



Politecnico di Torino

Ingegneria Informatica

Tesi di Laurea Magistrale

# Tracking di precisione a supporto della manutenzione e riparazione in ambito industriale

**Relatore**

Andrea Sanna

**Candidato**

Giuseppe Ferraro

Luglio 2020

# Indice

<b>Elenco delle figure</b>	<b>V</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 La realtà aumentata</b>	<b>3</b>
2.1 Un pò di storia . . . . .	3
2.2 I dispositivi della realtà aumentata . . . . .	6
2.2.1 Sistemi di proiezione ottica e monitor . . . . .	7
2.2.2 Dispositivi portatili . . . . .	8
2.2.3 Sistemi di visualizzazione indossabili . . . . .	10
2.2.4 Microsoft Hololens . . . . .	15
<b>3 Stato dell'arte</b>	<b>24</b>
3.1 Tracking e impiego industriale . . . . .	24
3.2 Hardware e software . . . . .	25
3.2.1 Trimble . . . . .	25
3.2.2 Eos . . . . .	26
3.2.3 Vuforia e Unity . . . . .	26
3.2.4 VisionLib e Unity . . . . .	35
3.2.5 Deploy di un'applicazione AR . . . . .	40
<b>4 Sviluppo di una soluzione con Hololens</b>	<b>43</b>
4.1 Requisiti progetto . . . . .	43
4.2 Sviluppo dell'applicazione . . . . .	44
4.2.1 Definire un sistema di riferimento . . . . .	44
4.2.2 Definire un interfaccia . . . . .	45
4.2.3 Dettagli tecnici e ciclo di vita dell'applicazione . . . . .	46
4.2.4 Script principali . . . . .	47
4.2.5 Oggetti principali . . . . .	49
4.2.6 Utilizzo di un terzo punto . . . . .	50
<b>5 Sviluppo di una soluzione con Vuforia e VisionLib</b>	<b>51</b>
5.1 Vuforia . . . . .	51
5.1.1 Architettura e dettagli tecnici dell'applicazione . . . . .	52
5.1.2 Model Target Generator: limitazioni . . . . .	54
5.2 VisionLib . . . . .	54
5.2.1 Architettura e dettagli tecnici dell'applicazione . . . . .	55
5.2.2 VisionLib: differenze con Vuforia . . . . .	56
5.3 Risultati . . . . .	57

<b>6</b>	<b>Risultati</b>	<b>61</b>
6.1	Test in fase di progettazione . . . . .	61
6.2	Test soggettivi . . . . .	63
6.2.1	Scala di usabilità del sistema . . . . .	63
6.2.2	SSQ . . . . .	70
6.2.3	VRUSE e HAR . . . . .	72
<b>7</b>	<b>Conclusioni e sviluppi futuri</b>	<b>75</b>





# Elenco delle figure

2.1	Sensorama . . . . .	3
2.2	HMD del professore Ivan Sutherland . . . . .	4
2.3	Videoplace . . . . .	4
2.4	Continuum RV . . . . .	5
2.5	Gameplay AR Quake . . . . .	5
2.6	Spot della General Electrics . . . . .	6
2.7	Pokémon Go . . . . .	7
2.8	Esempio di Sistema di proiezione ottica . . . . .	8
2.9	Schema logico sistemi portatili . . . . .	8
2.10	Wayfair e Measure . . . . .	9
2.11	Schema see-through . . . . .	10
2.12	Google Glass . . . . .	11
2.13	Microsoft Hololens . . . . .	11
2.14	FOV sistema umano . . . . .	15
2.15	Optimal zone per Hololens . . . . .	16
2.16	Livello di sforzo dell'occhio umano . . . . .	17
2.17	Gesture tap a sinistra e bloom a destra . . . . .	17
2.18	Gesture frame . . . . .	18
2.19	Spatial mapping . . . . .	19
2.20	MRTK Logo . . . . .	21
2.21	Azure spatial anchors . . . . .	22
3.1	Trimble Catalyst . . . . .	25
3.2	Arrow Gold . . . . .	26
3.3	Esempio colorazione di un oggetto . . . . .	29
3.4	Home Page Model Target Generator . . . . .	30
3.5	Tipi di Guide View . . . . .	30
3.6	Esempio di Guide View . . . . .	31
3.7	Tipi di Database . . . . .	31
3.8	Creazione del Database . . . . .	32
3.9	Parametri Model Target su Unity . . . . .	32
3.10	Esempio di Guide View Custom . . . . .	33
3.11	Aggiunta di un oggetto 3D chiamato geometry . . . . .	33
3.12	Esempio di Rating . . . . .	34
3.13	Esempio di modifica . . . . .	35
3.14	Macchina a stati della calibrazione della camera . . . . .	36
3.15	Import del file calibration su Unity . . . . .	38
3.16	Esempio di oggetto con viste multiple . . . . .	39

3.17	Esempio di gerarchia . . . . .	40
3.18	Creazione di una License Key . . . . .	41
3.19	Differenza tra versione 2018 e 2019 . . . . .	41
3.20	Esempio di build su Unity . . . . .	42
4.1	Esempio di impianto industriale . . . . .	43
4.2	Immagini del Tutorial . . . . .	45
4.3	Icone principali . . . . .	45
4.4	Macchina a stati dell'applicazione . . . . .	46
4.5	Cursore dell'applicazione . . . . .	49
4.6	Impiego di un terzo punto . . . . .	50
5.1	Tutorial Applicazione . . . . .	51
5.2	Esempio di utilizzo . . . . .	52
5.3	Architettura dell'applicazione . . . . .	52
5.4	Colorazione Modello . . . . .	53
5.5	Licenza di utilizzo . . . . .	53
5.6	Dettagli Model Target . . . . .	54
5.7	Architettura dell'applicazione . . . . .	55
5.8	Dettagli VLCamera . . . . .	55
5.9	Risultati ottenuti . . . . .	57
5.10	Esempi di falsi positivi . . . . .	58
5.11	Esempi di falsi positivi . . . . .	59
5.12	Esempio di falso positivo . . . . .	60
6.1	Calcolo della distanza . . . . .	61
6.2	Risultati sull'accuratezza . . . . .	62
6.3	Score affermazioni 1 - 5 questionario SUS per Tablet . . . . .	65
6.4	Score affermazioni 6 - 10 questionario SUS e score finale per Tablet . . . . .	66
6.5	Grafico 1 con i risultati del test di usabilità per Tablet . . . . .	66
6.6	Grafico 2 con i risultati del test di usabilità per Tablet . . . . .	67
6.7	Score affermazioni 1 - 5 questionario SUS per Hololens . . . . .	67
6.8	Score affermazioni 6 - 10 questionario SUS e score finale per Hololens . . . . .	68
6.9	Grafico 1 con i risultati del test di usabilità per Hololens . . . . .	68
6.10	Grafico 2 con i risultati del test di usabilità per Hololens . . . . .	69
6.11	Pre test 1 SSQ . . . . .	70
6.12	Pre test 2 SSQ . . . . .	71
6.13	Test 1 SSQ . . . . .	71
6.14	Pre Test 2 SSQ . . . . .	71
6.15	Tabella con i risultati del test qualitativo per Hololens . . . . .	72
6.16	Tabella con i risultati del test qualitativo per tablet/smatphone . . . . .	73
6.17	Grafico con i risultati del test qualitativo per Hololens . . . . .	73
6.18	Grafico con i risultati del test qualitativo per tablet/smartphone . . . . .	74

# Ringraziamenti

Voglio ringraziare tutti coloro che sono stati al mio fianco durante tutto il percorso formativo. Ringrazio il mio relatore Prof. Andrea Sanna per avermi dato la possibilità di svolgere questo progetto in collaborazione con Illogic S.r.l. e per la costante presenza durante tutto il mio percorso di tesi. Il Suo aiuto è stato fondamentale nella stesura di questo documento. Un grazie a Giuseppe Donvito, direttore tecnico di Illogic, per avermi permesso di approfondire le conoscenze nell'ambito della realtà aumentata e per avermi guidato durante tutto il percorso di tirocinio. Ringrazio inoltre tutte le persone conosciute in Illogic durante lo svolgimento del progetto.

Desidero inoltre ringraziare i miei genitori per tutto quello che hanno fatto per me fino ad oggi e continueranno a fare; senza di loro sicuramente non sarei qui. Ringrazio mio padre per avermi trasmesso la voglia di svolgere con serietà e determinazione il proprio lavoro, qualunque esso sia. Ringrazio mia mamma per avermi sempre incoraggiato anche quando le cose non andavano per il verso giusto, per essere pronta a far battaglia per me e i miei fratelli e per il suo carattere molto espansivo. Ringrazio anche i miei fratelli per avermi insegnato cosa significa condividere spazi e beni comuni. Sono sicuro che rimarremo sempre uniti nonostante le scelte che decideremo di intraprendere.

Ringrazio i colleghi che ho incontrato in questo percorso e con cui ho condiviso lo sviluppo di progetti e momenti di svago. Ringrazio anche tutti gli amici e in particolare: Giuseppe I., Luca P., Fabio D., Gioele P. e Fabio Greco. Spero queste amicizie, nonostante le difficoltà, non crollino mai.

# Acronimi

<b>AR</b>	Augmented reality: arricchimento della percezione sensoriale umana mediante informazioni che non sarebbero percepibili con i cinque sensi.
<b>3D</b>	Tre dimensioni: associato ad altri termini ne indica una specializzazione. Ad esempio con modello 3D si intende una rappresentazione matematica di un oggetto tridimensionale.
<b>KARMA</b>	Knowledge based Augmented Reality for Maintenance Assistance: prototipo per supportare gli utenti nella manutenzione delle stampanti laser.
<b>HUD</b>	Head Up Display: tipo di display che permette la visualizzazione dei dati evitando che l'utente sposti lo sguardo dallo stesso.
<b>LCD</b>	Liquid Crystal Display: tipo di display che utilizza una tecnologia basata sui cristalli liquidi.
<b>GPS</b>	Global Positioning System: sistema di posizionamento e navigazione satellitare statunitense.
<b>GNSS</b>	Global Navigation Satellite System: sistema di posizionamento geo-spaziale a copertura globale basato sull'utilizzo di satelliti.
<b>HMD</b>	Head Mounted Display: schermo montato sulla testa dell'utente utilizzando un casco ad hoc
<b>VPL</b>	Virtual Programming Languages: linguaggio che evita l'impiego della sintassi scritta e consente la manipolazione grafica degli elementi per la programmazione.
<b>VR</b>	Virtual Reality: insieme dei fenomeni percettivi indotti da un'apparecchiatura cibernetica a più componenti che viene applicata a un soggetto umano.
<b>NDK</b>	Native Development Kit: software di sviluppo nativo che permette di inserire porzioni di codice nativo (C o C++).
<b>SDK</b>	Software Development Kit: insieme di strumenti per lo sviluppo e la documentazione di software.
<b>JDK</b>	Java Development Kit: insieme degli strumenti per sviluppare programmi da parte dei programmatori Java.

<b>AV</b>	Augmented Virtuality: simile alla realtà virtuale aggiungendo però alcuni oggetti reali.
<b>MR</b>	Mixed Reality:
<b>RV</b>	Reality Virtuality: simulazione di situazioni reali mediante l'utilizzo di computers e l'ausilio di interfacce appositamente sviluppate.
<b>ATAP</b>	Advanced Technology and Projects: team di sviluppo di Google
<b>OLED</b>	Organic Light Emitting Diode: tipo di diodo organico utilizzato per schermi e televisori.
<b>NASA</b>	National Aeronautics and Space Administration: Ente Nazionale statunitense per le attività Spaziali e Aeronautiche.
<b>HD</b>	High Definition: termine legato alla risoluzione degli schermi
<b>FOV</b>	Field of View: area ripresa dalla telecamera.
<b>LED</b>	Light Emitting Diode: tipo di diodo utilizzato per schermi e televisori.
<b>HPU</b>	Holographic Processing Unit: coprocessore presente in Hololens.
<b>TPM</b>	Trusted Platform Module: specifiche per la costruzione di un microchip deputato alla sicurezza informatica.
<b>USB</b>	Universal Serial Bus: interfaccia per la comunicazione seriale fra dispositivi elettronici.
<b>IMU</b>	Inertial Measurement Unit: implementa il sistema di navigazione inerziale.
<b>MP</b>	Megapixel: unità di misura multipla del pixel.
<b>Bluetooth LE</b>	Bluetooth Low Energy: Bluetooth a consumo energetico e costo notevolmente ridotto pur mantenendo un intervallo di comunicazione simile a quello classico.
<b>RAM</b>	Random Access Memory: memoria volatile di un elaboratore.
<b>UFS</b>	Universal Flash Storage: specifica per le memorie di tipologia flash.
<b>DRAM</b>	Dynamic Random Access Memory: tipo di RAM che richiede di essere periodicamente rinfrescata, pena la perdita dei dati.
<b>LPDDR</b>	Low Power Double Data Rate: tipo di memoria ad accesso casuale che consuma meno energia.
<b>UWP</b>	Universal Windows Platform: piattaforma software che aiuta a sviluppare applicazioni universali per Windows 10, Windows 10 Mobile, Xbox One e Hololens.

<b>2D</b>	Due Dimensioni: associato ad altri termini ne indica una specializzazione. Ad esempio con modello 2D si intende una rappresentazione matematica di un oggetto bidimensionale.
<b>GPU</b>	Graphics Processing Unit: processore che si contraddistingue per essere specializzato nella resa grafica d'immagini.
<b>FPS</b>	Frames per Second: numero di fotogrammi al secondo.
<b>AC</b>	Alternating Current: corrente elettrica caratterizzata dal fatto di invertire la polarità elettrica continuamente nel tempo.
<b>MRTK</b>	Mixed Reality Tool Kit: kit contenente al suo interno diversi script, component e prefab molto importanti per lo sviluppo di applicazioni Hololens.
<b>SSQ</b>	Simulator Sickness Questionnaire: test usato per misurare problemi fisici o psicofisici legati l'utilizzo di un dispositivo.
<b>SUS</b>	System Usability Scale: test usato per misurare l'usabilità di un sistema
<b>HAR</b>	Handheld Augmented Reality: test usato per misurare l'usabilità di un sistema di realtà aumentata.
<b>VRUSE</b>	Usability evaluation of virtual/synthetic environment systems: test usato per misurare l'usabilità di un sistema virtuale.

# Capitolo 1

## Introduzione

La realtà aumentata sta lentamente modificando i modelli per la manutenzione e formazione di personale in ambito industriale, l'addestramento delle reclute in ambito militare e più in generale sta cambiando la routine della società. Si pensi all'impatto e alla diffusione che ha avuto Pokémon Go: applicazione di realtà aumentata per smartphone che ha l'obiettivo di catturare i Pokémon sparsi per il mondo. Con l'avvento della "quarta rivoluzione industriale" detta anche Industry 4.0 si ha uno sforzo maggiore da parte di vari settori ad investire nell'AR. Molti ambiti sono interessati ad approfondire le potenzialità della realtà aumentata: in ambito militare sarebbe utile addestrare le reclute simulando uno scenario di guerra, in ambito medico sarebbe conveniente formare il personale o avere un supporto durante le operazioni chirurgiche, in ambito industriale si potrebbe dare un supporto agli operatori sul campo, in ambito museale[6] si potrebbe mostrare una ricostruzione virtuale delle strutture antiche o migliorare l'esperienza dei visitatori lungo i percorsi e così via. Come si vede i campi applicativi sono molteplici e, in questo lavoro di tesi, si attenderà principalmente quello industriale. I vantaggi che le aziende possono trarre dall'utilizzo di questa tecnologia sono molteplici:

- velocizzare le operazioni di manutenzione: talvolta gli operatori devono consultare una documentazione complessa e articolata e ciò rallenta il lavoro dell'operatore. Tramite la tecnologia AR è possibile rendere smart la consultazione dei documenti ed accelerare il processo di manutenzione
- migliorare i processi produttivi in termini di efficienza ed efficacia: è possibile che avendo mansioni molto complesse l'operatore rischi di sbagliare. L'AR consente di realizzare sistemi che guidino l'utente nelle varie fasi del processo industriale e nel caso l'operatore abbia dei dubbi su come procedere lo aiutano evitando che faccia scelte sbagliate.
- formazione di nuovo personale e alla specializzazione di quello già formato: i neo assunti da un'azienda il più delle volte devono essere istruiti su quello che devono fare e ciò richiede l'impiego di utenti specializzati. Si può far risparmiare tempo agli utenti con esperienza realizzando un sistema di AR che istruisca i nuovi dipendenti o aggiorni sulle ultime novità tecnologiche i dipendenti con esperienza.

