'architettura di un componente deputato all'estrazione di questi riferimenti: `SpatialInfoExtractor`.

**SpatialInfoExtractor** Attraverso le classi che incapsulano l'effettiva libreria utilizzata, questo componente computa la trasformazione geometrica come aggregazione di posizione, rotazione e scala, e la rende disponibile come struttura dati aggregata, modellata attraverso la classe `Trasformation3D`. Il suo utilizzo è alla base dell'impiego del pattern template method applicato alla gerarchia dei `IFramesLoader`. Infatti, le due implementazioni del loader condividono un parte di codice del metodo pubblico `LoadFrames`, che ereditano

entrambi da `AcquisitionFramesLoader` e costituisce la parte invariante di interazione sui diversi file all'interno della cartella DICOM, ma si distinguono per la maniera in cui rispondono al metodo `LoadFrame`, che assegna la posizione, la rotazione e le dimensioni al `DICOMFrame` da rendere disponibile all'applicazione. Nel caso impiegato in questo prototipo, il loader delega la responsabilità di determinare la posizione al `PixelExtractor`, dal momento che è necessario vincolare il frame al sistema di riferimento del paziente. Nel prototipo base, invece, il frame era posizionato liberamente, nel punto in cui voleva l'utente.

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class OrientedFramesLoader : AcquisitionFramesLoader
{
    protected SpatialInfoExtractor sie;

    public OrientedFramesLoader(string directoryPath) :
        base(directoryPath)
    {
    }

    protected override DICOMFrame LoadFrame(string filename)
    {
        this.sie = new SpatialInfoExtractor(new
            DICOMDataset(filename));
        return new DICOMFrame(pp.GetFrameAsTexture(),
            sie.GetTransformation());
    }
}
```

Listato 5.1: Codice C# di `OrientedFramesLoader`, ovvero della classe responsabile del caricamento dei frame e della loro collocazione nel sistema di riferimento del paziente

**IFramesLoader, strategy pattern e template method** Questa modellazione tramite template method rende possibile applicare anche il pattern Strategy al fine di predisporre all'utilizzatore due diverse implementazioni di `IFrameLoader`, perfettamente intercambiabili, ed indipendenti dall'utilizzatore. Nel caso di questa applicazione, l'utilizzatore è il `FramesController`. La

sua applicazione evita quindi l'utilizzo di blocchi condizionali e un controllo centralizzato in un'unica classe.

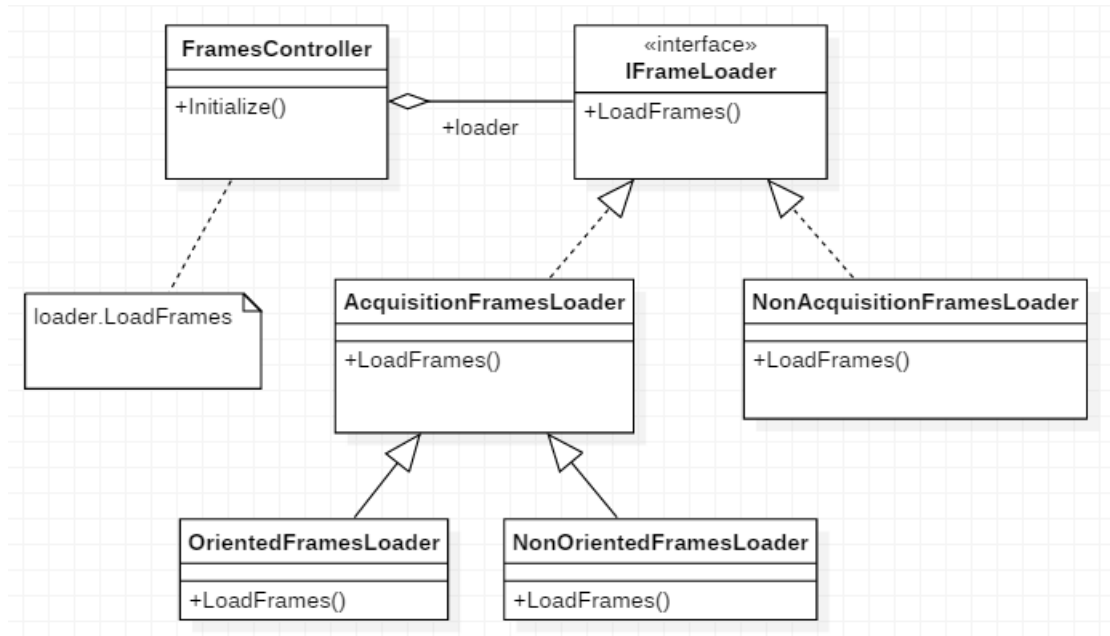


Figura 5.2: Strategy pattern in collaborazione con template method

**ARcore scene** Data la loro modularità, alcuni prefab, script e gameobject provengono dalla versione precedente del prototipo.

