

ALMA MATER STUDIORUM - UNIVERSITA' DI BOLOGNA  
CAMPUS DI CESENA

---

Dipartimento di Informatica – Scienza e Ingegneria  
Corso di Laurea in Ingegneria e Scienze Informatiche

# Il ruolo dello Spatial Computing nei Sistemi di Realtà Mista

*Relatore:*

**Prof. Alessandro Ricci**

*Co-Relatore:*

**Prof. Angelo Croatti**

*Presentata da:*

**Luca Bonini**

---

Anno accademico 2019-2020



# Indice

<b>Introduzione</b> .....	<b>vii</b>
<b>1. Introduzione alla realtà mista</b> .....	<b>1</b>
1.1 Cos'è la realtà mista?.....	1
1.2 Ologrammi.....	4
1.3 Un altro tipo di immersione.....	7
1.4 Tipologie di Mixed Reality .....	8
1.4.1 Enhanced environment.....	8
1.4.2 Blended environment.....	9
1.4.3 Immersive environment .....	9
1.5 Tecnologie basate su Mixed Reality .....	10
1.5.1 Dispositivi hardware per realtà mista.....	10
1.5.1.1 Microsoft HoloLens / HoloLens 2 .....	11
1.5.1.2 Magic Leap 1.....	12
1.5.2 Frameworks e software per realtà mista .....	14
1.5.2.1 Mixed Reality Toolkit (MRTK) .....	14
1.5.2.2 Lumin .....	14
1.5.2.3 Unity .....	15
1.5.2.4 Google ARCore.....	15
1.6 Approccio al rendering della Mixed Reality .....	16
1.7 Difficoltà e sfide .....	17

<b>2. Spatial Computing</b> .....	<b>19</b>
2.1 Cos'è lo Spatial Computing? .....	19
2.2 Un differente design process.....	20
2.2.1 Bodystorming .....	21
2.2.2 Acting .....	22
2.2.3 Storyboarding .....	22
2.3 Coinvolgere il team.....	24
<b>3. Posizionare gli ologrammi nello spazio</b> .....	<b>25</b>
3.1 Sistema di coordinate spaziali.....	25
3.1.1 Stationary frame of reference .....	26
3.1.2 Attached frame of reference .....	27
3.1.3 Stage frame of reference .....	27
3.2 Ancore spaziali .....	28
3.2.1 Persistenza delle ancore spaziali .....	30
3.3 Errori di Tracking .....	31
<b>4. Analizzare e comprendere l'ambiente</b> .....	<b>33</b>
4.1 Mappatura spaziale .....	33
4.2 Qualità della mappatura spaziale .....	35
4.2.1 Occlusione ambientale .....	36
4.2.2 Fisica .....	37
4.2.3 Navigazione .....	38
4.2.4 Visualizzazione .....	38
4.3 Mesh processing .....	39
4.4 Mesh caching .....	41
4.5 Rendering e prestazioni .....	43
<b>Conclusioni</b> .....	<b>45</b>

**Bibliografia.....47**



# Introduzione

Al giorno d'oggi la realtà mista è un argomento molto discusso e con molte potenzialità in svariati settori, per questo motivo sempre più sviluppatori cercano di approcciare questo tipo di tecnologia al fine di creare applicazioni innovative in specifici campi di interesse. Basti pensare al modo in cui potrebbe assistere un medico durante un'operazione mostrando utili informazioni sul paziente direttamente davanti ai suoi occhi, oppure in un'azienda manifatturiera la possibilità di un progettista di vedere il prodotto finale davanti agli occhi prima ancora che entri in produzione per fare degli aggiustamenti; le possibilità sono molteplici. Nella società odierna tutti hanno un telefono in tasca che sostanzialmente risulta essere un computer in miniatura che ci permette di compiere azioni che fino a 20 anni fa non erano possibili, ma non solo, con il passare degli anni questi cellulari sono diventati sempre più compatti, sempre meno costosi e sempre più performanti. Attualmente i dispositivi che permettono di vivere esperienze di realtà mista sono scarsi, molto costosi ed ingombranti, ma con il passare del tempo anch'essi saranno probabilmente accessibili e portatili come lo sono i cellulari adesso. Il progresso però non deve essere fatto solo sul lato dell'hardware, ma anche da quello di design, progettazione e sviluppo software. Infatti, quando si desidera creare un applicativo in realtà mista bisogna affrontare diverse sfide e difficoltà dettate da una natura completamente diversa di interazione tra macchina e uomo. È proprio questo concetto che questa tesi cerca di aiutare a comprendere meglio, di fatti l'obiettivo principale è, non solo esporre le differenze principali tra la concezione di spazio tra realtà mista e un'applicazione tradizionale, ma anche dare una vista sul modo differente in cui si approccia tale sviluppo ed il modo in cui è possibile creare contenuti che vanno a posizionarsi all'interno del mondo fisico come se ne facessero veramente parte.

Quindi questo testo sarà suddiviso in quattro capitoli di cui il primo è un'introduzione alla realtà mista con qualche spiegazione di cosa sono gli ologrammi, le differenze tra AR (Augmented Reality), VR (Virtual

Reality) e MR (Mixed Reality), le varie tipologie di realtà mista, una panoramica sui dispositivi hardware e tecnologie software che abilitano la creazione di questo tipo di esperienze, ed una visione di quali sono le principali difficoltà da affrontare e le possibili sfide da superare. Il secondo capitolo si concentra sul differente approccio al design che deve essere fatto quando si vuole creare un'applicazione in realtà mista ponendo enfasi al pensiero spaziale e all'importanza di coinvolgere diversi membri del team anche nelle prime fasi di progetto. Il terzo capitolo è una panoramica sulle differenti tecniche con le quali è possibile posizionare gli ologrammi nello spazio in base al loro scopo con una nota sulla possibilità di salvare, caricare e condividere la posizione di essi con altri utenti. Infine viene trattato come comportarsi quando non è possibile mantenerne la posizione nello spazio come in caso di perdita di tracking. Il quarto ed ultimo capitolo invece parla della principale tecnica che permette di rendere realistico il posizionamento degli ologrammi all'interno del mondo fisico, ed è la mappatura spaziale. Viene descritto il modo e sotto quali aspetti la qualità di questa mappatura è importante, come raffinarla per avere risultati più consistenti, come mantenerla in memoria in modo efficiente ed infine come essa viene utilizzata per renderizzare gli ologrammi con piccoli accorgimenti per non degradare le prestazioni dei dispositivi di realtà mista.



# Capitolo 1

## Introduzione alla realtà mista

In questo primo capitolo viene fatta una panoramica su cosa si intende per realtà mista distinguendola dalla realtà aumentata e virtuale, per poi passare a definire cosa sono gli ologrammi. Dopo di che vengono esposte le varie tipologie di realtà mista e successivamente viene fatta una panoramica dei dispositivi abilitanti a questo tipo di esperienze e i principali framework su cui sviluppare.

### 1.1 Cos'è la realtà mista?

Per realtà mista, o mixed reality, si intende, letteralmente, una fusione tra ciò che è reale e ciò che è virtuale.

Questa unione di due mondi apre un'ampia gamma di nuove possibili interazioni tra l'utente, la macchina e l'ambiente che li circonda. Il termine venne coniato per la prima volta nel 1994 da *Paul Milgram* and *Fumio Kishino* come: "...anywhere between the extrema of the virtuality continuum"<sup>1</sup>.

Dal 1994 ad oggi la mixed reality è stata sempre più analizzata e una disciplina in particolare, nota come *HCI* (Human-computer interaction), è alla base della ricerca sulle interazioni fra le macchine e gli umani ed al giorno d'oggi è sempre più approfondita.

---

<sup>1</sup> Mixed Reality, [https://en.wikipedia.org/wiki/Mixed\\_reality](https://en.wikipedia.org/wiki/Mixed_reality)

<sup>2</sup> HCI, [https://en.wikipedia.org/wiki/Human%E2%80%93computer\\_interaction](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction)

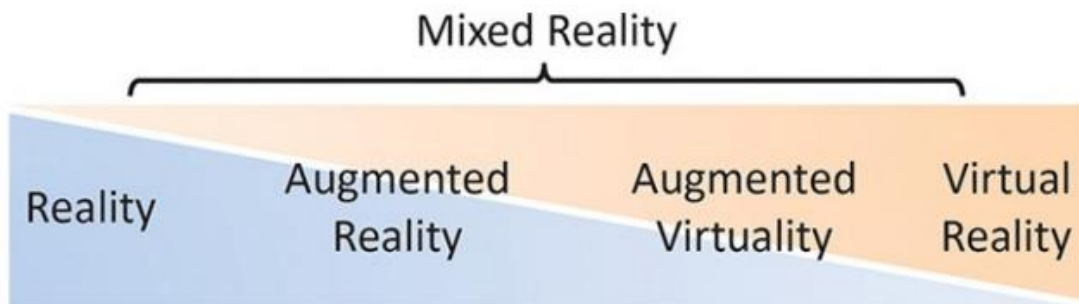


Figura 1.1 - mixed reality continuum. (tratta da [7])

Grazie ai progressi fatti nel campo dei sensori e dell'elaborazione sono state create delle API al fine di offrire un'interfaccia su cui sviluppare.

Le possibilità offerte da queste API in termini di cattura di input dall'ambiente sono molteplici, come per esempio permettono di catturare la posizione di una persona nell'ambiente tramite il tracciamento di movimento della testa (nel caso di un wearable headset come *HoloLens* o *Magic Leap*), le superfici ed i confini (mapping spaziale e riconoscimento della scena), l'illuminazione ambientale, l'audio ambientale, il riconoscimento di oggetti e la posizione di essi a prescindere dalla loro prospettiva e rotazione.

Combinando la capacità di elaborazione dei computer odierni con l'input umano e l'input ambientale reso possibile grazie a queste API è possibile realizzare esperienze di Mixed Reality.

Al giorno d'oggi sono disponibili diversi tipi di esperienze digitali come la realtà aumentata e la realtà virtuale, entrambe basate sull'utilizzo di smartphone e dispositivi dedicati, ma distinguendosi da quello che dovrebbe offrire la realtà mista.

La realtà aumentata, o augmented reality, offre un tipo di esperienza che non permette di interagire con il mondo digitale, sovrapponendo grafica su flussi video del mondo reale, mentre la realtà virtuale, o virtual reality, permette l'interazione con il mondo digitale, ma per rendere ciò possibile viene creato un mondo totalmente virtuale, oscurando quello fisico.

Entrambe le precedenti esperienze hanno i propri pro e contro in base all'utilizzo che si desidera farci, per questo è stato necessario creare

un nuovo tipo di esperienza che unisse i vantaggi di entrambi i mondi, da questo deriva la mixed reality, o realtà mista.