Ognuno di questi vantaggi rappresenta un risparmio di risorse (in senso economico) da parte di un'azienda. Il lavoro di tesi si concentra nello sviluppo di un sistema

---

che migliori i processi produttivi dei dipendenti. Verranno messe a confronto diverse tecnologie che supportano la realtà aumentata come gli Hololens di prima generazione e smartphone. Una brevissima sintesi della storia della realtà aumentata e di alcuni suoi concetti fondamentali sono proposti nel capitolo 1. A seguire verranno mostrate quali tecnologie sono oggi utilizzate (capitolo 2). Nel capitolo 3 si presenta invece la descrizione tecnica della soluzione realizzata: Hololens. Il capitolo 4 propone una soluzione basata sul dispositivo mobile come tablet o smartphone, mette in luce pregi e difetti in relazione alla soluzione del capitolo 3. Il capitolo 5 riguarda i test di usabilità delle soluzioni realizzate. La trattazione si conclude con il capitolo 6, ovvero le conclusioni e i ringraziamenti.



# Capitolo 2

## La realtà aumentata

### 2.1 Un pò di storia

Le origini della realtà aumentata si hanno già all'inizio del 1960 con la progettazione di prototipi, ma già da prima alcuni romanzi di fantascienza iniziano a parlarne (ricordando che non era ancora stato coniato il termine realtà aumentata). Il romanzo di Frank L. Baum "The Master Key: An Electrical Fairy Tale" sembra anticipare quello che saranno gli occhialini per la realtà aumentata raccontando la storia di un ragazzo che entra in possesso di un paio di occhialini capaci di leggere i sentimenti delle persone. Uno dei primi brevetti si ha nel 1962: Sensorama del fotografo Morton Heilig. Tale dispositivo cercava di dare stimoli visivi, sonori, olfattivi e percettivi per rendere l'esperienza quanto più vicina alla realtà.



Figura 2.1: Sensorama

Più avanti nel 1966 si hanno i primi head-mounted display o HMD del professore Ivan Sutherland. Era in grado di mostrare immagini 3D che si sovrappongono a

scene reali. Questo dispositivo, ben lontano da quelli attuali, aveva un peso non sostenibile da una persona tanto che venne soprannominato spada di Damocle e richiede per questo motivo un braccio meccanico che lo sostenesse. Un altro esempio

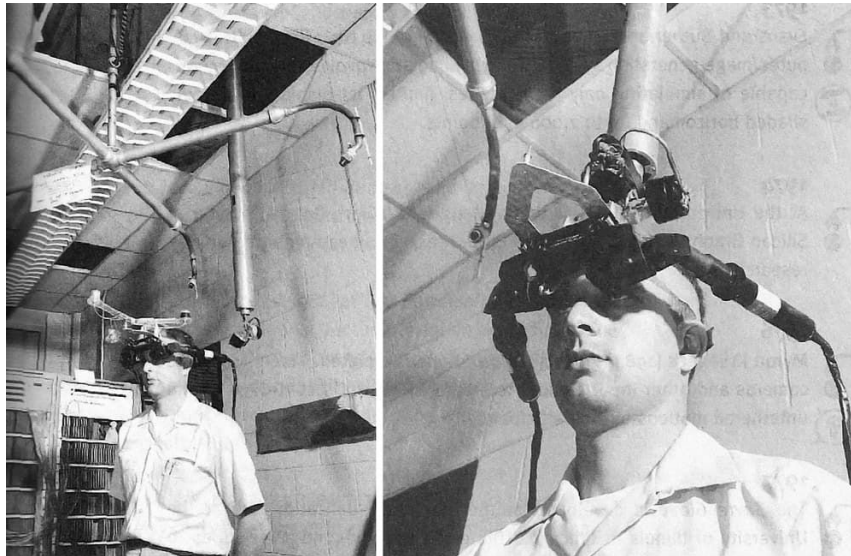


Figura 2.2: HMD del professore Ivan Sutherland

si ha nel 1975 da parte del Dott. Myron Krueger e il suo laboratorio di realtà artificiale chiamato Videoplace. Tale esperienza è molto importante poiché libera l'utente da ogni ingombro: stavolta c'è una camera che si occupa di registrare i movimenti dell'utente, un computer che si occupa di analizzare tali dati e un video proiettore che mostra come l'utente interagisce con gli oggetti virtuali. Ma solo

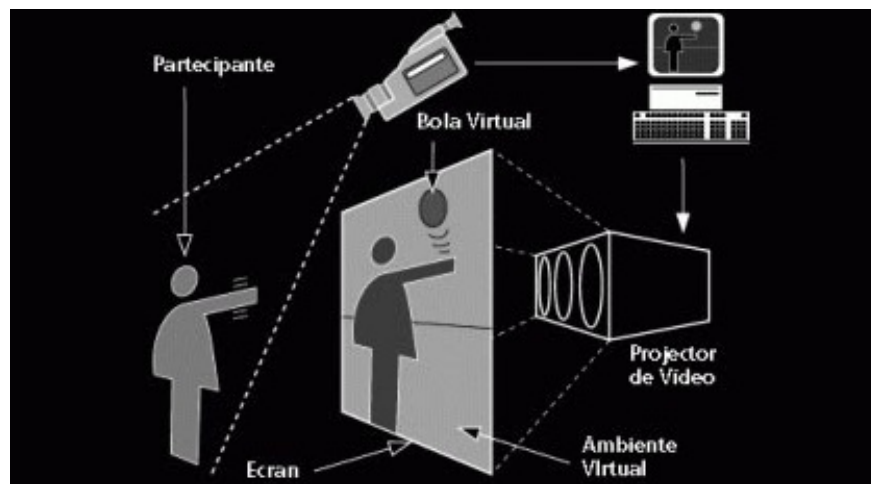


Figura 2.3: Videoplace

nel 1989 viene fondata la VPL Research, la prima azienda che si occupa di realtà virtuale, e subito dopo un anno dalla fondazione della stessa viene coniato il termine realtà aumentata (Thomas P. Caudell). Da questo momento in poi l'ambito medico e militare si interessa a questa tecnologia: verranno sviluppati dei visori Head-up display (HUD) per gli aviatori e sistemi che permettono ai medici di osservare un feto direttamente all'interno delle pazienti incinte. Bisogna parlare di alcuni concetti importanti introdotti nel 1994 da Paul Milgram e Fumio Kishino (Mixed Reality).

Si è già accennato a cosa si intende per AR e VR ma non si è parlato di augmented virtuality (AV) e mixed reality(MR). Per poter capire questi due concetti si faccia riferimento allo schema sottostante. Come mostrato in figura 2.4 realtà virtuale e mondo reale stanno agli opposti e tra i due stanno AR e AV che rientrano nella realtà mista. L'obiettivo dell'AR è quello di aggiungere oggetti virtuali alla realtà ma senza sostituirsi ad essa mentre l'AV è molto più vicina alla realtà virtuale mantenendo moltissimi oggetti virtuali e pochi oggetti reali. Un'applicazione che aggiunge un avatar virtuale in una stanza reale può essere considerata un app AR mentre un videogioco in cui tutta la scena è virtuale e l'unica cosa reale è la mano del protagonista e poche altre cose può essere considerata un app AV. Tutto quello che si trova in mezzo tra il reale e il virtuale puro è la realtà mista (MR): che si può vedere con il passaggio da uno "stato" verso l'altro. Ma la svolta si ha nel 1999: il

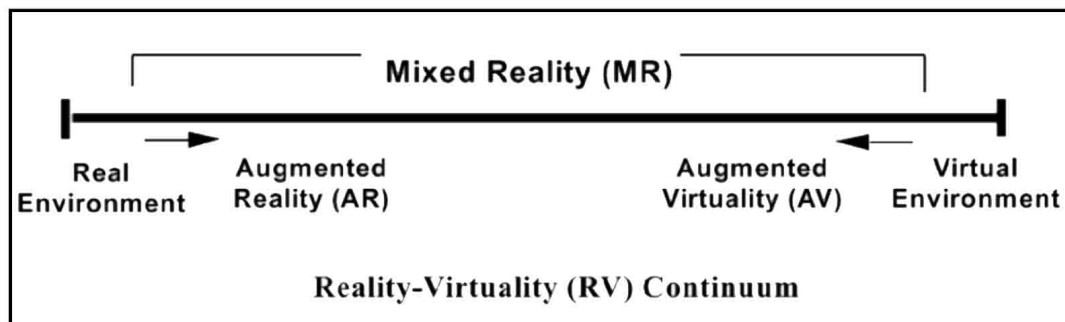


Figura 2.4: Continuum RV

professore Hirokazu Kato del Nara Institute of Science and Technology sviluppa il software ARToolKit (open source). Ciò porta molti programmatori ad avvicinarsi alla realtà aumentata e a sviluppare nuove soluzioni. Nel 2000 si ha il primo esempio con AR Quake di Bruce H. Thomas.



Figura 2.5: Gameplay AR Quake

Il sistema realizzato comprendeva diversi oggetti: un computer (tenuto in uno zaino), un giroscopio, un sensore, un GPS e un Head-Mounted display. Tratto dall'omonimo sparatutto il sistema consente di ripercorrere i labirinti del gioco originale nel mondo reale. Nel 2009 la General Electric, multinazionale statunitense, realizza il suo nuovo spot pubblicitario di realtà aumentata. Di fatto usando un marker e inquadrandolo con la camera del proprio computer è possibile “ricostruire” il nuovo impianto per la produzione di energia elettrica ecologica. L'utente poteva anche far girare le pale olografiche soffiando sul microfono.



Figura 2.6: Spot della General Electric

Un altro esempio viene da Ikea che nel 2014 propone il catalogo online di realtà aumentata: l'applicazione per smartphone si chiama Ikea Place e consente di arredare la propria casa aggiungendo gli ologrammi nei punti della stanza indicati dall'utente. Anche in ambito videoludico non mancano gli esempi: agli inizi di luglio 2016 viene pubblicato Pokémon Go, applicazione di realtà aumentata per smartphone che consente di andare a catturare Pokémon in giro per il mondo. Si noti che la realtà aumentata deve parte della sua diffusione agli smartphone: questi dispositivi ormai alla portata di tutti sono in grado di supportare videocamere digitali, sistemi di localizzazione GPS, bussole, connessione alla Rete, quindi riescono a far girare applicazioni di realtà aumentata senza grossi problemi.

## 2.2 I dispositivi della realtà aumentata

Si parlerà ora degli strumenti fisici che si occupano di fare AR. Come prima cosa bisogna dividerli in base alle caratteristiche peculiari e poi si descriveranno i dettagli tecnici. Le grandi famiglie sono le seguenti:



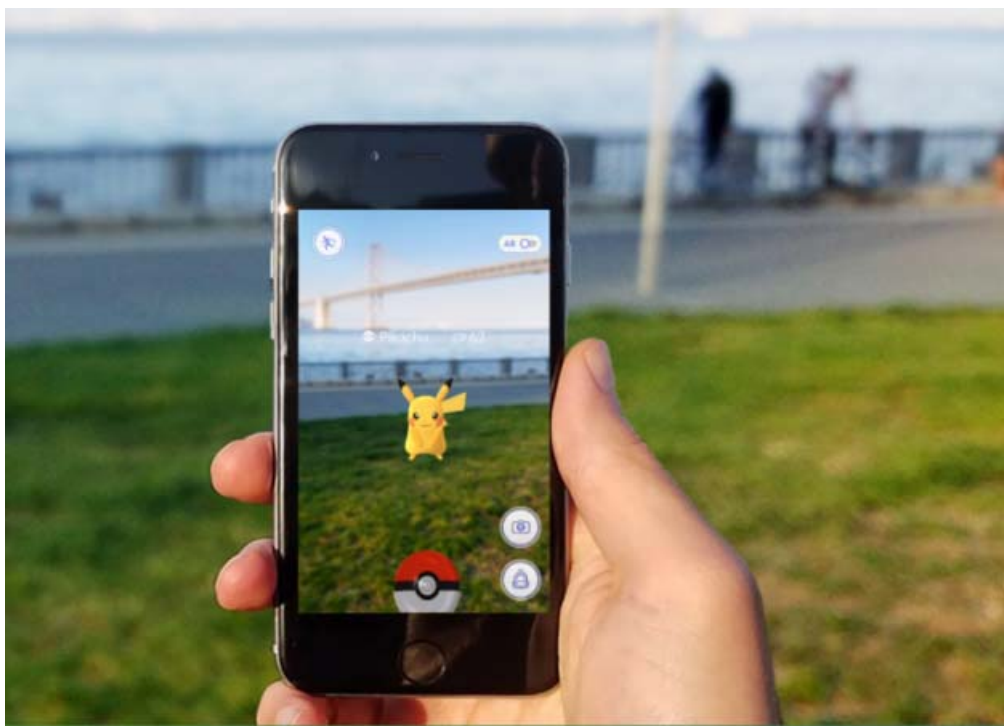


Figura 2.7: Pokémon Go

- sistemi di proiezione ottica e monitor
- dispositivi portatili
- sistemi di visualizzazione indossabili

Nei prossimi paragrafi verranno descritte soprattutto le ultime due categorie poiché riguardano da vicino il lavoro di tesi.

### 2.2.1 Sistemi di proiezione ottica e monitor

La prima categoria riguarda sia i monitor e webcam che i proiettori. Il caso citato prima dello spot della General Electrics rientra in questa categoria: si ha una webcam per acquisire la scena reale, un calcolatore che analizza i dati e un monitor che mostra degli ologrammi. Nel secondo caso sarà necessario l'utilizzo di un proiettore che si occuperà di aggiungere i modelli 3D nell'ambiente reale: questa tecnologia viene utilizzata principalmente in ambienti chiusi, controllati e con adeguata illuminazione. Il sistema di proiezione viene molto utilizzato dalle aziende automobilistiche. Come si intuisce dalla figura sottostante lo scopo è visualizzare gli ologrammi (componenti interni dell'automobile) direttamente sulla superficie esterna di un veicolo. Questo può essere molto utile quando si devono eseguire delle riparazioni e si vuole sapere cosa andare a cercare senza dover smontare tutto. Purtroppo tale tecnologia soffre l'elevata intensità luminosa (in ambienti aperti gli ologrammi proiettati rischiano di non vedersi) ma soprattutto non consente all'utente di potersi muovere liberamente nella scena.



Figura 2.8: Esempio di Sistema di proiezione ottica

### 2.2.2 Dispositivi portatili

Tale categoria, in cui rientrano cellulari e tablet, è la più impiegata. Oggi giorno infatti un po' tutti hanno uno smartphone dotato di fotocamera e schermo per l'acquisizione e la riproduzione dei contenuti virtuali. Il fatto di non avere costi elevati, di essere piccolo e facilmente trasportabile, di avere un'enorme versatilità (chiamare, collegarsi a internet, inviare mail e messaggi etc) rende questa categoria molto più fruibile dalla stragrande parte della popolazione. Come si vedrà più avanti le altre tecnologie hanno un costo più elevato, sono nate per fare realtà aumentata (non puoi chiamare qualcuno o inviare un messaggio), richiedono una fase di addestramento per imparare i dispositivi di input (le gesture per gli Hololens ad esempio) che in un dispositivo mobile non è richiesta, sono più ingombranti e talvolta possono provocare danni psicofisici. Nella figura sottostante si mostra uno schema concettuale ad alto livello dei dispositivi hand-held.

