



POLITECNICO DI TORINO
Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Interfacce 3D per la realtà virtuale e la realtà mista

Relatore

prof. Fabrizio Lamberti

Candidato

Luca Micelli

Matricola: 241926

SESSIONE DI DICEMBRE 2019

Sommario

Ad oggi ci sono molte tecnologie che permettono di interfacciarsi con la Realtà Virtuale: esistono dispositivi di Augmented Reality (AR), Virtual Reality (VR) e Mixed Reality (MR). L'interattività con l'ambiente virtuale gioca un ruolo importante nel definire il grado di immersione dell'utente e accrescere il suo senso di presenza.

Lo scopo di questo lavoro di tesi riguarda lo studio, l'approfondimento e lo sviluppo di nuovi modi di interagire in un ambiente virtuale.

A seconda della tecnologia utilizzata, i contenuti multimediali vengono presentati all'utente in maniera differente, e di conseguenza le modalità di interazione con essi cambiano.

Quindi, per proporre una soluzione più adatta alla determinata tecnologia, sono stati condotti 2 lavori, uno in MR e l'altro in VR.

La Mixed Reality è una tecnologia che combina le caratteristiche di AR e VR: permette di visualizzare elementi virtuali (VR) che oltre ad essere proiettati sul mondo reale (AR), interagiscono con esso.

Microsoft HoloLens è il primo computer indossabile per la Mixed Reality olografica, ed è un ottimo esempio col quale condurre lo studio sull'interazione in MR. Grazie alla sua innovativa tecnologia e versatilità d'utilizzo, HoloLens è oggetto di sviluppo e ricerca in diversi ambiti. Ha introdotto un nuovo metodo di interazione con l'ambiente virtuale che permette all'utente di mantenere le mani libere da controller, ma allo stesso tempo fornisce delle azioni poco naturali ed intuitive. Alcuni ricercatori hanno proposto interessanti soluzioni a questo problema, molte delle quali però prevedono l'utilizzo di tecnologie esterne che ne limitano lo spazio d'azione.

L'obiettivo del primo lavoro è quello di permettere all'utente di interagire con gli elementi virtuali utilizzando il proprio braccio con maggior naturalezza.

Verranno elencati i vari passi che hanno portato alla creazione di un Wearable Custom Device che estende le funzionalità di interazione di HoloLens.

Avendo in mente alcuni obiettivi, come non limitarne le proprietà wireless e

contenere la complessità del sistema, verranno discusse le scelte implementative adottate e come si sono evolute nelle varie iterazioni di progettazione. L'idea alla base è quella di tracciare la rotazione del braccio e avambraccio dell'utente in modo da creare un modello virtuale che segua quello reale. Per fare ciò sono stati utilizzati due sensori di movimento (IMUs) collegati ad Arduino, che si occuperà di elaborare i dati di rotazione dei sensori e inviarli ad Hololens via WiFi. Grazie alle dimensioni compatte e la sua versatilità, Arduino è la piattaforma ideale per una prototipazione veloce. Verranno discussi i vari passi effettuati per l'implementazione della comunicazione fra Hololens e Arduino: dovendo escludere una connessione cablata per motivi di incompatibilità, sono state testate connessioni Bluetooth e WiFi, creando un pattern di pacchetti personalizzato.

Si è utilizzato Unity come motore grafico 3D per creare un ambiente virtuale d'esempio dove provare le capacità del dispositivo creato. Comparando i tempi necessari a svolgere un task utilizzando dapprima solamente Hololens con le gestures integrate, poi Hololens con il custom device, si è notata una maggiore velocità oltre che naturalezza nell'eseguire i movimenti, sebbene resti ancora un prototipo. Inoltre, l'informazione aggiunta riguardo la posizione del braccio nello spazio virtuale, apre la possibilità all'implementazione di nuove Interfacce Utente 3D.

Quando si parla di Virtual Reality, ci si riferisce ad esperienze totalmente immersive: l'utente è circondato da un ambiente completamente virtuale, simulato da un computer, e non è in grado di vedere il mondo reale perché indossa un visore oclusivo (HMD - Head Mounted Display). Fornire un buon senso di immersione è fondamentale per questa tecnologia, e i fattori che lo determinano sono vari. L'interattività risulta essere uno di questi e le dinamiche di interazione fra l'utente e il mondo virtuale differiscono da quelle in AR, in quanto non è possibile vedere il mondo reale e quindi il proprio corpo.

Diverse ricerche dimostrano come un altro fattore, che viene spesso sottovalutato, gioca un ruolo molto importante nel determinare il grado di immersività: il feedback tattile. Infatti, molte tecnologie VR forniscono stimoli unicamente audiovisivi, trascurando il senso del tatto. I dispositivi aptici possono fornire un feedback in diversi modi: ad esempio esercitando una forza che si oppone al moto dell'utente, oppure semplicemente trasmettendo informazioni sulla loro forma, superficie, durezza ecc. Vengono catalogati come Active Haptics qualora facciano queste azioni in maniera controllata da un computer o utilizzando motori di vario genere. Viceversa, nel caso fossero energeticamente passivi, vengono chiamati appunto Passive Haptics, e

sono generalmente caratterizzati da una minore complessità costruttiva. L'obiettivo del secondo lavoro è quello di sviluppare un nuovo metodo di interazione in VR che possa aumentare il senso di immersione dell'utente grazie a feedback tattili, ponendo particolare interesse ai Passive Haptics. Verranno elencati i lavori di diversi ricercatori che negli ultimi anni hanno proposto soluzioni interessanti in questo ambito. In questo modo è possibile capire quali considerazioni abbiano portato alla soluzione che verrà proposta, e alla scelta della tecnologia di tracciamento da utilizzare con HTC Vive. Diversi studi hanno dimostrato che non è richiesta un'alta fedeltà nella somiglianza fra un oggetto reale e la sua versione virtuale affinché l'utente si accorga di impugnare un oggetto diverso. Detto ciò, l'idea alla base della soluzione è quella di creare degli oggetti (proxy) passivi facilmente riconfigurabili e riutilizzabili in diversi scenari. La linea di prodotti Lego Mindstorm soddisfa pienamente questi requisiti, fornendo diversi strumenti per costruire oggetti di vario tipo in maniera facile e veloce. Affinché ogni oggetto virtuale segua il movimento della controparte reale è necessario un sistema di tracciamento. A tal fine, sono stati utilizzati degli HTC Vive Trackers: sensori che collegati in maniera solidale con gli oggetti da tracciare, forniscono informazioni di posizione e rotazione accurate grazie alla suite HTC Vive. Il passo successivo consiste nella scelta del sistema di tracciamento delle mani. Affinché il sistema progettato sia utilizzabile, le mani devono infatti essere libere per poter afferrare gli oggetti. Sono state studiate diverse tecnologie, e inizialmente si è deciso di utilizzare il Leap Motion. È un dispositivo ben collaudato nel mondo della realtà virtuale. Grazie a 2 telecamere e 3 LED Infrarossi, il Leap Motion è in grado di tracciare la posizione delle mani e delle dita poste di fronte alla camera. Purtroppo però, quando si impugna un oggetto, il tracciamento diventa instabile e i suoi emettitori ad infrarossi interferiscono col tracciamento degli HTC Vive Trackers usati per tracciare gli oggetti. Per questo motivo si è deciso di testare un'altra tecnologia, più recente, che si basa su principi completamente differenti: Manus VR. Sono un paio di guanti wireless dotati di 2 IMUs (per la rotazione del polso e del pollice) e 5 estensimetri (uno per ogni dito) che ne misurano la flessione. La fedeltà del tracciamento è inferiore a quella del Leap Motion, ma la sua robustezza alle occlusioni e la compatibilità con i Vive Trackers li ha resi adatti allo scopo. Verranno descritti i processi di calibrazione che hanno ridotto il grado di errore presente nel tracciamento. Successivamente troviamo le fasi di progettazione e implementazione di uno scenario dove testare il sistema. Studiando la forma e la funzionalità dei

proxy necessari, è stato possibile contenere il numero di oggetti da creare, portando 2 vantaggi principali: il riutilizzo dello stesso oggetto per funzioni diverse (senza che l'utente ne sia consapevole) e un minor numero di Vive Tracker necessari.

Le due soluzioni realizzate vengono presentate come prototipi di sistemi d'interazione che, utilizzando componenti semplici e facilmente accessibili come Arduino e i blocchi Lego, espandono le capacità delle tecnologie immersive esistenti, in particolare Realtà Mista e Realtà Virtuale.

Indice

Elenco delle figure	10
1 Introduzione	12
1.1 Definizioni	12
1.1.1 Realtà Virtuale	13
1.1.2 Realtà Aumentata	13
1.1.3 Realtà Mista	13
1.1.4 Spettro Realtà-Virtualità	14
1.2 Obiettivi	15
2 Stato dell'arte	17
2.1 Realtà Mista	18
2.2 Realtà Virtuale	20
2.2.1 Dispositivi Aptici Attivi	21
2.2.2 Dispositivi Aptici Passivi	23
I Interazione in Realtà Mista	25
3 Progettazione	27
3.1 HoloLens	27
3.1.1 Hardware	27
3.1.2 Software	28
3.1.3 Spatial Mapping	29
3.1.4 Connettività	30
3.1.5 Metodi d'interazione	30
3.2 Progetto	31
3.2.1 Analisi della tecnologia	31
3.2.2 Idea e principio di funzionamento	32

3.2.3	Strumenti e architettura	33
4	Realizzazione	35
4.1	Hardware	35
4.1.1	HoloLens	35
4.1.2	Arduino	36
4.1.3	Espansioni	36
4.2	Ambiente di sviluppo	40
4.2.1	HoloLens e Unity 3D	40
4.2.2	Arduino IDE	44
4.2.3	Connessione con IMU	45
4.2.4	Istruzioni per lettura dei dati	46
4.3	Comunicazione di rete	50
4.3.1	Bluetooth	50
4.3.2	WiFi	53
4.4	Implementazione su HoloLens	58
4.4.1	Calibrazione	59
4.4.2	Scenario	60
II	Interazione in Realtà Virtuale	63
5	Progettazione	65
5.1	HTC Vive	65
5.2	Idea ed obiettivo	66
5.3	Architettura	67
6	Realizzazione	71
6.1	Applicazione	71
6.1.1	Ambiente di sviluppo	71
6.1.2	Struttura	72
6.2	Strumenti aptici	73
6.2.1	Interfacce	74
6.2.2	Tracciamento degli oggetti	76
6.3	Tracciamento delle mani	76
6.3.1	Leap Motion	77
6.3.2	Manus VR	79
7	Conclusioni	83

Elenco delle figure

1.1	Spettro AR-MR-VR di Milgram [9]	14
2.1	Schema dei tipi di dispositivi aptici attivi: a) <i>grounded</i> , b) <i>ungrounded</i> e c) <i>wearable</i> . Le frecce blu indicano le forze applicate all'utente, quelle rosse le forze di reazione. [7]	22
3.1	Microsoft HoloLens	28
3.2	Schema del sistema di tracciamento	33
4.1	Modulo MPU-6050 implementato nella scheda integrata GY-521 per Arduino	37
4.2	Modulo Bluetooth HC-06 per Arduino	38
4.3	Modulo WiFi Adafruit HUZZAH ESP8266	39
4.4	Componenti utilizzati dell'HoloToolkit	42
4.5	Esempio di <i>sketch</i> nell'Arduino IDE	45
4.6	Schema di collegamento del modulo GY-521 (IMU) alla scheda Arduino UNO	46
4.7	Esempio di <i>sketch</i> per il la lettura dei dati RAW dal modulo GY-521	47
4.8	Valori RAW letti in output dal sensore GY-521	48
4.9	Valori calibrati di orientamento in gradi letti in output dal DMP del sensore GY-521	49
4.10	Schema di collegamento del Modulo HC-06 con Arduino Uno	50
4.11	<i>Sketch</i> per la configurazione dei parametri iniziali del modulo HC-06	51
4.12	Schema di connessione diretta dell'IMU GY-521 alla scheda Adafruit Huzzah ESP8266 WiFi	54
4.13	Schema di connessione di due IMU alla scheda Adafruit Huzzah ESP8266 WiFi	56
4.14	Tabella che raffigura le parti di cui è composto un <i>OSC Bundle</i> contenente due o più messaggi	57

4.15	Schema della struttura del pacchetto utilizzato. ID è un <i>char</i> (1 byte) e identifica l'IMU.	58
4.16	Fase di calibrazione degli IMU con HoloLens	60
4.17	Comparazione dei sistemi d'interazione	61
5.1	Componenti del sistema HTC Vive	66
5.2	Schema dell'interfaccia di interazione tattile in Realtà Virtuale	68
6.1	Schermate dei vari livelli dell'applicazione	73
6.2	Oggetti fisici e virtuali	75
6.3	A sinistra la composizione degli oggetti fisici, a destra i corrispettivi elementi virtuali	77
6.4	Esempio di utilizzo dei Manus VR con l'interfaccia d'interazione tattile	79
6.5	Pannello di calibrazione dell'orientamento dei Manus VR	81

Capitolo 1

Introduzione

Con l'avanzamento della tecnologia, la Realtà Virtuale è diventata sempre più diffusa e accessibile a tutti. L'esperienza di immersione che essa è in grado di fornire ha aperto le strade a innumerevoli nuove applicazioni, prima impensabili. Oggi è un concetto che viene applicato in ogni ambito: educazione, addestramento, medicina, turismo, intrattenimento e così via. Inoltre, con l'avanzamento della Computer Vision, ha visto evoluzioni come la Realtà Aumentata e poi la Realtà Mista. La modalità di fruizione di questi contenuti si allontana dal tradizionale monitor con mouse e tastiera, e richiede nuove interfacce di comunicazione appositamente ridisegnate. L'interazione fra l'utente e l'ambiente virtuale è oggetto di continue e numerose ricerche negli ultimi tempi, in quanto è un elemento molto importante per determinare l'immersività dell'esperienza. In questo elaborato di tesi verranno presentati due lavori che hanno portato alla creazione di nuove soluzioni di interazione con un ambiente virtuale. Nella prossima sezione verranno brevemente descritti alcuni termini fondamentali, utili a comprendere al meglio il resto dell'elaborato e le scelte effettuate.

1.1 Definizioni

In numerosi studi, articoli e pubblicazioni si utilizzano spesso questi termini: Realtà Virtuale, Realtà Aumentata e Realtà Mista. Purtroppo, a causa delle continue evoluzioni di queste tecnologie, può succedere di confonderne il significato. Perciò verranno descritte brevemente qui di seguito, in modo da capire quali caratteristiche le accomunano e in cosa invece si differenziano.

1.1.1 Realtà Virtuale

La Realtà Virtuale (alla quale d'ora in poi si farà riferimento con VR) descrive quelle esperienze che vedono l'utente totalmente circondato da un mondo virtuale (VE, Virtual Environment). L'equipaggiamento di un sistema VR solitamente prevede un visore (HMD, Head Mounted Display), una o più basette per il tracciamento e dei controller. L'HMD è di tipo totalmente occlusivo, cioè non permette all'utente di vedere il mondo reale, ma solamente immagini renderizzate con un PC. Esempi di dispositivi VR più diffusi sono HTC Vive e Oculus Rift.

1.1.2 Realtà Aumentata

La Realtà Aumentata (AR) consiste nel sovrapporre informazioni generate da un computer (testo, immagini, video) sulla visione del mondo reale. Per fare questo l'utente può utilizzare HMD parzialmente occlusivi oppure tradizionali display per esperienze non immersive. Lo scopo di queste tecnologie è quello di agevolare l'utente nella fruizione di informazioni digitali, proiettandole sul mondo reale. Uno dei primi esempi di tecnologia di questo tipo sono i Google Glasses [*referenza]: un paio di occhiali con un sistema di proiezione integrato che una volta indossati permette all'utente di vedere informazioni digitali (orario, meteo, notifiche ecc.) direttamente proiettate sulle lenti. Successivamente, con l'evoluzione dei dispositivi mobili come smartphone e tablet, l'AR è stato reso accessibile anche su quest'ultimi: varie applicazioni fanno uso della fotocamera integrata per sovrapporre contenuti multimediali sull'immagine del mondo reale. E' importante sottolineare che in questo caso, gli elementi digitali non sono ancorati al mondo reale e quindi non sono in grado di interagire con esso.

1.1.3 Realtà Mista

La Realtà Mista (MR - Mixed Reality), a volte chiamata anche Realtà Ibrida (hybrid reality), può essere considerata come l'unione del VR e AR. E' la tecnologia di più recente evoluzione che permette di generare elementi virtuali, tipici del VR, e proiettarli nel mondo reale come se fossero parte di esso. A differenza dell'AR, gli oggetti digitali interagiscono con quelli reali e l'ambiente fisico in tempo reale. Per provare questo tipo di esperienza sono necessari degli HMD parzialmente occlusivi, che permettano quindi all'utente di continuare a vedere il mondo reale circostante mentre vengono renderizzati elementi virtuali. Questi dispositivi devono inoltre essere in grado di

scansionare l'ambiente reale per fornire al sistema le informazioni necessarie a produrre elementi digitali che si integrino con la realtà. A causa della novità e complessità di questa tecnologia, i dispositivi di MR oggi presenti sul mercato sono ancora pochi e non destinati al mercato commerciale. Un esempio di dispositivo che rappresenta al meglio questa tecnologia è Microsoft HoloLens: un visore parzialmente oclusivo in grado di proiettare elementi virtuali nel mondo reale sotto forma di ologrammi. Altri esempi di dispositivi usciti successivamente e unicamente dedicati alla MR sono Meta 2 e Magic Leap One.

1.1.4 Spettro Realtà-Virtualità

Sebbene VR, AR e MR si basano su tecnologie differenti e offrono esperienze diverse, a volte vengono catalogate nella stessa categoria. Una prima relazione proviene ad esempio dallo studio di Wilson [27], il quale categorizza l'AR come una forma di VR. Nel lavoro di Drascis e Milgram [9] invece vediamo la definizione di queste tecnologie in termini di *"Reality-Virtuality Continuum"*, rappresentato in Figura 1.1. Questo spettro è ancora considerato come punto di riferimento per la classificazione di esperienze che uniscono il mondo reale e quello virtuale. Secondo questa definizione troviamo una vasta gamma di applicazioni fra un ambiente completamente reale ed uno unicamente virtuale.

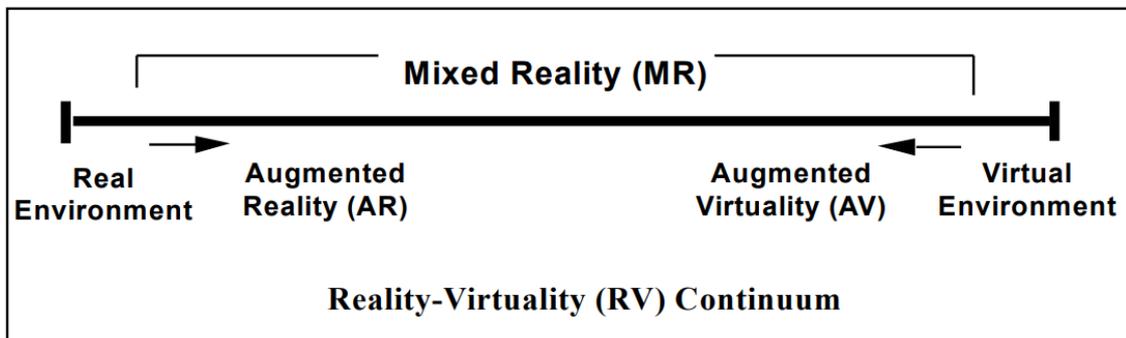


Figura 1.1: Spettro AR-MR-VR di Milgram [9]

Nonostante il concetto di Realtà Mista viene visto come un insieme che racchiude le varie tecnologie nel mezzo, nel corso degli anni ha subito variazioni dovute allo sviluppo di nuovi dispositivi. In questo lavoro di tesi si farà riferimento alla MR seguendo la definizione descritta nella sezione precedente.