Come nelle implementazioni già descritte, il funzionamento generale della scena è gestito dal gameobject ARAppController, che presenta uno script responsabile di gestire la logica con la quale vengono istanziati i due oggetti virtuali. Contemporaneamente, l'altro script di questo gameobject amministra il ciclo di vita dell'applicazione.

**Game object FramesContainer** Dal punto di vista dei componenti della scena, la necessità di integrare le due rappresentazioni si riflette sulla modellazione del componente FramesController. Il suo compito nella versione base era semplicemente quello di visualizzare il frame scelto dall'utente, osservandone il comportamento, ossia la sua intenzione di cambiare frame (Observer pattern). Se prima i due oggetti virtuali erano visualizzati in scene diverse, ora la loro integrazione fa sì che questo componente debba monitorare la posizione del modello, per poter registrare i frame di conseguenza, cambiando solamente la texture visualizzata, ma aggiornando la posizione.

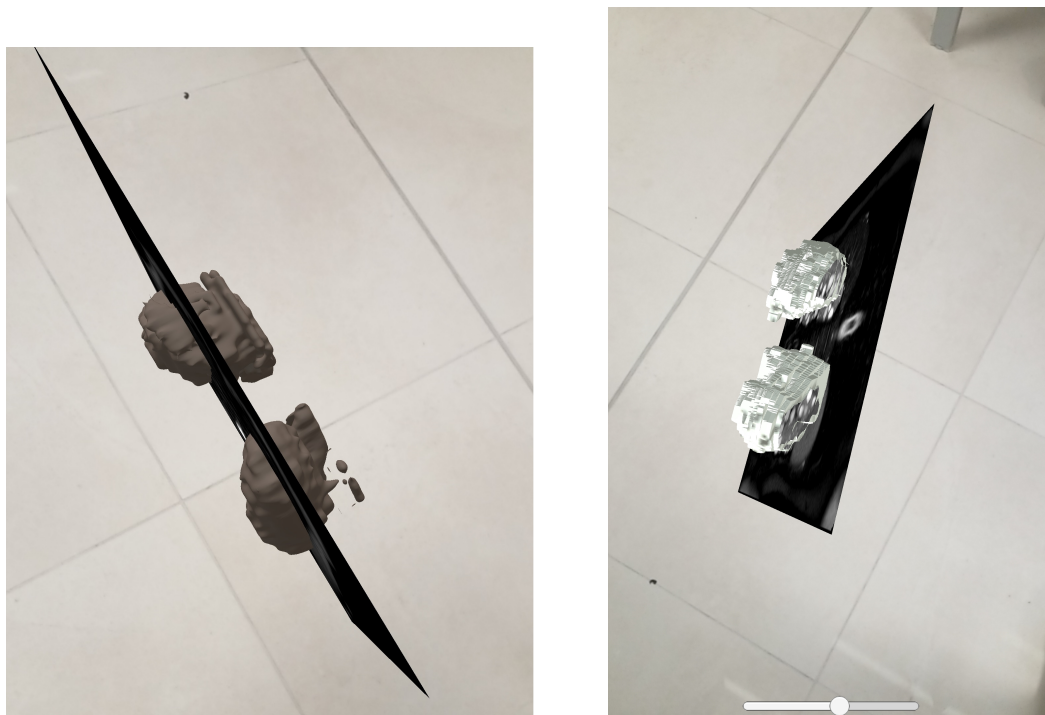


Figura 5.3: L'intersezione fra il modello tridimensionale e la slice DICOM bidimensionale. A sinistra il risultato ottenuto a partire dalla mesh prodotta con Slicer, a destra con quella fornita dai nostri collaboratori

Anche in questo caso la gestione dell'input è a carico di Lean Touch, che contiene la logica di traslazione del modello nell'ambiente aumentato. Quando esso si sposta, interviene il componente FramesController che ricolloca i frame mantenendo le rappresentazioni allineate.