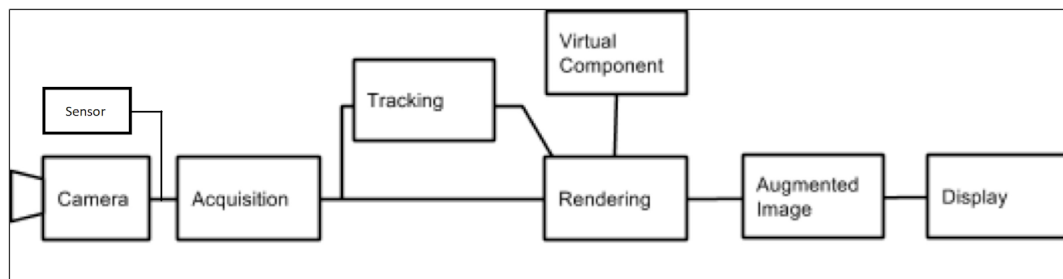


Figura 2.9: Schema logico sistemi portatili

Il sistema riesce a collocare il dispositivo nello spazio attraverso le informazioni della camera e del sensore; avendo una mappa precisa dell'ambiente circostante e la posizione del device all'interno dello stesso sarà possibile collocare in maniera precisa anche altri oggetti nella scena. Si noti che la qualità del tracking dipenderà fortemente dalla camera: maggiore è la qualità della fotocamera, minori saranno le distorsioni introdotte e quindi si otterrà un tracking maggiore. Questo aspetto si vedrà anche nei sistemi di visualizzazione indossabili. Le prestazioni degli Hand-held device però, in media (considerando tablet o smartphone non di ultima generazione), sono inferiori a quelle dei sistemi di visualizzazione indossabili. Inoltre

per visualizzare il contenuto 3D l'utente deve avere le mani impegnate a inquadrare correttamente l'ambiente e lo sguardo sempre rivolto verso lo schermo: ciò non è vero per quanto riguarda la terza categoria: sia mani che punto di vista sono liberi di muoversi liberamente. Si mostra un esempio concreto di device appartenente a questa categoria parlando di funzionalità offerte e criticità. Nel giugno del 2014 si ha la prima release di project Tango (sviluppato da Advanced Technology and Projects (ATAP) team di Google) che sfruttando la computer vision abilità il dispositivo a calcolare la sua posizione nello spazio circostante. La novità sta nel non utilizzare una tecnologia basata sul GPS o su segnali esterni consentendo di avere una buona esperienza in ambienti chiusi dove i segnali GPS hanno difficoltà a passare. La piattaforma ebbe successo e google nel 2016 annuncia la collaborazione Lenovo per rilasciare uno smartphone con supporto per Tango e rivolto ai consumatori. Più nello specifico venivano offerte le seguenti funzionalità:

- Area learning: consente di analizzare la stanza e utilizzare tali informazioni per rendere possibile l'interazione tra oggetti reali e virtuali.
- Motion tracking: utilizzando vari dati raccolti dalla camera, accelerometro e giroscopio è possibile individuare i movimenti del dispositivo nello spazio e conoscere, frame per frame, la posizione e l'orientamento dell'utente e quindi del dispositivo.
- Depth perception: valutando dimensioni, distanze e superfici dell'oggetto cerca di rendere l'esperienza dell'utente quanto più realistica possibile. Utilizzando queste funzioni possono essere sviluppate diverse app come Wayfair, che consente di arredare la propria stanza o Measure che consente con pochi passi di calcolare la dimensione in metri degli oggetti reali.



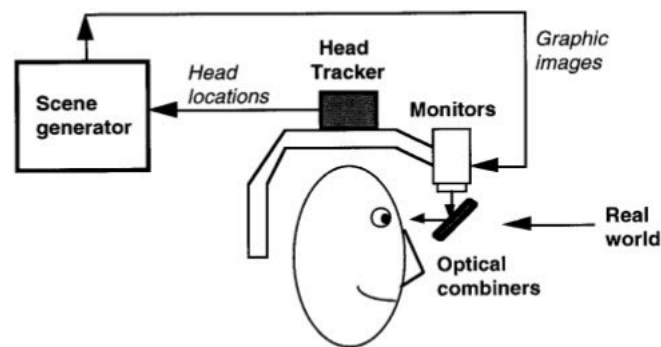
Figura 2.10: Wayfair e Measure

Successivamente però lo sviluppo di tango venne bloccato e Google si concentrò sul progetto ARCore, pubblicato il 2018 e tuttora usato dagli sviluppatori. La novità di quest'ultimo sta nella compatibilità nettamente maggiore con diversi dispositivi rispetto project Tango, che era stato pensato per tablet e smartphone Lenovo e Asus. Un altro esempio lo si ha con Vuforia e VisionLib, a cui sono dedicati i capitoli successivi.

### 2.2.3 Sistemi di visualizzazione indossabili

Detti anche head mounted display (HMD) poiché indossati come degli occhiali: essi consentono di unire i contenuti virtuali al mondo reale. Nella figura sottostante si mostra uno schema concettuale ad alto livello.

#### Optical see-through HMD



15

Figura 2.11: Schema see-through

I contenuti reali, frame per frame, vengono analizzati per far in modo che gli oggetti 3D presenti nell'ambiente siano consistenti o più in generale non si spostino dal punto dove sono stati posizionati. Più nello specifico le camere presenti nel dispositivo scansionano l'ambiente e mandano queste informazioni a un elaboratore. Una volta elaborate le informazioni provenienti dall'ambiente reale e quelle relative alla scena virtuale si può passare a renderizzare gli ologrammi nella scena. La lente è trasparente per consentire all'utente di vedere l'ambiente reale e ologrammi contemporaneamente. Tra i tipi di display vanno ricordati quelli OLED e LCD entrambi composti da una serie di schermi sovrapposti. I display OLED sfruttano la corrente elettrica e diodo organico a emissione di luce per mostrare nel piccolo schermo davanti gli occhi dell'utente i contenuti olografici. Inizialmente si aveva un anodo, un catodo e una sostanza organica in grado di emettere luce luminosa quando veniva applicata una tensione ai due elettrodi. Oggi di fatto si hanno diversi strati ma il concetto di fondo è rimasto il medesimo: si ha uno strato trasparente protettivo, un altro che funge da anodo, più strati per i materiali organici in grado di emettere luce e uno strato che funge da catodo. Ognuno di questi strati deve essere trasparente. I display LCD utilizzano la retroilluminazione. Il principio è simile a quello visto prima ma la differenza sostanziale è la seguente: i cristalli liquidi fanno da filtro facendo passare solamente la componente cromatica desiderata non emettono luce. In entrambi i casi le immagini che verranno mostrate a schermo saranno diverse per dare l'illusione stereoscopica ai due occhi. Per il resto del paragrafo si parlerà dei device appartenenti a questa categoria focalizzando l'attenzione su Microsoft HoloLens, utilizzato per sviluppare la soluzione proposta nel prossimo capitolo, e del suo successore HoloLens di seconda generazione. Tra i primi dispositivi creati, degni



di nota, quello di cui si è parlato maggiormente nel corso degli ultimi anni è senza alcun dubbio Google Glass. Progettato dal team X Lab di Google, è essenzialmente un occhiale da indossare, che visualizza informazioni e dati all'interno del campo visivo, mediante l'impiego di un piccolo display posizionato sopra l'occhio dell'utente. La profonda integrazione con i dispositivi Android è certamente il punto di forza principale degli occhiali, perché comunicando con smartphone e tablet sono in grado di mostrare all'utente notifiche, messaggi, indicazioni stradali, aggiornamenti provenienti dai social network e molto altro ancora, senza obbligarlo a toccare il touchscreen.



Figura 2.12: Google Glass

Nel 2016 in occasione dell'evento dedicato alla piattaforma Windows 10, Microsoft ha sorpreso tutti annunciando Microsoft HoloLens. In questo caso l'approccio alla fruizione e all'interazione con i contenuti è differente, poiché si fa ricorso all'impiego di ologrammi, ovvero immagini tridimensionali visualizzate attraverso uno speciale display nell'ambiente circostante, tenendo conto della giusta prospettiva e del corretto punto di osservazione. Il visore, basato sulla tecnologia Windows Holographic, è stato progettato in collaborazione con la NASA.



Figura 2.13: Microsoft Hololens

Come si vede dall'immagine il dispositivo è più invasivo e pesante di Google Glass ma offre molte più funzionalità. Si prosegue vedendo più nello specifico i dettagli hardware di HoloLens dividendoli in 7 gruppi: display, sensori, input/output e connettività, potenza, processore, memoria e peso.

Display:

- Lenti olografiche see-through
- 2 HD 16:9 producono 2.3M di punti luce totali
- >2.5k radianti (punti luce per radianti)
- FOV di 30° orizzontali e 17.5° verticali

Sensori:

- 1 unità di misure inerziali (IMU)
- 4 camere l'analisi spaziale
- 1 camera di profondità
- 1 2MP photo / HD video camera
- 4 microfoni
- 1 sensore di luce ambientale

Input/output e connettività:

- Speakers integrato
- Audio 3.5mm jack
- Volume up/down
- Luminosità up/down
- Power button
- Livello della batteria a LED
- Wi-Fi 802.11ac
- Micro USB 2.0
- Bluetooth 4.1 LE

Potenza:

- 2-3 ore di utilizzo
- 2 settimane in modalità standby
- Tutte le funzionalità garantite sotto carica

Processore:

- Architettura Intel 32-bit con supporto TPM 2.0
- Microsoft Holographic Processing Unit (HPU 1.0)

Memoria:

- 64 GB Flash
- 2 GB RAM

Peso:

- 579 grammi

Come si nota dalle specifiche il dispositivo ha un peso non indifferente ciò potrebbe risultare fastidioso dopo un paio d'ore di utilizzo. Altro aspetto da non sottovalutare è il FOV piuttosto basso rispetto altri dispositivi come Meta 2 che verrà visto in seguito. Per il resto la qualità e le funzionalità offerte rendono tale dispositivo molto versatile. Grazie il collegamento WIFI e un'indipendenza dall'alimentazione l'operatore che utilizza Hololens potrà spostarsi liberamente all'interno dell'area di copertura del WIFI. Tutte le camere invece servono ad offrire funzionalità come lo spatial mapping e gli speaker servono per i comandi vocali. La resa grafica viene garantita anche da componenti custom di Microsoft come HPU. Si vedrà, nel paragrafo successivo, nello specifico tutte le funzionalità e le relative criticità. Un suo diretto rivale è stato sicuramente Meta 2, presentato dall'azienda Meta e disponibile dal 2016. Questo dispositivo presenta diversi vantaggi rispetto gli Hololens:

- FOV di circa 90° (circa il triplo di Hololens)
- Possibilità di indossare occhiali da vista
- Costo di circa 1000 dollari
- Flessibilità nella scelta delle gesture

Ci sono anche svantaggi:

- Alimentazione costante
- Prestazioni leggermente inferiori

Si fa notare che nonostante il numero maggiore di vantaggi Meta due e Hololens restano comunque paragonabili: la necessità di avere alimentazione costante riduce di molto lo spettro di possibilità applicative. Inoltre avere alimentazione costante vuol dire avere un filo attaccato al dispositivo e ciò può risultare ingombrante in base ai movimenti effettuati dall'utente. Inoltre il costo maggiore di Hololens si può giustificare dall'hardware in più e dalla resa leggermente superiore. Il vero problema è sicuramente il FOV: determinate applicazioni se richiedono un visione globale dell'area di lavoro hanno necessariamente bisogno di un FOV ampio. Si fa notare infine che gli Hololens non possono essere usati insieme agli occhiali da vista e richiedono per questi soggetti l'uso delle lentine; Meta 2 a differenza di Hololens che sfrutta un set fisso di gesture utilizza gesture personalizzabili e quindi più vicine a quelle usate tutti i giorni. Per finire è importante ricordare che Microsoft ha annunciato Hololens di seconda generazione abbreviato Hololens 2. Anche in questo caso, come fatto per il suo predecessore, vengono elencate le specifiche hardware per capire quali sono stati i miglioramenti.

Display:

- Lenti olografiche see-through
- 2k 3:2 light engines
- >2.5k radianti (punti luce per radianti)
- Display ottimizzato per la posizione 3D degli occhi
- FOV di 52°

Sensori:

- 4 camere di luci ambientali
- 2 camere a infrarossi (IR)
- 1-MP Time-of-Flight sensori di profondità
- Accelerometri, giroscopi, magnetometri
- Camera da 8-MP , 1080p30 video

Audio:

- 5 canali
- Built-in spatial sound

Memoria e connettività:

- Microsoft Holographic Processing Unit (HPU 2.0)
- 4-GB LPDDR4x system DRAM
- 64-GB UFS 2.1
- 802.11ac 2x2
- Bluetooth 5.0
- USB Type-C

Potenza:

- 2-3 ore di utilizzo
- 2 settimane in modalità standby
- Tutte le funzionalità garantite sotto carica

Peso:

- 566 grammi

Si nota un notevole miglioramento computazionale del dispositivo: si passa da 2 giga di RAM e 4, da 2 megapixel di camera a 8, da un HPU di prima generazione a una di seconda generazione. Anche il numero di microfoni è aumentato da 4 a 5. La cosa più importante resta sicuramente l'allargamento del FOV. I campi applicativi e di utilizzo di queste tecnologie sono pressochè infiniti, basti pensare ad applicazioni mediche per il supporto del personale medico durante le operazioni chirurgiche o la visualizzazione di parametri relativi al paziente, applicazioni militari per la formazione del personale che simulano un campo di battaglia riducendo così i costi legati all'utilizzo di munizioni o mezzi militari oppure applicazioni industriali per la visualizzazione di parametri operativi in tempo reale, la supervisione di processi industriali e il supporto nella manutenzione.

### 2.2.4 Microsoft HoloLens

Si parla ora più nello specifico del dispositivo usato nel capitolo successivo: Microsoft HoloLens. Si è discusso di realtà mista e di come questa permette di fare interagire gli attori virtuali all'ambiente reale. Un esempio potrebbe essere la simulazione fisica di una pallina di gomma: questa quando collide con un muro o il pavimento deve rimbalzare proprio come farebbe una pallina reale. HoloLens si vuole spostare più su questo campo che su quello della realtà aumentata che si limita a sovrapporre oggetti virtuali nel mondo reale. Si sono già viste le specifiche tecniche nel precedente paragrafo. Si noti che le applicazioni sviluppate per HoloLens si basano su Universal Windows Platform (UWP): questa architettura consente la creazione di applicazioni universali ovvero eseguibili su gran parte dei sistemi creati da Microsoft. Gli HoloLens sono in grado di far girare, oltre le applicazioni olografiche per la realtà mista, anche applicazioni 2D come Skype oppure app per la visualizzazione di foto. Un aspetto molto importante, come detto più volte, è il FOV. Per capirne l'importanza bisogna riprendere il sistema visivo umano.