1.2 Obiettivi

L'argomento principale trattato in questo lavoro di tesi riguarda l'interazione uomo-computer con tecnologie di realtà virtuale. Queste tecnologie emergenti (VR, AR e MR) sono caratterizzate da una particolare differenza rispetto ai tradizionali computer desktop, sia dal lato costruttivo che nel contesto di utilizzo. Esse spesso richiedono interazioni molto diverse da quelle a cui siamo abituati con i classici computer. Quando dobbiamo interagire con un ambiente virtuale che ci circonda, le tecniche di interazione che prevedono mouse e tastiera non sono più sufficienti. Perciò è necessario progettare paradigmi di interazione uomo-macchina specificatamente per queste nuove tecnologie immersive. Questo argomento presenta una sfida particolarmente interessante, infatti è oggetto di numerosi studi e ricerche negli ultimi anni.

L'obiettivo di questa tesi di laurea è quello di studiare e approfondire i principi di interazione in un ambiente virtuale, col fine di progettare e sviluppare una nuova soluzione.

Studiando i lavori inerenti a questo argomento, ci si è resi conto di come queste tecnologie emergenti presentano sfide che sono legate alla particolare tecnologia. Sebbene VR, AR e MR sono accomunate dal concetto di "realtà" alternativa, esse offrono esperienze basate su principi e tecnologie differenti. Di conseguenza, i modelli di interazione necessitano una progettazione che tenga in considerazione le peculiarità di ogni tecnologia, le finalità di utilizzo e le limitazioni derivanti. Per questo motivo sono state scelte due tecnologie in particolare sulle quali condurre lo studio e la progettazione di nuovi metodi di interazione: la Realtà Mista e la Realtà Virtuale.

Nel primo caso l'obiettivo è quello di fornire uno strumento che permetta un'interazione più naturale, dato che come vedremo dalla letteratura nel prossimo capitolo, questo è risultato essere un punto di interesse per i dispositivi di Realtà Mista.

Nel secondo caso lo scopo è quello di migliorare il senso di immersività nelle esperienze di Realtà Virtuale puntato su un'interfaccia d'interazione che valorizzi il senso del tatto.

Questi lavori verranno esposti rispettivamente nella Parte I e Parte II dell'elaborato, mentre nel prossimo capitolo si discuteranno gli studi più rilevanti presenti in letteratura, analizzando le tecnologie utilizzate, gli obiettivi e le limitazioni evidenziate.

Capitolo 2

Stato dell'arte

Nel capitolo precedente abbiamo visto quali sono le caratteristiche che contraddistinguono le tecnologie di Realtà Virtuale, Aumentata e Mista. In letteratura sono presenti diversi studi volti a confrontare queste tecnologie, puntando principalmente ai vantaggi offerti da ognuna di esse.

Per esempio, uno studio condotto da Boud [5] vede come obiettivo il confronto delle tecnologie di VR, AR e MR usate come strumento di addestramento. Con lo scopo di istruire l'utente ad effettuare un task di assemblaggio, è emerso come il VR abbia facilitato la fruizione dell'esperienza grazie ad un ambiente totalmente simulato, mentre in AR e MR erano necessarie le componenti fisiche sulle quali eseguire le operazioni. D'altro canto però, l'esperienza VR ha dimostrato difficoltà nel fornire un buon senso di immersione a causa della mancanza di un feedback tattile, problematica assente nel caso dell'AR e MR. Quest'ultima però, oltre a richiedere la disponibilità dei componenti fisici (non sempre possibile in una catena di montaggio) ha presentato difficoltà nel tracciare correttamente gli elementi del mondo reale, come gli oggetti o l'utente stesso.

Un' altro esempio interessante lo troviamo nel lavoro di Tang [24], dove viene messo a confronto il senso di presenza fornito dalle varie tecnologie. Sono stati presi in considerazione fattori come *Presenza Spaziale* e *Coinvolgimento e Naturalzza* al fine di effettuare una valutazione. In quanto a Presenza Spaziale, le esperienze di AR/MR hanno registrato un maggior numero di votazioni favorevoli rispetto al VR. Questo viene attribuito al fatto che in AR l'utente mantiene il contatto col mondo reale ed il suo corpo, mentre in VR la mancanza di un avatar realistico in un'esperienza in prima persona, interferisce negativamente col sistema di percezione visivo-spaziale

del cervello umano. Ciò che stupisce è invece la poca differenza di valutazione rispetto a Coinvolgimento e Naturalezza. Sebbene l'ambiente AR/MR riceve valutazioni migliori di quello VR, le differenze non sono significative. Questo viene in parte attribuito al fatto che nonostante l'utente sia portato a movimenti più naturali grazie alla percezione del proprio corpo, il sistema AR/MR non è in grado di tracciarli correttamente e renderizzare elementi virtuali in maniera solidale al mondo reale.

Si evince come VR e MR presentino punti di forza e debolezze specifici per la particolare tecnologia. In particolare emerge come in MR ci sia la necessità di riconoscere i movimenti naturali dell'utente. Mentre in VR, la mancanza di un feedback tattile dell'ambiente virtuale, incide negativamente sul senso di immersione dell'utente.

Nelle prossime sezioni vedremo alcuni studi volti a risolvere le particolari criticità evidenziate rispettivamente in MR e in VR.

2.1 Realtà Mista

A differenza dei sistemi VR, in un'esperienza di Realtà Mista l'utente è in grado di percepire sia il mondo reale che il proprio corpo. Si è osservato come questo doni agli utenti una maggior sicurezza ad eseguire movimenti rispetto al VR. Questa maggiore coscienza del proprio corpo, agevola movimenti più naturali. Questo è sicuramente un aspetto importante da tenere in considerazione durante la progettazione di interfacce di interazione col sistema di Realtà Mista.

Un buon esempio di dispositivo in grado di offrire un'esperienza di MR e sul quale condurre gli studi, è Microsoft HoloLens: headset composto da lenti olografiche e un innovativo insieme di sensori che gli permette un tracciamento spaziale autonomo. Questo approccio è chiamato *inside-out tracking* in quanto permette di determinare la sua posizione nel mondo reale (e di conseguenza in quello virtuale) grazie ad una scansione continua dell'ambiente circostante tramite camere ad infrarossi ed RGB. Questo fornisce agli utenti una maggior libertà di movimento rispetto ai tradizionali sistemi VR limitati dall'area definita dalle basette di tracciamento. HoloLens fornisce inoltre diverse soluzioni di interazione come riconoscimento vocale o gestures a mani libere, ma discuteremo questi metodi più nel dettaglio nella Parte I dell'elaborato.

Grazie alle sue innovative caratteristiche, Microsoft HoloLens è stato utilizzato per condurre studi e sviluppare applicazioni di Realtà Mista in diversi

ambiti: Romaniuk [20] dimostra la sua utilità nell'agevolazione dell'apprendimento scolastico, Evans [10] presenta le sue potenzialità nell'addestramento di operatori industriali, oppure ancora Kors [14] propone soluzioni di intrattenimento. Ha riscontrato particolare interesse nello sviluppo di applicazioni in ambito medico. Adabi [3] sviluppa una soluzione che permette di impiegare HoloLens come strumento di misurazione di parti del corpo, al fine di agevolare i dottori nella pianificazione di un'operazione chirurgica. Tepper [26] presenta interessanti risultati sui benefici portati da interazioni a mani libere con HoloLens per visualizzare dati in una sala operativa.

Abbiamo visto come questo strumento di Realtà Mista può essere applicato negli ambiti più vari. Ora ci concentreremo sulle meccaniche di interazione con esso. HoloLens permette interazioni di selezione grazie al tracciamento di *gestures* predefinite con la mano che avvengono nel campo visivo della camera. Questo sistema di interazione, unito al cursore di puntamento legato all'orientamento della testa, può influire negativamente sull'esperienza utente, portando ad un affaticamento del braccio e del collo. Quindi, sebbene i vantaggi di avere interazioni a mani libere siano notevoli, queste interazioni risultano limitanti se messe a confronto con le azioni effettuate nel mondo reale, che invece sfruttano l'intero arco di movimento del braccio, anche fuori dal campo visivo dei sensori.

Bisogna quindi utilizzare delle tecnologie esterne per tracciare questo movimento più ampio del braccio, e inviare queste informazioni ad HoloLens affinché possa produrre un feedback adatto. Ci sono diverse tecnologie che è possibile utilizzare per tracciare il movimento del braccio: si può ricorrere a sistemi di elaborazione di immagini [25], esoscheletri [22] oppure tute per il *motion-capture* [11]. Alcuni lavori presenti in letteratura si sono occupati di arricchire le capacità di interazione di HoloLens sfruttando tecnologie aggiuntive che tracciano il corpo umano, col fine di fornire forme di interazione più intuitive.

Davies [8] propone un interessante approccio che vede l'utilizzo di un Microsoft Kinect ¹, in combinazione con HoloLens, per un tracciamento *full-body*. In questo modo, non solo è possibile implementare nuove *gestures* e paradigmi di manipolazione, ma si potrebbero mappare i movimenti dell'utente su un avatar virtuale per simulare la presenza in sessioni di collaborazione remota. Lo svantaggio principale però è dovuto al riconoscimento

¹<https://developer.microsoft.com/it-it/windows/kinect>

basato su una camera fissa: questa restringe i movimenti dell'utente all'interno del suo campo visivo, limitando notevolmente le funzionalità wireless di HoloLens.

Un altro esempio viene dal lavoro di Kervegant [13]: sono stati integrati un Leap Motion² per un tracciamento più accurato della mano, e un *touch development kit* di Ultrahaptics³ per fornire un feedback tattile. Questi dispositivi, in combinazione con HoloLens, offrono un'esperienza sicuramente interessante dal punto di vista interattivo, ma anche in questo caso si incorre nelle limitazioni spaziali dovute al cablaggio e hardware aggiuntivo. Leap Motion, così come il kit Ultrahaptics, non possono essere collegati direttamente all'HMD, ma devono utilizzare un pc come proxy.

Queste limitazioni sull'area di interazione, sono più o meno importanti in base al contesto in cui vengono applicate. Ci sono scenari che non risentono dei vincoli di mobilità, ma non limitare l'area d'interazione di HoloLens è stato uno degli obiettivi tenuti in considerazione durante il lavoro di ricerca, come vedremo nel Capitolo 3.

2.2 Realtà Virtuale

La tecnologia di Realtà Virtuale offre immersione ed interazioni attraverso un ambiente completamente generato dal computer, cercando di riprodurre un'esperienza simile alla vita reale con la stimolazione di diversi sensi dell'uomo. Un aspetto caratterizzante di un'esperienza VR è il senso di presenza che il sistema offre all'utente. A differenza dell'AR, dove è possibile mantenere contatto visivo col mondo reale, in VR questo senso viene stimolato unicamente da immagini renderizzate. Risulta necessaria quindi una maggiore attenzione e complessità nella generazione di segnali che stimolano i sensi dell'utente e lo inducono a credere di essere "realmente presente" nel mondo virtuale. Nell'articolo di Schueme [21], troviamo delle considerazioni interessanti riguardo il legame fra il concetto di presenza e l'interazione in VR. Scheume riporta come alcuni aspetti dell'interazione, ad esempio la velocità di risposta dell'ambiente virtuale ai comandi dell'utente, contribuiscono sicuramente a determinare il livello di presenza, ma fino ad un certo punto. Bisognerebbe analizzare anche altri aspetti, come la quantità e le modalità di input messi a disposizione dell'utente, tenendo in considerazione

²<https://www.leapmotion.com/technology/>

³<https://www.ultrahaptics.com/products-programs/>

l'importanza del senso di presenza durante la progettazione di interfacce di interazione uomo-macchina.

La componente d'interazione delle interfacce utente in VR richiede quindi il coinvolgimento di tutti i sensi. La riduzione dei costi delle tecnologie VR ha portato ad una rapida diffusione di questi dispositivi a livello commerciale, dando la possibilità a chiunque di provare la realtà virtuale. Ad oggi la maggior parte delle innovazioni in ambito commerciale fanno leva soprattutto sul comparto audio e video dei dispositivi VR, lasciando largo margine di miglioramento sulla stimolazione degli altri sensi. Come evidenziato da Boud [5], alcuni studi dimostrano come la mancanza di stimoli tattili sia un fattore importante che incide negativamente sull'immersività di un'esperienza VR. Lo studio e lo sviluppo di interfacce per produrre feedback tattile per la realtà virtuale è quindi un tema particolarmente importante dal punto di vista della ricerca, che negli ultimi anni vede sempre più interesse dagli studiosi.

I dispositivi pensati per stimolare il senso del tatto, sono chiamati dispositivi aptici (*haptic devices*) e possono essere classificati come **attivi** o **passivi** a seconda della tecnologia utilizzata.

- *Dispositivi aptici attivi*: dispositivi energeticamente attivi, che per esercitare forze sull'utente utilizzano attuatori controllati elettronicamente (Zenner [28]).
- *Dispositivi aptici passivi*: dispositivi energeticamente passivi che forniscono un feedback attraverso la loro forma, il peso o altre proprietà fisiche. Swanson [23] li definisce come interfacce che usano passivamente degli attuatori che possono solo rimuovere, conservare o reindirizzare l'energia cinetica all'interno di un sistema.

2.2.1 Dispositivi Aptici Attivi

I dispositivi attivi sono generalmente più sofisticati e tecnicamente più complessi. In base alla modalità con cui esercitano forze sull'utente, questi dispositivi possono essere catalogati ulteriormente come *grounded* se esercitano una forza da terra, *ungrounded* se basate sul corpo dell'utente oppure *wearable* se non esercitano forze sulle articolazioni (Figura 2.1).

Uno dei primi e più famosi esempi di dispositivi aptici attivi è il PHANTOM [16]. Come buona parte dei dispositivi attivi, è di tipo *grounded* e fu il primo a fornire una feedback su 3DOF: esercita una forza vettoriale controllata elettronicamente sulla punta delle dita dell'utente. Sebbene riesce a

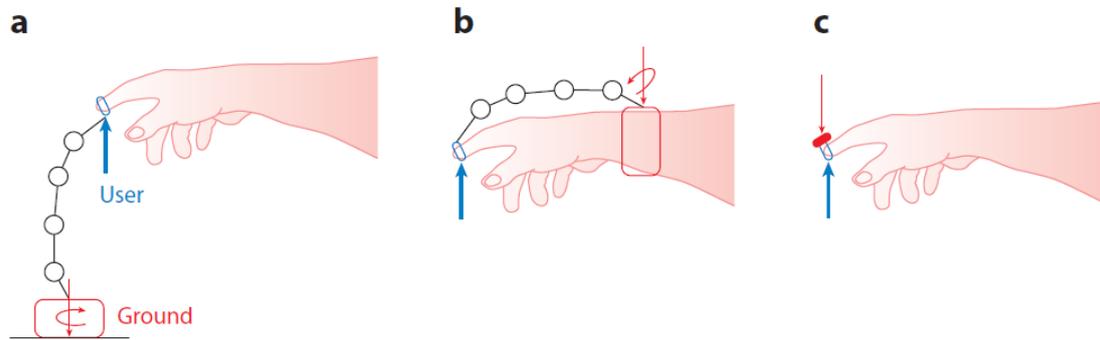


Figura 2.1: Schema dei tipi di dispositivi aptici attivi: a) *grounded*, b) *ungrounded* e c) *wearable*. Le frecce blu indicano le forze applicate all'utente, quelle rosse le forze di reazione. [7]

produrre forze controllate in maniera molto precisa, queste sono limitate ad una piccola regione di spazio descritti dal movimento del polso.

Esistono anche vari esempi di dispositivi ungrounded e wearable. Esempi tipici sono i cosiddetti *esoscheletri*. Aiple [4] ha presentato CyberGrasp: un esoscheletro per la mano realizzato per esercitare forze sulle dita dell'utente in modo da bloccarne il moto e simulare la presenza di un oggetto fisico. L'esoscheletro pesa 350g ed è composto da una serie di attuatori meccanici in grado di applicare una forza massima di 12N per dito, e da un'unità di controllo esterna al quale è collegato. Sullo stesso principio di funzionamento, Hinchet [12] ha creato un guanto aptico più leggero e flessibile. Su ogni dito sono presenti delle fascette metalliche elastiche che generano una forza d'attrito controllata elettricamente.

Il principale vantaggio dei dispositivi aptici attivi risiede nella loro capacità di esercitare forze in maniera precisa e controllata. D'altro canto però, oltre a permettere movimenti relativamente limitati nello spazio, sono dispositivi complessi meccanicamente e computazionalmente. Sono necessari algoritmi sofisticati per calcolare la giusta quantità di forza da applicare, ed essendo attivi, la sicurezza è un fattore critico in quanto un malfunzionamento potrebbe nuocere all'utente.

2.2.2 Dispositivi Aptici Passivi

I dispositivi passivi, a differenza di quelli attivi, sono caratterizzati da una minore complessità costruttiva e di funzionamento. Non vengono coinvolti attuatori controllati elettronicamente e non si esercitano forze esterne sull'utente.

Analogamente a quelli attivi, possiamo catalogare i dispositivi passivi in tre categorie in base al loro principio di funzionamento:

- *Wearable*: attrezzature indossabili dall'utente
- *Graspable*: strumenti da tenere in mano
- *Physical props*: oggetti fisici esterni all'utente

Un semplice esempio ma molto rappresentativo della prima categoria di dispositivi aptici passivi indossabili viene presentato da Achibet con Elastic-Arm[1]: è basato su armatura elastica che collega la mano dell'utente con la sua spalla. A questo punto, all'aumentare dell'estensione del braccio, aumenta la resistenza avvertita dall'utente sulla mano. Questo principio, applicato alle tecnologie di Realtà Virtuale, apre la possibilità ad innumerevoli tecniche di interazioni 3D che coinvolgono il senso del tatto.

Fra i dispositivi *graspable* un interessante soluzione viene presentata sempre da Achibet con il "Virtual Mitten" [2]. Questo approccio consiste nel tenere in mano un oggetto appositamente progettato per essere impugnato che esercita una forza verso l'esterno quando viene stretto, grazie a delle molle con cui è costruito. Utilizzando dei tracker ottici esterni per tracciare il grado di apertura di questo strumento, è possibile far sapere al sistema quando l'utente sta stringendo o meno il polso. In questo modo sarà possibile creare l'illusione di aver afferrato un oggetto e di poterlo manipolare nello spazio mentre si avverte un feedback tattile nel palmo della mano.

Un esempio simile lo troviamo nel lavoro di Pihuit con "Virtual Clay"[19]. Viene mostrato come è possibile simulare la sensazione di pressione con la mano utilizzando una pallina di spugna. Su quest'ultima sono stati applicati 5 sensori di pressione (uno per ogni dito), e quando l'utente la impugna sarà possibile misurare la forza applicata e restituire feedback visivi coerenti.

Sebbene siano tutte soluzioni relativamente semplici ed accessibili, uno svantaggio degli esempi trattati finora risiede nella mancanza di versatilità. Ogni soluzione permette di provare uno specifico tipo di feedback tattile ("spingere" e "tirare" nel primo esempio, oppure "stringere" negli altri

due) mantenendo le sue mani occupate da un determinato strumento ed impedendogli di interagire con l'ambiente virtuale in modo differente.