Al giorno d'oggi le esperienze di mixed reality sono possibili principalmente tramite l'utilizzo di dispositivi dedicati. Altri dispositivi come gli smartphone generalmente non dispongono di alcuna capacità di percezione ambientale, quindi non sono in grado di offrire esperienze di Mixed Reality, ma solo di Augmented Reality, e con l'aiuto di supporti esterni come *Google Cardboard*<sup>3</sup> o di supporti più sofisticati come *Samsung Gear VR*<sup>4</sup> è possibile avere esperienze di Virtual Reality.

Attualmente in commercio esistono due principali tipi di headset in grado di offrire esperienze di Mixed Reality:

- I dispositivi olografici, i quali sono caratterizzati dalla capacità del dispositivo di creare ologrammi e di collocarli all'interno del mondo reale.
- I dispositivi immersivi, i quali sono caratterizzati dalla capacità del dispositivo di creare un mondo virtuale, oscurando il mondo fisico, al fine di creare un'esperienza digitale.

Esempi di dispositivi olografici sono i *Microsoft HoloLens*<sup>5</sup> e i *Magic Leap*<sup>6</sup>, mentre esempi di dispositivi immersive sono l'*HTC VIVE*<sup>7</sup>, l'*Oculus Rift*<sup>8</sup>, *Oculus Quest*<sup>9</sup>, *Valve Index*<sup>10</sup>, *Samsung HMD Odyssey+*<sup>11</sup>.

---

<sup>3</sup> Google Cardboard, <https://arvr.google.com/cardboard/>

<sup>4</sup> Samsung Gear VR, <https://www.samsung.com/global/galaxy/gear-vr/>

<sup>5</sup> Microsoft HoloLens, <https://www.microsoft.com/en-us/hololens>

<sup>6</sup> Magic Leap, <https://www.magicleap.com/>

<sup>7</sup> HTC VIVE, <https://www.vive.com/>

<sup>8</sup> Oculus Rift, <https://www.oculus.com/>

<sup>9</sup> Oculus Quest, <https://www.oculus.com/>

<sup>10</sup> Valve Index, <https://store.steampowered.com/valveindex>

<sup>11</sup> Samsung HMD Odyssey +, <https://www.samsung.com/us/support/owners/product/hmd-odyssey-plus-mixed-reality>

Tutti questi dispositivi offrono una grande libertà di movimento a sei gradi con rotazione e traslazione.

Questi dispositivi possono essere standalone oppure aver bisogno di un PC separato (tramite cavo o Wi-Fi) a cui essere collegati in quanto non dispongono di capacità di elaborazione, ma ciò non indica se un dispositivo sia olografico oppure immersivo. La presenza, o l'assenza, di questo collegamento determina solo il grado di libertà di movimento offerto dal singolo dispositivo, quindi entrambe le tipologie dei sopra citati dispositivi possono essere sia con, che senza, tethering.

Il progresso tecnologico degli ultimi anni ha permesso la creazione delle prime esperienze di mixed reality, tuttavia lo spettro di tale realtà è molto più ampio ed al giorno d'oggi non sono ancora disponibili dispositivi in grado di offrire un'esperienza completa, cioè che copra l'intero spettro della realtà mista. I dispositivi odierni si stanno evolvendo molto rapidamente in quanto le possibilità e gli impieghi offerti sono molto promettenti, tuttavia per il momento sono in grado di offrire un'esperienza che copre solo una porzione dello spettro della mixed reality, ma in futuro le due tipologie di dispositivi che supportano la realtà mista inizieranno a fondersi in modo che i dispositivi olografici diventino più immersivi, mentre quelli immersivi diventino più olografici, fino a che non esisterà un'unica tipologia di dispositivo in grado di offrire un'esperienza completa.

## 1.2 Ologrammi

I dispositivi di realtà mista permettono di visualizzare ologrammi, che consistono in oggetti virtuali di luce e suoni i quali vengono generati come proiezioni all'interno delle lenti di un visore per mixed reality e si sovrappongono al mondo reale come se essi stessi ne fossero parte.

Gli ologrammi non sono inanimati, ma reagiscono a vari input dell'utente come sguardo, movimenti, comandi vocali e possono anche interagire con le superfici del mondo reale, il tutto per renderli il più realistici e credibili possibile.

Essi sono renderizzati direttamente all'interno del frame olografico davanti agli occhi dell'utente, ma ciò è fatto tenendo conto dell'ambiente circostante per fare in modo che le proiezioni sembrino di fronte all'utente e non a pochi centimetri dagli occhi.

Visto che gli ologrammi sono costituiti da luce e proiettati in modo additivo al mondo fisico, essi possono solo essere aggiunti all'interno dell'ambiente e non sottratti, quindi un ologramma completamente nero verrà visto come un oggetto completamente trasparente. Le forme e gli aspetti di questi ologrammi possono essere varie, infatti possono assumere forme realistiche e solide oppure stilizzate e semitrasparenti, possono essere anche elementi dell'interfaccia utente.



Figura 1.2 - proiezione di un ologramma di una doppia elica di DNA in ambito medico. (tratta da [9])

Inoltre possono produrre suoni generati in modo spaziale per essere percepiti come parte dell'ambiente, per questo i dispositivi che supportano la realtà mista in genere non isolano le orecchie ma possiedono delle casse audio che sono posizionate a pochi centimetri da esse in modo da non sopprimere i suoni provenienti dall'ambiente,

ma aggiungendo in modo additivo i suoni degli ologrammi a quelli del mondo reale.

Quando si decide una posizione per un ologramma, esso rimane fisso in quella posizione anche se l'utente dovesse muoversi all'interno della stanza; è anche possibile, utilizzando una *ancora spaziale* [cap. 3.2], di fissare un oggetto tale che il sistema si ricordi della sua posizione anche se si cambia stanza e si torna in un secondo momento.

Infine possono anche essere posizionati in modo assoluto rispetto all'utente, facendo ciò l'ologramma resterà sempre nella stessa posizione relativa a quella dell'utente a prescindere da dove quest'ultimo andrà.

Ora che gli ologrammi sono stati descritti in termini tecnici possiamo parlare di quanto essi siano veramente spettacolari, soprattutto al primo impatto. La prima volta che si indossa un visore per la realtà mista è impossibile non rimanere stupefatti; la sensazione di poter vedere e toccare, anche se in modo figurato, qualcosa che non esiste nel mondo fisico ma si mischia così bene con esso fa pensare a quanto sia fantastico il progresso tecnologico che abbiamo raggiunto. Cercando di illustrare cosa si prova ad utilizzare uno di questi visori possiamo immaginare l'ologramma di un cane, che non abbiamo mai avuto, e che improvvisamente è all'interno della nostra stanza. Questo cane si muove all'interno della stanza come se fosse veramente presente, cercando di evitare gli ostacoli che incontra o magari anche saltando su un tavolo. Ma anche altri comportamenti che lo fanno sembrare ancora più reale come inseguire e riportare una pallina che gli abbiamo lanciato. Una cosa che si nota quando si utilizzano gli ologrammi è che essi sono fondamentalmente una contraddizione. Sono palesemente digitali ma allo stesso tempo sembrano estremamente reali, sono a grandezza naturale ma possono essere ridimensionati, sono vicini a noi e sembrano toccabili fisicamente ma non lo sono. Gli ologrammi non sono per forza ristretti ad essere oggetti di tutti i giorni ma possono assumere qualsiasi forma e suono vogliamo. È possibile creare una finestra e al di là di essa vedere un mondo totalmente diverso dal nostro oppure creare una figura astratta che produce suoni, infine anche un buco nel pavimento che

sembra avere profondità infinita. Sanno essere estremamente convincenti al punto poter avere anche timore ad avvicinarsi.

## 1.3 Un altro tipo di immersione

Mentre la realtà virtuale (VR) immerge completamente l'utente all'interno di un'altra realtà, la realtà mista (MR) mantiene la percezione del mondo reale attorno all'utente aggiungendo ologrammi in sovrapposizione, ciò crea totalmente un altro tipo di reazione.

Una differenza che potrebbe sembrare superflua ma che alla fine non è in quanto può condizionare la nostra esperienza con questi visori, è la possibilità, nei visori VR, di sentirsi disorientati o avere nausea, o motion sickness. Questa comune condizione è causata dal modo in cui il nostro cervello funziona, in quanto non riesce a connettere l'input visivo e quello sensoriale in modo sensato. Ciò è dovuto al fatto che potrebbero esserci delle differenze, anche se molte volte piccole, tra il nostro corpo reale e quello virtuale che modificano la nostra percezione dell'ambiente, come per esempio avere un corpo leggermente più grande o alto, oppure piccolo o basso.

In altri casi, come quando si prova a guidare una macchina in VR, il nostro sistema nervoso si aspetta del feedback di forze applicate al nostro corpo fisico che non arriverà mai, per esempio mentre si fa una curva oppure una salita o discesa. Anche se molte tecniche e nuove tecnologie stanno venendo impiegate per contrastare questo problema è comunque più efficiente evitarlo completamente utilizzando un visore di mixed reality (da tenere però in considerazione che le esperienze che un dispositivo MR può offrire non saranno mai potenzialmente così "assurde" come quelle che può offrire un dispositivo VR se voluto). La realtà mista è quindi più terrena a livello di immersione e non fa sentire il corpo a disagio perché il mondo intorno a noi è sempre visibile e

inalterato rispetto a come lo percepiamo abitualmente con i nostri sensi. Essere ancorati al mondo reale mette a proprio agio il nostro corpo e infierisce poco sulla percezione dei nostri sensi andando a diminuire o eliminare totalmente la nausea.

L'approccio della realtà mista diventerà probabilmente il futuro dei visori e dell'interazione con i computer, e man mano che la tecnologia avanza ci si aspetta che la linea che separa VR e MR diventi più sottile e le due tipologie di dispositivi diventino sempre più simili.

## **1.4 Tipologie di Mixed Reality**

Quando si crea un'applicazione in realtà mista bisogna decidere quale tipologia si desidera creare nelle prime fasi di progetto. I visori per la mixed reality supportano differenti esperienze e gli strumenti di sviluppo offrono tools in grado di creare applicazioni che mirano lungo tutto lo spettro della realtà mista. È quindi compito del team di sviluppo decidere in quale parte di questo spettro il prodotto finale dovrà essere. I principali tipi di esperienze sono tre:

### **1.4.1 Enhanced environment**

Probabilmente la tipologia di applicazioni mixed reality più famosa, quella del Enhanced environment permette di mischiare il mondo reale con quello virtuale nel modo più naturale possibile. Infatti questo tipo di esperienza permette di collocare oggetti virtuali nel mondo reale mantenendo però la percezione dell'utente verso la realtà inalterata. Per esempio in queste tipologie di app è possibile piazzare un televisore su un muro vuoto, creare un modello tridimensionale di un palazzo sul



proprio tavolo oppure semplicemente proiettare un'interfaccia o delle note sulla scrivania. Gli enhanced environment sono il modo più potente per vivere la realtà mista.