# Capitolo 6

## Validazione e sviluppi futuri

### 6.1 Validazione

Essendo il risultato di una collaborazione con il team di ingegneri biomedici più volte citato, il sistema è stato sviluppato con il loro costante supporto, sin dalle fasi iniziali. Attraverso numerosi incontri, sono stati dapprima concordati gli obiettivi in termini di funzionalità da realizzare ed in seguito valutati i risultati, attraverso un processo iterativo che ha portato a rivedere le aspettative iniziali, riadattandole agli sviluppi intermedi. Ciò ha permesso di distinguere un iniziale insieme base di funzionalità da predisporre all'utente, che poi è stato successivamente rivisto ed ampliato, ed ha posto le basi per le osservazioni del capitolo 4. Al fine di validare il prototipo e di raccogliere opinioni sul percorso intrapreso, abbiamo coinvolto anche un medico specialista, che ha testato sia le funzionalità di base del sistema, sia le estensioni proposte al capitolo 4.

Conosciuto grazie alla gentile intermediazione dei professori di ing. biomedica, la sua opinione è stata subito molto chiara e molto positiva in merito all'utilità dell'impiego della mixed reality al contesto ospedaliero ed ha fornito importanti indicazioni in merito all'applicazione di un sistema del genere alle branche specialistiche di diagnosi e terapia cardiologica. Il dott. ha fornito opinioni molto confrontanti sulle potenzialità di sistemi software di questo tipo, quasi come se gli sviluppi in questa direzione siano già attesi da tempo. Il suo interesse in questo ambito è motivato dall'importanza diagnostica e terapeutica di poter visualizzare in 3 dimensioni l'organo esaminato (nello specifico: il cuore); questa modalità di visualizzazione è stata individuata come molto più "indicativa" rispetto a quella tradizionale, data dalla consultazione di referti ed immagini bidimensionali. Riferendosi alla sua esperienza, in ambito cardiaco, è stato sottolineato come i raggi non vengano più utilizzati negli ultimi anni, a favore di sistemi, detti "di mappaggio", il cui compito è quello di ricostruire tridimensionalmente gli organi sotto esame sulla base di parametri, fra i quali segnali

elettrici registrati da appositi elettro-cateteri. Riferendosi alla sua esperienza, nel caso di procedure mininvasive come l'ablazione trans-catetere, i sistemi attualmente a sua disposizione forniscono già rappresentazioni tridimensionali a partire da un'acquisizione real-time di immagini del paziente, anche se proiettate su superfici piatte come monitor. La responsabilità di cambiare il punto di vista attraverso il quale osservare i modelli generati è attualmente a carico dell'ingegnere che risiede alla console, al fine di permettere ai medici di mantenere le proprie mani e la propria strumentazione sterilizzate per motivi di igiene intra-operatoria. La possibilità di manipolare degli ologrammi in maniera autonoma (senza rischio di contaminare la propria strumentazione) permetterebbe di rinunciare a quella intermediazione, consentendo al medico di agire in prima persona nella visualizzazione delle ricostruzioni. Rispetto alle funzionalità descritte nel capitolo 4, il medico ha inoltre accolto positivamente la possibilità di integrare il dato DICOM bidimensionale e la rappresentazione tridimensionale, individuandola come uno strumento molto utile a fini terapeutici. A suo parere, alle diverse modalità di visualizzazione del modello corrispondono infatti diverse finalità: la manipolazione con le proprie mani del modello ha un'importanza soprattutto diagnostica, mentre la possibilità di posizionarlo in punto fisso e poterci girare attorno come se fosse un oggetto reale ha una maggiore rilevanza dal punto di vista terapeutico.