La terza categoria di dispositivi aptici passivi prevede l'utilizzo di oggetti fisici esterni come fonte di feedback aptico. Nello specifico, verranno utilizzati degli oggetti fisici veri e propri per simulare le proprietà del corrispondente oggetto virtuale. Ovviamente è necessario che l'oggetto reale abbia caratteristiche simili a quello virtuale in quanto a dimensioni, forma e peso. Ma per capire con esattezza quali proprietà fisiche influiscono maggiormente sulla percezione dell'utente, sono state condotte diverse ricerche ed esperimenti. Martin [15] nel suo studio, si pone l'obiettivo di stabilire quanto la fedeltà degli oggetti reali sia rilevante. L'esperimento prevedeva lo spostamento di oggetti nello scenario di una catena di montaggio in Realtà Virtuale, sostituendo di volta in volta i modelli fisici con altri più o meno simili alla controparte virtuale. I risultati mostrano come al crescere della fedeltà dell'oggetto fisico, non corrisponde necessariamente un miglioramento delle prestazioni dell'utente. Inoltre è interessante notare come alcune proprietà fisiche siano più rilevanti di altre: ad esempio è risultato che simulare il corretto peso del modello fisico, migliora l'esperienza soltanto se questo assomiglia anche nella forma. Quest'ultima proprietà risulta quindi più influente della prima.

Uno svantaggio di questa tecnica di dispositivi aptici passivi, è la mancanza di generalità dei modelli fisici, richiedendo diversi oggetti reali per altrettanti oggetti virtuali.

Un'interessante soluzione a questo problema viene proposto da Cheng con "iTürk"[6]: mostrano come due oggetti fisici (una palla sospesa ed un modello ripiegabile) possono essere riutilizzati nell'ambiente virtuale per simulare diversi oggetti a seconda della scena. Inoltre, l'esperienza è stata progettata per far sì che l'utente non si accorga di riutilizzare più volte gli stessi oggetti fisici.

Parte I

Interazione in Realtà
Mista

Capitolo 3

Progettazione

In questo capitolo verrà esposto il lavoro svolto nel campo della Realtà Mista, illustrando nel dettaglio le tecnologie utilizzate e le varie scelte di progettazione.

Il dispositivo principale sul quale è stato condotto lo studio è Microsoft HoloLens, che come visto nella [sezione 2.1](#), risulta essere uno dei migliori esempi di tecnologia in grado di fornire un'esperienza di Realtà Mista. Per comprendere al meglio le scelte progettuali che verranno illustrate, è necessario conoscere le funzionalità di questo dispositivo.

Nella prossima sezione verranno brevemente descritte le principali caratteristiche di HoloLens, sia dal punto di vista hardware che software, ponendo particolare attenzione alle modalità di interazione che esso offre.

3.1 HoloLens

HoloLens è un dispositivo di Realtà Mista sviluppato e prodotto da Microsoft, commercializzato nel 2016. È stato il primo computer indossabile "all-in-one" provvisto di display olografico. Al suo interno, infatti, integra tutti i componenti necessari al corretto funzionamento: lenti semitrasparenti dove proiettare le immagini, altoparlanti, processore, videocamere, sensori di movimento e batteria. Le principali innovazioni tecnologiche risiedono proprio nel display olografico e nella capacità di tracciamento spaziale.

3.1.1 Hardware

Come è possibile vedere nell'immagine [3.1](#), sulla parte frontale ci sono delle lenti semitrasparenti, dove vengono proiettate le immagini renderizzate dal



Figura 3.1: Microsoft HoloLens

processore a bordo, denominato HPU (Holographic Processing Unit), ed un insieme di videocamere per il tracciamento spaziale. Quest'ultimo è possibile grazie ad un elaborato processo chiamato "inside-out tracking": utilizzando le informazioni raccolte da 2 camere ad infrarossi, 2 camere di profondità, 1 camera RGB ed un sensore di movimento (IMU), è in grado di tracciare la sua posizione ed orientamento in relazione all'ambiente circostante, senza la necessità di basette di tracciamento esterne. Questa autonomia permette quindi di utilizzarlo in qualsiasi ambiente senza limitazioni spaziali dovute a cavi o confini prefissati. Completano l'allestimento hardware 4 microfoni per il riconoscimento vocale, una batteria per un utilizzo attivo stimato di 2-3 ore, 64Gb di memoria di archiviazione e 2Gb di RAM.

3.1.2 Software

Il sistema operativo montato a bordo è basato su Windows 10, ed era inizialmente chiamato Windows Holographic. Con la successiva introduzione di nuovi dispositivi di Realtà Virtuale da parte di Microsoft, il sistema operativo si è arricchito con nuove funzioni utili ai nuovi headsets immersivi, diventando una piattaforma comune per Realtà Mista e Virtuale, denominato Windows Mixed Reality. La versione eseguita su HoloLens rimane comunque l'unica in grado di supportare gli ologrammi e i vari sistemi di ancoraggio al mondo reale. Le applicazioni possono essere eseguite in 2 modalità principali:

- **Finestra:** in questa modalità le applicazioni vengono proiettate in finestre virtuali 2D. In questo modo possono essere eseguite quasi tutte le applicazioni UWP (Universal Windows Platform), quindi sviluppate con gli stessi strumenti utilizzati per creare app per Windows 10 desktop o Windows Phone. La finestra virtuale dispone anche di apposite "maniglie" che permettono di spostarla nello spazio 3D, ruotarla e ridimensionarla. Esempi sono il browser di navigazione internet Edge, oppure le varie finestre di Impostazioni.
- **Immersiva:** Questa è detta anche modalità Olografica 3D. L'applicazione occupa l'intero spazio che circonda l'utente e può utilizzare le API di Windows Mixed Reality. Può accedere alle funzionalità di Spatial Mapping ([sottosezione 3.1.3](#)), alle informazioni di posizione e orientamento del dispositivo e renderizzare elementi grafici 3D. Le applicazioni possono usare dei motori grafici 3d, come ad esempio Unity, oppure implementarne uno proprio tramite la libreria DirectX. Non è possibile eseguire più di 1 applicazione 3D contemporaneamente.

Lo studio verrà quindi condotto sviluppando applicazioni in modalità **immersiva**, in modo da poter accedere a tutti gli strumenti richiesti per un'esperienza di Realtà Mista.

3.1.3 Spatial Mapping

Una delle funzioni che ha contraddistinto HoloLens come dispositivo di Realtà Mista, è la sua capacità di Spatial Mapping. Questa funzionalità permette di ricreare una rappresentazione dettagliata delle superfici del mondo reale che circondando il dispositivo. In questo modo le applicazioni avranno accesso ad un maggior numero di informazioni ambientali, creando ologrammi più realistici che interagiscono coerentemente con il mondo reale. La ricostruzione delle superfici in 3D è possibile grazie all'elaborazione dei dati forniti dalle telecamere di profondità e quelli di movimento del sensore inerziale. La comprensione del mondo circostante permette alle applicazioni moltissime opportunità implementative, e fra quelle più importanti ed utili in questo progetto, troviamo:

- **Posizionamento:** Conoscere la posizione e le dimensioni delle superfici intorno, come pavimento, pareti e tavoli, permette di posizionare gli ologrammi in relazione ad essi. Questo aiuta l'utente ad orientarsi e percepire meglio la distanza con gli oggetti. Inoltre, in presenza di oggetti

in movimento, è possibile applicare un motore fisico per farli muovere e collidere in modo molto realistico.

- **Occultamento:** Un'altra importante possibilità è quella di poter "nascondere" gli ologrammi. Questo concetto è tanto semplice quanto fondamentale al fine di fornire credibilità ed immersione. Se un ologramma è dietro un oggetto del mondo reale, potrà essere occluso e aiuterà l'utente a percepire meglio la sua posizione nello spazio.

3.1.4 Connettività

HoloLens è equipaggiato con connettività wireless WiFi 802.11ac e Bluetooth 4.1 Low Energy. Essendo un dispositivo che punta principalmente alla mobilità e libertà di movimento, è dotato di una sola porta micro USB 2.0, utilizzabile per la ricarica e il collegamento con un PC per lo scambio dati. Non è possibile utilizzare tale porta come interfaccia di ingresso, per collegare ad esempio un dispositivo o sensore esterno.

3.1.5 Metodi d'interazione

Il modello d'interazione che Microsoft ha creato per utilizzare HoloLens è pensato per funzionare senza la necessità di dispositivi fisici come mouse, tastiere o controller: l'utente potrà quindi avere le mani libere. L'interfaccia prevede tre elementi chiave per interagire:

- **Gaze:** è la direzione in cui lo sguardo dell'utente è rivolto. Il sistema di HoloLens conosce, in ogni istante, la posizione e l'orientamento della testa dell'utente. Il *Gaze* è quindi la direzione di un raggio emesso dal centro del punto di vista (raycast). Questo strumento fornisce quindi al sistema l'abilità di capire dove stiamo guardando (o almeno puntando con la testa) in modo da concentrare la successiva azione sull'oggetto interessato. Può essere visto come il corrispondente puntatore 3D del classico mouse da computer.
- **Gesture:** è il metodo fondamentale col quale si invia un comando. HoloLens è in grado di riconoscere 2 gesture in particolare: "Air-Tap" e "Bloom". Il primo serve a registrare un'azione, come ad esempio un click del mouse tradizionale, e il secondo viene utilizzato per uscire da una qualsiasi applicazione e tornare alla home. Affinchè questi movimenti vengano riconosciuti, HoloLens utilizza le telecamere a disposizione per

riconoscere la posizione delle mani, che devono essere quindi all'interno del loro campo visivo.

- **Voice:** è possibile attivare dei comandi tramite il riconoscimento vocale. La serie di microfoni integrati è calibrata per rilevare al meglio la voce dell'utente, ed è possibile associare l'esecuzione di qualsiasi azione a determinate parole o frasi. Dato che HoloLens è un dispositivo che monta Windows 10, è disponibile un completo motore vocale per gli sviluppatori.

Questi elementi chiave sono molto utili per lo sviluppo di esperienze in Realtà Aumentata. La loro combinazione mette a disposizione un'interfaccia d'interazione sicuramente innovativa e più naturale rispetto ai tradizionali controller, mouse e tastiere, solitamente denominata "GGV" (Gaze, Gesture e Voice).

3.2 Progetto

Nella precedente sezione abbiamo illustrato quali sono le principali caratteristiche e modalità di utilizzo di Microsoft HoloLens. Sebbene offra una serie di strumenti d'interazione che permettono di interfacciarsi in maniera innovativa con la Realtà Mista, ci siamo chiesti se fosse possibile espandere le sue capacità.

Analizzando lo standard GGV (3.1.5) e dopo aver utilizzato HoloLens per diverso tempo, ci si è resi conto di come la modalità di puntamento e selezione basata sull'orientamento della testa e il riconoscimento della mano possano portare a risultati indesiderati. In particolare, durante sessioni di utilizzo prolungate, è possibile avvertire un senso di fatica sul collo e sul braccio, in quanto bisogna costantemente puntare in maniera precisa con lo sguardo e mantenere la mano all'interno del range di tracciamento dei sensori affinché vengano riconosciute le gestures.

L'idea su cui si basa il progetto è di espandere le capacità di tracciamento di HoloLens fornendo nuove informazioni di input e tentare di risolvere alcune criticità appena descritte.

3.2.1 Analisi della tecnologia

Si è visto come il metodo principale di interazione con HoloLens sia basato sul paradigma di Gestures. Di base il dispositivo è in grado di registrare un totale

di 2 gesti differenti: Bloom e Tap. Il primo è utilizzato con l'unico scopo di uscire dalle applicazioni oppure aprire il menu principale: per questo motivo non è un gesto esposto agli sviluppatori e quindi non è possibile implementare funzionalità personalizzate con esso. Il gesto del Tap rimane l'unica modalità di registrazione di input nelle applicazioni utilizzando le mani: HoloLens è in grado di tracciare questo comando solo quando la mano è all'interno di un determinato range visivo del dispositivo, e le dita sono in una specifica posizione.

L'impossibilità di conoscere la posizione relativa della mano rispetto al visore, oppure la capacità di tracciare la mano solo in una determinata posa, sono dei fattori piuttosto limitanti dal punto di vista di uno sviluppatore.

Per questo motivo sono state esplorate altre tecnologie e soluzioni, già esistenti, in grado di espandere il tracciamento di HoloLens e permettere nuove modalità di interazione.

Come descritto nella sezione 2.1, ci sono diversi dispositivi che avrebbero aiutato a raggiungere il nostro scopo, ma ognuna introduce particolari limitazioni e compromessi.

Leap Motion ad esempio è perfetto per rilevare molto dettagliatamente ogni articolazione delle dita e della mano, ma non è possibile collegarlo direttamente ad HoloLens. Ciò richiederebbe l'introduzione di una terza macchina e restrizioni derivate dal cablaggio.

Utilizzare il Kinect, invece, avrebbe reso possibile un tracciamento Full Body dell'utente, limitandone però il range di movimento al campo visivo della telecamera.

3.2.2 Idea e principio di funzionamento

Dopo aver valutato i lavori pubblicati in questo ambito, e analizzato tecnologie e limiti presenti fino ad oggi, si è scelto di progettare e realizzare uno strumento che permetta ad HoloLens di rilevare il movimento del braccio dell'utente.

L'idea alla base è quella di tracciare il movimento dell'arto tramite dispositivi di misurazione inerziali (IMUs). Indossando queste unità direttamente sul braccio, sarebbe possibile raccogliere informazioni di orientamento dello stesso e quindi ricreare un modello virtuale che ne segua i movimenti.

Questo approccio avrebbe portato a due principali vantaggi:

- Non si è limitati a mantenere la mano nel campo visivo delle telecamere di HoloLens per rilevare un'azione di selezione. Sarà possibile tracciare l'intero arco di movimento del braccio, e non bisognerà tenerlo

continuamente disteso frontalmente.

- L'orientamento della testa non sarà più l'unico strumento di puntamento disponibile, in quanto si potrà utilizzare il braccio, mitigando così il senso di fatica su testa e collo.

Inoltre, la mancanza di sensori di tracciamento esterni, come telecamere o basette ad infrarossi, rende il sistema libero da vincoli di movimento e perciò in linea con le capacità spaziali di HoloLens.

3.2.3 Strumenti e architettura

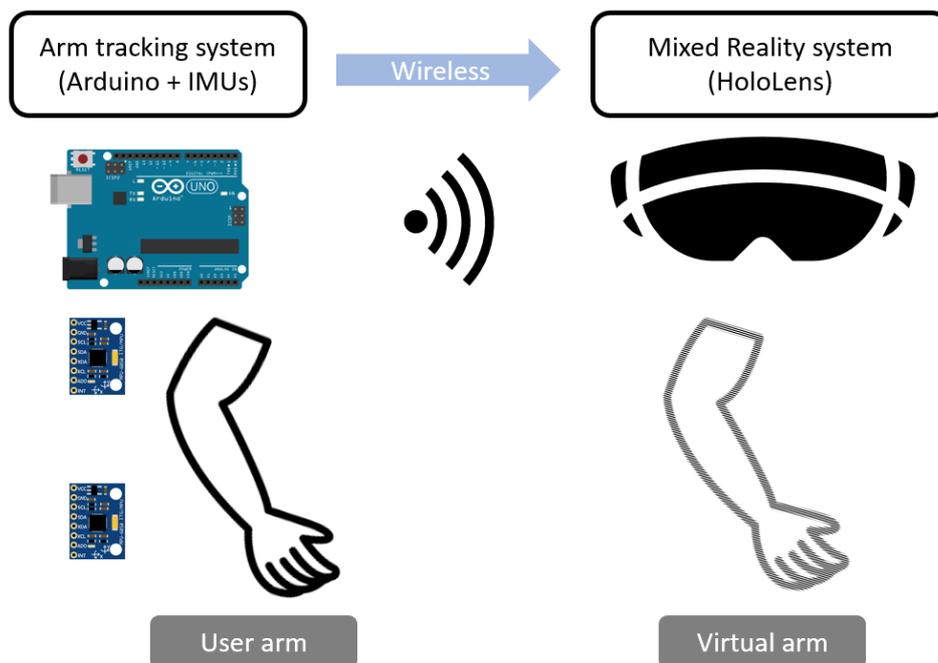


Figura 3.2: Schema del sistema di tracciamento

Per raccogliere ed elaborare informazioni di orientamento, si è scelto di utilizzare una board Arduino insieme a dei sensori di misurazione inerziali, chiamati IMU.

Arduino¹ è una piattaforma elettronica open-source dotata di un microprocessore e controllori programmabili. La board è equipaggiata con una vasta

¹<https://www.arduino.cc>

serie di ingressi e uscite, sia digitali che analogiche, che possono interfacciarsi con un'infinità di sensori, attuatori ed altri circuiti. Alcuni modelli possiedono anche interfacce di comunicazione seriali, come l'USB, utilizzati per lo scambio di dati e la riprogrammazione. È lo strumento ideale per la prototipazione di nuovi dispositivi digitali grazie alla rapidità di programmazione ed espandibilità che offre. Esistono infatti un'infinità di sensori e moduli aggiuntivi che possono essere usati per catturare informazioni dall'ambiente circostante e ritrasmetterle verso altri dispositivi. Il microcontrollore a bordo può essere programmato usando linguaggi di programmazione come C e C++.

I sensori di misurazione inerziale, detti anche IMU (Inertial Measurement Unit), sono delle unità elettroniche in grado di rilevare forze, velocità angolari ed a volte anche l'orientamento di specifici corpi. Grazie alla presenza di uno o più accelerometri, misurano l'accelerazione lineare di un corpo su 3 assi, mentre con dei giroscopi ne rilevano la velocità angolare. Alcuni integrano anche dei magnetometri che permettono di fissare una direzione di riferimento secondo il campo magnetico terrestre. La combinazione delle informazioni processate da questi componenti, permette di calcolare in maniera piuttosto accurata la velocità, l'accelerazione e soprattutto l'orientamento del dispositivo. Quest'ultimo valore è infatti l'informazione di maggiore interesse per il nostro progetto. Misurando l'orientamento del braccio è possibile tracciarne i movimenti e perciò applicarli su un modello virtuale.

Un altro modulo chiave da utilizzare con Arduino, è quello relativo alla connettività wireless. Affinché il dispositivo sia portatile e non introduca restrizioni di movimento, è importante che trasmetta queste informazioni al sistema senza cavi. Bluetooth e Wi-Fi sono due standard molto diffusi per le comunicazioni wireless, e sono presenti diversi moduli di espansione per Arduino che permettono di implementare queste tecnologie.

Nel prossimo capitolo verrà discusso nel dettaglio il processo di realizzazione e implementativo di questa soluzione. Saranno descritte le varie tecnologie utilizzate e i passaggi effettuati per arrivare a creare un primo prototipo funzionante.

Capitolo 4

Realizzazione

Nel precedente capitolo è stato presentato nel dettaglio HoloLens, il dispositivo attorno al quale è stato sviluppato il progetto, ed è stata illustrata ad alto livello l'idea e l'architettura pensata per sviluppare un dispositivo che ne aumenti le capacità d'interazione.

In questo capitolo si procederà a descrivere esattamente le tecnologie coinvolte nel progetto, sia dal punto di vista hardware che software, ed elencare i vari passaggi evolutivi per la realizzazione della soluzione.

4.1 Hardware

4.1.1 HoloLens

La versione di HoloLens utilizzata per il progetto è la Developer Edition ¹, prima iterazione del dispositivo olografico completamente indipendente creato da Microsoft. Questo modello è ideato per gli sviluppatori ed è stato rilasciato il 30 Marzo 2016 solo negli Stati Uniti ed in Canada. In Italia è stato reso disponibile dal Dicembre 2017, insieme ad altri 29 Paesi. La versione del sistema operativo montata a bordo di HoloLens è la 10.0.014393.0, chiamata anche Windows 10 Holographic Anniversary Update.

¹<https://www.microsoft.com/it-it/p/microsoft-hololens-development-edition/8xf18pqz17ts>

4.1.2 Arduino

I prodotti Arduino vantano una vasta gamma di modelli disponibili in commercio che si differenziano in quanto a dimensioni, funzionalità, facilità d'uso e prezzo. I vari modelli possono essere classificati in 5 macro-categorie in base allo scopo di utilizzo: Livello Base, Funzionalità Avanzate, IoT, Educazione oppure Indossabili.