### **1.4.2 Blended environment**

Un'applicazione per realtà mista con blended environment mira a fondere il mondo reale con quello virtuale ed il modo in cui ci riesce è grazie all'abilità di riconoscere l'ambiente ed i vari oggetti intorno all'utente e di creare un layer sovrapposto a tutto ciò. L'applicazione potrebbe anche decidere di modificare alcuni elementi del mondo reale, tramite questo layer, per costruire un ambiente più immersivo per l'utente. Le possibilità offerte da questo tipo di esperienza permettono di modificare a piacimento l'ambiente intorno a sé, per esempio, nel campo del design degli interni è possibile modificare il colore dei muri per trovare la tonalità desiderata oppure nel settore industriale è possibile modificare la struttura di una determinata macchina per cercare di accomodare al meglio il cliente.

### **1.4.3 Immersive environment**

Le applicazioni per realtà mista di tipo immersive environment sono invece adatte a coprire completamente il mondo reale simile all'esperienza ottenuta utilizzando un dispositivo immersive invece di uno olografico. Questi ambienti hanno danno molte possibilità in termini creativi agli sviluppatori in quanto permettono la creazione di un vasto range di tipologie di app. In contrasto ad un blended environment il visore dopo aver identificato e definito lo spazio intorno all'utente può decidere di rimpiazzarlo completamente con un ambiente totalmente digitale come per esempio la possibilità di vedere il pianeta Terra mentre si è in piedi sulla superficie lunare oppure la possibilità di creare un

videogioco con il suo mondo di gioco ed immergere l'utente completamente nel suo level design.

## **1.5 Tecnologie basate su Mixed Reality**

Le tecnologie che negli ultimi anni hanno veramente rivoluzionato le piattaforme da noi usate sono poche ed includono per esempio il desktop, il web, i dispositivi mobili ed il cloud.

Possiamo considerare i visori olografici come la prossima rivoluzionaria piattaforma da noi creata e probabilmente anche quella più complessa a livello di sviluppo e design per sfruttarla al meglio.

### **1.5.1 Dispositivi hardware per realtà mista**

Al giorno d'oggi l'unico mezzo che abbiamo per vedere ed interagire con gli ologrammi è utilizzare dispositivi con tecnologia all'avanguardia anche se essi non sono ancora abbastanza avanzati da essere piccoli e leggeri al punto di non rendersi nemmeno conto di indossarli.

Anche se in molti film di fantascienza è possibile vedere proiezioni di ologrammi nel mondo reale senza l'utilizzo di alcun tipo di dispositivo, purtroppo oggi non siamo ancora arrivati a quel punto quindi dobbiamo accontentarci di dispositivi che sono più grossi, pesanti e costosi (ma comunque di dimensioni, peso e costo ragionevoli).

I principali dispositivi per la realtà mista sono i seguenti:

### **1.5.1.1 Microsoft HoloLens / HoloLens 2**

Microsoft HoloLens, conosciuto durante lo sviluppo come Project Baraboo, sono il primo paio di smart glasses per la realtà mista sviluppati da Microsoft e introdotti sul mercato all'inizio del 2016. Attualmente è il visore più all'avanguardia nel campo degli Head Mounted Display (HMD) e possiede hardware dedicato che gli permette di analizzare l'ambiente con i sensori, eseguire software ed applicazioni senza il bisogno di alcun dispositivo esterno per funzionare. I Microsoft HoloLens sono dotati di cuscinetti regolabili all'interno ed un peso bilanciato lungo tutta la sua struttura per permettere un confortevole utilizzo anche con sessioni prolungate.

La parte frontale del dispositivo è dove alloggia la maggior parte dell'hardware come il processore, le fotocamere e le lenti. Una delle componenti più interessanti della struttura del dispositivo sono le lenti che sono un paio di vetri trasparenti combinate in cui vengono proiettati gli ologrammi che poi l'utente percepirà come parte del mondo fisico. Come tutti i visori MR anche questi vanno calibrati per assecondare la propria distanza focale degli occhi per avere la migliore esperienza possibile ed evitare di causare nausea.

Gli HoloLens possiedono anche un paio di speaker per l'audio 3D posizionati vicino all'orecchio ma non appoggiati ad esso per evitare di occludere i suoni provenienti dal mondo fisico. L'audio spaziale è una componente fondamentale dell'esperienza che permette all'utente di percepire i suoni fissati in un punto virtuale all'interno dell'ambiente per massimizzare l'immersione. Gli HoloLens includono anche dei pulsanti hardware per regolare la luminosità delle proiezioni per adattarsi a tutte le condizioni di luce. Includono anche porte utilitarie come porte USB e jack 3.5mm.

Ciò ovviamente ha anche dei contro come una batteria integrata nel dispositivo che ne aumentano il peso e un processore limitato in potenza per non consumare esageratamente la batteria e non mandare il dispositivo in thermal throttling (limitazione delle prestazioni dovuta ad una eccessiva temperatura della CPU).



Figura 1.3 - Microsoft HoloLens 2. (tratta da [10])

Dopo qualche anno, Microsoft decise di introdurre sul mercato il suo nuovo paio di mixed reality smart glasses e così fece uscire gli HoloLens 2 a fine 2019. Questo nuovo dispositivo porta con sé dei miglioramenti per l'esperienza complessiva dell'utente come un maggiorato campo visivo (FOV) che passa da 34° a 52° mantenendo la risoluzione di 47 pixel per grado. Ma anche altri miglioramenti come la quasi raddoppiata potenza di calcolo, il doppio bandwidth di memoria DRAM e miglioramenti ai vari sensori.

### **1.5.1.2 Magic Leap 1**

Magic Leap 1, come per HoloLens, punta sull'immersività offerta all'utente per scenari di realtà mista, ma al contrario di quest'ultimo ci prova in modo leggermente diverso. A partire dal design del dispositivo che ricorda più un paio di occhiali che un visore, il suo principale punto di forza è il peso decisamente inferiore rispetto ai suoi competitors che

è di soli 316g il che lo porta a pesare circa come un paio di cuffie ad arco. Uno dei motivi primari che permette un peso così ridotto è la scorporazione dell'unità di calcolo, memoria e memoria di massa da quella delle lenti e dei sensori, infatti da una parte abbiamo il visore che comprende le lenti dove vengono proiettati gli ologrammi, dall'altra parte, generalmente fissato sul fianco dell'utente, abbiamo un box di forma sferica appiattita in cui risiede l'unità di calcolo centrale. Le due componenti sono poi collegate tra di loro tramite un apposito cavo che passa lungo la schiena.

Magic Leap 1 possiede anche un discreto campo visivo di 50° e un display ad alta frequenza di aggiornamento di 120Hz. Include anche un piccolo controller ideato per essere tenuto comodamente in una mano per permettere controlli più precisi del mondo virtuale, ciò sacrifica un poco l'immersività ma garantisce un controllo maggiore sugli elementi proiettati.



Figura 1.4 - Magic Leap 1. (tratta da [11])

## **1.5.2 Frameworks e software per realtà mista**

Ora che i principali dispositivi hardware sono stati introdotti, bisogna citare alcuni framework e software basati sulla realtà mista che abilitano lo sviluppo sui dispositivi sopra citati per permettere a questo tipo di esperienze di prendere vita. Di seguito vengono descritti alcuni di questi frameworks:

### **1.5.2.1 Mixed Reality Toolkit (MRTK)**

Il Mixed Reality Toolkit, o MRTK, è un framework open-source per sviluppo cross-platform di applicazioni per realtà mista rilasciato da Microsoft insieme alla prima iterazione di HoloLens e si è evoluto fino ad oggi, grazie al contributo della comunità di sviluppatori in tutto il mondo, arrivando alla sua versione odierna l'MRTK v2. Esso offre tools per poter costruire un ambiente che permette interazioni spaziali e gestisce gli input dell'utente per creare esperienze di realtà mista. Questo progetto mira principalmente a ridurre la barriera di conoscenze necessarie per iniziare a sviluppare in questo tipo di ambiente ed è costruito in maniera modulare per permettere di organizzare meglio i progetti e favorire il supporto cross-platform.

### **1.5.2.2 Lumin**

Lumin come MRTK è un framework che mira ad abilitare lo sviluppo di applicazioni per realtà mista. Esso è stato creato dal team di sviluppo di Magic Leap e rilasciato insieme al lancio di Magic Leap 1 per offrire una piattaforma di sviluppo per il proprio device.

Lumin SDK mette a disposizione tools, analogamente a MRTK, per creare ambienti che abilitano interazioni spaziali gestendo gli input dell'utente al fine di farlo interagire con i contenuti e generare feedback sia visivo che sonoro.

### **1.5.2.3 Unity**

Unity è un game engine sviluppato da Unity Technologies che è stato rilasciato per la prima volta nel 2005 e da allora si è evoluto notevolmente arrivando ad includere dal 2017 il supporto alla realtà mista. Unity è un engine multiplatforma che supporta svariati sistemi operativi e differenti piattaforme, può essere usato per creare progetti 2D e 3D classici ma anche in realtà aumentata, virtuale e mista. L'engine è utilizzabile per creare qualsiasi tipo di esperienza anche al di fuori di quella videoludica, infatti è possibile realizzare anche altri tipi di produzioni come i film. Il comportamento del motore grafico è dettato principalmente tramite scripting in C# ma è anche possibile, tramite plugin esterni, di utilizzare visual scripting tramite blueprint, ma non è limitato solo a tale. Unity, come altri engine competitori, offre un array di possibilità creative molto esteso permettendo di creare applicazioni con grafica anche complessa in poco tempo grazie alla sua gestione intuitiva di asset, illuminazione, animazioni e tante altre componenti. Infine Unity, insieme ad altri engine come Unreal, supporta l'integrazione con i framework MRTK e Lumin abilitando e agevolando lo sviluppo di applicazioni in realtà mista.

### **1.5.2.4 Google ARCore**

ARCore è una piattaforma sviluppata da Google per supportare esperienze in realtà aumentata su dispositivi mobili. Le API che permettono il supporto dell'AR sono multiplatforma e funzionano sia su Android che iOS; esse permettono al dispositivo mobile di sfruttare i sensori per percepire l'ambiente circostante, determinare la sua posizione in esso ed interagire con le informazioni. In linea di massima ARCore è in grado di percepire la propria posizione all'interno dell'ambiente identificando punti attraverso la fotocamera e facendo un tracking di come si muovono nel tempo, nel frattempo utilizza

l'accelerometro e il giroscopio per determinare l'orientamento. Una volta raccolte le informazioni necessarie si passa poi all'analisi e infine alla stima della posizione del dispositivo nello spazio.