Sono stati discussi anche gli eventuali problemi causati dall'impiego della MR in ambito medico, allo stadio prematuro in cui si trova attualmente, e quanto questi inciderebbero sulla diagnosi e sulle attuali modalità di lavoro. Come descritto nel capitolo 1, fra questi, una questione rilevante è costituita dall'eventuale disallineamento della rappresentazione tridimensionale rispetto alla reale anatomia della struttura che rappresenta, dovuto sia alla scarsa risoluzione disponibile, come ad eventuali errori nella segmentazione, sia ad effettivi cambiamenti nella fisiologia del paziente intercorsi dalla data di acquisizione delle immagini a quella visualizzazione. A parere del professionista, tali problematiche sarebbero secondarie rispetto ai potenziali benefici, poiché i parametri presi in considerazione prima di effettuare procedure chirurgiche anche solo mini invasive (in questo caso cardiache) sono numerose e di natura differente e non esclusivamente visuali (impedenza e voltaggio cardiaci). In questo senso, leggere imprecisioni nel modello sarebbero dunque tollerate, poiché compensate da un numero insieme di fattori.

In merito alle eventuali difficoltà del personale medico a sostituire le metodiche tradizionali in favore di una tecnologia così innovativa come la MR, è emerso come questo non sia un ostacolo così importante, poiché paragonabile ad altre transizioni tecnologiche affrontate nel passato ed oggi pienamente realizzate ed operative. A questo proposito si è fatto riferimento alla fruttuosa introduzione dei sistemi di mappaggio nella pratica microchirurgica cardiaca,

dapprima visti come estranei ed ora invece indispensabili: in quest'ottica la MR dovrebbe rappresentare, a parere del professionista, lo step successivo.

## 6.2 Problematiche

Il prototipo presenta qualche problematica:

- L'utilizzo della versione per Android, che sfrutta l'SDK di ARCore, necessita che l'ambiente fisico nel quale si testa il prototipo rispetti determinate condizioni strutturali e di illuminazione. In presenza di superfici irregolari o chiare, scarsa illuminazione etc., il riconoscimento dei piani può essere tardivo. Questa caratteristica non è facilmente modificabile, perché è direttamente imputabile ad ARCore.
- Per quanto riguarda la versione estesa per Android, quella che predispose all'utente la visualizzazione integrata del modello e dei piani che lo intersecano, nel caso in cui il piano riconosciuto sia di piccole dimensioni si verifica un disallineamento fra le due rappresentazioni. Ciò non accade se gli stessi script sono eseguiti in locale su un pc desktop o nel caso in cui vengano piazzati nella scena prima dell'esecuzione. Sarà necessario indagare i meccanismi con cui Arcore concilia l'intenzione di istanziare un oggetto in un punto specifico con il requisito che quello sia un piano riconosciuto e con l'evoluzione della posizione dell'oggetto data dal riconoscimento dinamico dell'ambiente.