La versione utilizzata inizialmente per questo progetto è Arduino UNO, il primo e più popolare modello della famiglia Arduino. Questa è la migliore piattaforma per iniziare ad interfacciarsi con l'elettronica in quanto è una board molto robusta. Grazie all'enorme diffusione di questo modello, ogni funzionalità è vastamente documentata e gode di un ottimo supporto dalla community. Inoltre è facilmente reperibile ovunque ed con un costo contenuto, solitamente inferiore a 20 Euro.

Arduino Uno usa una board basata sul microcontrollore Atmel ATmega328P. Possiede 14 pins per input ed output digitali, 6 analogici, una connessione USB, connettore di alimentazione ed un tasto di reset. Ha tutto il necessario per funzionare: basta collegarlo con il cavo USB ad un PC oppure alimentarlo con una batteria per iniziare a funzionare. Sebbene le dimensioni non siano fra le più contenute, la sua struttura modulare permette di interfacciarsi con qualsiasi chip, sensore o espansione.

Una volta installato l'ambiente di sviluppo ed aver collegato la board al PC, si è pronti per iniziare a sperimentare.

4.1.3 Espansioni

Come abbiamo anticipato nella sezione precedente, esistono moltissimi sensori, moduli e kit di espansione che possono essere collegati ad Arduino. Le categorie di maggior interesse per questo progetto sono i sensori di misurazione inerziali, e le interfacce di rete wireless. Nella prossime sezioni illustreremo nel dettaglio quagli moduli in particolare sono stati utilizzati, le loro specifiche e le loro modalità di utilizzo.

IMU

Il sensore al cuore di questo progetto è sicuramente quello di misurazione inerziale, anche detto IMU (3.2.3).

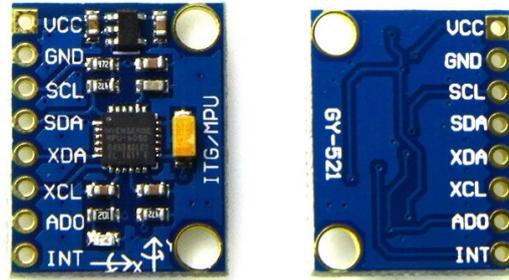


Figura 4.1: Modulo MPU-6050 implementato nella scheda integrata GY-521 per Arduino

Il modello di IMU utilizzato è l’InvenSense MPU-6050²: è un sensore a basso costo e compatto, quindi ideale per creare un sistema di tracciamento indossabile. L’MPU-6050 possiede un giroscopio a 3 assi e un accelerometro a 3 assi sullo stesso chip, insieme ad un Processore Digitale di Movimento (DMP) che processa e combina informazioni su un totale di 6 assi. Le caratteristiche principali sono:

- Chip con Convertitore AD a 16 bit Integrato
- Range di misura giroscopio: ± 250 , 500, 1000 e 2000°/s
- Range di misura accelerometro: +2, +4 , +8 , +16 g
- Interfaccia: I²C
- Alimentazione: da 3V a 5V

Per semplicità di utilizzo e configurazione, è stata utilizzata la scheda *Sodial GY-521* [Figura 4.1]: è un modulo che integra l’MPU-6050 in un circuito che facilita l’interconnessione con Arduino. Questa scheda è più conveniente da utilizzare rispetto al singolo sensore per vari motivi:

- Possiede un regolatore interno che alimenta l’MPU-6050 alla tensione richiesta di 3.3V rispetto ai 5V erogati da Arduino
- Pone a massa il pin AD0, utile a determinare l’indirizzo del bus I²C usato per la comunicazione

²<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>

- Integra un LED per segnalare la corretta alimentazione della scheda

Va fatta una piccola precisazione sui 6DOF dichiarati per questo sensore. I 6 gradi di libertà potrebbero essere intesi come 3 gradi per l'orientamento e 3 gradi per la posizione, come accade per i controller di tracciamento nei sistemi di Realtà Virtuale come HTC Vive o Oculus Rift. In realtà i 6 gradi di libertà derivano dalla somma della misura dell'accelerazione su 3 assi e della velocità angolare sempre sugli stessi 3 assi. Di conseguenza, con un sensore di questo tipo sarà possibile tracciare la sua rotazione nello spazio, ma non la posizione.

Rete

Per quanto riguarda la comunicazione di rete, è importante scegliere una tecnologia wireless per permettere ad Arduino e HoloLens di non avere restrizioni dovute al cablaggio. Sono state testate entrambe le tecnologie più diffuse in questo campo: Bluetooth e WiFi.

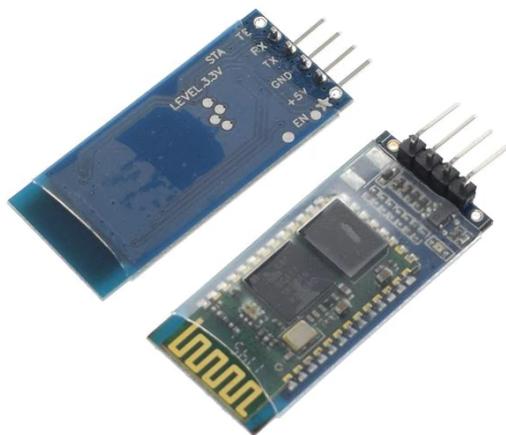


Figura 4.2: Modulo Bluetooth HC-06 per Arduino

Per abilitare Arduino ad inviare informazioni tramite lo standard Bluetooth, è stato utilizzato un modulo HC-06. Anche in questo caso si è scelti una scheda integrata che ne facilita la connessione con Arduino. Il modulo HC-06 è un modulo Bluetooth SPP (Serial Port Protocol): questo significa che comunica con Arduino attraverso la porta di Comunicazione Seriale (conosciuta anche come porta UART). Fra le caratteristiche principali, troviamo:

- Sensibilità (Bit error rate) fino a -80dBm
- Antenna integrata a 2.4Ghz
- Tensione operativa 3.1V - 4.2V
- Modulo EDR per bitrate variabile da 2Mbps a 3Mbps

Possiede 4 pin, 2 per l'alimentazione e 2 per comunicazione seriale con Arduino, come illustrato in Figura 4.2.

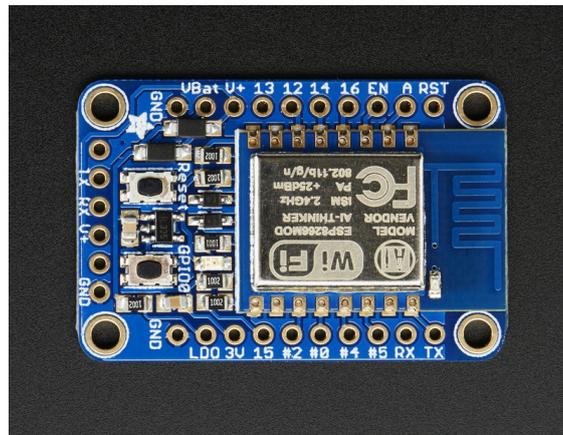


Figura 4.3: Modulo WiFi Adafruit HUZZAH ESP8266

Fonte: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-huzzah-esp8266-breakout.pdf>

Per quanto riguarda la connettività WiFi è stato utilizzato il modulo ESP8266. Il processore ESP8266, prodotto da Espressif, ha un microcontrollore a 80Mhz con un'interfaccia WiFi 802.11 b/g/n completa e supporta il protocollo TCP/IP, anche con DNS. Ha la capacità, quindi, di fungere sia da Client che da Access Point. Data la complessità di utilizzo di questo chip, anche in questo caso si è optato per una scheda integrata: la HUZZAH ESP8266. Prodotta da Adafruit, questa scheda è progettata per facilitare l'interfacciamento e la programmazione di questo modulo. Fra le varie funzionalità, troviamo:

- Regolatore di tensione a 3.3V, 500mA
- Antenna integrata da +19.5dBm in modalità 802.11b
- Due diodi per proteggere gli input di alimentazione

- RAM da 64KB per le istruzioni e 96KB per i dati
- Memoria Flash da 4MB
- Certificazione FCC, tasto Reset e LED di stato

Anche in questo caso, la comunicazione con Arduino avviene tramite la porta seriale UART.

4.2 Ambiente di sviluppo

In questa sezione verranno elencati e descritti i programmi utilizzati per sviluppare gli applicativi del progetto. In particolare, si seguiranno le evoluzioni del software per capire nel dettaglio le modalità implementative, quali difficoltà si sono incontrate e come sono state affrontate.

4.2.1 HoloLens e Unity 3D

Nella sezione 3.1.2 sono stati illustrati i due tipi di applicazioni che è possibile sviluppare per HoloLens: Finestra o Immersiva. Nell'ambito di questo progetto, la modalità Immersiva è sicuramente la scelta da fare, in quanto è necessario poter accedere a tutte le funzionalità spaziali del dispositivo. Inoltre, vista la complessità delle informazioni da gestire ed i tipi di input da registrare, si è scelto di sviluppare l'applicazione per HoloLens utilizzando il motore grafico Unity 3D.

Grazie a questo si ha già accesso a diversi strumenti che agevolano la gestione di modelli poligonali e automatizzano le varie fasi di rendering, così da poter concentrarsi maggiormente sulle interfacce di interazione.

A partire dalla versione 5.6, Unity ha aggiunto il supporto allo sviluppo di applicazioni per HoloLens. Visto che le applicazioni di Realtà Mista richiedono un approccio sia implementativo ma anche progettuale, molto diverso rispetto alle tradizionali applicazioni Desktop, Microsoft ha rilasciato degli strumenti molto utili a riguardo: *Holographic Academy* [17] ed *HoloToolkit* [18].

Holographic Academy è una documentazione online dove Microsoft espone diversi criteri e consigli per guidare lo sviluppo di un'esperienza AR con HoloLens sin dalla fase di progettazione. Si fa riferimento ad alcuni concetti chiave come il comfort utente, il sistema di riferimento ed i colori in un sistema di proiezione. Questa è un'ottima sezione da dove iniziare per chi è

alle prime armi con lo sviluppo di applicazioni in Realtà Mista. Si possono apprendere le nozioni base e trarre spunto per progettare in maniera ottimale sia con HoloLens sia con qualsiasi altro dispositivo MR e AR.

Nella prossima sezione entreremo nel dettaglio dello sviluppo software con HoloLens. Verrà prima illustrato l'HoloToolkit, lo strumento principale per trarre vantaggio delle funzionalità avanzate di HoloLens tramite Unity, e poi le varie iterazioni per integrare una nuova interfaccia d'interazione.

HoloToolkit

HoloToolkit per Unity è una suite software rilasciata da Microsoft per agevolare lo sviluppo con HoloLens. Permette una prototipazione più veloce grazie alla modalità di simulazione in-editor per testare istantaneamente le modifiche apportate. In sua assenza, infatti, bisognerebbe effettuare la procedura di "Build and Deploy" su HoloLens ogni volta che si vuole verificare una modifica. Questa procedura richiede più di 2 minuti per essere eseguita, rallentando notevolmente il processo di sviluppo.

Vista la novità e particolarità del genere di piattaforma, HoloToolkit è una libreria molto dinamica ed in continua evoluzione. Ogni mese vengono rilasciati aggiornamenti dagli sviluppatori guidati dai feedback della community per risolvere problemi noti oppure implementare nuove funzionalità.

In questo progetto è stata utilizzata la versione 2017.2.0p1 MRT4 dell'HoloToolkit³ insieme alla versione dell'Editor di Unity 2017.1.2⁴. Per creare applicazioni per HoloLens con questa versione, è necessario puntare all'SDK 10.0.15063 del sistema operativo Windows Holographic.

Per iniziare, è sufficiente importare l'HoloToolkit (nel formato *unitypackage*) in un nuovo progetto Unity 3D. Il progetto deve puntare alla piattaforma UWP (Universal Windows Platform) e alla corretta versione dell'SDK. Una volta importato il toolkit, sarà possibile trovare negli Asset una serie di classi, funzioni e modelli pronti all'uso. Sono presenti anche delle scene d'esempio che aiutano a capire come impostare correttamente i vari parametri e dipendenze.

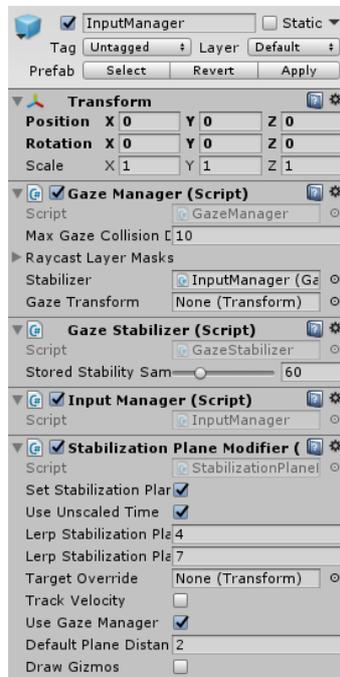
Dopo una fase di esplorazione e testing, i moduli relativi alle funzionalità più rilevanti al nostro scopo sono i seguenti:

³<https://github.com/microsoft/MixedRealityToolkit-Unity/releases/tag/v1.2017.2.0>

⁴<https://unity3d.com/unity/whats-new/unity-2017.1.2>



(a) HoloLensCamera



(b) InputManager



(c) Cursors

Figura 4.4: Componenti utilizzati dell'HoloToolkit

HoloLensCamera Questo è l'oggetto principale che si occupa di impostare correttamente i vari parametri di rendering del frame. Prende il posto della *MainCamera* nella scena Unity e si assicura di valorizzare correttamente i campi:

- **Transform:** La componente che contiene le informazioni di posizione, rotazione e scalamento dell'oggetto. Ai fini del corretto utilizzo con HoloLens, è importante che questo oggetto si trovi nell'origine del sistema di riferimento, quindi Posizione e Rotazione azzerati, e Scalamento unitario. Quando l'applicazione viene avviata su HoloLens, questa componente rappresenterà la posizione e rotazione del dispositivo e servirà ad avere un riferimento dei suoi movimenti rispetto al mondo reale. Come si vedrà più avanti, quest'informazione sarà necessaria per dislocare correttamente il braccio virtuale.

- **Camera Background:** Rappresenta il colore o il materiale col quale renderizzare lo sfondo. È importante che sia settata in modalità *Solid Color* e colore *Nero* in quanto, in un sistema di proiezione (come appunto HoloLens) "renderizzare" questo colore significa "non emettere luce", di conseguenza risulterà trasparente e permetterà all'utente di vedere correttamente il mondo reale attraverso le lenti.
- **Field of View:** Per allineare la vista del mondo virtuale con quello reale, deve rispettare le caratteristiche tecniche del dispositivo. Va settato con un valore di 20 gradi.
- **Manual Gaze Control:** Questo è uno script legato all'oggetto Camera che permette di simulare il movimento di HoloLens all'interno dell'Editor di Unity. Molto importante per verificare immediatamente le modifiche apportate alla scena senza dover creare una Build, installarla ed eseguirla sul dispositivo continuamente. Permette di simulare il movimento dell'HoloLens virtuale tramite mouse e tastiera, oppure un joystick collegato.

InputManager Questo oggetto si occupa di tutti i processi necessari ad interagire con l'applicazione tramite il modello GVV visto nella Sezione 3.1.5. Gli script che compongono questo elemento sono:

- **Gaze Manager:** Si occupa di gestire la dinamica di selezione oggetti durante l'esecuzione. Utilizzando la Transform della HoloLensCamera, proietta un raggio virtuale uscente dal punto di vista. Tenendo traccia di tutti gli elementi che vengono intersecati da questo raggio, lo script è in grado di sapere quale oggetto stiamo guardando, così da settare il focus su di esso e quindi indirizzare correttamente i successivi input oppure eseguire degli eventi che lo riguardano. Rappresenta quindi il blocco alla base del paradigma d'interazione GVV di HoloLens.
- **Gaze Stabilizer e Stabilization Plane:** Come si intuisce da l nome, sono oggetti che si occupano di stabilizzare il puntamento. Il , infatti, dipende unicamente dal movimento del collo e della testa che, per natura umana, effettuano continuamente piccole oscillazioni involontarie. Queste vibrazioni si riflettono inevitabilmente sul sistema di puntamento, rendendo più difficile l'interazione con oggetti di dimensioni ridotte o lontani. Con questo script, si cerca di "smussare" queste vibrazioni tramite algoritmi di interpolazione lineare, rendendo il puntatore più fluido ma leggermente meno reattivo.

- **Cursor:** Serve a restituire un feedback visivo all'utente riguardo lo stato di registrazione degli input. É composto da elementi grafici personalizzabili, che aiutano l'utente a sapere se il sistema stia puntando esattamente lungo il suo sguardo e se abbia registrato i suoi input di selezione. Nella Figura 4.4c sono illustrati rispettivamente il cursore di puntamento, quello della mano in posizione *Ready* e quello in posizione *Tap* oppure *Click*. Il cursore sarà posizionato inoltre in prossimità dell'oggetto puntato. L'utente potrà percepire questa distanza grazie alla visione stereoscopica di HoloLens.

4.2.2 Arduino IDE

Per sviluppare un progetto con la board Arduino Uno è necessario utilizzare l'Arduino Software. É un IDE open-source che permette di scrivere programmi e caricarli sulle board Arduino.

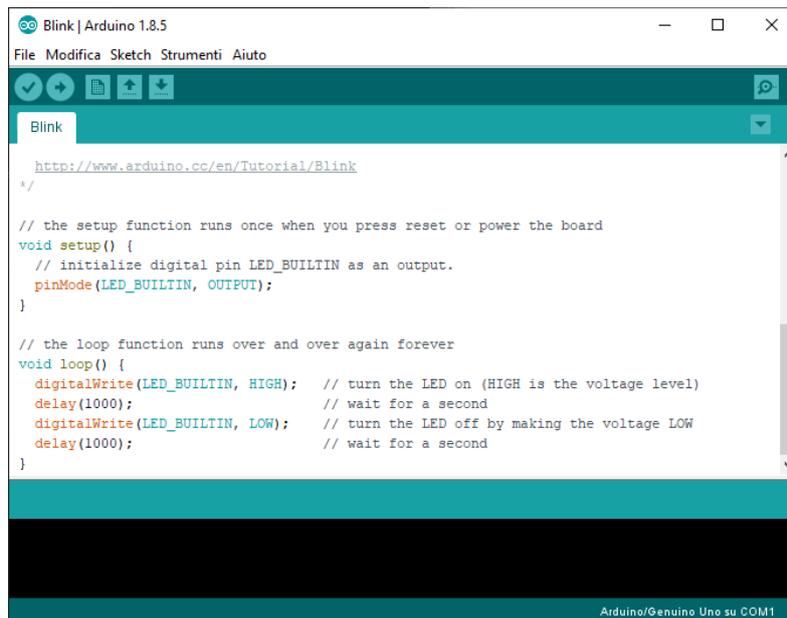
In questo progetto è stata utilizzata la versione Arduino IDE 1.8.5⁵ per Windows. Per verificare che l'ambiente sia correttamente installato e funzionante, è sufficiente collegare la board Arduino Uno al Computer tramite il cavo A-B USB. Il cavo è necessario per programmare la board oltre che alimentarla. Arduino Uno è in grado di alimentarsi automaticamente sia tramite la porta USB sia attraverso una fonte di energia esterna, come pile o batterie. Una volta collegato ed alimentato, è possibile vedere un LED verde vicino la voce *PWR* della board che indica il corretto funzionamento.

I documenti che contengono le istruzioni da eseguire sono denominati *sketch* ed il linguaggio di programmazione utilizzato è un derivato del C/C++. L'Arduino IDE è in grado di compilare e caricare il programma eseguibile sulla scheda Arduino in un unico passaggio, rendendo il processo di prototipazione molto veloce.