Anche se in una versione più primitiva, è comunque possibile la creazione di esperienze di realtà mista su mobile grazie a questo framework e ad altri competitori come ARKit.

## **1.6 Approccio al rendering della Mixed Reality**

Per poter generare gli ologrammi all'interno del mondo reale è necessario prima eseguire un'accurata analisi del mondo fisico perché altrimenti gli oggetti verrebbero generati senza tener conto della posizione nell'ambiente e ciò porterebbe ad avere situazioni come per esempio un vaso fluttuante nell'aria oppure una pallina che rimbalza senza toccare il terreno. Quindi è importante che prima venga effettuata un'analisi per determinare lo stato attuale del mondo fisico ed insieme a questa va anche effettuata un'analisi dello stato del mondo virtuale che dovrà poi essere sovrapposto a quello reale.

Successivamente bisogna proiettare i contenuti del mondo virtuale in sovrapposizione a quello reale tenendo conto dell'analisi precedentemente effettuata, in questo modo è possibile posizionare correttamente gli elementi all'interno della scena così l'utente è in grado di percepire l'ambiente così generato come parte del mondo fisico.

Possiamo quindi astrarre i concetti sopra citati nei seguenti punti:

- Cattura ed analisi della geometria dello spazio
- Comprensione della prospettiva dell'utente all'interno dello spazio
- Analisi, comprensione e feedback delle azioni esercitate dall'utente.
- Rendering della scena



## 1.7 Difficoltà e sfide

La sfida principale che viene posta quando si desidera creare un'applicazione in realtà mista è proprio quella di fondere il mondo reale con quello virtuale nel modo più realistico e naturale possibile. Per fare ciò è necessario che i dispositivi che supportano questa tecnologia abbiano un quantitativo di sensori adeguato e una potenza di calcolo sufficiente per garantire un'esperienza piacevole all'utente, ed il motivo è perché generare ologrammi che si fondano in modo realistico al mondo fisico è molto complesso.

I calcoli e le stime a livello di luce e occlusione sono computazionalmente molto pesanti e le tecniche per effettuare tali stime sono ancora studiate perché anche se il progresso tecnologico per quanto riguarda i motori grafici come Unity e Unreal Engine, nei quali è possibile progettare applicazioni di mixed reality, è stato veramente incredibile negli ultimi anni al punto di poter creare simulazioni estremamente vicine alla realtà, è comunque difficile riprodurre tali realtà quando bisogna adattarsi alle condizioni di luce del mondo reale. Ovviamente il problema non è solo questo, ma anche altri aspetti come la necessità di comprendere pienamente la geometria dell'ambiente circostante e la prospettiva dell'utente al fine di poter renderizzare i contenuti virtuali in modo che siano percepiti come parte della realtà. Oppure il problema dato dalla simulazione fisica, spesso necessaria per molti contenuti, che richiede un costo computazionale non indifferente e una comprensione dell'ambiente molto precisa per evitare comportamenti anomali degli oggetti.

Un'altra considerazione molto importante da fare è l'abituarsi a pensare diversamente, in modo più orientato allo spazio, per costruire una applicazione che sfrutti al meglio le possibilità della realtà mista. Infatti, come poi descritto in questa tesi, bisogna fare diverse considerazioni fin dal principio con un processo di design adeguato per poi successivamente utilizzare i sistemi di coordinate adatti a ciò che si vuole creare, per finire con una mappatura precisa dell'ambiente per renderizzare i contenuti.

Quando ci si appropria a sviluppare un'applicazione in realtà mista è quindi importante comprendere come approcciarsi allo sviluppo in quanto un metodo tradizionale che si può utilizzare per un'app bidimensionale non si applica bene in questo contesto.

# Capitolo 2

## Spatial Computing

In questo capitolo viene analizzato cosa si intende per Spatial Computing in merito alla realtà mista e il differente approccio al design, più orientato allo spazio, di applicazioni per questa tecnologia. Infine viene descritto come il coinvolgimento di diverse figure del team di sviluppo nelle prime fasi di progetto possa offrire una prospettiva più ampia a livello di design.

### 2.1 Cos'è lo Spatial Computing?

Lo spatial computing è una tecnologia che permette ai dispositivi che la supportano di interagire con l'utente come mai prima d'ora. Microsoft HoloLens e Magic Leap sono i due leader nella creazione di hardware che abilita questo tipo di esperienza in quanto grazie ai loro sensori e fotocamere riescono a capire dove si trovano nello spazio costruendo una loro comprensione dell'ambiente circostante e dell'utente. Questa capacità di tali visori permette alle esperienze di realtà mista di prendere vita e quando vengono collegati al cloud è anche possibile condividerle con altri utenti ed avere una visione unificata del mondo virtuale. Questi contenuti digitali danno la possibilità di percepire il mondo intorno a noi sotto una nuova luce aumentando il livello di ricchezza e significato che diamo all'ambiente. Di fatto questa trasposizione di asset, dati ed istruzioni in oggetti fisici permette un nuovo tipo di creatività ed utilità fino a qualche anno fa ritenuta fantascienza, come per esempio proiettare un televisore su un'intera parete e guardare un film insieme agli amici come se si fosse al cinema, oppure proiettare un modello di una

macchina ancora in sviluppo per permettere ai progettisti di vedere se il prodotto finale soddisfa le aspettative.

Le applicazioni in realtà mista sono molto differenti rispetto a qualsiasi altra applicazione al giorno d'oggi, quindi portano con sé una serie di sfide a livello di design che vanno affrontate correttamente. Bisogna tenere in considerazione molti più fattori e variabili che possono influire sull'esperienza finale per poter creare un applicativo che possa essere utile senza avere problematiche.

## **2.2 Un differente design process**

Quando bisogna affrontare un problema è sempre buona norma utilizzare un approccio il più aperto possibile, condividendo le proprie idee con i membri del team e di valutarle collettivamente. Da questo punto di vista il processo di design di una applicazione di realtà mista non differisce perché è comunque necessario avere un mix di developers, designers, researchers e managers a collaborare per iniziare la progettazione rimanendo aperti a idee da parte di tutto il team, magari mentre si iniziano a disegnare i primi sketch per definire le necessità tecniche o a definire i potenziali comportamenti dell'utente. Tuttavia con la mixed reality non è possibile mantenere un approccio classico, ovvero come si farebbe per una normale applicazione desktop o mobile, ma bisogna imparare un altro tipo di tecnica e modo di pensare per creare un buon prodotto.

Come prima barriera si incontra sicuramente il processo di design che tradizionalmente viene effettuato in 2D ma che inizia a mostrare le sue limitazioni in un ambiente prevalentemente 3D. Un'altra problematica è la difficoltà per la maggior parte delle persone di concepire sketch direttamente in tre dimensioni e l'utilizzo di strumenti tradizionali come carta e penna possono generalmente limitare le idee e le possibilità. Per questo il modo più efficiente per affrontare questa problematica è

l'utilizzo diretto di strumenti per la creazione tridimensionale come quelli per il gaming o l'engineering, tuttavia essi richiedono spesso un alto livello di abilità e non risultano molto pratici per un brainstorming di idee. Sono state dunque definite delle tecniche per supportare il processo di generazione di idee che non richiedono particolari abilità in modellazione 3D, in questo modo è possibile fare una buona ricognizione prima di procedere con la creazione dei primi prototipi.

## **2.2.1 Bodystorming**

La prima parte di questo design process è chiamata bodystorming, una tecnica che è stata uno standard nel design industriale per anni a questa parte ed è anche il primo passo che un team di mixed reality deve usare per uscire dal classico pensiero 2D dei dispositivi desktop o mobile. Questa tecnica consiste nell'utilizzo di oggetti da sceneggiatura nel mondo reale, in questo modo si possono costruire elementi digitali come interfacce utente, animazioni e oggetti di vario tipo per avere un'idea di come essi possano risultare all'interno di un futuro prototipo. Mentre creare delle storyboard accurate e prime versioni del software richiedono l'impiego di personale qualificato, l'utilizzo di semplici bastoncini di plastica e pezzi di cartone permettono ai membri del team di contribuire alla creazione di idee, di iniziare a pensare in un ambiente tridimensionale e di creare nuove opportunità. L'approccio migliore con questo tipo di tecnica è quello di provare a creare tante idee per avere un quadro ampio delle possibilità di implementazione; può anche essere molto utile cercare di includere persone che non sono familiari con lo sviluppo di applicazioni in realtà mista perché potrebbero offrire un pensiero differente.

A questo punto dello sviluppo non è necessario essere estremamente precisi o dettagliati in quanto è solamente necessario creare un prototipo minimale delle funzionalità che si credono essere importanti da includere nell'applicazione finale. L'obiettivo principale è quello di creare il maggior numero di idee possibile e di discuterne con il team per avere un numero

ridotto di funzioni da prototipare e testare ad uno stage successivo nel processo di sviluppo.

## **2.2.2 Acting**

La seconda parte di questo design process è chiamata acting e consiste nel simulare il comportamento dell'utente con le funzionalità che sono state ideate nello stage precedente. Questa fase è molto importante per comprendere quale sarà l'esperienza dell'utente mentre utilizzerà la suddetta applicazione, infatti durante la acting phase si simula la prospettiva di chi utilizza il dispositivo di realtà mista permettendo anche a persone esterne di osservare il comportamento che poi avrà l'utente finale. Può risultare molto importante anche includere esperti del campo in cui si stanno testando le funzionalità, per esempio in un contesto ospedaliero avere un medico che spiega il comportamento del personale in determinate situazioni può essere considerato estremamente utile. Ciò non è limitato solo a contesti specifici ma anche ad altri più generalizzati come la scelta se generare alcuni asset più dettagliati o meno al fine di avere più margine di performance.

In generale questo tipo di esperienza permette agli esperti di avere una visione di insieme su quali saranno i casi d'uso senza che ci sia bisogno di creare nessun prototipo specifico, ma non solo, infatti il processo di acting è incredibilmente leggero come carico di lavoro.

In ogni caso decidere l'approccio migliore in questa fase dipende molto dallo scopo che l'applicazione deve avere.

## **2.2.3 Storyboarding**

La terza parte di questo design process si chiama storyboarding e consiste nel disegnare dei particolari scenari per illustrare il

funzionamento dell'applicazione. Spesso presentare nuove idee ai membri del proprio team può essere fatto riproponendo delle recitazioni viste nel paragrafo precedente, ma quando bisogna introdurre specifici scenari ai clienti è necessario essere più accurati mostrando delle storyboard. Esse sono la base per moltissimi altri tipi di produzioni come film o videogiochi perché sono in grado di mostrare l'evoluzione di un'esperienza rapidamente e trasmettendo le sensazioni che l'utente finale dovrebbe avere. Lo storyboarding è estremamente utile anche per mostrare le idee del team sulle funzionalità e l'utilizzo dell'applicazione ai clienti in modo da evitare di dovere fare passi indietro durante le fasi successive dello sviluppo portando ad un grosso spreco di risorse. Ci sono due tipologie di storyboard, quelle a bassa fedeltà e quelle ad alta fedeltà.