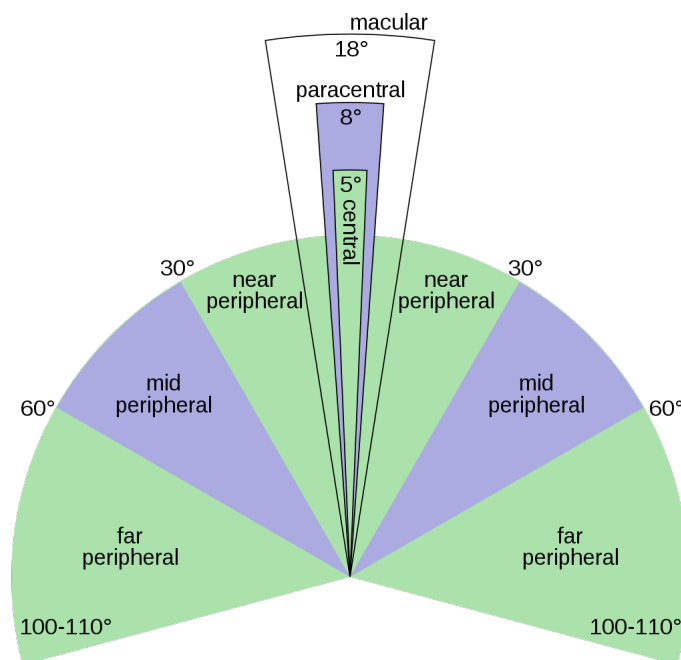


Figura 2.14: FOV sistema umano

Il campo visivo dell'uomo è di 200-220° in orizzontale e 130-135° verticali: come si vede dalla figura però non tutte le aree coperte dal campo visivo sono uguali ma più ci si allontana dall'asse centrale più l'occhio ha "difficoltà" nel focalizzare gli oggetti; la zona di comfort è di 150°. Da questi concetti si possono ricavare due informazioni importanti: la prima è che un dispositivo che offre un FOV maggiore di 220° sviluppa funzionalità inutili per l'occhio umano e la seconda è che se il dispositivo ha un FOV molto piccolo allontana l'utente da un'esperienza immersiva. Purtroppo si fa notare come si è ancora lontani dal raggiungimento di un valore apprezzabile da parte degli HoloLens. Oltre al campo visivo bisogna parlare di convergenza e accomodazione. La convergenza si riferisce al movimento degli occhi verso un oggetto che si intende guardare mentre l'accomodazione si riferisce al processo all'interno dell'occhio per mettere a fuoco oggetti a distanza finita: questi due aspetti sono molto importanti e collegati tra loro. Nei dispositivi di questo tipo solitamente quello che accade è un disallineamento innaturale tra convergenza e accomodazione che se eccessivo può provocare fastidi: nello specifico gli occhi dell'utente convergeranno in base alla distanza dell'ologramma nello spazio ma avranno un'accomodazione in base alla distanza focale del visore. In particolare gli HoloLens sono stati calibrati per avere ologrammi a distanza di 2 metri. Questo vuol dire che l'utente vedrà in maniera nitida oggetti che si trovano in un intorno di questa distanza: nella figura viene mostrata questa area detta Optimal zone.

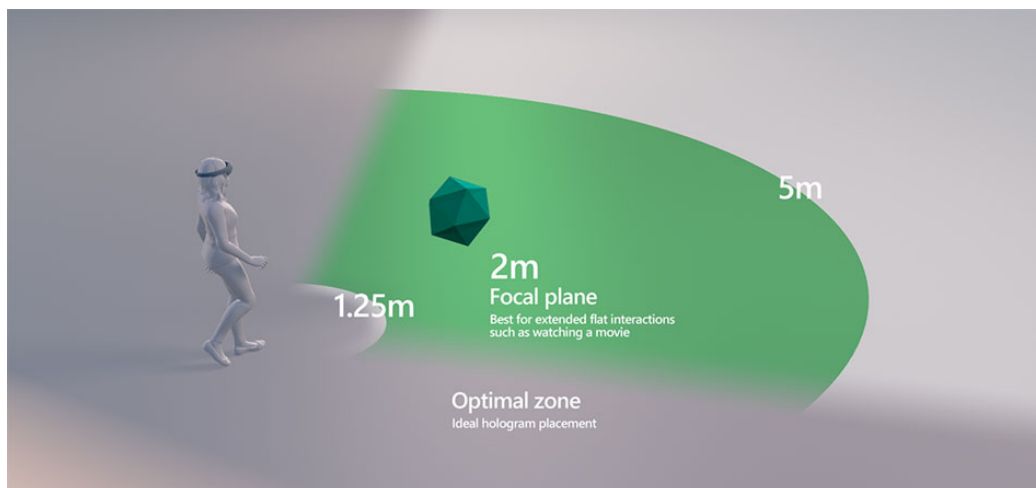


Figura 2.15: Optimal zone per HoloLens

Si noti infine che l'occhio non avverte in maniera lineare il fastidio ma segue determinate curve: mostrate in figura. Come si nota l'occhio soffre molto distanze molto basse (da 0 a 40 cm) e meno quelle grandi (da 5 - 10 m). Si fa notare che a distanze molto prossime (da 0 a 15 cm) il dispositivo non è più in grado di mostrare ologrammi ed eseguire lo spatial mapping. Altro aspetto molto importante sono gli spostamenti cioè l'avvicinamento e l'allontanamento di ologrammi rispetto la posizione dell'utente: ciò accentua la disparità tra accomodazione e la convergenza. Oggetti a distanza di un metro circa possono essere sopportati dall'occhio se sono statici. Anche il movimento delle braccia, del collo e degli occhi non deve essere costantemente ripetuto e non deve superare certe soglie:

- sguardo non oltre 10° al di sopra dell'orizzonte (spostamento verticale)

- sguardo non oltre  $60^\circ$  al di sotto dell'orizzonte (spostamento verticale)
- rotazioni del collo non oltre  $45^\circ$  rispetto al centro (spostamento orizzontale)
- evitare di avere la mano alzata per l'intero utilizzo dell'applicazione

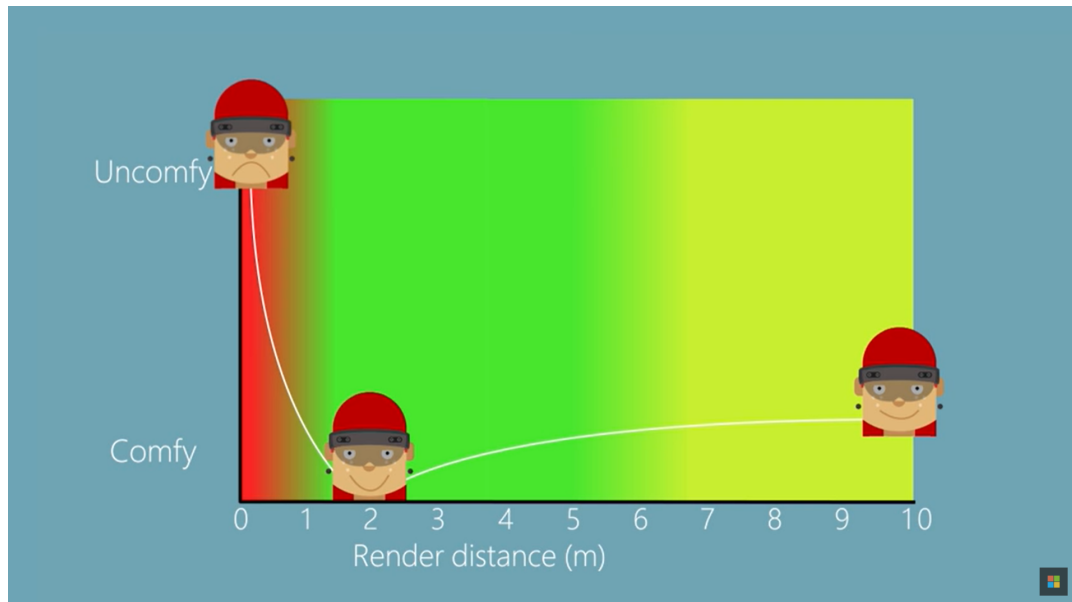


Figura 2.16: Livello di sforzo dell'occhio umano

Si parla adesso delle modalità di input di Hololens. Come detto in precedenza si possono effettuare gesture o vocal input. Le gesture sono principalmente air tap e il bloom mostrati in figura.

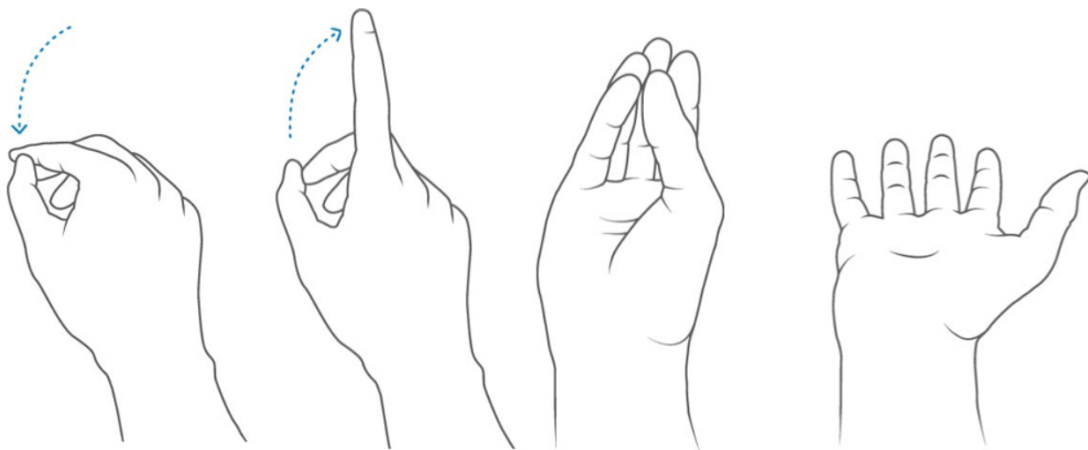


Figura 2.17: Gesture tap a sinistra e bloom a destra

A sinistra la gesture air tap che ha le stesse funzionalità del click del mouse e può essere usata per trascinare o ridimensionare oggetti e a destra il bloom che serve ad aprire il menu principale. Hololens è ottimizzato per riconoscere questi movimenti della mano: c'è un'area entro cui devono avvenire queste gesture che si trova davanti l'utente. Queste funzionalità possono essere sostituite da input vocali: ad esempio l'air tap può essere sostituito dal comando vocale select.

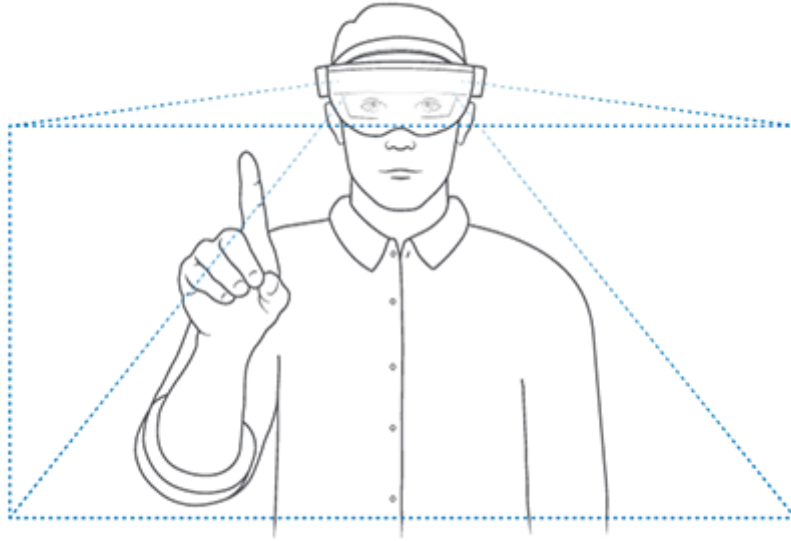


Figura 2.18: Gesture frame

### Interpretazione dello spazio

Si analizza adesso un aspetto molto vicino al tema centrale del lavoro di tesi: come Hololens gestisce la posizione degli oggetti nello spazio. L'idea alla base è quella di trovare un modo per allineare coordinate del mondo virtuale e coordinate del mondo fisico (reale) cioè trovare un sistema di coordinate spaziali. Il sistema di riferimento a cui si fa ricorso è quello cartesiano con 3 assi (X,Y,Z) perpendicolari tra di loro e aventi la convenzione destrorsa o mancina: entrambe hanno l'asse y come vettore perpendicolare al pavimento dove ci si trova (orientato verso l'alto) e l'asse x rivolto alla destra dell'utente (orientato verso la destra dell'utente). L'unico asse che cambia è l'asse z: nel caso destrorso ha orientamento positivo davanti l'utente, viceversa nel caso mancino (verso positivo dietro l'utente). Tre concetti fondamentali sono lo Stationary frame of reference, Attached Frame of reference e Spatial anchors. Il primo viene utilizzato per mantenere fissa la posizione degli ologrammi nell'ambiente reale. Di fatto si può pensare questo stationary frame of reference come l'origine di riferimento in un sistema di coordinate cartesiano: la posizione di tutti gli elementi verrà dunque determinata in base alla distanza con l'origine (stationary frame of reference). Come detto prima questo sistema di coordinate è strettamente legato al mondo reale tanto che non è detto l'origine rimanga fissa nel corso dell'esecuzione dell'applicazione; in particolare quando l'utente percorre distanze maggiori di 5 metri Hololens acquisisce nuove informazioni e tenderà a spostare tale origine causando lo spostamento di alcuni oggetti all'interno della scena. Attached Frame of reference si riferisce all'utente e serve principalmente per mantenere ologrammi a una distanza fissa dall'utente. Da non confondere questa tipologia con quella descritta prima: quando si è parlato di stationary frame of reference si fa riferimento a oggetti che vengono posizionati nello spazio e devono rimanere fermi, adesso si parla di oggetti che devono seguire l'utente e che devono trovarsi a una certa distanza da quest'ultimo. Il problema legato all'area relativamente piccola e allo spostamento degli ologrammi viene risolto utilizzando le spatial anchor cioè sistemi di riferimento locali che assicurano agli ologrammi vicini stabilità.



### Spatial mapping

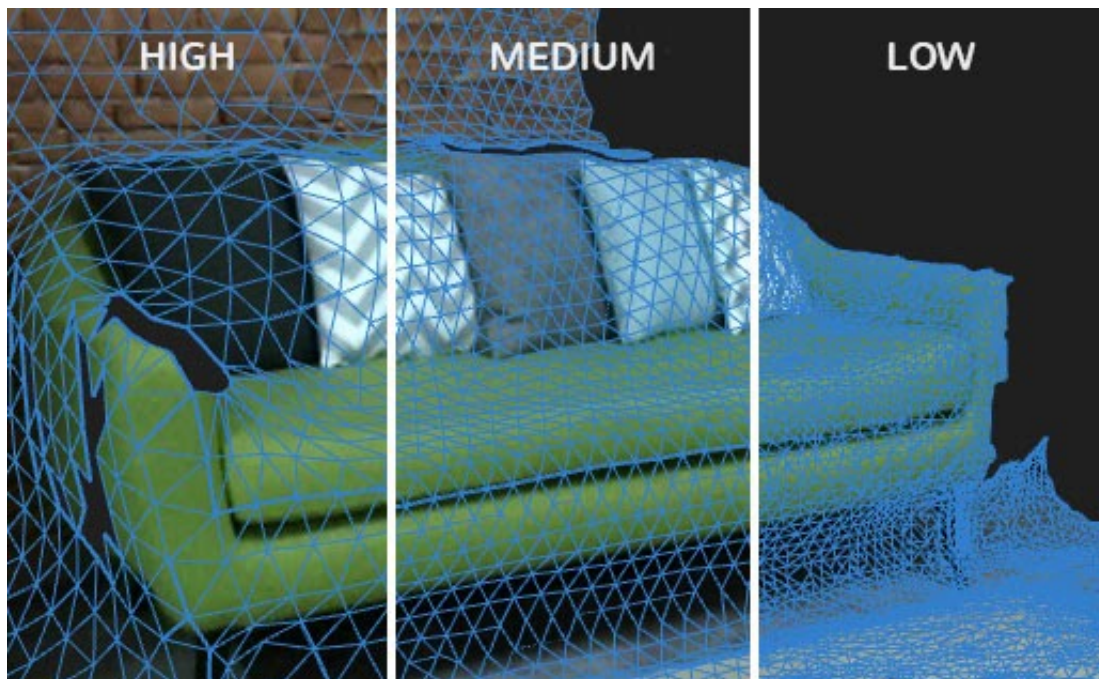


Figura 2.19: Spatial mapping

Lo spatial map è un processo della realtà aumentata che tenta di creare una mesh dell'ambiente reale: questa apparirà come una rete di triangoli con differente dimensione. Normalmente l'ambiente reale non è statico ma cambia in continuazione: movimento di esseri umani o la presenza di oggetti in movimento. Le funzionalità che si ottengono dallo spatial mapping sono molteplici:

- posizionamento ovvero la possibilità di posizionare oggetti virtuali su oggetti reali. Semplicità e precisione sono le caratteristiche molto importanti: si vuole che l'utente riesca facilmente a eseguire questa funzionalità e allo stesso tempo l'oggetto venga messo dove l'utente si aspetta che sia. Una sedia olografica che sprofonda nel pavimento o un libro virtuale non parallelo al tavolo su cui è poggiato rendono l'esperienza lontana dalla realtà.
- occlusione cioè la copertura di oggetti lontani da parte di altri oggetti più vicini. Se un ologramma si trova dietro un muro reale non deve essere visto dall'utente: questo aiuta l'utente a capire la distanza che intercorre tra lui e l'ologramma. Talvolta però ci sono oggetti che devono essere sempre visibili, in questo caso si potrebbe pensare di cambiare il loro colore per far capire la distanza all'utente.
- fisica: dare la possibilità ad oggetti virtuali di avere il comportamento di un oggetto reale. Un esempio è quello di una pallina di gomma che rimbalza sul pavimento: quanto più sarà semplice l'area in cui la pallina rimbalza quanto più sarà realistica. Aree con mesh mancanti o con buchi rischiano di rendere la simulazione fisica disastrosa.
- navigazione: dare la possibilità ad avatar di muoversi all'interno dell'ambiente reale. Le sfide sono molteplici in questo campo: il rischio che l'avatar si blocchi

quando non è necessario, attraversarsi gli oggetti reali (come una porta chiusa), non riesca a calcolare un percorso valido. La funzione in Unity NavMesh non può essere integrata poiché i dati spaziali non sono noti a priori.