Gli *sketch* richiedono un minimo di due funzioni per essere eseguiti correttamente:

- **void setup():** metodo eseguito solo una volta, all'avvio della scheda, utilizzato per le impostazioni iniziali che non cambieranno durante l'esecuzione.
- **void loop():** il contenuto di questo metodo viene eseguito continuamente, finchè la scheda non viene spenta.

⁵<https://github.com/arduino/Arduino/releases/tag/1.8.5>

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.8.5". The menu bar includes "File", "Modifica", "Sketch", "Strumenti", and "Aiuto". The toolbar contains icons for opening, saving, and running. The main editor area displays the following code:

```
http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

The status bar at the bottom right indicates "Arduino/Genuino Uno su COM1".Figura 4.5: Esempio di *sketch* nell'Arduino IDE

Con l'installazione dell'IDE, vengono forniti anche degli *sketch* d'esempio. Per verificare il corretto funzionamento della scheda Arduino UNO, si è caricato l'esempio in Figura 4.5. Finito l'upload, il LED del pin 13 inizia a lampeggiare confermando l'esito dell'operazione.

Si può procedere quindi col collegamento dei moduli di espansione come IMU e interfaccia Wireless.

4.2.3 Connessione con IMU

Si procede ora quindi a collegare il modulo GY-521 ad Arduino Uno per recuperare le informazione di orientamento del modulo.

Schema di collegamento

Come è possibile vedere in Figura 4.6, per connettere il modulo GY-521 alla scheda Arduino Uno è sufficiente collegare quattro cavi:

- VCC a 3.3V (rosso)
- GND a GND (nero)
- SCL al pin A5 (giallo)

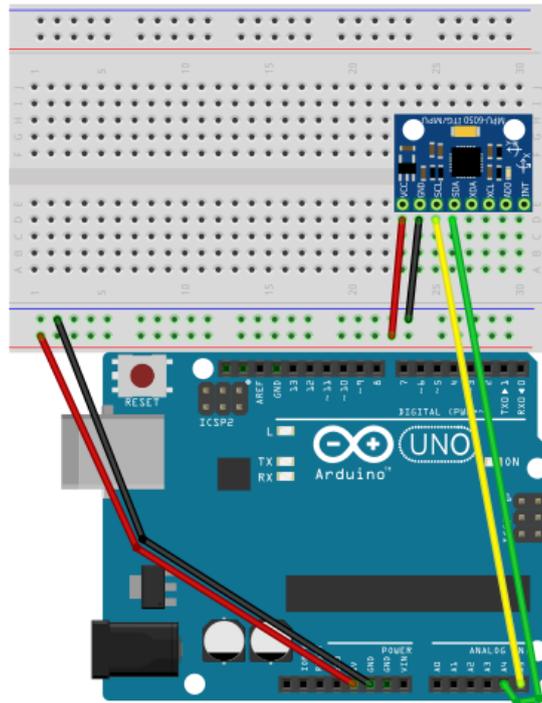


Figura 4.6: Schema di collegamento del modulo GY-521 (IMU) alla scheda Arduino UNO

- SDA al pin A4 (verde)

I pin *VCC* e *GND* sono utilizzati per alimentare il modulo. I pin *SCL* e *SDA* invece sono i due canali utilizzati per la comunicazione seriale I²C (Inter Integrated Circuit). Questo tipo di comunicazione seriale necessita, infatti, di due canali di comunicazione: uno *Master*, in questo caso SDA (**S**erial **D**Ata) per il trasferimento dei dati, ed uno *Slave*, in questo caso SCL (**S**erial **C**Lock), per la sincronizzazione.

All'avvio di Arduino, il LED rosso sulla scheda GY-521 indicherà la corretta alimentazione del modulo.

4.2.4 Istruzioni per lettura dei dati

A questo punto si può procedere con la scrittura delle istruzioni necessarie a leggere i dati dell'IMU. In Figura 4.7 è riportato un esempio di codice per effettuare una prima lettura di dati. I passaggi chiave sono:

```

imu_example
1 #include<Wire.h>
2 const int MPU_addr=0x68; // I2C address of the MPU-6050
3 int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
4 void setup() {
5   Wire.begin();
6   Wire.beginTransmission(MPU_addr);
7   Wire.write(0x6B); // PWR_MGMT_1 register
8   Wire.write(0); // set to zero (wakes up the MPU-6050)
9   Wire.endTransmission(true);
10  Serial.begin(9600);
11 }
12 void loop() {
13   Wire.beginTransmission(MPU_addr);
14   Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
15   Wire.endTransmission(false);
16   Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
17   AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
18   AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
19   AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
20   Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
21   GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
22   GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
23   GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
24   Serial.print("AcX = "); Serial.print(AcX);
25   Serial.print(" | AcY = "); Serial.print(AcY);
26   Serial.print(" | AcZ = "); Serial.print(AcZ);
27   Serial.print(" | Tmp = "); Serial.print(Tmp/340.00+36.53); //equation for temperature in degrees C from datasheet
28   Serial.print(" | GyX = "); Serial.print(GyX);
29   Serial.print(" | GyY = "); Serial.print(GyY);
30   Serial.print(" | GyZ = "); Serial.println(GyZ);
31   delay(333);
32 }

```

Figura 4.7: Esempio di *sketch* per il la lettura dei dati RAW dal modulo GY-521

1. Includere la libreria *Wire.h* per stabilire una connessione fra Arduino Uno e il sensore GY-521. L'Arduino IDE include già questa libreria poichè comunemente usata per le comunicazioni seriali I²C.
2. Settare l'indirizzo del modulo MPU-6050. Se il pin AD0 è posto su "Alto", l'indirizzo sarà 0x68, altrimenti è 0x69.
3. Inizializzare le variabili dove verranno salvati i dati relativi ad accelerometro, giroscopio e temperatura.
4. Nella funzione *setup()* stabilire la connessione seriale. Iniziare anche la prima trasmissione I²C verso la scheda GY-521 per svegliarla dallo stato di *sleep-mode*
5. Nella funzione *loop()* si vanno a leggere (e stampare) ciclicamente i seguenti valori del sensore:
 - 3 valori dell'accelerometro

- 3 valori del giroscopio
- 1 valore della temperatura

Se il tutto è stato connesso e configurato correttamente, una volta caricato il codice su Arduino, l'output sarà simile a quello riportato in Figura 4.8.

```

aX = -3128 | aY = -44 | aZ = 16408 | tmp = 31.12 | gX = -458 | gY = -109 | gZ = -87
aX = -2972 | aY = -28 | aZ = 16312 | tmp = 31.31 | gX = -430 | gY = -88 | gZ = -106
aX = -2928 | aY = 40 | aZ = 16152 | tmp = 31.49 | gX = -442 | gY = -98 | gZ = -105
aX = -3028 | aY = -80 | aZ = 16316 | tmp = 31.92 | gX = -467 | gY = -128 | gZ = -108
aX = -2956 | aY = -756 | aZ = 16196 | tmp = 31.40 | gX = -938 | gY = -54 | gZ = 2631
aX = 9976 | aY = -1404 | aZ = 12932 | tmp = 31.17 | gX = -3792 | gY = -15059 | gZ = -415
aX = -596 | aY = 492 | aZ = 17032 | tmp = 30.93 | gX = -479 | gY = 7178 | gZ = 1163
aX = 3164 | aY = -12648 | aZ = 10588 | tmp = 30.79 | gX = -5358 | gY = -8152 | gZ = -3161
aX = 12444 | aY = -5420 | aZ = 10272 | tmp = 30.69 | gX = 350 | gY = -4088 | gZ = -1461
aX = 16224 | aY = -2800 | aZ = 4916 | tmp = 30.65 | gX = 910 | gY = -2784 | gZ = -693
aX = 13968 | aY = -3528 | aZ = 8248 | tmp = 30.46 | gX = 2363 | gY = 14582 | gZ = 3272
aX = -6260 | aY = -1392 | aZ = 15124 | tmp = 30.41 | gX = 464 | gY = 10636 | gZ = 1093
aX = -14332 | aY = -628 | aZ = 7792 | tmp = 30.41 | gX = -188 | gY = 1615 | gZ = 137
aX = -9348 | aY = -1164 | aZ = 13604 | tmp = 30.41 | gX = -308 | gY = -18053 | gZ = -2686

```

Figura 4.8: Valori RAW letti in output dal sensore GY-521

Da notare che le informazioni recuperate, per ogni asse, sono dei valori "grezzi" salvati nei registri del modulo MPU-6050. Questo significa che il sensore funziona correttamente, ma per ottenere le informazioni di orientamento in un formato comprensibile, come ad esempio in gradi, bisogna effettuare qualche altro passaggio.

Come precedentemente descritto nella Sezione 4.1.3, il sensore MPU-6050 possiede un DMP, ovvero un Processore Digitale di Movimento. Il DMP è in grado di effettuare velocemente dei calcoli direttamente sul sensore stesso. Questo permette di ridurre il carico computazionale del microcontrollore di Arduino e di combinare in maniera più complessa i dati grezzi forniti dal sensore. Il sensore possiede un buffer FIFO da 1024 byte ed il DMP è programmato per salvare qui i valori elaborati.

Per comunicare correttamente col DMP si è fatto ricorso alla libreria open-source *i2cdevlib*⁶. La I²C Device Library è una collezione di classi ben documentate che fornisce interfacce semplici ed intuitive verso i dispositivi I²C. In particolare contiene delle classi ottimizzate per Arduino e compatibili con vari IMUs. Le principali novità introdotte dall'utilizzo di questa libreria sono:

⁶<https://github.com/jrowberg/i2cdevlib>

- **Calibrazione del sensore:** Quando si utilizzano sensori di movimento, come l'MPU-6050, si prevede una fase iniziale di calibrazione. Durante questo processo, che può durare dai 5 ai 10 secondi, il sensore deve essere tenuto in una posizione stabile. In questo modo sarà possibile calibrare con i giusti offset i valori letti ed evitare sfasamenti nel tempo.
- **Funzioni del DMP:** Sono previste delle interfacce di comunicazione diretta con il processore a bordo del sensore di movimento. In questo modo è possibile trarre vantaggio di algoritmi specializzati alla combinazione delle informazioni di accelerometro e giroscopio per ottenere un valore di orientamento 3D pulito ed affidabile. Fra le funzioni di recupero dati dal DMP, ci sono:
 - `dmpGetQuaternion()` restituisce un quaternione per l'orientamento
 - `dmpGetYawPitchRoll()` restituisce l'orientamento in gradi
 - `dmpGetAccel()` restituisce l'accelerazione

Se la connessione con il DMP è avvenuta correttamente, si potranno leggere i valori di orientamento del sensore sotto forma di gradi per ogni asse (Yaw, Pitch e Roll) come nell'esempio in Figura 4.9.

```
Initializing I2C devices...
Testing device connections...
MPU6050 connection successful

Send any character to begin DMP programming and demo:
Initializing DMP...
Enabling DMP...
Enabling interrupt detection (Arduino external interrupt 0)...
DMP ready! Waiting for first interrupt...
ypr 47.10 50.21 -39.67
ypr 47.03 50.18 -39.69
ypr 46.84 50.12 -39.74
ypr 46.66 50.07 -39.77
ypr 46.51 50.05 -39.78
ypr 46.38 50.03 -39.79
ypr 46.25 50.02 -39.79
ypr 46.09 49.98 -39.81
```

Figura 4.9: Valori calibrati di orientamento in gradi letti in output dal DMP del sensore GY-521

Fra i vari valori di output disponibili dal DMP, il formato più interessante è sicuramente il quaternione. Selezionando questa opzione verranno fornite le componenti x,y,z e w, ideali per la rappresentazione di rotazioni in Unity 3D.

4.3 Comunicazione di rete

Una volta che le informazioni di rotazione sono state acquisite, bisogna trovare il modo con cui inviare ad HoloLens in maniera Wireless. Arduino Uno, di base, permette di leggere l'output seriale della console tramite la porta USB. Se si vogliono raccogliere informazioni in altro modo, bisogna utilizzare correttamente i pin analogici e digitali in modalità di output.

In questa sezione verrà descritto il procedimento effettuato per implementare la funzionalità wireless ad Arduino ed instaurare una comunicazione di rete per inviare informazioni. Le tecnologie utilizzate sono quella Bluetooth e quella WiFi.

4.3.1 Bluetooth

Il modulo utilizzato per implementare la connettività Bluetooth su Arduino, è il sensore HC-06.

Schema di connessione

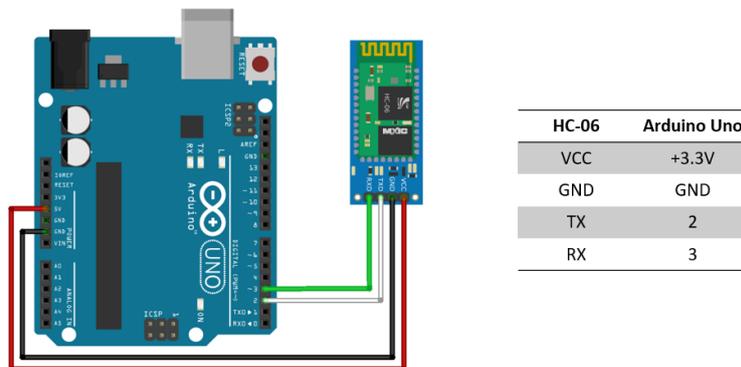


Figura 4.10: Schema di collegamento del Modulo HC-06 con Arduino Uno

Come è possibile vedere in Figura 4.10, per collegare il modulo Bluetooth sono sufficienti 4 cavi: 2 sono utilizzati per l'alimentazione e 2 per l'invio e ricezione dei dati. I pin 2 e 3 di Arduino Uno supportano la comunicazione seriale.

Configurazione del modulo

Per utilizzare il modulo HC-06 è necessaria una prima fase di configurazione. In questo processo vengono settati parametri come il SSID (il nome della rete), il codice PIN di accoppiamento e la velocità di comunicazione, anche detta *baudrate*. Per permettere la configurazione, il modulo HC-06 deve essere alimentato ma non associato: si può verificare questo stato se il LED della scheda risulta lampeggiante.

Lo *sketch* in Figura 4.11 mostra le istruzioni da caricare su Arduino per avviare una comunicazione col modulo HC-06 che permetta la configurazione dei parametri.

1. Includere la libreria *SoftwareSerial.h* utilizzata per la comunicazione seriale.
2. Specificare i pin che saranno utilizzati per la connessione seriale col modulo HC-06: in questo caso 2 e 3.
3. Nella funzione di *setup()* inizializzare i due canali seriali: uno fra la console del PC ed Arduino (per inviare e leggere i comandi) e l'altra fra Arduino e il modulo Bluetooth.
4. La funzione *loop()* conterrà invece delle semplici istruzioni di "inoltro" di comandi fra un canale seriale e l'altro.

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial hc06(2,3);
3 void setup(){
4   Serial.begin(9600);
5   Serial.println("ENTER AT Commands:");
6   hc06.begin(9600);
7 }
8 void loop(){
9   if (hc06.available()){
10    Serial.write(hc06.read());
11  }
12  if (Serial.available()){
13    hc06.write(Serial.read());
14  }
15 }
```

Figura 4.11: *Sketch* per la configurazione dei parametri iniziali del modulo HC-06

Per inviare i parametri di configurazione al modulo HC-06, bisogna usare dei comandi AT attraverso il monitor seriale dell'Arduino IDE. I comandi AT (abbreviazione di ATtention) sono delle istruzioni testuali nel formato "AT+comando". Nella configurazione di questo modulo, quelli più importanti sono:

- **AT+NAME***nomemodulo*: per modificare il nome visualizzato nella ricerca BLuetooth
- **AT+PIN***xxxx*: per modificare le 4 cifre del PIN di accoppiamento
- **AT+BAUD***x*: per modificare la velocità di comunicazione del modulo. I valori di *x* corrispondono alle seguenti velocità:
 - 1 per 1200
 - 2 per 2400
 - 3 per 4800
 - 4 per 9600
 - 5 per 19200
 - 6 per 38400
 - 7 per 57600
 - 8 per 115200

Una volta inviato il comando col tasto *Enter*, si riceverà l'esito dell'esecuzione con un messaggio nel formato "**OK***testo*".

Associazione ed utilizzo del modulo

Una volta configurato il modulo, è possibile procedere con l'associazione verso il dispositivo Bluetooth desiderato. Effettuando una scansione dei dispositivi Bluetooth dalle impostazioni di HoloLens, è possibile vedere il modulo HC-06 ma purtroppo il collegamento non è andato a buon fine. Per tentare di risolvere il problema sono state provate diverse configurazioni: si è provato a cambiare la velocità di comunicazione del canale, a cambiare o rimuovere il PIN di accoppiamento, anche a provare un secondo modulo HC-06. In ogni caso, HoloLens non è riuscito ad associarsi direttamente con questa scheda.

A questo punto si è provato a collegare il modulo HC-06 ad un Computer per verificarne il funzionamento. L'associazione tramite BLuetooth con un

PC non ha incontrato nessun problema, e la comunicazione fra i 2 dispositivi è avvenuta correttamente. Come possibile soluzione, si è pensato di utilizzare il PC come ponte per inoltrare le informazioni da Arduino ad HoloLens.

Sebbene questa soluzione funzionasse, ci sono due aspetti principali da considerare:

1. La velocità di comunicazione delle informazioni di rotazione è piuttosto lenta. Utilizzando il baud-rate consigliato per il modulo HC-06, cioè 9600, si è riscontrata una ricezione di 2-3 quaternioni per secondo. Aumentando il baud-rate si è potuta raggiungere una ricezione di 25-26 pacchetti al secondo, che può essere accettabile per mostrare un movimento fluido. Purtroppo a questa velocità di connessione, corrispondono anche frequenti disconnessioni del modulo HC-06 ed un'instabilità generale del sistema.
2. Sebbene il collegamento fra i vari dispositivi rimane pur sempre wireless, l'introduzione di una terza macchina con funzionalità di proxy diventa comunque un vincolo aggiuntivo. La qualità della connessione dipenderà dalla distanza fra Arduino ed il PC, e non fra Arduino ed HoloLens.

4.3.2 WiFi

In questa sezione verrà illustrato il procedimento effettuato per implementare la funzionalità WiFi su Arduino ed inviare le informazioni di orientamento tramite questa interfaccia.

Il modulo utilizzato per la connettività Wifi è l'ESP8266.

ESP8266

La scheda integrata utilizzata che implementa il modulo WiFi ESP8266, è la Adafruit HUZZAH ESP8266. Questa scheda, oltre ad integrare un'interfaccia completa WiFi, come visto in Sezione 4.1.3, è dotata anche di un microcontrollore a 80Mhz e diversi pin di input ed output. Sebbene possa essere collegata ad Arduino in maniera simile alla scheda Bluetooth BH-06, cioè utilizzando i pin TX e RX per instaurare una connessione seriale, questa scheda integrata ha la capacità di essere riprogrammata per eseguire delle istruzioni, proprio come Arduino stesso.

Grazie alla documentazione e alle librerie fornite dal produttore⁷ è possibile rendere la scheda compatibile con l'Arduino IDE. Bisogna riprogrammare il bootloader, che viene fornito di fabbrica con un interprete *NodeMCU's Lua*, con lo *sketch* che si desidera eseguire.