Le storyboards a bassa fedeltà sono le più rapide ed efficienti per esporre dei concetti al proprio team e possono essere anche disegnate in modo estremamente semplice perché l'importante è condividere l'idea. Però bisogna tener conto del fatto che non tutte le persone riescono a comprendere pienamente le intenzioni tridimensionali quando viene mostrato uno sketch bidimensionale.

Le storyboards ad alta fedeltà, invece, sono uno strumento molto più efficace se si vuole mostrare l'idea del team ad un cliente oppure si vuole esporre il feeling che una certa situazione deve possedere. Esse possono essere utilizzate per mostrare momenti chiave dell'esperienza in cui è importante una più approfondita comprensione del prodotto finale. Generalmente, però, la costruzione di tali storyboard può richiedere un certo livello di abilità per cui diventa necessario che sia un membro del team, con buone capacità artistiche e buone capacità ad esporre idee difficile da descrivere, a procedere con il disegno.

## 2.3 Coinvolgere il team

Il processo produttivo di un'applicazione di realtà mista risulta essere differente dalle classiche applicazioni desktop e mobile come abbiamo visto nei paragrafi precedenti, quindi modificare l'approccio di comunicazione ed interazione del proprio team è il modo migliore per produrre risultati migliori. Risulta quindi importante che ogni membro della squadra vada al di fuori del proprio campo di abilità per cercare un coinvolgimento maggiore attraverso diverse fasi di sviluppo.

Interazione e comunicazione sono essenziali per offrire un buon design soprattutto quando la natura di questa tecnologia risulta essere relativamente nuova e gli strumenti a disposizione ancora grezzi.

Migliorare le abilità dei designer in aree più tecniche, per esempio l'utilizzo di engine come Unity, li aiuta a comprendere meglio l'approccio alla realtà mista e permettere la creazione di prototipi più funzionali, ma allo stesso modo membri del team che conoscono meglio gli aspetti implementativi possono aiutare i designer offrendo aiuti concreti sulle modalità in cui certi oggetti verranno poi codificati.

Attualmente gli strumenti per sviluppare applicazioni di mixed reality non sono ancora abbastanza evoluti per permettere un ottimo workflow di sviluppo, quindi spesso i team si ritrovano ad acquisire personale che abbia esperienze in settori come quello dei videogiochi o film, in quanto avere persone qualificate ad usare engine come Unity, con i vari strumenti che include, è di vitale importanza.

Le applicazioni in questo campo richiedono abilità molto più vicine all'architettura ed alla psicologia invece di design di interfacce, ma non solo; perché gli impieghi per questa tecnologia sono potenzialmente infiniti e si sta già pensando ai possibili utilizzi in campi come medicina e ingegneria dove i dati in real-time sono essenziali. Il campo della realtà mista è ancora nei primi stadi della sua vita quindi anche i processi produttivi sono in costante evoluzione, con sempre maggiore enfaticizzazione sul cloud per permettere una condivisione delle esperienze di mixed reality fra più persone.



## Capitolo 3

# Posizionare gli ologrammi nello spazio

In questo capitolo vengono trattati i vari sistemi di coordinate spaziali, con i vari casi d'uso, in quanto sono essenziali per comprendere come piazzare i contenuti all'interno dello spazio. Infine viene trattato il comportamento delle applicazioni in caso di perdita di tracking per mantenere l'immersione dell'utente. All'interno del capitolo sono presenti anche riferimenti ad alcuni aspetti legati allo sviluppo con Unity in quanto esso è diventato uno standard dello sviluppo in realtà mista.

### 3.1 Sistema di coordinate spaziali

Le applicazioni in realtà mista per sembrare realistiche hanno bisogno di piazzare i vari ologrammi all'interno del mondo fisico in posizioni corrette al contesto e alla prospettiva dell'utente. Per fare ciò bisogna usare un sistema di coordinate adatto alla mixed reality che sia in grado di piazzare gli ologrammi nelle giuste posizioni e che siano orientati correttamente. Esistono tre tipologie primarie di sistemi di coordinate: il sistema cartesiano, il sistema sferico ed il sistema cilindrico, tuttavia generalmente viene utilizzato il primo. Questo sistema di coordinate viene chiamato sistema di coordinate spaziali, oppure spatial coordinate system.

Generalmente le applicazioni 3D classiche come possono essere i giochi del giorno d'oggi utilizzano il sistema cartesiano per le coordinate

dei vari oggetti presenti nel mondo virtuale. Esso utilizza tre assi perpendicolari per rappresentare lo spazio: X, Y, Z.

Con la realtà mista è necessario definire uno standard come unità di misura per permettere la corretta fusione tra mondo reale e virtuale, per questo generalmente si utilizza il metro, quindi avere due oggetti che sono a 2 unità di distanza sull'asse X significa che sono distanti due metri. Prendendo come esempio uno dei sistemi di mixed reality più diffuso come Windows mixed reality, abbiamo X positivo verso destra, Y positivo verso l'alto e Z positivo verso l'utente.

### **3.1.1 Stationary frame of reference**

Quando un'applicazione di mixed reality si limita ad un'esperienza più stazionaria, con poco movimento da parte dell'utente come una seated-scale experience, il sistema di coordinate utilizzato in uno stationary frame of reference risulta essere il più indicato per mantenere stabili gli ologrammi vicino all'utente tenendo anche conto dei movimenti della testa. Nei game engine come Unity esso è definito come il "world origin" per cui gli oggetti piazzati utilizzando questo frame of reference prendono il nome di *world-locked content* e fanno riferimento alle coordinate del mondo reale come quelle del mondo virtuale. Generalmente quando viene lanciata un'applicazione il frame of reference viene creato durante lo startup al centro del campo visivo dell'utente tenendo in considerazione la posizione e l'orientamento del visore, e mantenuto per tutto il tempo di utilizzo. Nel caso in cui il cambio di scena sia drastico, come per esempio il cambio di stanza, è comunque possibile ri-centrare l'origine con una nuova posizione. Il sistema si adatta all'ambiente circostante ed impara nel tempo a determinare in modo più accurato le distanze tra i punti di interesse reali su cui si basa il tracking, ma ciò ha delle limitazioni, infatti nel caso in cui dei contenuti dovessero essere renderizzati ad una distanza maggiore di 5 metri gli ologrammi diventerebbero instabili e inizierebbero a shiftare di posizione, allora per questi casi è necessario utilizzare altri tipi di frame of reference.

### 3.1.2 Attached frame of reference

Quando un'applicazione in mixed reality necessita che l'utente si sposti nel tempo allora un attached frame of reference risulta adatto a generare ologrammi mantenendone la stabilità, infatti esso mantiene un orientamento fisso che viene definito quando l'applicazione viene lanciata. Gli oggetti creati utilizzando questo sistema di coordinate prendono il nome di *body-locked content*. Questo tipo di frame of reference risulta essere estremamente utile in situazioni in cui il dispositivo non riesce a comprendere la sua posizione nel mondo fisico, per cui risulta essere l'unico sistema di coordinate in grado di offrire stabilità agli ologrammi. Può essere utilizzato anche per generare UI e fallback UI che consiste nel mostrare all'utente messaggi indicando che il dispositivo ha perso traccia della sua posizione nel mondo fisico.

### 3.1.3 Stage frame of reference

Quando un'applicazione in mixed reality necessita di andare oltre una esperienza seated-scale allora è possibile utilizzare uno stage frame of reference che offre una libertà di movimento superiore rispetto ad uno stationary frame of reference. Lo stage è definito dalla stanza precedente definita dall'utente durante il setup e da essa viene creato uno "stage origin" posizionato sul pavimento (il quale sarà definito come piano in  $Y = 0$ ) dove si baserà il sistema di coordinate. Tuttavia visto che questo frame of reference definisce un solo sistema di coordinate fisso è possibile solamente creare esperienze room-scale contenute entro 5 metri dall'utente, in quanto andare oltre comporterebbe una incrementale instabilità degli ologrammi renderizzati. In applicazioni VR questo problema non esiste perché si conosce in anticipo la geometria degli oggetti e si standardizza un unico sistema di coordinate da utilizzare per tutta la scena, ma in mixed reality non è possibile questo tipo di approccio in quanto il dispositivo aggiorna continuamente la propria percezione del mondo fisico modificando di conseguenza anche

il sistema di coordinate. Quindi per poter creare un'esperienza spaziale che si estenda oltre questi 5 metri è necessario utilizzare un altro tipo di tecnologia chiamata spatial anchors, o ancore spaziali.

## 3.2 Ancore spaziali

Le ancore spaziali, o spatial anchors, sono una tecnologia di Microsoft estremamente utile per i sistemi di realtà mista e servono per rappresentare vari punti d'interesse all'interno del mondo reale in modo che il sistema ne tenga traccia nel tempo. Per assicurarsi che ogni ologramma sia piazzato nel punto giusto e mantenga la sua posizione nel tempo anche quando l'utente si muove nella stanza è necessario collocarli in posizione relativa ad una spatial anchor. Quest'ultime sono un sistema di coordinate che varia nel tempo in base alla posizione di altre ancore oppure del frame of reference. Questo sistema permette di aggiustare la posizione degli ologrammi continuamente per avere sempre una posizione corretta dei suddetti in relazione al mondo fisico.

Quando un ologramma deve essere piazzato all'interno del mondo reale bisogna scegliere se è più consono utilizzare un frame of reference, che sia fisso o dinamico, oppure utilizzare una ancora spaziale:

- Ologrammi che vengono piazzati in uno stationary frame of reference mantengono delle relazioni rigide tra di loro, però nel caso in cui l'utente si sposti nella stanza, al fine di mantenere gli ologrammi stabili, il sistema di coordinate potrebbe modificarsi in base al mondo reale.
- Ologrammi che vengono piazzati in uno stage frame of reference mantengono anch'essi delle relazioni rigide tra di loro, tuttavia essi rimangono posizionati in base alla loro origine fisica predefinita, al

contrario di quelli piazzati in uno stationary frame of reference. Tuttavia ci sono anche limitazioni, infatti nel caso di ologrammi renderizzati oltre 5 metri dall'utente si potrebbe avere scarsa stabilità di essi finché non rientrano nel raggio utile.

- Infine gli ologrammi piazzati utilizzando spatial anchors condividono la propria posizione tra di loro al fine di migliorare il piazzamento delle varie ancore. Ciò permette al sistema di correggere continuamente sia la posizione degli ologrammi che delle spatial anchors.