- visualizzazione: mostrare all'utente il volume delle superfici reali. Normalmente questa funzionalità rende la scena più confusionaria e aggiunge lavoro in più al dispositivo. Però può essere d'aiuto all'utente quando si vuole posizionare un ologramma su un muro.

Si capisce che questa funzionalità richiede un impegno enorme di risorse per questo motivo vengono adottate tecniche per minimizzare i costi. Sia a livello di calcolo per valutare i cambiamenti dell'ambiente che a livello di memoria impiegata per salvare le mesh questa funzionalità risulta problematica e le soluzioni cercano un trade off tra questi due aspetti. L'idea alla base è quella di indovinare quali mesh serviranno nell'immediato futuro e quali invece posso buttare via poiché non servono: se l'utente muove un attimo la testa per poi tornare alla posizione originale quello che si aspetta è che le mesh di partenza non vengano buttate ma vengano memorizzate nella cache e riusate non appena servono. Ciò mira principalmente a ridurre i movimenti ridondanti dell'utente all'interno della stanza. Si sfruttano due osservatori: uno "pigro" che copre una grande area e un'altro che copre un'area più piccola, interna all'osservatore grande. Il primo serve a mantenere traccia di tutti quegli elementi che si presuppone appariranno nuovamente nella scena e il secondo invece segue costantemente i movimenti dell'utente. Più ad alto livello si può parlare di tempi di risposta e risoluzione dello spatial mapping: il dispositivo si troverà ad aggiornare molto frequentemente la mesh che ricopre l'ambiente reale; un ambiente molto complesso (con molti oggetti con forme particolari) può diventare problematico. Tanto più è complesso l'ambiente tanto più appesantisce il dispositivo. Bisogna dunque trovare un trade-off tra tempi di risposta, legate ai calcoli computazionali, e risoluzione dello spatial mapping. In base all'applicazione che si intende realizzare si potrà preferire avere tempi di risposta molto bassi e una risoluzione non perfetta (ad es nei videogiochi) o viceversa (ad esempio applicazioni in cui bisogna avere precisione molto elevata). Un modo per ottenere tempi di risposta molto alti potrebbe essere aumentare la dimensione dei triangoli quindi diminuita la densità dei triangoli in una determinata area: ciò però riduce la qualità delle superfici cioè approssima forme complesse degli oggetti. Si può anche ridurre la frequenza di aggiornamento delle mesh se ad l'ambiente è statico o non cambia velocemente. Un'altra strategia è quella di determinare un ordine di rendering ovvero iniziare dalle mesh molto vicine per poi finire con quelle più lontane: questa strategia è molto efficiente nei casi in cui le mesh vicine occluda le mesh lontane. Il risparmio è sulla GPU che non dovrà mostrare elementi lontani occlusi da altri. Tutti quelli trattati sono considerazioni legate ai costi dello spatial mapping. Purtroppo oltre i costi si hanno limitazioni per quanto riguarda il riconoscimento di volumi: non tutti gli elementi di un ambiente reale vengono riconosciuti. Bisogna analizzare nello specifico come Hololens si comporta in base all'ambiente che ha di fronte. Le criticità maggiori si hanno nei seguenti casi:

- intensità della luce: le camere vengono utilizzate per effettuare il tracking dell'utente e la quantità di luce ideale per loro si aggira intorno i 500 - 1000 lux (unità di misura dei luxmetro). Le camere si possono immaginare come occhi umani: se c'è troppa luce non si riesce a distinguere più i colori e allo

stesso modo se c'è buio non si vedono i colori. Il risultato nel caso di ambiente troppo luminoso o troppo buio è lo stesso: il dispositivo non riesce a creare la mesh quindi si crea un buco.

- tipi di sorgenti luminose: in questo caso si fa riferimento a sorgenti di luce artificiali come ad esempio le lampadine. Come si sa queste lavorano a corrente alternata AC e hanno una certa frequenza. Ora nel caso la frequenza sia di 60Hz non ci sono particolari problemi poiché tale frequenza si allinea ai 30 FPS degli HoloLens, ma se la frequenza è di 50 Hz si potrebbero avere immagini più luminose di altre e altre meno luminose.
- oggetti nell'ambiente: un ambiente molto spoglio senza muri e con pochi elementi al suo interno non è l'ideale: HoloLens si aspetta di raccogliere quante più informazioni possibili per poter calcolare la sua posizione. L'ideale sarebbe una stanza con dei muri pieni di quadri e con una scrivania piena di oggetti. Da evitare anche i pattern ripetuti poiché mandano in confusione il dispositivo.
- oggetti in movimento: in un ambiente molto caotico in cui gli attori reali si spostano rapidamente il dispositivo rischia di non vedere tutti i movimenti. Se ad esempio una persona reale entra ed esce subito da una stanza può capitare che la sua mesh rimanga anche quando lui non è più presente. Questo fantasma andrà ad occludere oggetti che dovrebbero essere visibili.

Microsoft ha messo a disposizione degli sviluppatori un kit chiamato Mixed Reality Tool Kit che può essere scaricato gratuitamente e importato su Unity.



Figura 2.20: MRTK Logo

Questo kit contiene al suo interno diversi script, component e prefab molto importanti per lo sviluppo di applicazioni HoloLens. Va ricordato ad esempio il prefab Spatial Mapping che ha come script Spatial Mapping Observer, Spatial Mapping Manager e Object Surface Observer: questi servono per settare i parametri descritti prima come la densità di triangoli per area oppure la frequenza di aggiornamento delle mesh.

### **Spatial anchor**

Il sistema delle ancore è molto diffuso nelle applicazioni di realtà aumentata e serve ad evitare che gli oggetti virtuali che vengono introdotti dall'utente o già presenti

all'avvio dell'applicazione subiscono dei movimenti indesiderati. Ad esempio se si posiziona un oggetto su un tavolo andrebbe evitato che questo dopo un po' non sia più dove era originariamente. Viene data anche la possibilità di condividere le ancore al fine di far vedere a più utenti contemporaneamente lo stesso oggetto o oggetti nella stessa posizione come mostrato in figura.

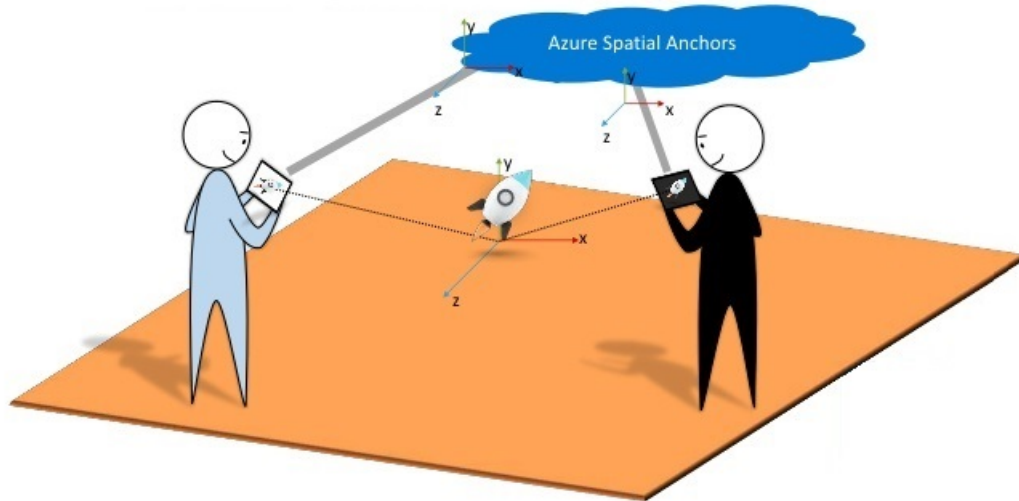


Figura 2.21: Azure spatial anchors

Le ancore vengono salvate localmente o in un cloud remoto in base alle necessità applicative. Le ancore funzionano molto bene per oggetti statici e molto vicini all'origine dell'ancora: oggetti che si trovano a più di 5 metri dall'ancora iniziano a non funzionare bene mentre per oggetti in un raggio di 3 metri circa rimangono stabili. Una caratteristica importante delle ancore è che non possono essere modificate cioè una volta create posso solo caricarle dall'archivio dove si trovano: quindi se vi è la necessità di modificare la posizione di un'ancora l'unica possibilità che si ha è quella di eliminarla e crearne una nuova. Avere come vincolo una distanza relativamente corta dall'ancora e l'obbligo di usare oggetti statici non va bene. Si deve poter avere oggetti che si spostano da una stanza ad un'altra e in ognuna di queste stanze gli oggetti statici non devono spostarsi dalla posizione originale. Entrambi i problemi vengono risolti dall'impiego di più ancore: di fatto avendo più ancore avrò la possibilità di garantire la stabilità di tutti gli oggetti statici e al tempo stesso cambiando l'ancora di riferimento degli oggetti in movimento otterrò la loro stabilità. Ad esempio se ho due mercati (entrambi ancorati) che si scambiano oggetti posso pensare di cambiare l'ancora di riferimento. Nello specifico quando un oggetto parte dal primo mercato assegno l'ancora di tale mercato, quando è in prossimità del secondo mercato assegnerò quella del secondo (nell'ottica di avere gli oggetti sempre vicini all'ancora di riferimento). Tipicamente per ottenere buoni risultati in caso di oggetti virtuali distanti tra di loro si assegna ad ognuno di essi un'ancora. La tentazione può essere quella di creare una griglia di ancore, tutte distanziate di 8 - 10 metri così da avere un ambiente altamente stabile. Ma si sa che il costo computazionale e di archiviazione di queste ancore diventerebbe inutilmente elevato. Quindi l'idea sarebbe usare meno ancore possibili magari atenzionando determinati punti dell'ambiente (aree in cui l'utente svolge diverse azioni rispetto le zone di passaggio).

### **Spatial Sound**

Una funzionalità che rende l'esperienza Hololens ancora più immersiva è la possibilità di far credere ai sensi che il suono provenga da una sorgente olografica che si trova in una determinata posizione. Questo avviene modulando in maniera opportuna il suono che arriva alle orecchie e i vantaggi possono essere molteplici: l'utente ha la possibilità di orientarsi meglio nell'ambiente e può sfruttare anche altri sensi oltre la vista per individuare gli ologrammi.

# Capitolo 3

## Stato dell'arte

### 3.1 Tracking e impiego industriale

La realtà aumentata è in costante sviluppo e le aziende ormai da tempo la sfruttano per migliorare la produttività o per istruire i dipendenti. L'idea di impiegare l'AR in ambito industriale non è affatto una novità: già negli anni novanta se ne parlava [8]. Caudell e Mizell capendo che la produzione di aerei richiede moltissimi componenti propongono un sistema che aiuti gli operatori nella produzione. L'idea è quella di avere un dispositivo che consenta all'utente di vedere sia oggetti reali che modelli 3D e lo accompagni per tutta la produzione: magari indicandogli dove posizionare determinati componenti o mostrando animazioni su come procedere. Solamente un anno dopo Steven Feiner, Blair Macintyre e i collaboratori svilupparono un prototipo, KARMA [13], per supportare gli utenti nella manutenzione delle stampanti laser: in questo caso si sfruttarono tracker acustici ovvero tracker che usano ultrasuoni a bassa frequenza per determinare la posizione e orientamento dell'utente. Il sistema prevede un visore see-through e supporta solo primitive come linee o poligoni e testo e aggiornava le immagini 3D in base al movimento dell'utente. Diversi furono i prototipi e le iniziative [3][2][5][11] e il miglioramento tecnologico ha sicuramente incentivato lo sviluppo: a questo si sono aggiunti dei risultati incoraggianti riguardanti il miglioramento produttivo. A questo proposito Steven J. Henderson e Steven Feiner [15] hanno condotto un'analisi interessante. Prendendo in esame il mantenimento dei veicoli corazzati in ambito militare hanno deciso mettere a confronto varie tecnologie a supporto degli operatori della manutenzione: vengono prese in esame 3 tecnologie HUD, AR e LCD. I risultati che vengono fuori sono prevalentemente positivi: l'AR riesce ad accelerare determinati task come ad esempio la localizzazione di particolari componenti (risulta più veloce del 56% rispetto l'HUD e del 47% del LCD) ma come era prevedibile risulta meno facile da usare da parte degli operatori (circa mezzo punto di distacco dall LCD) e gli operatori riescono a completare interamente i task più velocemente grazie LCD. A questi dati bisogna ricordare però che l'AR, nonostante sia più difficile da usare (non essendo una tecnologia diffusa come altre) risulta intuitivo (4,5 punti in una scala da 1 a 5) e molto soddisfacente (quasi 5 in una scala da 1 a 5). Questi valori possono essere spiegati ad esempio dal vantaggio che l'utente a nel non dover spostare in continuazione la testa avendo tutti le informazioni "sempre" dentro il raggio visivo. Nel caso dell'LCD invece l'utente vedrà le informazioni e il modello 3D nello schermo e dovrà andare a individuarlo nella realtà: in caso di modelli complessi dovrà

ripetere l'operazione più volte. Diversi sono stati gli studi condotti per confrontare le varie tecnologie. Oggi grazie la notevole capacità computazionale in dispositivi sempre più piccoli si parla di tracking multiplo di oggetti (multi-object tracking) [12] e tracking di oggetti composti [10]. Mengyun Yi e He Xu del dipartimento di ingegneria elettrica dell'università di Tsinghua hanno condotto uno studio per un algoritmo che riuscisse a migliorare i risultati esistenti. La sfida che si pone è quella di sviluppare un sistema che riesca a ridurre i falsi positivi e falsi negativi in scenari complessi: infatti gli algoritmi standard riescono fare tracking in scenari semplici. Nonostante la soluzione proposta porti risultati migliori rispetto quelle esistenti si tratta sempre di funzioni probabilistiche.

## 3.2 Hardware e software

Verranno mostrati ora degli strumenti che si occupano di fare tracking di precisione e che vengono utilizzati oggi. Questi si dividono per strategie adottate: esiste la possibilità di sfruttare le frequenze satellitari (GPS-based) per ricavare la posizione esatta di un oggetto nello spazio oppure impiegare il machine learning per calcolare con esattezza la posizione. Iniziando con la prima categoria: Trimble Catalyst e Arrow Gold.