## 6.3 Lavori futuri

L'obiettivo della tesi è stato quello di ottenere una controparte virtuale di DICOM all'interno della realtà aumentata e mista, a partire dalla constatazione che le effettive implementazioni di sistemi di questo tipo attualmente disponibili siano poche e riservate quasi esclusivamente a software proprietari. Una serie di aspetti sicuramente significativi, relativi fra gli altri all'automatizzazione del processo software descritto, è stata messa in secondo piano e deve essere sicuramente affrontata in lavori futuri. A queste considerazioni si deve affiancare certamente un'attività di ricerca volta ad individuare nuove funzionalità, poiché le effettive potenzialità di MR nel settore medico sono ancora da scoprire nella loro completezza e la loro individuazione deve necessariamente passare dal confronto con professionisti del settore, come avvenuto per lo sviluppo di questo progetto. Gli aspetti da considerare in futuro sono numerosi. Anche sulla base del confronto con i prof. di ingegneria biomedica e con lo specialista, sono stati riconosciuti come principali quelli elencati di seguito.

**Automatizzazione della pipeline tecnologica** Non è stato un obiettivo primario del progetto, ma sarà comunque necessario in futuro rendere automatico e programmabile l'insieme degli step necessari al raggiungimento delle specifiche del sistema. Per quanto riguarda l'importazione del dato morfologico bidimensionale DICOM, il prototipo è già predisposto al caricamento automatico a tempo di esecuzione, mediante i componenti software discussi nel capitolo 4. L'aspetto invece tuttora non completamente automatizzato riguarda l'estrazione e il caricamento della mesh in Unity. Sebbene il processo di segmentazione rappresenti l'impedimento più importante in tal senso, è comunque possibile inglobare nel nucleo attualmente automatico (cioè la cui gestione avviene tramite codice `c# Unity`):

- sia la procedura di conversione della mesh dai dati raw costituiti da vertici e facce ad un formato compatibile con Unity,
- sia l'importazione a run-time della stessa all'interno dell'esperienza aumentata.

Questo sarebbe importante al fine di sottoporre all'utente (a tempo di esecuzione) la scelta di quale stack di acquisizione DICOM e quale modello visualizzare, che, allo stato attuale, devono invece essere predisposti prima di avviare l'applicazione, utilizzando gli script matlab già implementati e descritti nel capitolo 4. Questa direzione non è stata intrapresa anche perché avrebbe comportato un costo computazionale che sarebbe gravato direttamente sul dispositivo sul quale è stato eseguito il deployment, che nel caso del tablet *samsung galaxy tab S4*, avrebbe probabilmente minato al funzionamento dell'applicazione. Per quanto riguarda il file matlab contenente la mesh fornitaci dal dip. di ingegneria biomedica, è possibile accedere ai dati relativi ai vertici e alle facce anche in ambiente .NET e quindi da Unity. In questo caso, basterebbe convertire gli script che al momento sono scritti in matlab e quindi utilizzarli per ottenere un volume in formato .obj da Unity. Da questo punto in poi, la conversione a .fbx è già realizzabile automaticamente mediante l'interfaccia python che Blender espone, e si dispone già dello script. Disponendo a questo punto del modello in un formato compatibile con Unity, l'unico step rimanente è quello di importare in Unity a run-time la mesh, funzionalità che la piattaforma predispone.

**Intersecare il modello con un piano arbitrario** L'intento è quello di estendere le funzionalità del capitolo 4, giungendo ad una versione del prototipo nella quale sia possibile non solo visualizzare i piani acquisiti, ovvero i frame prodotti direttamente dal dispositivo di acquisizione, ma anche di ricostruire dapprima i 3 piani ortogonali, e successivamente qualsiasi piano arbitrario,

permettendo all'utente di sceglierlo, orientandolo a piacere nello spazio. Questa direzione di sviluppo predispone l'utilizzo di tecniche di interpolazione, già esplorate ed implementate tramite matlab (anche se non incluse nel progetto), ma che sono facilmente convertibili a python, ad esempio, impiegando librerie apposite (es. `scipy`).