I sistemi di piazzamento degli ologrammi sopra descritti non sono necessariamente una scaletta dal peggiore al migliore ma hanno i loro utilizzi specifici in diverse situazioni. Per esempio lo stationary frame of reference prioritizza la stabilità degli ologrammi vicino all'utente mentre gli altri due tipi prediligono la stabilità vicino alle origini. Gli stage frame of reference e spatial anchors quindi cercano di mantenere la stabilità degli ologrammi nel tempo a discapito di una incrementale instabilità di essi man mano che si allontanano dai punti di origine, ciò è dovuto dal fatto che le modifiche alla posizione e l'orientamento dello stage o delle varie anchors provoca un effetto amplificato in proporzione alla distanza da tali origini.

Tenendo in considerazione Unity come game engine di riferimento, per poter creare una spatial anchor è necessario creare un *GameObject*<sup>12</sup> a cui aggiungere una *WorldAnchor* e quando essa non serve più è possibile eliminarla in qualunque momento. Tuttavia Unity non supporta lo spostamento di un *GameObject* nella scena mentre gli è stata assegnata una *WorldAnchor* quindi è necessario eliminare quest'ultima dal *GameObject* prima di eseguire tale operazione, dopodiché effettuare una trasposizione del *GameObject* con un vettore ed infine assegnare nuovamente l'anchor. Esiste la possibilità che una *WorldAnchor* non sia localizzabile all'interno del mondo fisico in un determinato momento, in queste situazioni Unity non aggiornerà più gli

---

<sup>12</sup> Un *GameObject* in Unity è una classe base per tutte le entità della scena.

oggetti legati a quella determinata spatial anchor risultando in una visualizzazione errata degli ologrammi ad essa ancorati. Per questa specifica situazione Unity ci offre un modo per controllare i cambiamenti dello stato delle spatial anchors.

### **3.2.1 Persistenza delle ancore spaziali**

Le spatial anchors possono essere utilizzate anche per mantenere vari riferimenti al mondo reale attraverso diverse sessioni, infatti è possibile salvare esse in vari modi per poi ricaricarle in un secondo momento. Quando queste ancore vengono salvate o caricate gli viene assegnata una chiave identificatrice al quale è anche possibile associare diversi ologrammi per essere caricati allo stesso tempo.

Ci sono diversi modi per salvare delle spatial anchors in base all'utilizzo che dovrà poi avere la specifica applicazione:

- Possono essere salvate localmente sulla memoria di massa per poi essere caricate in un secondo momento ricalcolando la stessa posizione in relazione al mondo reale.
- Possono essere salvate nel cloud per essere poi caricate su diversi dispositivi. Ciò permette anche di condividere le spatial anchors con altri utenti, i quali le caricheranno negli stessi punti del mondo reale per permettere un'esperienza mixed reality condivisa in real time.
- Possono essere salvate nel cloud, ma a differenza del punto precedente, utilizzate per una persistenza asincrona su diversi dispositivi. In questo modo è possibile condividere esperienze in mixed reality con altri utenti senza che essi siano presenti nella stessa stanza allo stesso tempo.

Per applicazioni in realtà mista che si limitano ad esperienze con distanze di rendering ridotte (entro 5 metri), tuttavia, è più consono

utilizzare lo stage frame of reference come sistema di coordinate. Invece per esperienze che necessitano un rendering su distanze maggiori è necessario utilizzare le spatial anchors.

Sempre considerando Unity come game engine è possibile gestire la persistenza delle WorldAnchor utilizzando un componente chiamato *WorldAnchorStore* che permette di tenere traccia di tutti i GameObject che utilizzano le WorldAnchor. Dopodiché è possibile salvare le varie anchors assegnandole una stringa identificatrice dentro lo Store, e in modo analogo caricarle e riposizionarle all'interno dell'ambiente.

### 3.3 Errori di Tracking

I dispositivi di realtà mista basano la loro capacità di sfruttare le varie metodologie di tracking sopra descritte con l'utilizzo di vari sensori nella loro dotazione, però è possibile che in alcuni casi questi ultimi non riescano a catturare abbastanza dati per permettere al visore di mixed reality di comprendere la sua posizione nell'ambiente. Questo tipo di problema può essere causato da diversi tipi di fattori esterni come scarsità di luce o sensori ostruiti e comporta, successivamente, un posizionamento degli ologrammi non corretto o ne previene il piazzamento. Quando, per istanza, il dispositivo non riesce ad eseguire il tracking dell'ambiente a causa di un'insufficiente quantità di dati proveniente dai sensori come nelle due casistiche sopra descritte, diventa impossibile renderizzare correttamente contenuti world-locked quindi il device non riesce a comprendere dove si trovino le varie ancore spaziali, gli stationary frame of reference e lo stage frame of reference. Tuttavia è comunque possibile utilizzare l'attached frame of reference e quindi renderizzare contenuti body-locked in quanto questo sistema di coordinate ha posizione assoluta rispetto al dispositivo e quindi non necessita dati dai sensori. Un altro caso in cui si potrebbe avere un altro tipo di problema di tracking è quando l'ambiente ha cambiamenti

dinamici come persone che camminano all'interno della stanza in cui il dispositivo cerca di effettuare il tracking e ciò comporta instabilità agli ologrammi generati mentre il visore cerca di recuperare il piazzamento delle varie ancore. Anche quando l'utente cambia stanza mentre sta renderizzando contenuti relativi ad uno stationary frame of reference comporta instabilità nei contenuti generati mentre il dispositivo cerca di rimappare l'ambiente. In alcuni casi è possibile che il tracking sia errato a causa di aree identiche presenti nel mondo fisico e ciò può comportare il posizionamento errato di alcuni ologrammi che avevano come sistema di riferimento delle spatial anchors che sono state invertite di posto.

Unity di per sé non è in grado di risolvere le problematiche descritte, tuttavia offre metodi per permettere agli sviluppatori di decidere come l'applicazione si dovrebbe comportare in questi casi, e se non gestite portano il game engine a mostrare una splash image a schermo finché il tracking non è stato recuperato. Però è possibile gestire la perdita del tracking cambiando, per esempio, tutti i contenuti world-locked e facendoli diventare temporaneamente body-locked mentre si effettua polling su ogni frame della fotocamera per recuperare il tracciamento dell'ambiente.



# Capitolo 4

## Analizzare e comprendere l'ambiente

Dopo aver parlato di come è possibile piazzare gli ologrammi utilizzando diversi sistemi di riferimento spaziale, adesso viene descritto come fonderli al mondo reale, infatti in questo capitolo viene definita la mappatura spaziale dell'ambiente con approcci alla creazione di essa, descrivendo aspetti critici sull'importanza della qualità, ma anche dell'importanza di una gestione della memoria efficiente e delle prestazioni.

### 4.1 Mappatura spaziale

Con mappatura spaziale, oppure spatial mapping, si intende la capacità dei dispositivi di creare una dettagliata rappresentazione delle superfici del mondo reale all'interno di quello virtuale ed esso è la base dei dispositivi di realtà mista perché permette alle applicazioni di comprendere la geometria del mondo fisico per piazzare gli ologrammi all'interno di esso in modo realistico e credibile. Lo spatial mapping sfrutta le informazioni di profondità dell'ambiente per permettere, per esempio, di occludere gli oggetti renderizzati in base a cosa è presente intorno ad essi e la prospettiva dell'utente, quindi è molto importante per far percepire i contenuti virtuali come parte del mondo reale. Esso è costituito da due componenti principali che sono lo *Spatial Surface Observer* e le *Spatial Surfaces*. Le applicazioni dovranno fornire al primo delle zone di delimitazione per marcare le varie aree in cui si necessita la ricezione di dati per la mappatura spaziale e per ognuna di tali zone verranno ritornati dei set di superfici spaziali, ognuna delle quali sarà

costituita da una mesh di triangoli attaccati a dei world-locked content come parlato nel capitolo precedente. Ovviamente l'ambiente potrebbe cambiare nel tempo e quindi lo spatial mapping dovrà adattarsi a tutti questi cambiamenti creando nuove mesh, distruggendo le vecchie oppure modificandole.

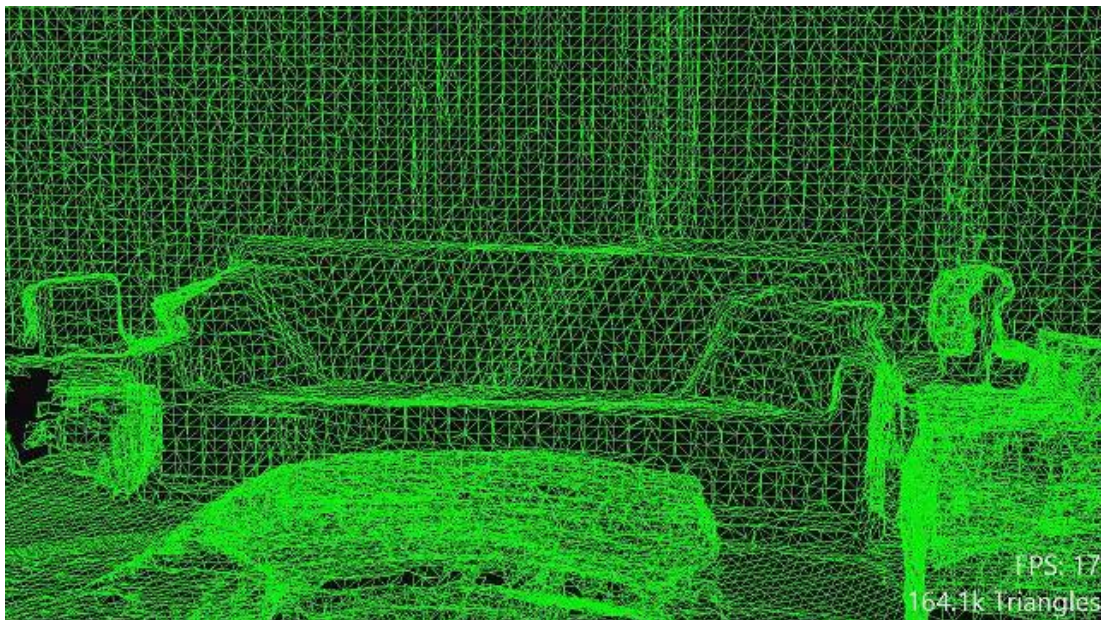


Figura 4.1 - Mesh di triangoli che ricopre una stanza. (tratta da [1])

Quindi quando si vuole mappare un ambiente è necessario prima di tutto creare un Surface Observer al quale passare delle figure tridimensionali che servono a definire delle zone nelle quali l'applicazione è interessata a ricevere una mappatura spaziale.