### 3.2.1 Trimble

Azienda fondata nel 1978 che fornisce servizi a industrie globali nel settore agricolo, geospaziale, edile, risorse naturali etc. Ha sviluppato Trimble Catalyst un dispositivo hardware con la finalità di calcolare la posizione geografica dell'utente in maniera molto precisa. Come si vede il dispositivo viene collegato a un dispositivo (android)



Figura 3.1: Trimble Catalyst

mobile come tablet o smartphone che mostrerà tutte le info di localizzazione al-



l'utente. La tecnologia usata è di tipo GNSS (global navigation satellite system): l'antenna, infatti, è capace di catturare frequenze L1 ed L2. L'utente oltre a comprare tale dispositivo dovrà anche abbonarsi ad uno dei servizi offerti dall'azienda: le variazioni di prezzo dipendono dall'accuratezza che l'utente vuole avere (dai metri ai centimetri).

### 3.2.2 Eos

Tale azienda è specializzata nello sviluppo di ricevitori GPS e GNSS Bluetooth ad alta precisione. Hanno presentato Arrow Gold che consente di calcolare la posizione geografica dell'utente in maniera molto precisa. Tale dispositivo riesce a catturare



Figura 3.2: Arrow Gold

anche le frequenze L5 (oltre a quelle L1 ed L2) ed è compatibile con diversi dispositivi mobili (come iOS e Android).

### 3.2.3 Vuforia e Unity

Unity è un motore grafico multiplatforma che unito a Vuforia, kit di sviluppo software di realtà aumentata, può sviluppare applicazioni per dispositivi mobili. Si passa da tecnologie GPS-based a tecnologie basate sul machine learning. In questo caso si sfrutta l'intelligenza artificiale per analizzare l'ambiente circostante. Da questa analisi sarà possibile avere un tracking preciso degli oggetti: in realtà la precisione dipenderà anche dalle caratteristiche specifiche degli oggetti. Un esempio di utilizzo [14] di tale tecnologia viene fornito da Andrew Orlando e Sandor Markon. Il loro sistema usa Unity e Vuforia e può girare su dispositivi android e ios. Si voleva guidare passo passo i dipendenti nella manutenzione dei macchinari: normalmente sarebbe servito un manuale per capire cosa andasse fatto. Grazie l'AR invece si è risparmiato tempo prezioso per reperire il manuale e cercare la parte interessate



al suo interno. Un altro aspetto molto importante è il costo relativamente basso relativo allo sviluppo di tale tecnologia. Vuforia è un kit di sviluppo software di realtà aumentata per dispositivi mobili. Ha un engine suo ed offre la possibilità di scaricare .unitypackage (pacchetto importabile in Unity e avente tutte le funzioni che servono per lo sviluppo di un'applicazione in realtà aumentata). Si hanno 3 strumenti messi a disposizione dal team di Vuforia:

- Vuforia Object Scanner
- Model Target Generator
- Vuforia Image Recognition

La prima è un app per smartphone che offre la possibilità di scannerizzare un oggetto reale, salvare le feature specifiche e utilizzarle su Unity per ottenere il riconoscimento dello stesso oggetto. Model Target Generator è un programma per il riconoscimento che non parte dalle feature di oggetti reali ma da cad realizzati (ad es in maya o blender) e infine Vuforia Image Recognition è un servizio messo a di dal team Vuforia ed è specializzato nel riconoscere immagini come foto o quadri.

### **Vuforia Object Scanner**

Vuforia mette a disposizione un app per android e un foglio che dovrà essere stampato e avrà l'origine (in basso a sinistra) e un'area bianca dentro cui ci si aspetta di trovare l'oggetto da scannerizzare. Bisognerà mettere l'oggetto entro l'area del file stampato. Tutto ciò che sta fuori non verrà visto. Questo può essere utile se si vuole prendere solo una parte dell'oggetto scelto. Durante la scansione l'oggetto non deve essere mosso. E' possibile fare una scansione successiva per migliorare il riconoscimento dell'oggetto desiderato. Nella stessa app si può vedere quanto è preciso il riconoscimento di tale oggetto: più punti è riuscito a catturare più il riconoscimento è stato accurato. Una volta soddisfatti del risultato ottenuto basterà caricare su unity il file prodotto. Purtroppo questa soluzione ha grosse limitazioni poiché allo stato attuale è possibile fare lo scanner solo di oggetti piccoli (che riescono ad entrare nel rettangolo bianco del foglio citato) e non fissi. Non è possibile infatti staccare i tubi o altri oggetti fissati nel terreno e scansionarli in un ambiente poco rumoroso. Si elencano le best practice per ottenere buoni risultati:

- L'ambiente non deve essere troppo luminoso (e neanche troppo buio) e bisogna evitare che l'oggetto sia colpito direttamente dalla luce.
- Evitare oggetti riflettenti poiché, colpiti dalla luce, rischiano di non essere visti completamente. Devo fare in modo che l'oggetto sia uniformemente illuminato: avere oggetti che fanno ombra sul target scelto dovrebbero essere evitati.
- L'ambiente dovrebbe essere quanto più spoglio possibile e lo stesso vale per lo sfondo. In questo modo evito che parti dell'oggetto vengano coperte o che vengano prese parti che non sono del target.
- Si dovrebbe avere la possibilità di girare attorno l'oggetto: un tavolino rotondo sarebbe l'ideale.

Allo stato attuale vuforia da la possibilità di riconoscere fino a due oggetti contemporaneamente e la dimensione degli oggetti che posso scansionare è limitata (oggetti piuttosto piccoli che stanno in un tavolo). Si ricordi infine che non tutti gli oggetti sono facilmente scansionabili: oggetti deformabili, oggetti con parti mobili, riflettenti sono sconsigliati. Si consiglia invece di usare oggetti rigidi, opachi e con pochissime parti mobili.

### **Model Target Generator(MTG)**

Le caratteristiche che dovrebbero avere i cad che si caricano nel software messo a disposizione dal team Vuforia (Model Target Generator). Le caratteristiche che si vanno a ricercare sono una somiglianza quanto più fedele del modello 3D all'oggetto reale, una complessità del modello non troppo elevata e una corrispondenza del colore. La cosa che bisogna attenzionare è la scala del modello e il numero di poligoni. Le caratteristiche consigliate sono le seguenti:

- non superare i 400,000 poligoni o triangoli;
- non avere più di 10 parti;
- evitare di mettere più di 5 texture;
- fare uso di sistema di coordinate opportuno (destrorso).

Gli stessi sviluppatori ricordano che questi parametri sono da prendere con i dovuti accorgimenti: esistono tantissimi software per modellare oggetti e ognuno ha la sua pipeline di lavoro che potrebbe allargare o stringere i parametri sopra elencati. Da qui emerge l'impossibilità nel dare consigli che riescano ad adattarsi a tutti i software di modellazione. Altre criticità legate ai modelli sono i grandi fori: questi possono portare a un overhead computazionale o addirittura a un rilevamento di feature inesistenti. Quando viene caricato il modello nel Model Target Generator può capitare che alcune parti vengano cancellate: sarà cura dello sviluppatore stare molto attento al modello che viene caricato. Anche le normali messe male possono portare a effetti disastrosi: rilevamento di parti interne del modello (che dovrebbero essere invisibili), creazione di feature inesistenti etc. Si è detto che si cerca di rilevare la forma dell'oggetto ma anche i colori possono aiutare: non è necessario che siano fedeli a quelli reali, anzi è importante che non lo siano affatto. Usare texture fotorealistiche o materiali che cercano di emulare effetti fisici particolari possono portare a un effetto opposto. La regola è la seguente: dare colori diversi a parti diverse dell'oggetto oppure assegnare colori diversi in base ai materiali dell'oggetto; si faccia riferimento all'immagine 3.3 come esempio.

Come si nota porte dell'armadio, tubo, maniglie e il resto dell'armadio sono di colori diversi e non realistici poiché l'idea è quella di aiutare il software a capire quanti componenti diversi ci sono nel modello. Si ricorda che i dispositivi che si useranno non hanno grosse capacità computazionali e appesantirli non è una buona idea quindi si può pensare di ridurre quanto più il numero di poligoni del cad: non bisogna esagerare altrimenti si rischia di non avere feature sufficienti per il riconoscimento. Ad esempio un oggetto con 500.000 poligoni può essere ridotto a 50.000 senza avere un degrado del riconoscimento. Per semplificare ulteriormente il cad si può pensare di:

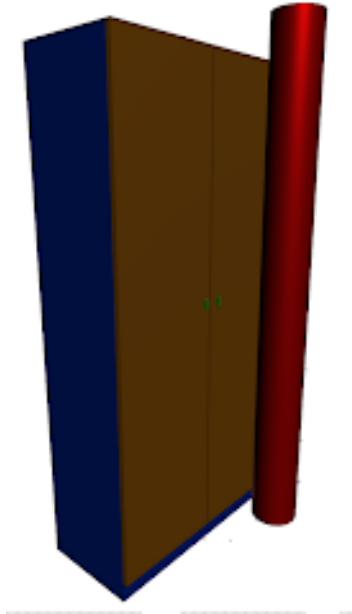


Figura 3.3: Esempio colorazione di un oggetto

- rimuovere parti mobili dell'oggetto (volante di una macchina o ruote)
- rimuovere parti che non sono subito visibili (interni di un cassetto)
- eliminare tutte le parti che non devono essere tracciate (oggetto che magari è nel cad ma non è parte della scena reale)

Se si hanno più oggetti bisogna assicurarsi che ognuno di essi sia ben distinguibile a livello di forma dagli altri per non avere falsi positivi. Per effettuare l'analisi di un modello 3D bisognerà:

- caricare un modello
- definire una Guide View
- creare un Model Target
- creare un database

Si tenga presente che i formati che possono essere processati sono Creo View (.pvz), Collada (.dae), FBX (.fbx), IGES (.igs, .iges), Wavefront (.obj), STEP (.stp, .step), STL (.stl, .sla), VRML (.wrl, .vrm).

Fatto ciò si può procedere con il creare una guide view dell'oggetto appena caricato. Le guide view rappresentano un punto di vista con cui l'utente vedrà il target. MTG (model target generator) mette a disposizione diverse opzioni come mostrato in figura 3.5. Il caso più restrittivo è 1 view, in cui si dovrà scegliere l'unica angolazione con cui si riconoscerà l'oggetto) a seguire si ha 2 views, in cui l'utente dovrà scegliere una vista e l'altra verrà calcolata in automatico dal software, fino ad arrivare a 6 view, in cui si sceglie una vista e le altre 5 verranno calcolare in automatico dal sistema. Questa è una limitazione: una volta scelta l'angolazione non sarà possibile cambiarla se non eseguendo di nuovo tutti i passi visti fino a questo momento. Quello a cui si aspira è poter riconoscere l'oggetto reale qualunque

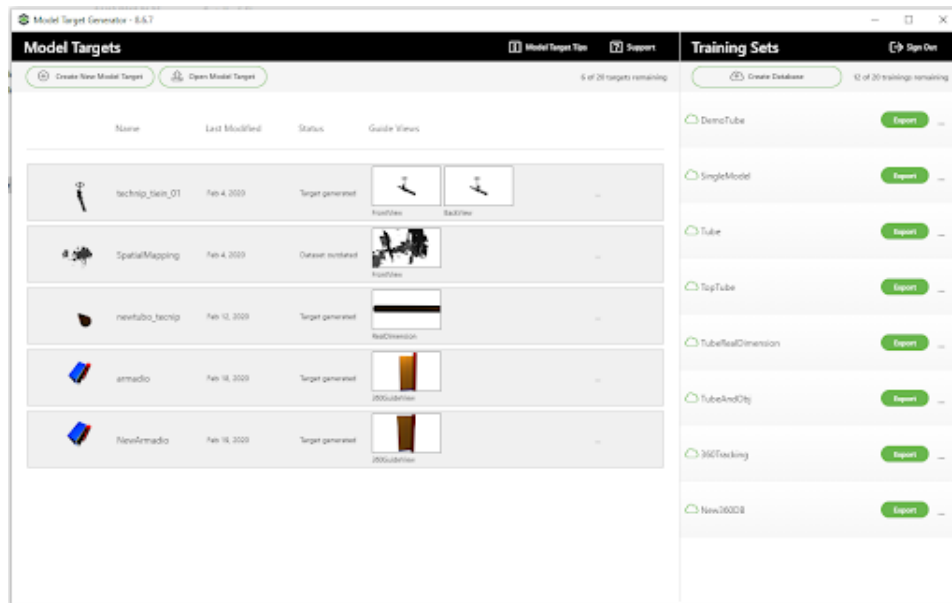


Figura 3.4: Home Page Model Target Generator

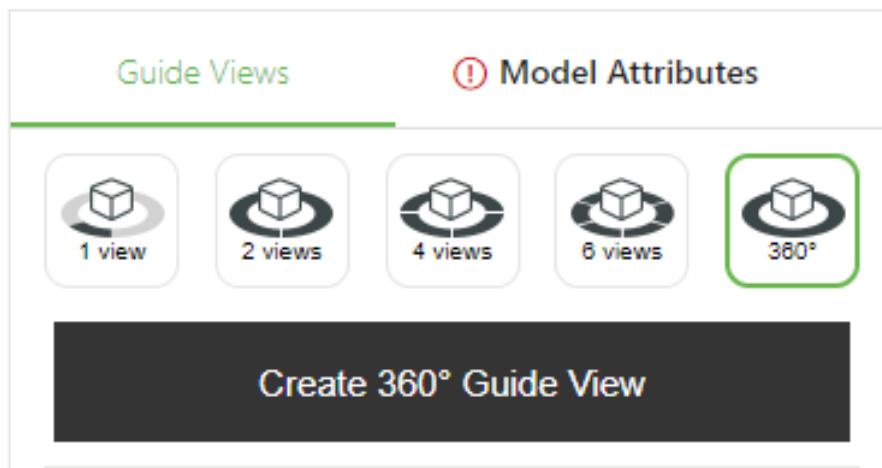


Figura 3.5: Tipi di Guide View

sia l'angolazione con cui lo aggancio. Questo è reso possibile dalla modalità 360. Fatto ciò si può procedere con la generazione del “model target”: la creazione di quest'ultimo è limitata a un massimo 20 utilizzi. Avendo un Model target si può procedere con la creazione del database: anche qui si ha un limite di 20 utilizzi cioè si possono creare 20 database. Si hanno 2 opzioni: Advanced o Advanced 360°. Il primo ti consente di riconoscere più oggetti contemporaneamente ma rispettando certi vincoli. Il secondo invece può riconoscere solo un oggetto alla volta senza avere vincoli stringenti. Vale quanto segue:

- Model target di tipo 1 view - 6 view sono inseribili nel database di tipo Advanced
- Model target di tipo 360° sono inseribile nel database di tipo Advanced 360°.