**Visualizzare in MR acquisizioni in dinamica** Per organi fisiologicamente caratterizzati da una notevole variazione di volume e dimensioni, come il cuore, le acquisizioni acquistano grande importanza soprattutto se ottenute in dinamica, ovvero se sono in grado di mostrare l'evoluzione della struttura del modello in istanti differenti. Nonostante non sia stato incluso nel prototipo, è stato già importato in Unity, attraverso una preliminare conversione tramite Blender, un modello animato che mostra le fasi del ciclo cardiaco di un ventricolo. Il modello è stato ottenuto a partire da una serie di mesh differenti acquisite ad istanti di tempo diversi. Per la segmentazione è stato usato un modello deformabile (level-set), che renderebbe complicata l'intersezione del modello stesso con uno oppure i 3 piani ortogonali.

**Disporre di un modello risultante da un processo di segmentazione online (in tempo reale)** Questa direzione è stata suggerita, oltre che dagli ing. biomedici, anche dallo specialista durante l'incontro con lui avuto. Il medico l'ha individuato come step successivo alla manipolazione della rappresentazione statica del modello già ottenuta con questo prototipo e alle funzionalità introdotte nel capitolo 4. I sistemi oggi utilizzati sono in grado di ricostruire indirettamente una visualizzazione di questo tipo, ma essa rimane disponibile unicamente su schermi piatti, per questo sarebbe importante proseguire su questa direzione.

**Includere nel modello anche la rappresentazione delle sue strutture interne** Anche questo aspetto è stato consigliato dal medico. La possibilità di "navigare" le strutture interne di un organo sarebbe fondamentale, in ambito cardiaco, per individuare le fibrosi, spesso fonti di aritmie. Questa visualizzazione è fondamentale perché la risonanza non permette di "scavare" l'interno del segmento del cuore interessato.

**Fornire il supporto anche per immagini RGB, multi-frame e per la decodifica dei pixel DICOM codificati in formato compresso** L'attuale implementazione non permette di estrarre dal file DICOM i dati sull'immagine nel caso questi siano memorizzati in formato compresso, attraverso codifiche quali, ad esempio, JPEG lossy o JPEG lossless. Sarebbe importante

includere questa funzionalità al fine di rendere più solido il sistema ed includere il supporto ad un più ampio numero di acquisizioni.

**Estendere la libertà di manipolazione della versione base anche alla seconda versione del prototipo** Nonostante la possibilità di piazzare il modello in un punto fisso dell'ambiente virtuale, potendoci girare attorno e ritrovandolo nello stesso punto dopo averlo abbandonato momentaneamente abbia un'importante valenza terapeutica, cioè di supporto alla cura della zona soggetta a malattia, sarebbe importante poter manipolare l'oggetto tridimensionale alla pari della versione base descritta nel capitolo 2. Ciò significa ruotare, zoomare e spostare l'oggetto tridimensionale assieme al piano che lo interseca, mantenendo inalterate le relazioni spaziali fra di loro. L'implementazione non dovrebbe essere fonte di troppi problemi.

**Registrazione e fusione di immagini provenienti da diverse acquisizioni** Uno dei possibili sviluppi accennati con i collaboratori di ing. biomedica è quello di costruire una rappresentazione tridimensionale a partire non una singola acquisizione, come effettivamente implementato in questo prototipo, bensì da acquisizioni multiple, raffiguranti la medesima struttura anatomica di interesse, o una parte. Questa ipotesi è sicuramente difficile da realizzare, anche per motivi extra-tecnologici, dovuti a esempio a compressioni dei tessuti derivanti dalla diversa posizione del paziente nelle varie acquisizioni. A rendere difficoltosa questa strada anche la necessità di riportare nello stesso sistema di riferimento i diversi frames e di uniformare la risoluzione dei pixel, nel caso questa vari fra le immagini estratte da diverse serie.