Queste zone, chiamate anche volumi spaziali, possono essere usate per identificare delle porzioni di mondo fisico utilizzando un sistema di coordinate world-locked, oppure identificare una porzione di spazio che si muove insieme all'utente utilizzando un sistema di coordinate body-locked. L'applicazione può anche effettuare del polling per verificare se ci sono cambiamenti nelle superfici spaziali di interesse, in generale utilizzato per zone dinamiche come quelle body-locked, oppure rimanere in ascolto per un evento che notificherà la modifica di tali superfici, in generale per zone statiche come quelle world-locked. Dunque è compito

dell'observer di tener traccia e classificare tutte le superfici spaziali che vengono create, modificate o eliminate.

## 4.2 Qualità della mappatura spaziale

La qualità della mappatura spaziale è influenzata da diversi fattori ed è quindi opportuno eseguire un'analisi accurata dell'ambiente per poi generare ologrammi che si fondano in modo naturale col mondo fisico. Generalmente il piazzamento dei contenuti su superfici crea un naturale mapping dal 3D al 2D, ciò riduce il quantitativo di dati da fornire all'applicazione e rende più intuitiva l'interazione con essi.

Bisogna però tener conto che quando l'applicazione cerca di dedurre la distanza degli oggetti, per esempio facendo il raycasting di un singolo raggio seguendo la direzione dello sguardo dell'utente per cercare la più vicina spatial surface, potrebbe portare a risultati imprevedibili a causa di raggi passanti attraverso piccoli buchi o geometria di cui l'utente non è a conoscenza, quindi una soluzione può essere l'utilizzo di un raycasting con più raggi in modo da avere risultati più prevedibili e precisi. Esiste anche la possibilità di utilizzare la forma e la direzione delle superfici per posizionare gli ologrammi per esempio utilizzando una simulazione fisica invece del metodo appena descritto, tuttavia anche in questo caso bisogna prestare attenzione a possibili problematiche come la compenetrazione della geometria oppure un oggetto che presenta poligoni irregolari potrebbe incastrarsi con la mesh del mapping spaziale quindi potrebbe essere necessario effettuare uno smussamento della superficie per permettere uno scorrimento fluido. In ogni caso la possibilità dell'applicazione di piazzare ologrammi su spatial surfaces è dipendente da se è stato fatto oppure no l'environment scan e in alcuni casi è considerabile una scelta migliore eseguire un piazzamento manuale, per esempio se un collocamento automatico porterebbe a posizionare un oggetto fuori dalla portata dell'utente.

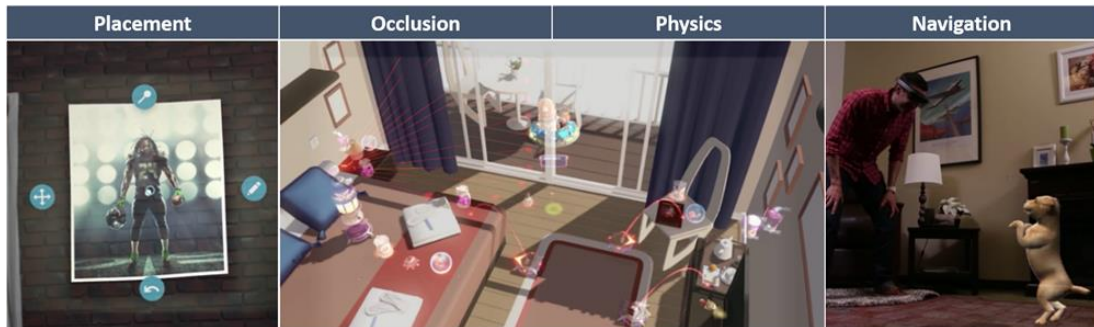


Figura 4.2 - Le caratteristiche della mappatura spaziale che permettono una buona fusione tra gli ologrammi ed il mondo fisico. (tratta da [1])

## 4.2.1 Occlusione ambientale

La primaria funzione dello spatial mapping potrebbe essere considerata quella dell'occlusione ambientale che consiste semplicemente nel nascondere, parzialmente o interamente, gli ologrammi dietro superfici. Questa tecnica aumenta sensibilmente la percezione che gli oggetti renderizzati facciano parte del mondo fisico e fornisce informazioni importanti sulla posizione spaziale di essi. L'occlusione fornisce anche feedback visivo sull'interazione che gli ologrammi dovrebbero avere con gli oggetti reali, per esempio un oggetto completamente coperto ci si aspetta sia dietro una superficie solida e quindi non dovrebbe passarci attraverso.



Figura 4.3 - Ologramma di un gatto senza occlusione (a sinistra) e con occlusione (a destra). (tratta da [12])

## 4.2.2 Fisica

Un altro modo per sfruttare un buon spatial mapping è tramite l'utilizzo di una simulazione fisica per aumentare la percezione di realismo degli ologrammi, infatti i vari oggetti generati dovrebbero, possibilmente, avere un comportamento fisico realistico ed interazioni basate sulla fisica con la mappatura ambientale. Tuttavia questo tipo di simulazione risulta essere complessa se lo spatial mapping non è stato raffinato, quindi potrebbe essere necessario utilizzare tecniche come il mesh processing [cap. 4.3] per raffinare alcune imperfezioni della mesh. Altre considerazioni devono essere fatte in base allo scanning dell'ambiente perché se un oggetto dovesse andare fuori dall'area mappata e quindi colpire una superficie a cui non è stata applicata una mesh, come dovrebbe comportarsi? E l'applicazione dovrebbe adattarsi a cambiamenti dinamici dell'ambiente?

### 4.2.3 Navigazione

Le applicazioni possono anche sfruttare la mappatura spaziale per permettere a diverse entità di muoversi all'interno dello spazio in modo consono alla struttura di esso, quindi per esempio un ologramma di una persona dovrebbe essere in grado di camminare evitando gli ostacoli ed evitando di passare sopra divani o tavolini. Le mappe di navigazione sono utili anche ad un utente non familiare con una certa zona a provvedere utili informazioni su quali sono i percorsi utilizzabili. Le principali difficoltà quando si cerca di mappare queste superfici sono riuscire a realizzare affidabili percorsi e adattamenti allo stato dell'ambiente, quindi per mitigare queste problematiche potrebbe essere necessario effettuare del mesh processing [cap. 4.3] per smussare queste superfici o altre tecniche per permettere un ottimale path-planning, quest'ultimo avente ricevuto una grande attenzione negli ultimi anni nello sviluppo di videogiochi. Purtroppo in Unity non è possibile utilizzare la sua funzionalità integrata di *NavMesh* con le superfici generate con lo spatial mapping ed il motivo è perché questa funzionalità necessita di processare i dati preventivamente da source assets, ma la mappatura spaziale non può essere effettuata prima del lancio dell'applicazione. Da notare anche il fatto che non è possibile ottenere informazioni di superfici troppo lontane dal dispositivo di realtà mista, quindi se si vuole utilizzare funzionalità di path-planning in un'area di maggiori dimensioni è necessario mappare tutta l'area.

### 4.2.4 Visualizzazione

Genericamente le superfici spaziali vengono nascoste all'utente in quanto sono informazioni irrilevanti per un'esperienza in realtà mista e, così facendo, si riduce anche il numero di elementi proiettati dal visore, tuttavia in alcune situazioni come, per esempio, quando si vuole piazzare una scatola sul pavimento, avere un'ombra proiettata sulla superficie spaziale che copre il terreno aiuta a comprendere meglio la

distanza da esso. Quando si vogliono renderizzare queste superfici, utilizzare tradizionali sistemi di calcolo della luce potrebbero rendere più visibili alcune imperfezioni delle mesh, quindi anche in questo caso effettuare del mesh processing [cap. 4.3] potrebbe aiutare, ma anche non utilizzare le “normali” (le normali, o normal mapping, sono delle texture utilizzate per simulare le interazioni della luce con le imperfezioni delle superfici e sono utilizzate per renderle più realistiche senza utilizzare ulteriori poligoni) e quindi proiettare texture pulite produce lo stesso risultato.

## 4.3 Mesh processing

Il processo di mesh processing consiste nell’apportare delle modifiche alle mesh, in questo caso quelle della mappatura spaziale, al fine di renderle più “pulite” e facilmente utilizzabili dall’applicazione come per esempio durante una simulazione fisica. Questa procedura, tuttavia, necessita di essere effettuata in modo accurato perché eccessive modifiche alle mesh al fine di renderle più lisce e consistenti potrebbe portare a perdere del dettaglio ambientale e ad allontanare la rappresentazione virtuale delle superfici da quella reale.

Ci sono diverse tecniche che possono essere utilizzate per effettuare del mesh processing:

- La prima consiste nel riempire vari “buchi” che si creano nella mesh durante lo scan dell’ambiente a causa per esempio di zone troppo scure per cui il dispositivo di realtà mista le interpreta erratamente. Ciò può comportare diverse problematiche come:

- A livello di occlusione potrebbe essere possibile vedere attraverso tali buchi anche se la superficie reale dovrebbe essere opaca.
- A livello di raycast potrebbe essere calcolata erratamente la distanza tra l'utente e la superficie a causa di un singolo raggio proiettato attraverso un buco. È possibile mitigare questa problematica proiettando diversi raggi e aggregando il risultato per avere un calcolo più consistente, tuttavia ciò ha un costo computazionale.
- A livello di fisica potrebbe succedere che un oggetto cada attraverso uno di questi buchi.

Quando si cerca di riempire queste imperfezioni delle mesh bisogna definire un algoritmo che sappia riconoscere i buchi immaginari da quelli reali in modo da non riempire oggetti come le finestre.

- La seconda tecnica consiste nel rimuovere le “allucinazioni” che si creano a causa dello scan ambientale in zone con riflessi e oggetti illuminati con luci particolarmente intense o in movimento. Anche questi creano problematiche come:
  - A livello di occlusione queste allucinazioni potrebbero occludere altri ologrammi.
  - A livello di raycast potrebbe essere calcolata erratamente la distanza tra l'utente e la superficie. Anche in questo caso una soluzione potrebbe essere quella di effettuare la proiezione di diversi raggi e aggregare il risultato.
  - A livello di fisica potrebbe succedere che oggetti colpiscano delle allucinazioni comportandosi quindi in modo imprevedibile.



Anche in questo caso è possibile definire un algoritmo per filtrare queste imperfezioni stando attenti a non rimuovere piccoli oggetti riflettenti che invece esistono nel mondo reale.

- La terza tecnica consiste nello spianare ed ammorbidire superfici che vengono mappate durante lo scan ambientale e che ritornano con del “rumore”. Anche in questo caso abbiamo problematiche associate come:
  - A livello di fisica potrebbero verificarsi comportamenti anomali causati da superfici irregolari.
  - A livello di rendering potrebbe succedere che una superficie con del “rumore” porti ad avere una texture normale alterata, il che comporta una rappresentazione della luce errata.