Una volta inseriti i modelli nel DB si potrà procedere con il training: tale operazione può richiedere, per ogni modello, dai dieci minuti circa(caso di 1 view) a

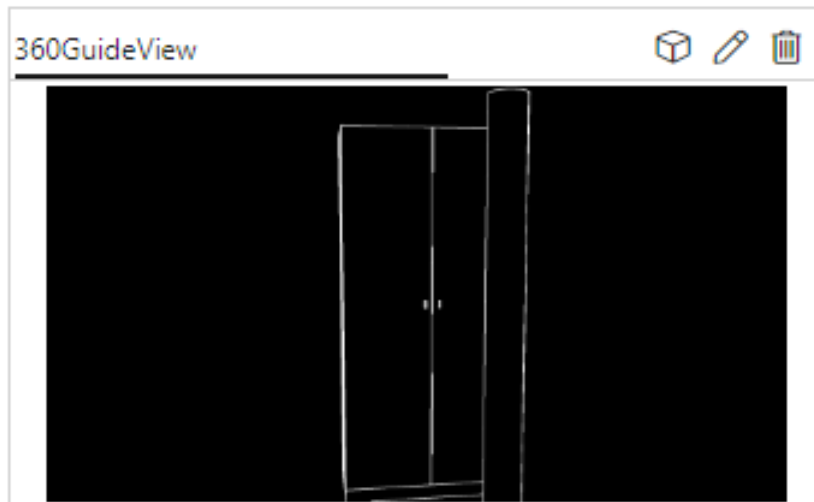


Figura 3.6: Esempio di Guide View

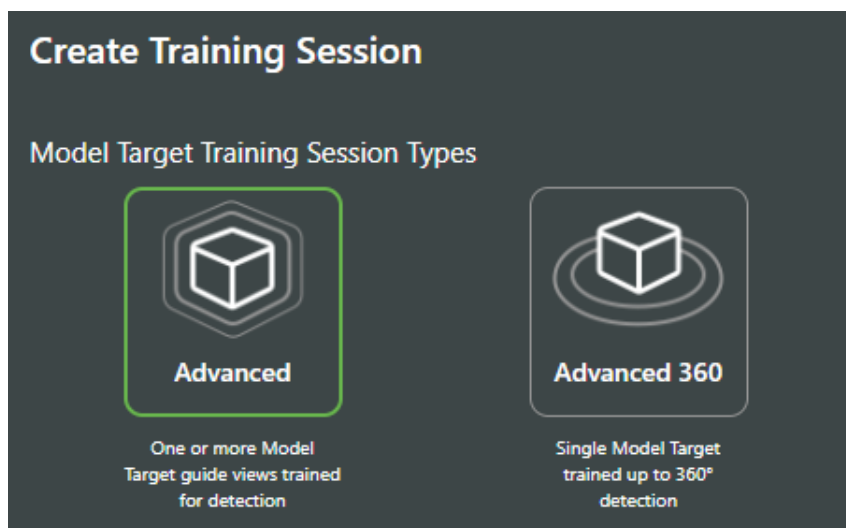


Figura 3.7: Tipi di Database

diverse ore (caso 360°). Alla fine di tale operazione verranno prodotte delle immagini, un file .dat, uno .XML e soprattutto uno .unitypackage. Quest'ultimo sarà quello che dovrà essere importato su Unity. Andando a selezionare Model Target si dovrà selezionare il database desiderato e il modello appartenente al database che si vuole riconoscere, come mostrato in figura 3.9. Prima però bisognerà assicurarsi di avere una license key valida: ottenibile registrandosi nel sito ufficiale di Vuforia. Infine è bene controllare che i valori del campo Physical Length, Width e Height (espressi in metri) siano corretti. Per oggetti che hanno guide view compresa tra uno e sei sarà possibile mostrare un'immagine che aiuterà l'utente a inquadrare correttamente l'oggetto reale. Per oggetti riconoscibili da qualunque angolazione (360°) non verrà caricata alcuna immagine di supporto. Sia che l'oggetto sia di tipo 1 view o 360° è possibile caricare una "Custom 2D GuideView Image" (figura 3.10), cioè un'immagine prodotta dallo sviluppatore che verrà mostrata per aiutare l'utente.

A questo punto se si intende aggiungere un oggetto virtuale in un punto rispetto l'oggetto tracciato basterà imparentare tale oggetto al model target appena creato 3.11.

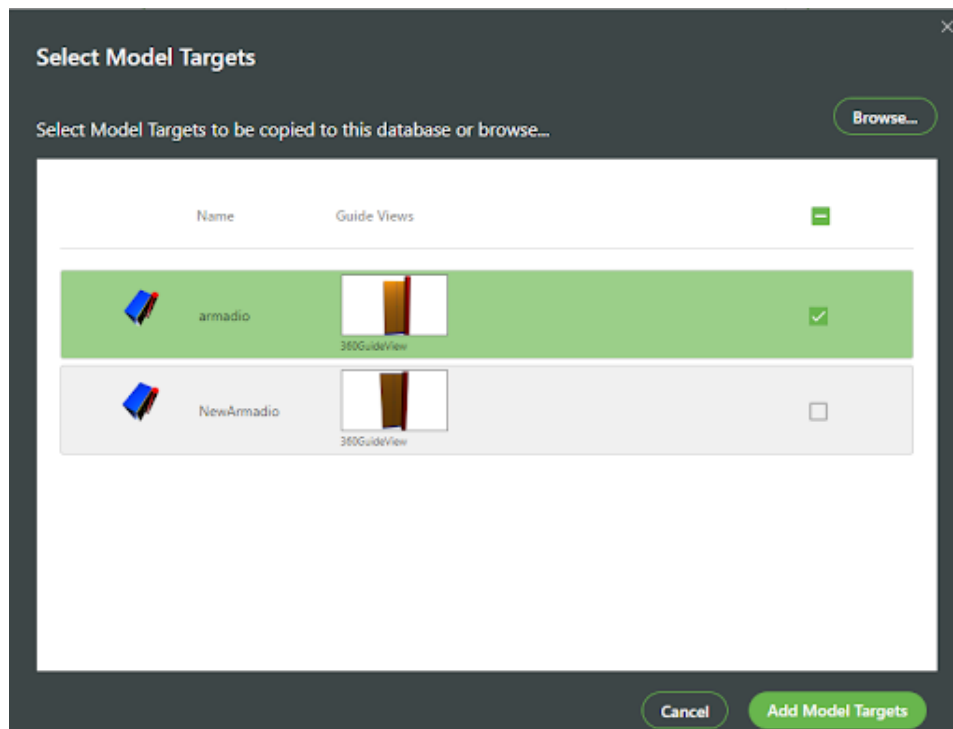


Figura 3.8: Creazione del Database

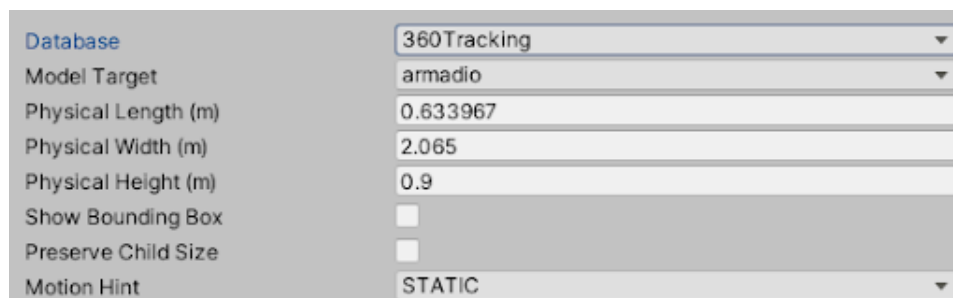


Figura 3.9: Parametri Model Target su Unity

### Vuforia Model Target App

Questa app (per android) è messa a disposizione dagli sviluppatori Vuforia e serve per testare quanto sia riconoscibile l'oggetto generato dal programma Model Target Generator. Dopo aver installato nel proprio smartphone ModelTargetsTest bisognerà caricare manualmente gli oggetti da testare. Solitamente all'installazione dell'app viene creata in automatico la cartella: `com.vuforia.engine.ModelTargetsTest`, se ciò non dovesse avvenire bisognerà crearla manualmente. Il path finale dovrebbe essere qualcosa del genere: `/sdcard/Android/data/com.vuforia.engine.ModelTargetsTest/files/ModelTargetData/YourDatabaseName/`. Una volta aggiunti i file (.dat e .xml), creati con Model Target Generator, nella cartella YourDatabaseName sarà possibile avviare l'applicazione: basterà inquadrare l'oggetto reale riferito al modello 3D e una piccola tabella riporterà la qualità dell'oggetto in termini di riconoscibilità.

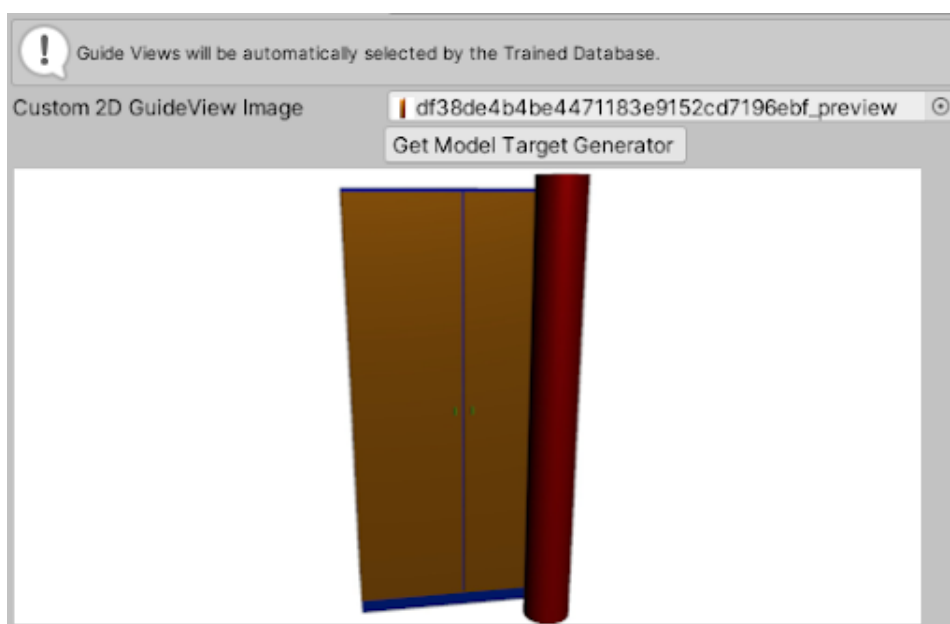


Figura 3.10: Esempio di Guide View Custom

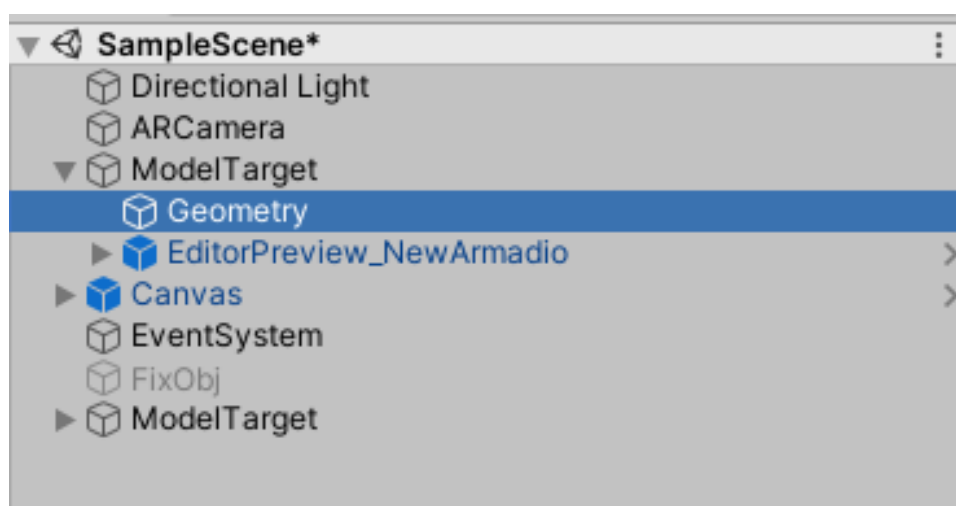


Figura 3.11: Aggiunta di un oggetto 3D chiamato geometry

### Vuforia Image Recognition

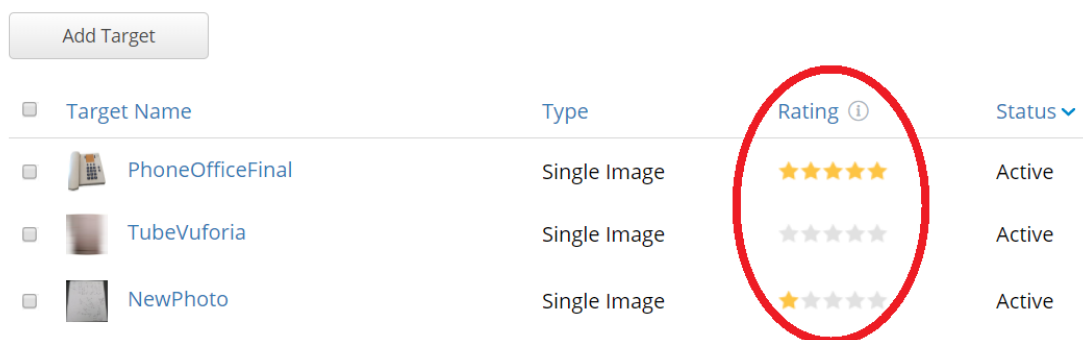
Il team Vuforia mette a disposizione degli sviluppatori il suddetto servizio: dopo la registrazione al sito ufficiale sarà possibile accedere alla sezione "develop", dove sarà possibile scaricare una licenza, caricare immagini e database. La licenza dovrà essere inserita su Unity e servirà per effettuare il deploy dell'applicazione. Il primo passo sarà la creazione di un database e il caricamento dell'immagine che si intende tracciare. Si noti a tal proposito che esistono immagini particolarmente facili da riconoscere e altre meno. Le caratteristiche desiderate sono le seguenti:

- Immagini ricche di dettagli: folla da stadio, una stanza molto disordinata o più in generale una moltitudine di cose o persone.
- Immagini con un buon contrasto cioè avere ogni regione dell'immagine ben illuminata. Bisogna evitare che ci siano intere zone oscurate o intere zone

estremamente illuminate.

- Immagini senza modelli ripetitivi come ad esempio i motivi ripetuti in una tovaglia da tavola oppure i pattern delle decorazioni.
- Le immagini caricate non devono superare la dimensione di 2MB e si accettano solo i formati PNG e JPG a 8 o 24 bit.

Dei punti sopra elencati solo l'ultimo è condizione necessaria per caricare le immagini: ognuna di queste avrà una valutazione (Rating) che va da 0 a 5 come mostrato in figura 3.12



The screenshot shows a web interface with a table of targets. At the top left is a button labeled 'Add Target'. The table has four columns: 'Target Name', 'Type', 'Rating', and 'Status'. There are three rows of data. The 'Rating' column is circled in red. The first row has a rating of 5 stars, the second has 4 stars, and the third has 1 star.

Target Name	Type	Rating	Status
PhoneOfficeFinal	Single Image	★★★★★	Active
TubeVuforia	Single Image	★★★★☆	Active
NewPhoto	Single Image	★☆☆☆☆	Active

Figura 3.12: Esempio di Rating

Il punteggio viene determinato, oltre che dalle caratteristiche elencate sopra, dalla distribuzione delle funzionalità: un'immagine che concentra tutti i suoi dettagli in una regione e lascia le altre senza non avrà una buona valutazione. Ad esempio una foto con una casa al centro e uno sfondo monocromatico concentra tutte le feature nei pressi della casa e lascia coperti tutti gli altri. Un altro aspetto molto importante sono le forme degli oggetti: le sfere e tutte le forme morbide non aiutano il riconoscimento delle immagini a differenza delle forme spigolose. Anche le dimensioni reali vanno tenute in considerazione: in particolare la dimensione è collegata alla distanza che l'utente avrà rispetto il target. La regola è moltiplicare le dimensioni per un fattore 10: se dimensione è 10 centimetri la distanza massima è 1 metro. Ad esempio se l'immagine da riconoscere è larga 30 centimetri l'utente non dovrà superare i 3 metri di distanza dal target altrimenti quest'ultimo non sarà più riconoscibile. Alcune considerazioni vanno fatte anche sul dispositivo che viene utilizzato, sulle condizioni del target reale e sul comportamento dell'utente:

- Una camera deve mettere a fuoco per poter tracciare l'immagine: esiste l'opzione FOCUS MODE CONTINUOUS AUTO che automatizza la messa a fuoco del dispositivo quando l'utente si muove all'interno della scena. Purtroppo alcuni vecchi dispositivi non sono abilitati per questa modalità quindi una buona alternativa può essere FOCUS MODE TRIGGERAUTO cioè abilitare la messa a fuoco attraverso il tocco dello schermo da parte dell'utente.
- Il target deve essere posto su un supporto rigido e non deve presentare piegature o deformazioni.
- Evitare ambienti troppo luminosi o troppo bui. L'ideale sarebbe scegliere ambienti chiusi e con una luce controllata. Se l'ambiente è molto buio si può pensare di utilizzare il flash del dispositivo.



- L'utente deve fare attenzione all'angolazione rispetto il target e le sorgenti luminose. Può capitare che si vengano a creare pesanti riflessioni che copriranno intere zone dell'immagine. La conseguenza è il non riconoscimento dell'immagine.