**Scenari di utilizzo condiviso (telemedicina)** Una delle possibili estensioni al sistema è quella di predisporre agli utenti la possibilità di condividere l'ambiente aumentato, che aprirebbe scenari di lavoro collaborativo fra i medici, anche da remoto. Come illustrato nel capitolo 1, in questo modo AR e MR andrebbero a supportare il filone della telemedicina, che è in costante sviluppo negli ultimi anni.

**Introdurre il tracking della strumentazione** In un ipotetico scenario di utilizzo in supporto all'attività chirurgica, sarà fondamentale affrontare il tema del tracking della strumentazione. Questa necessità deriva dal fatto che durante gli interventi chirurgici (prevalentemente quelli mininvasivi), a causa della scarsa visibilità, risulta fondamentale tenere sotto controllo in ogni momento la sua posizione e orientazione rispetto all'anatomia del paziente. Nonostante sia già disponibile nei sistemi attualmente impiegati in sala operatoria, il

tracking richiede l'utilizzo di complessi sistemi di registrazione e rappresenta uno dei temi più impegnativi da affrontare, poiché pone questioni in termini di occlusione e di risoluzione delle immagini.

**Applicare ulteriore tecniche di visualizzazione tridimensionale** Le tecniche di visualizzazione tridimensionale presentate ed implementate nei prototipi sono soltanto alcune di quelle presenti in letteratura. L'importanza di usufruire di una visione tridimensionale di un oggetto a partire da rappresentazioni bidimensionali ha prodotto uno sforzo importante in questa direzione da parte della comunità scientifica. Nonostante siano numerose, ad oggi tali tecniche risultano impiegate per la stragrande maggioranza su dispositivi a schermi piatti. Anche se non presentate in questo progetto di tesi, esiste una moltitudine di tecniche di visualizzazione volumetrica, racchiusa sotto il nome di "volume rendering" la cui implementazione risulta molto importante in un contesto biomedicale, poiché permette di osservare i dati volumetrici dell'interno del paziente senza richiedere un'elaborazione invasiva come la segmentazione, che ne determini una perdita di informazione così significativa. Tale funzionalità è già disponibile nei software per l'analisi e l'elaborazione di immagini mediche presentati (come Slicer e Invesalius), ma anche in numerosi altri (imageJ/fiji, paraview etc.) e ci è stata riferita anche da parte degli ing. di biomedica. Nonostante non sia stata inclusa nelle 2 versioni del prototipo, ho eseguito comunque un tentativo di implementazione del volume rendering in Unity, che ha testimoniato come il game engine lo supporti. Tuttavia, ha anche messo in evidenza come tali tecniche richiedano competenze più specifiche di computer graphics, quali la conoscenza di algoritmi espressi mediante linguaggi di programmazione non affrontati durante il mio percorso di studio. Lo step successivo sarebbe poi quello di integrare la visualizzazione planare a quella del rendering volumetrico, estendendo le riflessioni già esposte nel capitolo 4, consentendo l' "immersione" del modello segmentato contenuto al suo interno per poterlo confrontare con il dato originale, in questo caso non più bidimensionale, bensì volumetrico.





# Conclusioni

La tesi si è sviluppata dall'idea di applicare la mixed reality all'ambito medico, nello specifico alla visualizzazione delle immagini biomedicali prodotte da sistemi di acquisizione come quelli per risonanza magnetica e CT. Data la scarsità di implementazioni di questo tipo pubblicamente note, l'obiettivo principale era quello di capire se sistemi del genere potessero agevolare veramente l'attività dei medici. Attraverso la collaborazione con un team di ingegneri biomedici si è riflettuto sulle modalità con le quali presentare questa tecnologia al contesto medicale e si è implementato un prototipo che ha raccolto, fra gli altri, il parere di un medico. Il prototipo ha permesso di esplorare le potenzialità di un sistema di questo tipo in contesti ospedalieri e di fornire una prima base per implementazioni future. Oltre ad un set di funzionalità di base, sono state implementate estensioni più specifiche, nella direzione di supportare determinate attività, nello specifico per l'ambito cardiaco. Dopo aver eseguito la sperimentazione su un device non dotato dello specifico supporto alla realtà mista, si è riusciti a rendere fruibile il prototipo attraverso l'headset Meta 2, che si avvicina, i termini di resa, ai dispositivi di vertice in questo ambito tecnologico. L'alta specificità delle discipline e l'ampio spettro di applicazioni nel dominio medico pongono le basi per un grande lavoro futuro, le cui potenzialità sono ancora prevalentemente oscure agli addetti ai lavori. Le opinioni raccolte sono state confortanti e alla luce di queste considerazioni sarebbe molto promettente proseguire in questa strada, con la consapevolezza che i risultati ottenuti sono solo il punto di partenza.