Nuovo schema di connessione

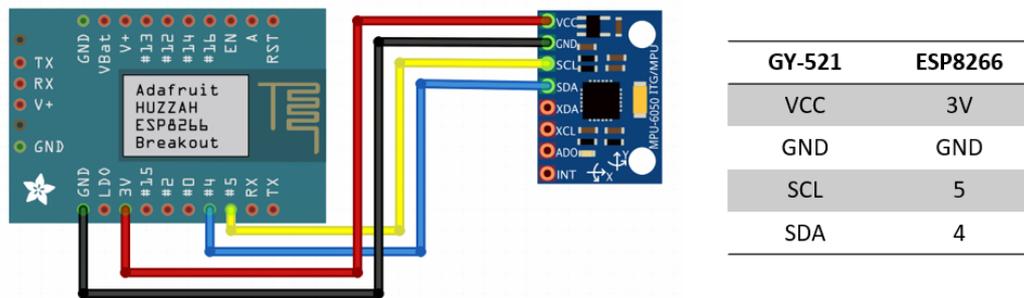


Figura 4.12: Schema di connessione diretta dell'IMU GY-521 alla scheda Adafruit Huzzah ESP8266 WiFi

Come si può vedere in Figura 4.12, è possibile interfacciare il sensore inerziale direttamente con la scheda ESP8266. Il microcontrollore a bordo della Huzzah ESP8266 (80Mhz) è in grado di eseguire un maggior numero di istruzioni al secondo rispetto a quello di Arduino Uno (16Mhz). Quest'ultimo vanta una maggiore disponibilità di porte I/O, ma quelle disponibili sull'ESP8266 sono sufficienti per il nostro progetto.

Così come su Arduino Uno, i pin 4 e 5 della scheda ESP8266 possono essere utilizzati per la comunicazione seriale I²C, rendendo il passaggio semplice ed immediato.

Collegamento di un secondo IMU A questo punto, si è deciso di aggiungere un secondo sensore inerziale di movimento per poter tracciare anche il movimento dell'avambraccio. È stato utilizzato un altro IMU dello stesso modello, cioè MPU-6050 integrato in una scheda GY-521.

⁷<https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/using-arduino-ide>

Sia Arduino, sia l'ESP8266, possono instaurare un solo canale di comunicazione I²C alla volta, in quanto solo i Pin 4 e 5 supportano la comunicazione con SDA e SCL. Per gestire la lettura di due o più IMU è necessario quindi aggiungere dei collegamenti di sincronizzazione. Per fare ciò si useranno due uscite aggiuntive disponibili sulle schede GY-521:

- **INT**: Sta per INTERRUPT e viene utilizzato per sincronizzare il clock del DMP della scheda in modo da sapere quando i dati di orientamento sono pronti per lettura nel buffer FIFO.
- **AD0**: Se questo pin è posto su *ALTO*, l'indirizzo di comunicazione I²C sarà 0x69. Se è posto su *BASSO*, l'indirizzo sarà 0x68.

Secondo quest'ultima informazione, sarebbe possibile tuttavia leggere le informazioni di 2 MPU contemporaneamente, avendo l'accortezza di settare AD0 su ALTO per il primo, e BASSO per il secondo.

Sebbene questa soluzione sia compatibile con la nostra configurazione, si è scelto di implementare un approccio più generale che offre la possibilità di collegare un numero arbitrario di IMU contemporaneamente. L'implementazione si basa su questo principio:

1. All'avvio del dispositivo, i pin AD0 di tutti gli IMU sono posti sullo stesso stato. Ai fini di questo esempio, si prende lo stato comune come *alto*, di conseguenza comunicheranno tutti sull'indirizzo 0x69.
2. Nella funzione *loop()* si imposta il pin AD0 del primo IMU su *basso* e si effettua la lettura dei dati sull'indirizzo 0x68. Terminata la lettura, si reimposta il pin AD0 di questo IMU su *alto* e si riesegue la stessa operazione col successivo.

Iterando ciclicamente su ogni IMU, si utilizzerà solo l'indirizzo 0x68 per la lettura dei dati, uno alla volta, condividendo il canale I²C sugli stessi pin (4 e 5). Nella Figura 4.13 è possibile visualizzare lo schema di connettività che include il secondo GY-521, implementando questa tecnica di comunicazione.

Invio dei dati

L'utilizzo delle funzionalità WiFi sulla scheda Huzzah ESP8266 richiede una prima fase di configurazione seguita dalla connessione vera e propria. Innanzitutto, bisogna scegliere se impostare la scheda come Access Point oppure Client. Nel primo caso verrà creata una nuova rete WiFi alla quale HoloLens

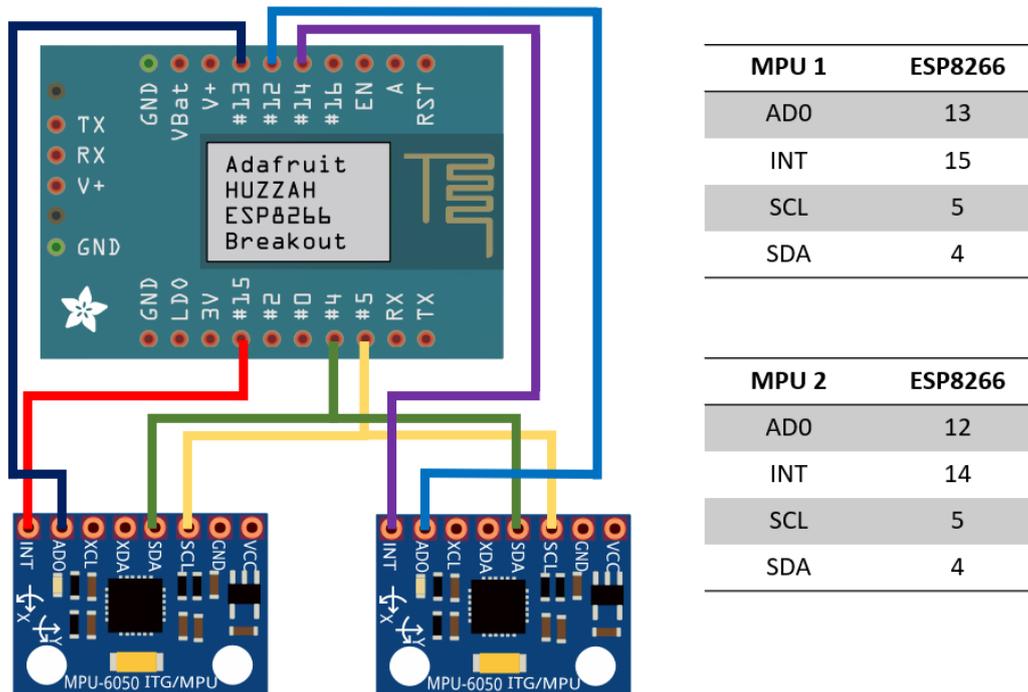


Figura 4.13: Schema di connessione di due IMU alla scheda Adafruit Huzzah ESP8266 WiFi

dovrà collegarsi per comunicare con i sensori, nel secondo caso la scheda si collegherà ad una rete WiFi già esistente e comunicherà con HoloLens tramite WLAN. Avendo HoloLens già collegato ad una rete, si è scelta la seconda modalità.

Fra i parametri richiesti in fase di configurazione troviamo:

- *SSID* e *Password* della rete WiFi alla quale bisogna connettersi
- Indirizzo IP e Porta dove verranno inviati i dati
- Includere le librerie *WiFi.h*, *WiFiClient.h*, *WiFiManager.h* e *WiFiUdp.h* per la gestione delle funzionalità wireless.

Grazie alla libreria *WiFiUdp.h* sarà infatti possibile inviare dati tramite il protocollo UDP. Per l'invio di una grande quantità di dati nel modo più veloce possibile, l'UDP è sicuramente il protocollo migliore da usare. Non si ha la necessità di instaurare ogni volta una connessione fra i dispositivi, ma invece è utile inviare più dati possibili, appena questi vengono generati.

Per inviare le informazioni di orientamento a HoloLens, i dati da elaborare corrispondono a 4 float: x, y, z e w del quaternion generato dall'IMU. A questi, va aggiunto un "tag" per identificare univocamente il sensore di provenienza. Sono stati utilizzati due metodi per formattare questi dati: il primo tramite il formato OSC, e il secondo tramite una formattazione personalizzata.

Primo metodo Open Sound Control (OSC) è un protocollo aperto basato su messaggi, sviluppato per la comunicazione fra computer e dispositivi multimediali. Ogni messaggio OSC segue una sintassi prestabilita secondo il modello, e l'unità di trasmissione è chiamato "pacchetto OSC".

È un protocollo molto utilizzato nei progetti di IoT, o più in generale, dove sono coinvolti circuiti integrati, microcontrollori e sensori. Infatti, è disponibile per Arduino una libreria open-source *OSCMessages.h* che permette di formattare i dati secondo questo protocollo. I tipi di dato supportati dai messaggi OSC sono *int32*, *float32*, *OSCstring* e *OSC-blob*. Si possono concatenare più tipi di dato in un unico "messaggio contenitore" chiamato *OSC Bundle* e la sua dimensione deve essere sempre un multiplo di 4 byte. In Figura 4.14 sono schematizzate le parti con cui un *OSC Bundle* è composto, formato che verrà utilizzato per concatenare i nostri dati.

Data	Size	Purpose
OSC-string "#bundle"	8 bytes	How to know that this data is a bundle
OSC-timetag	8 bytes	Time tag that applies to the entire bundle
Size of first bundle element	int32 = 4 bytes	First bundle element
First bundle element's contents	As many bytes as given by "size of first bundle element"	
Size of second bundle element	int32 = 4 bytes	Second bundle element
Second bundle element's contents	As many bytes as given by "size of second bundle element"	
etc.		Additional bundle elements

Figura 4.14: Tabella che raffigura le parti di cui è composto un *OSC Bundle* contenente due o più messaggi

Fonte: http://opensoundcontrol.org/spec-1_0

Considerando che il nostro *OSC Bundle* conterrà un *char* come identificatore per l'IMU (1 byte), quattro float del quaternion (16 byte), header e padding dovuti al protocollo OSC, la dimensione del pacchetto complessiva sarà di 56 byte.

Utilizzando questo metodo si è riusciti ad inviare correttamente le informazioni di orientamento degli IMU ad HoloLens. La simulazione del movimento

utilizzando i dati ricevuti è fluido e continuo. Si verifica però un ritardo non trascurabile fra il movimento dei sensori e le informazioni ricevute. Questo ritardo, inizialmente impercettibile, va ad aumentare con il tempo d’esecuzione del programma.

Ipotizzando che il ritardo possa essere attribuito ad un eccessivo carico computazionale della scheda ESP8266, oppure ad un’eccessiva dimensione del traffico di rete, si è deciso di implementare una soluzione personalizzata per la formattazione dei pacchetti.

Secondo metodo Allo scopo di risolvere il ritardo riscontrato con il primo metodo, si è deciso di optare per una soluzione personalizzata per la serializzazione dei dati.

Si è creata una struttura dati che permetta di scomporre un *float* in un array di bytes per poterli accedere singolarmente. A questo punto è stato possibile aggregare i bytes a piacimento, creando una sequenza personalizzata (Figura 4.15).

ID	W float				X float				Y float				Z float			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17 bytes																

Figura 4.15: Schema della struttura del pacchetto utilizzato. ID è un *char* (1 byte) e identifica l’IMU.

In questo modo non si è dovuto ricorrere a complesse conversioni di dato, riducendo sia il carico computazionale sia la dimensione del pacchetto finale.

Deserializzando correttamente i pacchetti lato ricezione su HoloLens, è stato possibile eliminare il ritardo osservato col primo metodo e quindi generare un’esperienza fluida e reattiva.

4.4 Implementazione su HoloLens

Indossando i due IMU sul braccio ed avambraccio dell’utente, si è ora in grado di tracciare i movimenti dell’arto ed inviarle ad HoloLens.

Per generare un modello virtuale del braccio che segua quello reale, si è proceduto in questo modo:

1. Generare un modello 3D umanoide da utilizzare come avatar virtuale

2. Vincolare la posizione dell'avatar con la posizione dell'HoloLens
3. Applicare le rotazioni ricevute dai sensori alle articolazioni di spalla e gomito dell'avatar

Questo approccio, tanto semplice quanto immediato, ha permesso sin da subito di verificare e testare il sistema. Inizialmente si è notato un eccessivo disallineamento fra il braccio reale e quello virtuale. Le motivazioni sono da attribuire principalmente alle dimensioni del modello virtuale e alla calibrazione dei sensori.

Nel primo caso, visto che come punto di ancoraggio dell'intero modello 3D viene utilizzata la posizione di HoloLens, dimensioni prefissate come grandezza del capo e distanza testa-spalla, giocano un ruolo importante su un corretto allineamento.

Nel secondo caso, la calibrazione dei sensori IMU si è dimostrata essere essenziale per un tracciamento fluido. Per funzionare correttamente, i sensori richiedono che al loro avvio vengano tenuti in una posizione stabile per circa 5-10 secondi. Si è verificato che in mancanza di una corretta calibrazione all'avvio, il tracciamento fosse soggetto ad instabilità ed errori come *drift* nel tempo.

4.4.1 Calibrazione

Per cercare di limitare o risolvere i problemi appena menzionati, si è implementata una fase di calibrazione che si svolge in questo modo:

1. Viene mostrato un pannello di fronte all'utente contenente un bottone virtuale *Calibrate* ed un timer di 10 secondi.
2. L'utente distende il braccio di fronte a sè in modo tale che la mano sia in corrispondenza del pannello
3. Allo scadere del timer, l'utente preme il tasto *Calibrate*

Questa procedura ha permesso di allineare in maniera considerevole il modello virtuale. Il timer aiuta a scandire il tempo necessario alla calibrazione elettronica dei sensori, durante il quale il braccio dell'utente è disteso e stabile. Il Tap sul tasto *Calibrate* innesca una duplice funzione.

Innanzitutto, serve a calibrare l'orientamento dei sensori sull'asse Y. L'MPU-6050, in mancanza di un magnetometro, non è in grado di rilevare un orientamento assoluto sull'asse Y, ma solo i gradi di rotazione rispetto ad un



Figura 4.16: Fase di calibrazione degli IMU con HoloLens

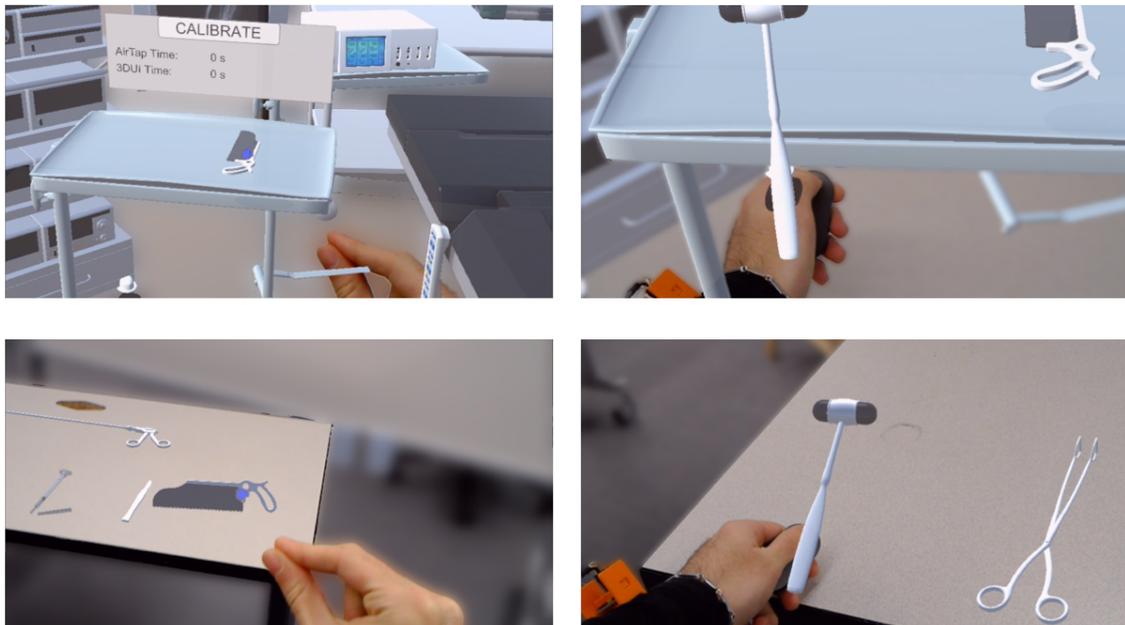
orientamento iniziale arbitrario. Gli altri due assi, X e Z, vengono invece correttamente calibrati autonomamente grazie alla dipendenza dalla forza di gravità, rilevata dall'accelerometro. Quando l'utente clicca sul tasto *Calibrate* di fronte a lui, viene calcolato l'offset fra la rotazione su Y di HoloLens e quella dei sensori. Sottraendo quest'offset alle rotazioni ricevute dai sensori, il modello risulterà orientato correttamente.

In secondo luogo, il pannello virtuale funge da punto di riferimento per rilevare la lunghezza del braccio. Infatti, calcolando la distanza fra HoloLens ed il pannello al momento del click, è possibile assumere questa quantità come lunghezza del braccio reale e di conseguenza applicarla al modello virtuale.

4.4.2 Scenario

È stato creato uno scenario dove poter utilizzare il sistema come metodo di interazione. L'applicazione rappresenta uno strumento di apprendimento ed è ambientata in una sala chirurgica popolata di accessori e macchinari medici in scala 1:1. Con lo scopo di migliorare la percezione degli spazi e la confidenza nel maneggiare attrezzature mediche, l'utente dovrà portare degli strumenti da un punto A ad un punto B.

Per eseguire queste azioni, sono state implementate due forme d'interazione. La prima si basa sul sistema GVV integrato in HoloLens, secondo il quale l'utente dovrà puntare con lo sguardo sullo strumento desiderato e con un Air-Tap trascinarlo verso la destinazione. La seconda modalità prevede



(a) Interazione col modello GGV di HoloLens (b) Interazione col tracciamento del braccio

Figura 4.17: Comparazione dei sistemi d'interazione

l'utilizzo del sistema d'interazione basato sul tracciamento del braccio creato durante il progetto. In questo caso sarà necessario che l'utente raggiunga fisicamente lo strumento, lo intersechi con la mano e lo trasporti fino alla destinazione.

In Figura 4.17 è rappresentato un esempio dei due tipi di interazione.

Parte II

Interazione in Realtà
Virtuale

Capitolo 5

Progettazione

In questa parte dell'elaborato verrà discusso il lavoro effettuato nel campo della Realtà Virtuale, descrivendo le tecnologie coinvolte e spiegando il processo che ha portato alla realizzazione di un sistema d' interazione.

5.1 HTC Vive

Il dispositivo che è stato utilizzato per condurre lo studio in Realtà Virtuale è HTC Vive: è un dispositivo progettato da Valve¹ in collaborazione con HTC², presentato a Marzo 2015 e rilasciato al pubblico a Giugno 2016. Esso permette di sperimentare esperienze VR in modalità "roomscale". Questo significa che il dispositivo offre la possibilità di generare un ambiente virtuale attorno all'utente il quale, oltre ad osservare, può spostarsi e camminare in esso, pur restando nei confini dell'area di tracciamento.

Le componenti principali di cui è composta la suite HTC Vive sono:

- **Vive Headset:** questo è il visore indossabile che contiene due pannelli OLED con frequenza di aggiornamento a 90Hz e risoluzione 1080x1200 ciascuno, posizionati in modo da creare un campo visivo di 110 gradi (Figura 5.1a). L'hardware include anche una serie di sensori come accelerometro, giroscopio e led infrarossi che permettono di tracciare sia l'orientamento che la posizione del visore nello spazio. Si collega al PC con un cavo HDMI per il flusso video ed un cavo USB per la sincronizzazione dei dati.