Quando si smussano superfici bisogna anche in questo caso stare attenti a non eliminare del dettaglio che invece si dovrebbe mantenere per permettere una precisa occlusione degli ologrammi. In alcuni casi se fosse necessaria solo una modifica estetica è possibile smussare i vertici delle texture normali invece delle posizioni dei vertici della mesh.

## 4.4 Mesh caching

Quando si utilizza lo spatial mapping per definire digitalmente un’area intorno all’utente bisogna anche pensare a come salvare, renderizzare e processare queste mesh. Visto che queste ultime sono composte da un’alta densità di triangoli, elaborarle richiede un quantitativo di risorse computazionali e di storage non indifferenti, quindi le applicazioni dovrebbero utilizzare un appropriato sistema di caching per ridurre

l'utilizzo di tali risorse. Quando si cerca di fare mesh caching bisogna fare diverse considerazioni in merito a come gestire le superfici spaziali in base all'utilizzo dell'applicazione, ma in generale quando deve essere interpretato un evento di modifica delle superfici spaziali generato dal surface observer, possono essere semplificate in tre punti:

- Se una superficie con un ID sconosciuto entra nel campo visivo del visore deve essere considerata una nuova superficie spaziale.
- Se una superficie con un ID conosciuto che ha subito un aggiornamento entra nel campo visivo del visore deve essere considerata una superficie spaziale aggiornata.
- Se una superficie con un ID conosciuto esce dal campo visivo del visore viene considerata una superficie spaziale rimossa.

Dopodiché le applicazioni devono fare delle considerazioni su come gestire al meglio le varie mesh per ognuno dei casi sopra citati.

Nel caso del primo punto, cioè quando entra nel campo visivo del visore una nuova superficie spaziale, la mesh viene richiesta immediatamente in quanto potrebbe fornire dati importanti per l'utente, tuttavia dovrebbe essere lasciata la priorità alle superfici vicino e direttamente di fronte al visore. Nel caso in cui, per vari motivi, la mesh non sia necessaria allora non dovrebbe essere richiesta per risparmiare risorse.

Nel caso del secondo punto, cioè quando entra nel campo visivo del visore una superficie spaziale che ha subito un aggiornamento, come nel punto precedente dovrebbe essere data priorità alle superfici vicino e direttamente di fronte al visore ed allo stesso tempo lasciare una priorità ancora maggiore al caricamento di nuove superfici spaziali. In questo caso, visto che gli update sono meno importanti di nuove zone, potrebbe essere considerato di limitare il bandwidth con cui processare questi cambiamenti oppure, nel caso di modifiche minori e di aggiornamenti al di fuori dell'area di interesse dell'utente, potrebbe essere considerato di non processarli proprio.

Infine nel caso del terzo punto, cioè quando esce dal campo visivo del visore una superficie spaziale, in generale le mesh vengono scartate immediatamente ma ci sono casi, come quando l'applicazione ipotizza che tale superficie possa ritornare nel campo visivo poco dopo, in cui può essere considerato di mantenerla in memoria per evitare successivi caricamenti con costo computazionale maggiore. In casi particolari in cui l'applicazione desidera di creare una mappatura ambientale di larga scala si dovrebbe considerare di non eliminare le mesh dalla memoria, tuttavia potrebbe essere necessario spostare parte di esse su disco a causa di limitazioni di spazio. In quest'ultimo caso, e più raramente in altri casi in cui non si desidera eliminare vecchie superfici, potrebbe essere incontrata una problematica in cui viene creata una nuova superficie spaziale in una posizione simile ad una precedente ma con diverso identificatore e ciò comporta la sovrapposizione di più mesh nella stessa zona. In linea di massima quando non è più utile mantenere delle superfici spaziali in memoria a prescindere dalle considerazioni sopra descritte, come per esempio quando la stanza viene rimpiazzata da un mondo completamente virtuale, esse dovrebbero essere scartate.

## 4.5 Rendering e prestazioni

Quando si ha una mesh ambientale che è stata generata e processata, l'applicazione può finalmente usarla per rendere gli ologrammi più realistici fondendoli con il mondo fisico. Ci sono tre principali modi in cui queste mesh vengono utilizzate per il rendering:

- Un primo modo è per la visualizzazione di superfici, per esempio al fine di proiettare un'ombra creata da un ologramma.
- Un altro modo è per occludere gli ologrammi dietro altri oggetti del mondo reale ed in questo caso la superficie spaziale viene utilizzata per modificare il depth buffer della scena che andrà a

cambiare il modo in cui tutti gli ologrammi successivamente renderizzati vengono visualizzati.

- Infine l'ultimo modo è per modificare come vengono visualizzati gli ologrammi che risultano occlusi da superfici, per esempio si potrebbe voler mostrare il bordo di questi ologrammi anche se sono dietro altri oggetti per permettere all'utente di sapere che sono stati renderizzati anche se sono nascosti.

Un'altra importante considerazione va fatta quando si vogliono renderizzare le mesh delle superfici spaziali, e sono le prestazioni. Visto che la fluidità di queste applicazioni non è un aspetto secondario ma un importante fattore che condiziona l'esperienza finale, è importante comprendere come gestire lo spatial mapping per fare in modo che non impatti troppo sugli FPS<sup>13</sup>.

Ci sono diversi modi in cui è possibile intervenire come per esempio quando l'applicazione necessita di una mesh di una superficie è possibile ridurre la densità dei triangoli in base alla distanza dall'utente o all'importanza di tale superficie. Questo semplice accorgimento aiuta a ridurre sia l'utilizzo della memoria che il costo computazionale del rendering dei vertici per la GPU. Un altro metodo utilizzabile viene chiamato *Frustum Culling* e consiste nell'eliminare le chiamate di rendering di oggetti che sono al di fuori del campo visivo del dispositivo, tuttavia questa tecnica è fatta su singole mesh spaziali, quindi potrebbe essere necessario dividerle in pezzi più piccoli per renderla più efficace stando attenti ad un piccolo particolare che più mesh si hanno e più draw calls devono essere fatte, il che potrebbe aumentare l'utilizzo della CPU. Infine, esiste un'altra tecnica per ridurre l'utilizzo della GPU e consiste nel saltare il rendering di geometria che è occlusa da altre superfici. Visto che gli oggetti renderizzati di fronte all'utente occludono tutto ciò che sta dietro, risulta essere un buon metodo renderizzare in ordine la scena, scartando tutto ciò che non può essere visto.

---

<sup>13</sup> FPS: Frames per second

# Conclusioni

Per la visione odierna la realtà mista è una tecnologia mirata al futuro, per questo motivo investire nella ricerca e nello sviluppo dei dispositivi e del software che abilitano questo tipo di esperienza è importante per avvicinarsi sempre di più a tale futuro. Ovviamente dal lato hardware possono essere fatti grossi passi in merito alle dimensioni, alle prestazioni ed al costo di questi visori, ma anche a livello di sensori per permettere un tracking più semplice ed efficace a livello software, e parlando di quest'ultimo, è possibile anche qui fare miglioramenti come ad esempio nelle tecniche di sviluppo. Lo scopo di questa tesi è stato analizzare come si approccia lo sviluppo e la creazione di contenuti nello spazio per applicazioni di realtà mista, per questo motivo avere una panoramica sulle peculiarità e sfide poste da queste esperienze è il modo migliore per iniziare sviluppare. Visto il processo produttivo differente di queste applicazioni, essere a conoscenza delle metodologie di design porta vantaggi nelle successive fasi di progetto, in quanto correggere errori derivanti da fasi precedenti ha un costo incrementale man mano che si procede con lo sviluppo. Aver visto i possibili sistemi di coordinate con cui posizionare gli ologrammi serve per capire i modi efficaci con cui posizionare i contenuti nello spazio nelle varie situazioni in cui uno di essi è più adatto degli altri. Di fatto per un tipo di esperienza in realtà mista da seduto è più indicato utilizzare uno stationary frame of reference, se si vuole utilizzare l'intera stanza potrebbe essere utile lo stage frame of reference, mentre se si desidera avere un'esperienza di vaste dimensioni le ancore spaziali sono un ottimo modo per mantenere stabili gli ologrammi su lunghe distanze. Infine avere una panoramica di come si analizza lo spazio attraverso l'utilizzo di una mappatura spaziale aiuta a comprendere come fondere gli ologrammi con il mondo reale rispettando le interazioni fisiche che tutti noi conosciamo. In aggiunta, vista l'importanza di una gestione efficiente della memoria e delle prestazioni, viene anche descritto come gestire i dati di queste mappe, come vengono utilizzate per il rendering e come manipolarle per mantenere prestazioni adeguate. Per concludere quindi, la speranza è che sempre più sviluppatori inizino a creare contenuti per questo tipo di

tecnologia in modo che venga effettuata sempre più ricerca e vengano investite sempre più risorse in questo tipo di progetti per permettere ad un ipotetico futuro dove tutti possiedano ed utilizzino questo tipo di visori su base giornaliera per migliorare la nostra percezione del mondo, incrementare la nostra produttività ed espandere i nostri limiti creativi.

# Bibliografia

- [1] Microsoft, “*Microsoft Mixed Reality Developer Portal*”,  
<https://docs.microsoft.com/en-gb/windows/mixed-reality/>
- [2] Magic Leap, “*Magic Leap 1 Developer Portal*”,  
<https://developer.magicleap.com/en-us/home>
- [3] M. Pell, “*Envisioning Holograms*”, 2017
- [4] Google, “*Google ARCore*”,  
<https://developers.google.com/ar/discover>
- [5] E. Costanza, A. Kunz, M. Fjeld, “*Mixed Reality. A survey*”, 2009
- [6] A. Bulling, O. Cakmakci, K. Kunze, J. M. Rehg, “*Eyewear Computing – Augmenting the Human with Head-Mounted Wearable Assistants*”, 2016
- [7] D. Schmalstieg, T. Höllerer, “*Augmented-Reality - Principles and Practice*”, 2016
- [8] S. Aukstakalnis, “*Practical Augmented Reality*”, 2016
- [9] Microsoft, figura 1.2,  
<https://blogs.windows.com/windowsexperience/2018/03/08/how-mixed-reality-is-changing-the-game-for-healthcare-from-performing-live-surgeries-to-delivering-ultrasounds-in-3d/>
- [10] Haptic, figura 1.3,  
<https://www.teamviewer.com/en-us/augmented-reality/smartglasses/>
- [11] ifixit, figura 1.4,  
[https://it.ifixit.com/Device/Magic\\_Leap\\_One](https://it.ifixit.com/Device/Magic_Leap_One)

[12] techgig, figura 4.3,

<https://content.techgig.com/arcore-toolkit-gets-depth-api-to-create-ar-depth-maps-and-occlusion/articleshow/72741542.cms>