# Ringraziamenti

Vorrei ringraziare la mia famiglia, a cui spero di restituire l'aiuto ricevuto finora, e soprattutto negli ultimi mesi. Un ringraziamento di cuore ai professori Alessandro Ricci e Angelo Croatti, con cui sarebbe un piacere collaborare anche in futuro. Infine, vorrei ringraziare i nostri collaboratori di ing. biomedica per il loro gentile contributo.



# Bibliografia

- [1] <https://www.wikipedia.org>.
- [2] <https://www.dicomstandard.org>.
- [3] RT Azuma. A survey of augmented reality. 1997.
- [4] RT Azuma. The road to ubiquitous consumer augmented reality systems. 2019.
- [5] B. Rosen” ”B. Spiegen. ”immersive virtual reality intervention for non-opioid pain management: A randomized controlled trial”. 2017.
- [6] Y. Baram. Closed-loop augmented reality apparatus. 2000.
- [7] CM. Botella. Mixing realities? an application of augmented reality for the treatment of cockroach phobia. 2005.
- [8] Alan Craig. *Understanding Augmented Reality*. 2013.
- [9] Cui, Kharei, and Gruev. Augmented reality with microsoft hololens holograms for near infrared fluorescence based image guided surgery. 2017.
- [10] Magee D., Y. Zhu, R. Ratnalingam, P. Gardner, and D Kessel. An augmented reality simulator for ultrasound guided needle placement training . medical & biological engineering & computing. 2007.
- [11] Davis. Virtual interactive presence in global surgical education: international collaboration through augmented reality. 2016.
- [12] HG. Hoffman. The analgesic effects of opioids and immersive virtual reality distraction: Evidence from subjective and functional brain imaging assessments. 2007.
- [13] Antonio Marttos, Fernanda M Kuchkarian, Emmanouil Palaaios, Daniel Rojas, Phillipe Abreu-Reis, and Carl Schulman. Surgical telepresence: the usability of a robotic communication platform. 2012.

- 
- [14] RT Navab, Blum, Wang, Okur, Wendler, and University of Munich. First deployments of augmented reality in operating rooms. 2012.
  - [15] F. Kishino P. Milgram. A taxonomy of mixed reality visual displays. 1994.
  - [16] Jon Peddie. *Augmented Reality: Where We Will All Live*. Il Saggiatore, Milano, 2017.
  - [17] Pianykh. *Digital Imaging and Communications in Medicine Dicom: A Practical Introduction and Survival Guide*. 2011.
  - [18] Ponce. Emerging technology in surgical education: combining real-time augmented reality and wearable computing devices. 2014.
  - [19] Dieter Schmalstieg. *Augmented Reality: Principles and Practice*. 2016.
  - [20] SR. Sharar. Applications of virtual reality for pain management in burn-injured patients. 2008.
  - [21] Speicher, Hall, and Nebeling. What is mixed reality. 2019.
  - [22] LM Su. Augmented reality during robot-assisted laparoscopic partial nephrectomy: toward real-time 3d-ct to stereoscopic video registration. 2009.
  - [23] Wake. Patient-specific 3d printed and augmented reality kidney and prostate cancer models: impact on patient education. 2019.