¹<https://www.valvesoftware.com/it/>

²<https://www.htc.com/it/>

- **Vive Controllers:** i controllers sono il metodo principale di input a disposizione dell'utente (Figura 5.1b). Includono un touch-pad, dei bottoni ed una batteria integrata ricaricabile. Comunicano col sistema via wireless e anch'essi contengono sensori inerziali e led infrarossi per il tracciamento spaziale.
- **Base Stations:** le stazioni base sono una coppia di cubetti neri che permettono di delimitare un ambiente virtuale attorno all'utente (Figura 5.1c). Esse sono alla base del funzionamento del sistema di tracciamento di HTC Vive: emettendo dei raggi ad infrarossi temporizzati, permettono al visore ed ai controller di tracciare la loro posizione rispetto ad essi con un alto grado di precisione (meno di 1 cm). Inoltre, esse definiscono e delimitano l'area spaziale che è possibile utilizzare per spostarsi nel mondo virtuale (circa 5m x 5m).
- **Vive Tracker:** è un accessorio ideato per essere ancorato ad un oggetto fisico reale in modo da poterne tracciare il movimento nel mondo virtuale (Figura 5.1d). Il principio di funzionamento è lo stesso del visore e dei controller: integrando sensori inerziali e led infrarossi, può essere tracciato all'interno dell'area delle stazioni base.



(a) Visore



(b) Controllers



(c) Base Station



(d) Tracker

Figura 5.1: Componenti del sistema HTC Vive

5.2 Idea ed obiettivo

Come descritto nella Sezione 2.2, la maggior parte delle tecnologie che offrono esperienze di Realtà Virtuale sono sviluppate per consentire la fruizione di contenuti audiovisivi e quindi si concentrano soprattutto nello stimolo dei sensi come vista ed udito. Diversi studi hanno dimostrato come il coinvolgimento di ulteriori sensi possa beneficiare notevolmente l'esperienza d'utilizzo

complessiva aumentando il senso d'immersione dell'utente. Fra questi, il senso del tatto si è dimostrato essere di grande importanza ed infatti è al centro di numerosi studi e ricerche.

Si è deciso, quindi, di approfondire questo aspetto e trovare un modo per stimolare il senso del tatto tramite la progettazione di un nuovo metodo d'interazione in Realtà Virtuale. Ad oggi esistono diverse interfacce che hanno lo scopo di fornire un feedback tattile, ma data la novità di questi metodi, le tecnologie coinvolte sono spesso complesse e costose. Inoltre, non tutte sono direttamente applicabili al mondo della Realtà Virtuale, introducendo limitazioni nei movimenti che ne compromettono l'esperienza.

Per questo motivo, il metodo d'interazione per la Realtà Virtuale basato sul feedback tattile è stato progettato sui seguenti requisiti:

- Bassa complessità del sistema: l'interfaccia sarà composta da elementi semplici, che non richiedono l'utilizzo di elettricità per fornire un feedback tattile (Dispositivi aptici passivi, Sezione 2.2.2)
- Reperibilità ed accessibilità dei componenti: utilizzo di strumenti che è possibile trovare in commercio e ad un costo relativamente contenuto
- Riconfigurabilità: avere la possibilità di riutilizzare gli stessi strumenti in più scenari differenti

L'idea alla base del sistema consiste nel posizionare ed utilizzare degli oggetti reali in corrispondenza delle controparti virtuali. Questi oggetti saranno progettati e realizzati in modo che le loro proprietà fisiche, come dimensioni, forma e peso, corrispondano a quelli virtuali con un certo grado di fedeltà. La corretta gestione e sincronizzazione di questi oggetti con i loro modelli virtuali porterà l'utente a credere che quel determinato modello virtuale abbia una consistenza fisica, e di conseguenza aumentarne l'immersione.

Sulla base di questi principi, verrà quindi progettata un'interfaccia di interazione 3D per la Realtà Virtuale che permetta all'utente di "sentire" gli elementi virtuali toccandoli ed interagendo con essi proprio come farebbe nel mondo reale.

5.3 Architettura

In Figura 5.2 è riportato uno schema dei macro blocchi di cui sarà composto il sistema:

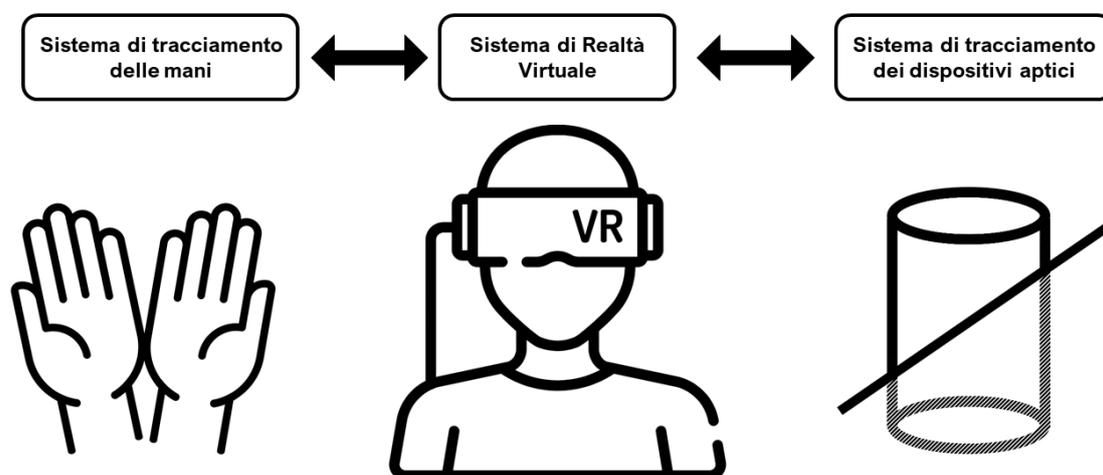


Figura 5.2: Schema dell'interfaccia di interazione tattile in Realtà Virtuale

- **Sistema di Realtà Virtuale:** questo è il cuore del sistema che genera un ambiente di Realtà Virtuale. Come anticipato nella Sezione 5.1, per questo progetto verrà usato il dispositivo HTC Vive. Con la combinazione di Headset e basette di tracciamento, è possibile delimitare uno spazio ed avviare un'esperienza di Realtà Virtuale e muoversi all'interno di esso. Questa è il sistema di partenza che verrà utilizzato comunicare con gli altri moduli e sincronizzare i dati.
- **Dispositivi aptici:** qui risiedono le componenti necessarie alla realizzazione di un'interfaccia di interazione in grado di fornire un feedback aptico. Per fare ciò, questo modulo prevede i seguenti passaggi:
 - Scelta dello strumento da utilizzare per la realizzazione degli oggetti fisici
 - Progettazione delle forme e del numero di oggetti da realizzare in maniera da coprire una più vasta gamma di interazioni fisiche possibili
 - Il tracciamento di questi oggetti realizzati affinché sia possibile sincronizzarne posizioni e movimenti delle controparti virtuali

- **Tracciamento delle mani:** questa è una conseguenza del principio su cui si basa questo metodo di interazione. Dato che l'utente dovrà fisicamente approssimare gli oggetti con i quali desidera interagire (es. prenderli, spostarli o anche solo toccarli), le sue mani dovranno essere libere da qualsiasi strumento. Questo significa che visto che in questo sistema non sarà possibile utilizzare i tradizionali Vive Controllers, sarà necessario un altro strumento per il tracciamento della posizione delle mani. Data la fisicità delle interazioni con gli oggetti, è importante che l'utente abbia un feedback visivo delle sue mani nel mondo virtuale per evitare errori e disorientamento.

Nel prossimo capitolo si entrerà nel dettaglio dell'implementazione del sistema appena presentato, descrivendo le tecnologie e gli strumenti utilizzati per la realizzazione.

Capitolo 6

Realizzazione

Per la realizzazione del sistema di interazione progettato nel capitolo precedente, si è dapprima definito uno scenario applicativo nel quale utilizzare i concetti presentati. La presenza di un contesto nel quale applicare questo paradigma d'interazione, aiuta a determinare quali siano le principali necessità che il sistema deve coprire e far fronte ad eventuali criticità che possono emergere.

6.1 Applicazione

Lo scenario scelto per l'implementazione dell'interfaccia di interazione tattile in Realtà Virtuale è quello di un Escape Room.

Questo è tipo di attività nel quale all'utente è richiesto di risolvere dei puzzle per poter avanzare nel gioco e terminarlo prima della scadenza di un tempo limite. Nella realtà, per risolvere un Escape Room, gli utenti devono fare molto affidamento al senso del tatto, in quanto è spesso richiesto di spostare degli oggetti in una determinata posizione oppure tirare o spingere leve.

Di conseguenza, questo lo rende un esempio ideale dove applicare e testare le capacità di un sistema d'interazione basata sul tatto.

6.1.1 Ambiente di sviluppo

Per lo sviluppo dell'applicazione è stato utilizzato il motore grafico Unity 3D, in particolare nella versione 2018.4.0f1.

Unity è compatibile con i principali sistemi di Realtà Virtuale disponibili, ma

per interfacciarsi correttamente con HTC Vive (utilizzato in questo progetto), è necessario che sul PC sia installata la suite Steam¹ e SteamVR². Infine, una volta creato un nuovo progetto Unity, è necessario importare il Plugin SteamVR, disponibile gratuitamente sullo Unity Asset Store³: questo plugin è l'SDK di SteamVR, cioè un insieme di librerie ufficiali distribuite da Valve in modo da facilitare lo sviluppo con i principali visori di Realtà Virtuale. Per questo progetto è stato utilizzato lo SteamVR plugin in versione 2.0.1.

6.1.2 Struttura

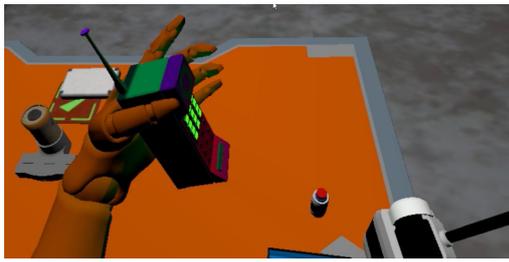
L'applicazione di Escape Room creata è ambientata in un laboratorio nucleare futuristico dove è in corso un'emergenza. All'inizio del gioco, le luci sono spente ed un telefono squilla nel buio, illuminando leggermente l'ambiente circostante. Durante la chiamata, una voce spiega all'utente la situazione di emergenza e vengono forniti dei suggerimenti per risolverla. La procedura prevede il superamento di tre livelli:

- Nel primo livello (Figura 6.1a) l'utente deve ripristinare la corrente del generatore di emergenza azionando tre volte una leva posta di fronte a lui. Visto che inizialmente la stanza è al buio, l'utente potrà aiutarsi solo con una leggera luce proveniente dallo schermo del telefono e sarà spinto a toccare gli oggetti vicini per identificarli.
- Il secondo livello (Figura 6.1b) è ambientato in una sala di controllo dove l'utente deve liberare la zona da rifiuti radioattivi utilizzando un braccio robotico a tre assi che può essere controllato con delle manopole. Inoltre, per attivare il braccio robotico, sarà necessario inserire ed orientare correttamente la batteria nell'inserito di alimentazione.
- Nel terzo livello (Figura 6.1c), ambientato nella sala del reattore nucleare, l'utente dovrà inserire un codice a 4 cifre per attivare le barre di contenimento e risolvere l'emergenza. Per scoprire il codice, dovrà guardarsi attentamente attorno alla ricerca di indizi, puntando un faro UV che ha a disposizione sul tavolo.

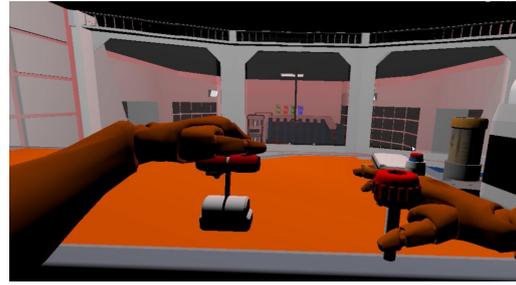
¹<https://store.steampowered.com/about/>

²<https://store.steampowered.com/steamvr>

³<https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>



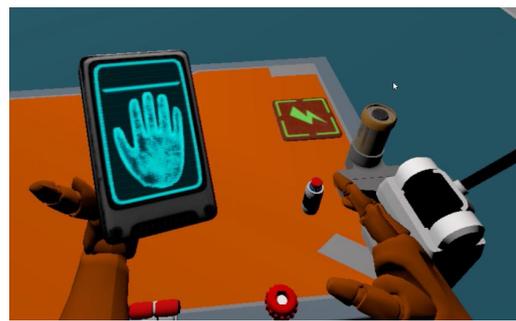
(a) Livello 1



(b) Livello 2



(c) Livello 3



(d) Sistema di teletrasporto

Figura 6.1: Schermate dei vari livelli dell'applicazione

A partire dal secondo livello, viene attivato un conto alla rovescia che resta visibile fino alla fine. Nel caso il tempo dovesse scadere prima dell'inserimento del pin corretto, il gioco termina con la simulazione di un'esplosione. Viceversa, una voce si congratula con l'utente, che viene portato in una zona sicura.

Per lo spostamento da un livello al successivo, si è prevista un'operazione di "teletrasporto" (Figura 6.1d). Quest'azione è stata studiata ed implementata in questo modo sia per le limitazioni spaziali di movimento dovute all'area di tracciamento dei sistemi di Realtà Virtuale, sia per organizzare al meglio l'utilizzo (e riutilizzo) degli oggetti aptici.

6.2 Strumenti aptici

Gli oggetti da utilizzare come strumenti fisici in corrispondenza degli oggetti virtuali, sono le componenti chiave per fornire un feedback tattile alla base di questa interfaccia d'interazione.

Per soddisfare i requisiti del progetto e quindi fornire uno strumento che permetta di realizzare gli oggetti in maniera semplice ed accessibile, si è scelto

di utilizzare i mattoncini della serie LEGO⁴.

L'utilizzo di blocchi della serie LEGO permette la creazione di dispositivi aptici in maniera facile ed immediata, offrendo grande flessibilità e riconfigurabilità. È possibile infatti costruire oggetti di qualsiasi forma e dimensione, permettendo di creare velocemente nuove interfacce in base alle necessità applicative.

Inoltre, è stata utilizzata una versione denominata LEGO Mindstorms EV3⁵: questo prodotto prevede l'utilizzo di un "mattoncino intelligente" che grazie ad un processore ARM9 con sistema operativo basato su Linux è in grado di acquisire dati da una vasta gamma di sensori. Con l'utilizzo di sensori come quello di pressione, o di rotazione, è possibile registrare le informazioni di stato o di rotazione ed inviarli via WiFi all'applicazione Unity per sincronizzare i movimenti dei corrispondenti elementi virtuali.

6.2.1 Interfacce

Una volta scelto lo strumento per la creazione delle interfacce aptiche, si è passati alla progettazione delle stesse: tenendo in considerazione lo scenario dimostrativo utilizzato, sono stati realizzati diversi oggetti, ognuno con la funzione di strumento fisico per uno o più elementi della scena virtuale.

Fra gli oggetti ideati e realizzati per questa applicazione, troviamo:

- Un oggetto cilindrico che viene utilizzato per diverse funzioni a seconda del livello. Per esempio, all'inizio del primo livello funge da telefono. Nel secondo livello viene utilizzato come batteria da inserire nel suo alloggiamento per attivare la corrente. Questa azione di inserimento della batteria nell'alloggiamento, fa sì che l'utente abbia inconsciamente generato un nuovo strumento che utilizzerà nel terzo livello: l'emettitore UV (Figura 6.2a).
- Un oggetto rettangolare che verrà utilizzato come tablet palmare per autorizzare il teletrasporto. Durante l'esperienza fornirà informazioni utili all'utente per capire lo stato del gioco, e potrà essere preso e spostato in qualsiasi momento (Figura 6.2a).
- Una leva equipaggiata con un elastico permetterà all'utente di esercitare delle forze come spingere e tirare (Figura 6.2b).

⁴<https://www.lego.com/>

⁵<https://www.lego.com/it-it/themes/mindstorms>



(a) Cilindro, alloggiamento batteria, unione cilindro ed alloggiamento, rettangolo



(b) Leva elastica, bottone, manopola e levetta

Figura 6.2: Oggetti fisici e virtuali

- Una levetta più piccola, una manopola ed un bottone, verranno utilizzati come strumenti di controllo in base al livello. Ad esempio, nel secondo

livello permettono di controllare il movimento del braccio robotico, mentre nel terzo saranno usati per inserire e confermare le cifre del codice di sicurezza (Figura 6.2b).

La serie di leve, manopola e bottone, è collegata al Mindstorms EV3 per raccogliere informazioni sul loro stato, e non vengono in nessun modo comandati attivamente per fornire un feedback tattile.

6.2.2 Tracciamento degli oggetti

Per far sì che gli elementi virtuali si muovano nello spazio in corrispondenza degli oggetti reali, è necessario fornire al sistema di Realtà Virtuale le informazioni di posizione ed orientamento di questi ultimi. È necessario quindi tracciare gli oggetti reali nello spazio, e per far questo sono stati utilizzati i Vive Trackers.

Come anticipato nella Sezione 5.1), i Trackers sono accessori della suite HTC Vive e quindi possono essere facilmente integrati e registrati. Dal punto di vista implementativo, si comportano come se fossero dei Controllers ma senza tasti per l'input. Essi vanno ancorati sull'oggetto che si vuole tracciare, e bisogna far corrispondere la posizione del Tracker rispetto al corpo dell'oggetto anche sul modello virtuale.

Per limitare il numero di Vive Tracker necessari e quindi la complessità del sistema, gli oggetti fisici sono stati studiati per questo. In particolare, alcuni oggetti come leve, manopola e bottone, sono stati ancorati sulla superficie di un tavolo, vicino al Mindstorm EV3 (sia per semplicità di cablaggio che per solidità). Di conseguenza, sarà necessario utilizzare un solo Tracker per tutti questi elementi, in quanto essi sono solidali l'uno con l'altro (Figura 6.3).

Il discorso cambia quando parliamo del cilindro utilizzato come telefono o batteria, oppure il rettangolo che funge da tablet. Questi dispositivi sono ideati per essere presi e spostati dall'utente, quindi la loro posizione deve essere tracciata singolarmente ed in maniera continua. Di conseguenza, ognuno di questi oggetti avrà un Tracker dedicato.

6.3 Tracciamento delle mani

L'utilizzo di oggetti fisici come strumento di interazione nel mondo virtuale, permette di abbandonare il paradigma d'interazione basato sui controllers ed attuare un approccio molto più naturale che vede l'utente interagire direttamente con l'oggetto che si intende utilizzare. Per far questo, non solo i

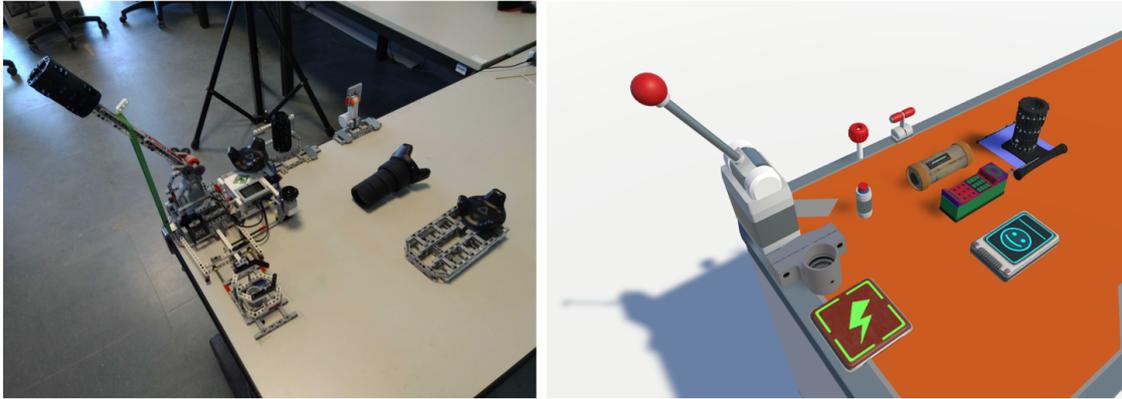


Figura 6.3: A sinistra la composizione degli oggetti fisici, a destra i corrispettivi elementi virtuali

controllers non sono più necessari, ma il loro utilizzo diventa un ostacolo in quanto l'utente non avrebbe le mani libere per poter afferrare i vari oggetti.

La mancanza dei controllers però pregiudica la capacità di tracciare la posizione e l'orientamento delle mani dell'utente, rendendo impossibile fornire un feedback visivo delle stesse. Questa mancanza risulta piuttosto disorientante e svantaggiosa, in quanto non solo l'utente non avrà un riferimento esatto della posizione delle sue mani, ma il fatto di doverle usare per toccare o afferrare degli oggetti, rende il compito ancora più difficile.

Per questo motivo sono state studiate delle tecnologie che permettessero il tracciamento delle mani dell'utente senza occuparle, potendo così sia utilizzare gli oggetti fisici sia ricevere un feedback visivo sulla posizione delle proprie mani.

6.3.1 Leap Motion

Descrizione

La prima tecnologia studiata e sperimentata all'interno del progetto è Leap Motion⁶, un dispositivo in grado di tracciare il movimento delle mani e delle dita utilizzando telecamere ad infrarossi.

Consiste di una periferica USB che è possibile montare sul visore di Realtà Virtuale ed attraverso 2 camere e 3 LED ad infrarossi è in grado di osservare l'area circostante fino alla distanza di circa 1 metro. Tramite il cavo USB,

⁶<https://www.leapmotion.com/>

questa periferica invia fino a 200 frames al secondo al PC dove è installato il Leap Motion Software che, tramite alcuni calcoli matematici, riesce ad estrapolare la posizione nello spazio 3D delle mani e delle dita di fronte alla camera.

Per integrarlo in un progetto Unity, è necessario aver installato il software Leap Motion Orion sul PC ed importare il pacchetto "Unity Core Assets" disponibile su sito⁷. L'SDK comprende alcuni esempi grazie ai quali è possibile implementare sin da subito le funzionalità di tracciamento delle mani.

Considerazioni

Questa è una soluzione sicuramente interessante, in quanto grazie al tracciamento ad infrarossi, le mani dell'utente restano totalmente libere da ogni vincolo o impedimento.

Durante i primi test di utilizzo in accoppiata al sistema di HTC Vive, il tracciamento è risultato molto accurato e soprattutto veloce.

Purtroppo però il comportamento non è risultato altrettanto soddisfacente quando lo si è utilizzato insieme al sistema d'interazione tattile. Le osservazioni principali sono:

- Quando si prende in mano un oggetto, il tracciamento di Leap Motion diventa molto instabile e vengono generati artefatti. La posa dello scheletro tracciata cambia in maniera molto rapida e spesso risulta sbagliata. Il problema sembra essere dovuto alla presenza di oggetti in mano, che probabilmente mette in difficoltà l'algoritmo di stima della posa.
- Mentre Leap Motion è montato sul visore ed è in funzione, il tracciamento dei Vive Tracker collegati agli oggetti risulta instabile. I rispettivi oggetti virtuali sembrano muoversi in maniera incontrollata e non rispecchiano la posizione di quelli reali. Probabilmente questo può essere dovuto ad un'interferenza fra i raggi infrarossi del Leap Motion e quelli utilizzati dal sistema di HTC Vive.

Quest'ultimo comportamento risulta particolarmente limitante tanto da compromettere l'intera esperienza e rendendo questo dispositivo non adatto all'utilizzo in questo caso.

⁷<https://developer.leapmotion.com/unity5436356>

6.3.2 Manus VR

Descrizione

La seconda tecnologia di tracciamento studiata è Manus VR⁸. Questa consiste in un paio di guanti wireless che contengono 2 IMUs e 5 sensori di flessione per ogni mano. Questo gli permette di tracciare l'orientamento della mano e la flessione di ogni dito. Il tracciamento della mano è basato su principi totalmente differenti da quelli di Leap Motion, ed inoltre non fornisce informazioni sulla posizione assoluta della mano nello spazio. Per ovviare a questa limitazione, i guanti vengono utilizzati in accoppiata ad un Vive Tracker per polso, in modo da avere un punto di riferimento spaziale di partenza sul quale applicare il modello della mano calcolato dai Manus.

Implementazione

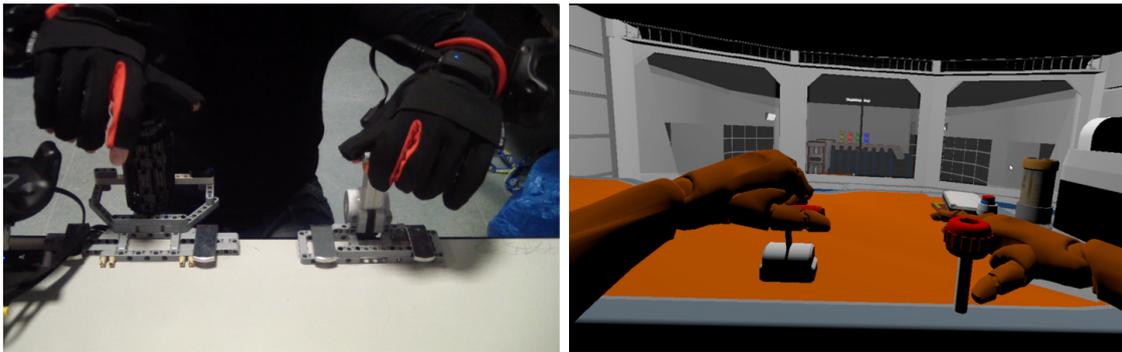


Figura 6.4: Esempio di utilizzo dei Manus VR con l'interfaccia d'interazione tattile

Il processo necessario per utilizzare i Manus all'interno di un applicazione Unity risulta leggermente più articolato rispetto a quello visto con Leap Motion.

Innanzitutto è necessario installare il Software Apollo, contenente l'SDK dei Manus. Apollo permette di testare la connettività con i guanti ma soprattutto di effettuare la calibrazione, che si suddivide in 2 fasi:

- Calibrazione dell'orientamento: cioè riguarda la calibrazione dei sensori inerziali (IMU)

⁸<https://manus-vr.com/>

- Calibrazione delle dita: attraverso il mantenimento di alcune pose della mano come mostrato su schermo, viene calibrato il grado di flessione per ogni dito su quell'utente.

Quest'ultima calibrazione andrebbe ripetuta ogni volta che i guanti vengono indossati da un utente diverso.

Terminato questo primo processo di calibrazione, è possibile utilizzare i Manus in un applicazione Unity. Come anticipato, vengono fornite solo le informazioni di rotazione del palmo e delle articolazioni delle dita. Perciò per ricavare la posizione sono stati utilizzati altri 2 Vive Trackers correttamente configurati.

Durante i primi test, l'accuratezza del tracciamento non era ottimale. I modelli virtuali delle mani risultavano leggermente sfasati sia riguardo la posizione che l'orientamento rispetto alle mani reali. Nonostante ciò, il loro utilizzo non ha compromesso l'utilizzo dell'interfaccia d'interazione tattile a differenza di quanto sperimentato con Leap Motion.

Calibrazione

Si è notato come l'errore sulla posizione delle mani virtuali era dovuto ad un errato scalamento del modello 3D. Non si hanno infatti informazioni sulla dimensione della mano dell'utente, ed applicando un modello 3D sempre delle stesse dimensioni, può portare ad incongruenze spaziali che rendono difficoltoso raggiungere ed interagire correttamente con le interfacce fisiche. Riguardo l'orientamento, invece, si è notato come la dipendenza dai sensori inerziali possa portare a leggeri sfasamenti nel tempo del polso. Per migliorare la precisione del tracciamento, quindi, è stata creata un'ulteriore fase di calibrazione all'interno dell'applicazione Unity, che prevede due processi:

- Il primo mira a minimizzare l'errore di posizione calcolando il giusto scalamento della mesh rispetto alla mano dell'utente. Per fare questo, viene utilizzato temporaneamente Leap Motion per calcolare la dimensione della mano ed applicandolo al modello dei Manus VR.
- Il secondo mira a correggere l'errato orientamento dei guanti utilizzando temporaneamente i Vive Controllers come punto di riferimento. Come mostrato in Figura 6.5, l'utente punta le mani nella stessa direzione dei Controllers e preme il grilletto. A questo punto viene calcolato ed

applicato l'orientamento di quest'ultimi alle mani. Inoltre, dato che indossando il visore è possibile vedere sia il modello ricostruito della mano sia i modelli dei Controllers, l'utente può ripetere quest'azione finché la posizione delle mani non sia coerente con quella dei Controllers.



Figura 6.5: Pannello di calibrazione dell'orientamento dei Manus VR

Capitolo 7

Conclusioni

Il progetto presentato in questo elaborato di tesi riguarda l'interazione uomo-macchina con sistemi di realtà virtuale.

Con il tentativo di ricreare oppure alterare la percezione del mondo reale attraverso un ambiente virtuale, le moderne tecnologie immersive, come Realtà Virtuale e Realtà Mista, hanno introdotto un nuovo punto di vista su come l'utente si relaziona con il dispositivo. Concetti del tutto nuovi, come *sensazione di immersione*, *presenza spaziale* e *naturalità* hanno dimostrato di essere fattori molto importanti nel determinare la qualità di un'esperienza immersiva.

Con l'obiettivo di contribuire alla ricerca e allo sviluppo dei sistemi d'interazione immersivi, sono state progettate e realizzate delle soluzioni che mirano a superare alcune limitazioni delle tecnologie esistenti.

Come si è potuto osservare in letteratura, i fattori (e le limitazioni) che caratterizzano un'esperienza immersiva, variano da tecnologia a tecnologia: ad esempio, si è notato come le interfacce di interazione esistenti in Realtà Mista, a differenza di quelle in Realtà Virtuale, possano risultare macchinose e poco naturali. Viceversa, i sistemi di Realtà Virtuale risentono della mancanza di un feedback tattile durante l'interazione con gli oggetti, fattore non applicabile in Realtà Mista in quanto si mantiene il contatto col mondo reale. Questa analisi ha portato alla scelta di ideare e creare due sistemi per affrontare al meglio le caratteristiche di ogni tecnologia.

Utilizzando Microsoft HoloLens come strumento di riferimento per le esperienze di Realtà Mista, è stato progettato e realizzato un prototipo di dispositivo indossabile che traccia il movimento di braccio ed avambraccio dell'utente tramite sensori inerziali.

Sfruttando queste informazioni per ricreare un braccio virtuale che simuli

quello reale dell'utente, è stato possibile slegare HoloLens dal modello d'interazione GGV (Gaze Gesture Voice, Sezione 3.1.5). Quest'ultimo, infatti, richiede all'utente di utilizzare costantemente la testa come metodo di puntamento e di mantenere la mano in una determinata posa di fronte al visore per selezionare, affaticando collo e spalle durante un utilizzo prolungato. Il prototipo indossabile realizzato, invece, permette di arricchire l'insieme di informazioni tracciate da HoloLens, offrendo la possibilità di implementare interazioni più semplici e naturali basate sul movimento e posizione dell'intero braccio dell'utente.

Per affrontare il problema della mancanza di feedback tattile in Realtà Virtuale, è stato progettato invece un sistema di interazione che prevede l'utilizzo di oggetti riconfigurabili al posto dei tradizionali controller. Questi oggetti sono stati creati e disposti nel mondo reale in maniera che corrispondano ad elementi del mondo virtuale.

Grazie al tracciamento spaziale sia degli oggetti che delle mani dell'utente, basato su sensori ottici ed inerziali, è stato possibile realizzare un sistema che permetta di *toccare* gli elementi virtuali.

Implementando uno scenario applicativo d'esempio, si è dimostrato come sia possibile fornire un feedback tattile tramite l'impiego di semplici oggetti opportunamente creati.

In un contesto come quello attuale, nel quale l'avanzamento delle tecnologie immersive propone dispositivi sempre più sofisticati e complessi, le due soluzioni prodotte vengono presentate come alternative basate su sistemi semplici e pratici.

Tramite l'impiego di strumenti a basso costo e facilmente accessibili (Arduino, sensori inerziali e blocchi LEGO) è stato infatti possibile ideare e realizzare dei sistemi prototipali che espandano le capacità delle tecnologie immersive esistenti, come appunto Realtà Virtuale e Realtà Mista.

Sebbene le soluzioni presentate siano allo stato sperimentale e richiedano ulteriori test per essere perfezionate ed esaminate, i principi su cui si basano le rendono interessanti per ulteriori studi ed approfondimenti.

Sono sicuramente presenti margini di miglioramento che è possibile raggiungere con sviluppi futuri ed il perfezionamento delle tecniche utilizzate.

Uno degli elementi che richiede maggiore attenzione riguarda l'accuratezza nel tracciamento spaziale. È evidente come la precisione con la quale è possibile rilevare movimenti ed oggetti ha un grosso impatto sull'esperienza immersiva finale. La soluzione presentata in Realtà Virtuale richiede un livello di precisione del tracciamento che non sempre è stato possibile raggiungere a causa di interferenze oppure oclusioni alle quali sono soggette le tecnologie

utilizzate, basate su una combinazione di sensori ottici ed inerziali.

Allo stesso modo, il grado di accuratezza del tracciamento del braccio in Realtà Mista, basato unicamente su sensori inerziali, potrebbe migliorare se abbinato ad una seconda fonte di informazioni spaziali, come ad esempio un magnetometro.

La progettazione di nuovi scenari e casi d'uso che sfruttino le soluzioni presentate, sarebbe certamente molto utile per raffinare le tecniche coinvolte ed identificare i campi applicativi che meglio si prestano ad implementare questi metodi d'interazione.

Bibliografia

- [1] Merwan Achibet, Adrien Girard, Anthony Talvas, Maud Marchal, and Anatole Lécuyer. Elastic-arm: Human-scale passive haptic feedback for augmenting interaction and perception in virtual environments. In *2015 IEEE Virtual Reality (VR)*, pages 63–68. IEEE, 2015.
- [2] Merwan Achibet, Maud Marchal, Ferran Argelaguet, and Anatole Lécuyer. The virtual mitten: A novel interaction paradigm for visuo-haptic manipulation of objects using grip force. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 59–66. IEEE, 2014.
- [3] Kian Adabi, Hayeem Rudy, Carrie S Stern, Katie Weichman, Oren Tepper, and Evan S Garfein. Optimizing measurements in plastic surgery through holograms with microsoft hololens. *Plastic and Reconstructive Surgery Global Open*, 5(9 Suppl), 2017.
- [4] Manuel Aiple and André Schiele. Pushing the limits of the cybergraspTM for haptic rendering. In *2013 IEEE international conference on robotics and automation*, pages 3541–3546. IEEE, 2013.
- [5] Andrew C Boud, David J Haniff, Chris Baber, and SJ Steiner. Virtual reality and augmented reality as a training tool for assembly tasks. In *1999 IEEE International Conference on Information Visualization (Cat. No. PR00210)*, pages 32–36. IEEE, 1999.
- [6] Lung-Pan Cheng, Li Chang, Sebastian Marwecki, and Patrick Baudisch. iturk: Turning passive haptics into active haptics by making users reconfigure props in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 89. ACM, 2018.
- [7] Heather Culbertson, Samuel B Schorr, and Allison M Okamura. Haptics: The present and future of artificial touch sensation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:385–409, 2018.
- [8] Cooper Davies, Jade White, Alec McAllister, Adam Saroka, Omar Addam, Fatemeh Hendijani Fard, and Frank Maurer. A toolkit for building collaborative immersive multi-surface applications. In *Proceedings of the*

-
- 2016 ACM International Conference on Interactive Surfaces and Spaces, pages 485–488. ACM, 2016.
- [9] David Drascic and Paul Milgram. Perceptual issues in augmented reality. In *Stereoscopic displays and virtual reality systems III*, volume 2653, pages 123–135. International Society for Optics and Photonics, 1996.
- [10] Gabriel Evans, Jack Miller, Mariangely Iglesias Pena, Anastacia MacAllister, and Eliot Winer. Evaluating the microsoft hololens through an augmented reality assembly application. In *Degraded Environments: Sensing, Processing, and Display 2017*, volume 10197, page 101970V. International Society for Optics and Photonics, 2017.
- [11] Yuki Fujimori, Yoshiyuki Ohmura, Tatsuya Harada, and Yasuo Kuniyoshi. Wearable motion capture suit with full-body tactile sensors. In *2009 IEEE International Conference on Robotics and Automation*, pages 3186–3193. IEEE, 2009.
- [12] Ronan Hinchet, Velko Vechev, Herbert Shea, and Otmar Hilliges. Dextres: Wearable haptic feedback for grasping in vr via a thin form-factor electrostatic brake. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, pages 901–912. ACM, 2018.
- [13] Cédric Kervegant, Félix Raymond, Delphine Graeff, and Julien Castet. Touch hologram in mid-air. In *ACM SIGGRAPH 2017 Emerging Technologies*, page 23. ACM, 2017.
- [14] Martijn JL Kors, Karel Millenaar, Erik D Van der Spek, Gabriele Ferri, Ben AM Schouten, and Tim Marsh. Shaping attitudes across realities. exploring strategies for the design of persuasive virtual, augmented and mixed reality games. In *International Conference on Entertainment Computing*, pages 497–501. Springer, 2017.
- [15] Néstor Andrés Arteaga Martin, Victor Mittelstädt, Michael Prieur, Rainer Stark, and Thomas Bär. Passive haptic feedback for manual assembly simulation. *Procedia CIRP*, 7:509–514, 2013.
- [16] Thomas H Massie, J Kenneth Salisbury, et al. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55, pages 295–300. Citeseer, 1994.
- [17] Microsoft. Holoacademy, mixed reality basics tutorials. <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-101>, 2019.
- [18] Microsoft. Holotoolkit-unity-v.1.2017.1.2. <https://github.com/Microsoft/MixedRealityToolkit-Unity/releases/tag/v1.2017.1>.

- 2, 2019.
- [19] Adeline Pihuit, Paul G Kry, and Marie-Paule Cani. Hands on virtual clay. In *2008 IEEE International Conference on Shape Modeling and Applications*, pages 267–268. IEEE, 2008.
 - [20] Michael Romaniuk, Jason Lamb, Josh Mitchell, Ilana Bayer, and Bruce Wainman. The promise of mixed reality in anatomy education. *The FASEB Journal*, 31(1_supplement):736–6, 2017.
 - [21] Martijn J Schuemie and CAPG Van der Mast. Presence: Interacting in virtual reality. In *Proceedings, Twente Workshop on Language Technology*, volume 15, pages 213–217, 1999.
 - [22] Zhibin Song and Shuxiang Guo. Development of a real-time upper limb’s motion tracking exoskeleton device for active rehabilitation using an inertia sensor. In *2011 9th World Congress on Intelligent Control and Automation*, pages 1206–1211. IEEE, 2011.
 - [23] Davin Karl Swanson. *Implementation of arbitrary path constraints using dissipative passive haptic displays*. PhD thesis, Citeseer, 2003.
 - [24] Arthur Tang, Frank Biocca, and Lynette Lim. Comparing differences in presence during social interaction in augmented reality versus virtual reality environments: An exploratory study. *Proceedings of PRESENCE*, pages 204–208, 2004.
 - [25] G Tao, Philippe S Archambault, and MF Levin. Evaluation of kinect skeletal tracking in a virtual reality rehabilitation system for upper limb hemiparesis. In *2013 international conference on virtual rehabilitation (ICVR)*, pages 164–165. IEEE, 2013.
 - [26] Oren M Tepper, Hayeem L Rudy, Aaron Lefkowitz, Katie A Weimer, Shelby M Marks, Carrie S Stern, and Evan S Garfein. Mixed reality with hololens: where virtual reality meets augmented reality in the operating room. *Plastic and reconstructive surgery*, 140(5):1066–1070, 2017.
 - [27] John R Wilson. Virtual environments and ergonomics: needs and opportunities. *Ergonomics*, 40(10):1057–1077, 1997.
 - [28] A. Zenner and A. Krüger. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(4):1285–1294, April 2017.