Bisogna ricordare che è possibile migliorare la valutazione che viene data all'immagine senza modificare il target reale:

- Taglio delle regioni superflue: se l'immagine ha i bordi che non portano informazioni utili si può ritagliare tali zone e lasciare solo quello che serve.
- Se il contrasto non è alto si può modificare con software come photoshop.
- Quando possibile è consigliabile modificare l'immagine reale affinché sia più semplice da riconoscere.

Nella figura 3.13 si ha un esempio di quanto detto.



Figura 3.13: Esempio di modifica

A sinistra viene mostrata l'immagine modificata e a destra quella originale. Ogni croce gialla è una feature che viene riconosciuta: l'immagine di sinistra ha molte più feature rispetto quella di destra. La differenza è il livello di contrasto.

### 3.2.4 VisionLib e Unity

Il rivale di Vuforia è VisionLib: creato da Visometry che è una startup e spin-off Fraunhofer molto giovane, fondata nel 2017. L'architettura è molto centralizzata e non prevede l'utilizzo di software esterni come nel caso di Vuforia: nello specifico bisognerà scaricare solamente un \*.unitypackage dal sito ufficiale e lì si troverà tutto il necessario per usufruire dei servizi offerti da VisionLib:

- Calibrazione della camera: deve essere effettuata sul dispositivo mobile che si intende utilizzare. Di fatto si raccolgono dati relativi alla capacità della camera del dispositivo.
- Riconoscimento di oggetti reali: partendo da un modello 3D dell'oggetto sarà possibile riconoscere lo stesso nella realtà.
- Riconoscimento di immagini: partendo dall'immagine digitale di un quadro o una foto sarà possibile riconoscere la stessa nella realtà.
- Viste multiple: è possibile definire diverse angolazioni dalle quali l'applicazione riconoscerà un oggetto reale.
- Riconoscimento multiplo di oggetti reali: partendo da modelli 3D degli oggetti sarà possibile riconoscere gli stessi nella realtà.

### Calibrazione della camera

Questo processo è fondamentale per ottenere un buon riconoscimento dell'immagine o modello quindi viene elencato cosa è bene evitare e cosa è importante attenzionare. La prima cosa da fare è andare nella cartella Camera Calibration, contenuta nel \*.unitypackage scaricato, aprire la scena presente (scene Simple Camera Calibration) ed eseguire il deploy. Dopo aver installato l'app sul cellulare si avrà bisogno della scacchiera (scaricabile dal sito ufficiale di VisonLib). Il funzionamento lo si può riassumere nella macchina a stati mostrata in figura.

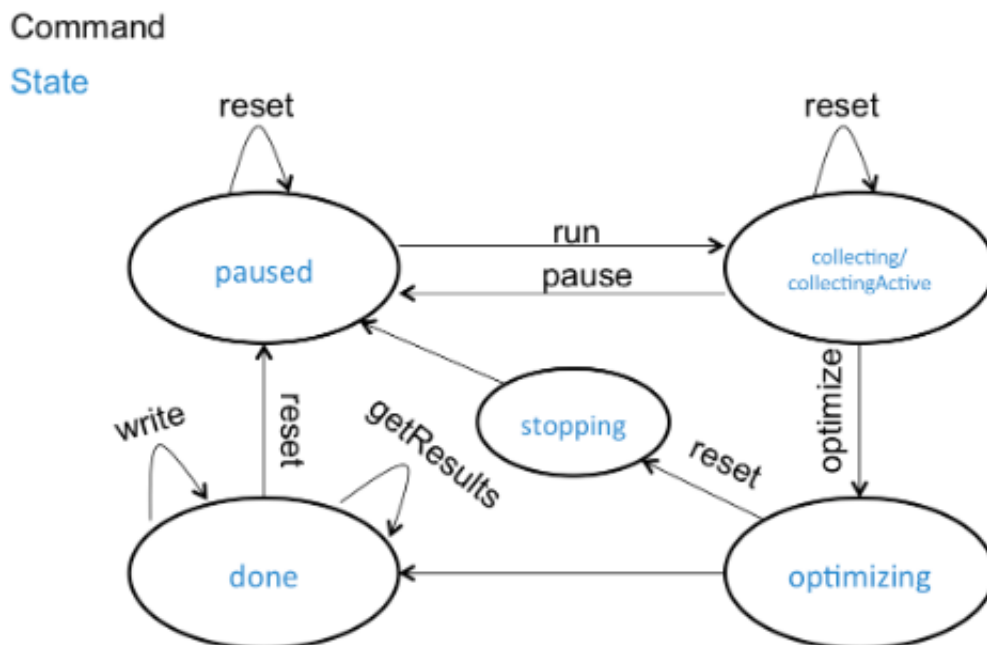


Figura 3.14: Macchina a stati della calibrazione della camera

All'avvio si è nello stato pause: non appena si è pronti ad iniziare basta premere il bottone di run. A questo punto apparirà un pattern e l'obiettivo sarà quello di

muovere il cellulare o tablet per allinearli ai segnalini che appariranno sulla scacchiera: `collecting` passerà a `collectingActive`. Sono necessarie diverse inquadrature della camera per passare allo stato `optimizing`. Verrà data una valutazione alla calibrazione fatta e si potrà decidere di ripartire dall'inizio il processo. Se non si è soddisfatti del risultato ottenuto si può usare bottone di reset oppure salvare i dati localmente. Verrà prodotto un file `.JSON` che dovrà essere importato in Unity e conterrà tutti i parametri ottenuti dalla calibrazione. Di seguito sono elencati i parametri per una buona valutazione:

- Lunghezza e Larghezza uguali: ovvero tutti i quadrati della scacchiera devono avere la stessa dimensione. Non è importante la grandezza assoluta dei quadrati (3x3 cm).
- Superficie rigida: quando si stampa il foglio bisogna assicurarsi che non abbia piegature: si consiglia di incollare il foglio su una superficie rigida per evitarle.
- Uso di uno schermo piatto: una soluzione alternativa alla stampa della scacchiera è l'utilizzo del monitor del pc. Il file pdf messo a disposizione nel sito rispetta molto bene il primo vincolo e assicura che la superficie sia rigida.
- Area di riconoscimento: quando viene eseguita la calibrazione bisogna ricordarsi che la scacchiera deve essere sempre visibile per intero dalla camera.
- Dimensioni dell'immagine: se si utilizza uno schermo si consiglia di avere la scacchiera quanto più grande possibile. Ridurre la dimensione è altamente sconsigliato poiché si rischia di non calcolare i parametri di distorsione.
- Angoli obliqui: quando si cercano di inquadrare le viste bisogna evitare di mantenere la camera sempre perpendicolare allo schermo. Bisogna avere la capacità di inclinare la camera senza tagliare fuori parti della scacchiera stessa.
- Movimenti delicati e lenti: quando ci si muove attorno la scacchiera con la camera non bisogna avere fretta; movimenti molto veloci rischiano di causare effetti negativi nei risultati di calibrazione. Le camere che hanno i dispositivi tablet o smartphone acquisiscono le immagini in maniera sequenziale, ciò significa che non è consigliabile avere notevoli differenze tra un'immagine e la successiva.

### Riconoscimento di oggetti reali

Per descrivere come sviluppare un'applicazione che riconosca gli oggetti reali è importante aver già seguito le seguenti istruzioni:

- scaricare dal sito ufficiale di VisionLib il file `*.unitypackage` e importarlo su Unity;
- effettuare la calibrazione della camera;
- importare correttamente `*.json`, prodotto dalla calibrazione, su Unity: come mostrato in figura 3.15.



Figura 3.15: Import del file calibration su Unity

Fatto questo bisognerà importare il cad dell'oggetto che si vogliono venga riconosciuto su Unity: bisognerà fare attenzione alle dimensioni dell'oggetto importato ricordando che ogni unità di Unity corrisponde a un metro nella realtà. Ci sono altri componenti importanti per far funzionare il tutto:

- VInitCamera che serve ad aiutare l'utente nel inquadrare l'oggetto reale
- file .vl che contiene tutte le info fondamentali per far capire alla camera quale sarà l'oggetto che deve riconoscere nella scena. I parametri più importanti di questo file sono: "modelURI": "project dir:NomeFile.FBX" con cui si indica il nome dell'oggetto da riconoscere e "metric": "mm" con cui si indica l'unità di grandezza.
- VLCamera che si occuperà di valutare il file \*.Json creato nella fase di calibrazione, il file \*.vl descritto sopra e la licenza.

Va precisato che VLCamera per leggere il file \*.vl ha bisogno di uno script che gli indichi dove prendere il suddetto file: la funzione principale è la seguente worker-Behaviour.StartTracking("path da seguire"). I formati dei modelli riconosciuti sono 3D, 3DS, AC, ASE, B3D, BLEND, BVH, COB, CSM, DAE, DXF, FBX, GLB, GLTF, HMP, IFC, IRR, IRRMESH, LWS, LXO, MD2, MD3, MD5, MDC, MDL, MDL, MS3D, NDO, NFF, NFF, OBJ, OFF, OGEX, PK3, PLY, Q3D, Q3S, RAW, SCN, SMD, STL, TER, VTA, X, XGL, XML e ZGL. Tra questi il più testato è il formato OBJ e PLY. Similmente a quello che accade per Vuforia se si vuole aggiungere ologrammi all'oggetto riconosciuto basterà aggiungere il modello 3D in prossimità del target. Una volta settato il tutto si può procedere con la Build dell'applicazione.

### Riconoscimento di immagini

Questa funzione può essere vista come una versione semplificata del riconoscimento di oggetti vista prima. Stavolta non si dovranno caricare modelli 3D ma immagini in formati riconosciuti come ad esempio png. Valgono infatti le stesse premesse viste prima e gli attori principali sono simili a quelli visti:

- VInitCamera non è necessaria poichè si parla di una figura 2D da riconoscere.
- Il file .vl contiene tutte le info fondamentali per far capire alla camera quale sarà l'oggetto che deve riconoscere nella scena. I parametri più importanti di questo file sono: "imageURI": "project dir:NomeImmagine.png" con cui

di indica il nome dell'immagine da riconoscere e "metric": "mm" con cui si indica l'unità di grandezza e "realWidth" con cui si indica la larghezza. Si noti che PNG e JPEG sono gli unici formati riconosciuti.

- VLCamera che si occuperà di valutare il file \*.Json creato nella fase di calibrazione, il file \*.vl descritto sopra e la licenza.

Come specificato prima, VLCamera ha bisogno di uno script per leggere il file \*.vl. Si noti inoltre che il parametro realWidth è strettamente associato a "metric": nello specifico in uno indico la dimensione e nell'altro l'unità di grandezza. Se si vuole aggiungere ologrammi basterà aggiungere il modello 3D nell'origine della scena. Una volta settato il tutto si può procedere con la Build dell'applicazione.

### Viste multiple

Questa funzionalità si può vederla come le guide view di Vuforia. Di fatto si dà la possibilità di impostare più punti di vista per riconoscere l'oggetto. Non si ha ancora una versione definitiva ma solo una beta. I protagonisti principali sono i prefab VLWorkspace e VLWorkspaceManager. Il primo serve per definire le pose o meglio i punti di vista consentiti e deve essere associato a un VLWorkspaceManager. I parametri principali di VLWorkspace sono i seguenti:

- la definizione di un up-vector cioè la definizione del vettore verticale del modello;
- la possibilità di mostrare o meno la linee tratteggiate tra punti di origine e destinazione;
- la possibilità di controllare le pose che verranno generate.

Si possono sviluppare tutte le pose possibili andando a settare opportunamente i parametri origin e destination di VLWorkspace: si può scegliere tra 4 forme (sfera, piano, linea e punto) e per ognuna di esse può scegliere il valore della larghezza, lunghezza, raggio (nel caso della sfera) e così via.

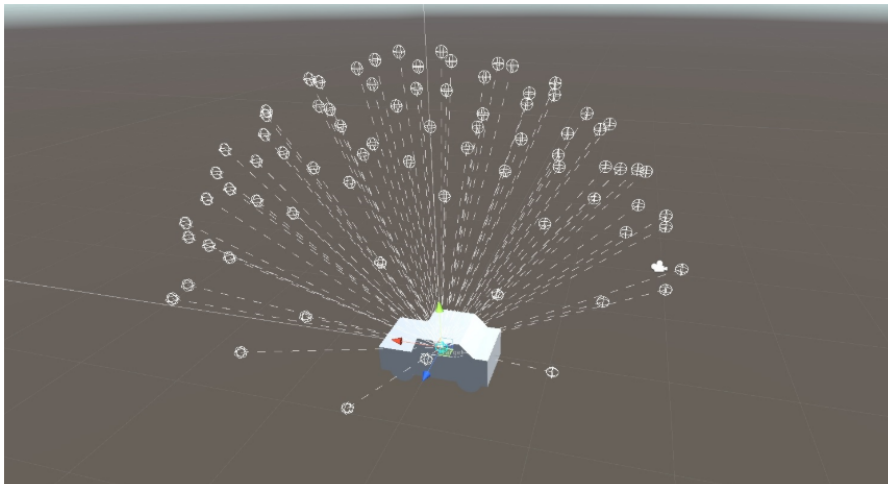


Figura 3.16: Esempio di oggetto con viste multiple

La beta si trova nella cartella ModelTracking, contenuta nel solito \*.unitypackage, e si chiama AutoInit. Si noti che non basta solo impostare correttamente tutti i parametri ma sarà necessaria anche una fase di apprendimento una volta effettuato il deploy. Non appena tutte le pose sono state analizzate verrà creato nel dispositivo utilizzato un file \*.binz che verrà caricato ogni avvio dell'applicazione. Di norma non è possibile ripetere questa operazione: l'unica opzione è quella di eliminare manualmente il file \*.binz seguendo il seguente path: localstorage-dir:/VisionLib/AutoInit/initDatafilename.binz.

### Riconoscimento multiplo di oggetti reali

Questa funzione non ha una versione definitiva ma soltanto una beta: essa racchiude molti concetti visti per VisionLib come VLWorkspace e Calibrazione della camera. Di fatto l'idea alla base è quella di avere per ogni oggetto che si intende riconoscere dei parametri specifici: per un dispositivo mobile più datato questo potrebbe risultare impossibile a livello computazionale. La versione beta di esempio si trova nella cartella ModelTracking e si chiama MultiModelTracking. La prima cosa da fare sarà impostare il parametro Coordinate System Adjustment di VLCamera a Multi Model Tracking. Dopo di che bisognerà introdurre VLTrackingAnchor: questo prefab è quello che rappresenterà ogni oggetto che si vuole riconoscere. In figura 3.17 si mostra la struttura gerarchica che bisogna realizzare. Come si nota per ogni

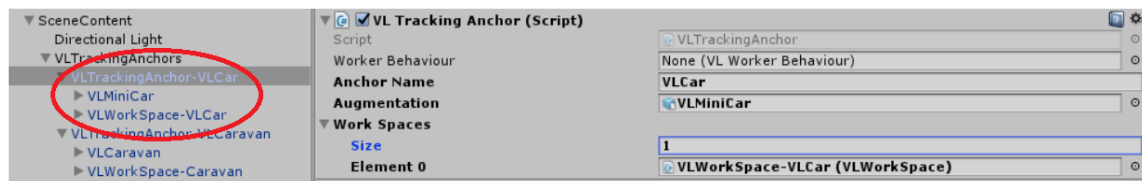


Figura 3.17: Esempio di gerarchia

VLTrackingAnchor si ha un modello 3D dell'oggetto (VLMiniCar) e il Workspace associato: il primo serve a determinare cosa deve essere riconosciuto e il secondo le pose consentite per il riconoscimento. Va ricordato che questa modalità non è disponibile per Hololens di prima generazione.

### 3.2.5 Deploy di un'applicazione AR

Per il deploy dell'applicazione viene utilizzato Unity. In questo caso la versione che si utilizza può portare a problemi di compatibilità: in particolare Vuforia con le versioni di Unity precedenti alla 2019, in alcuni casi, aveva problemi nel deploy dell'applicazione. Di fatto il codice di licenza che veniva fornito genera problemi con le vecchie versioni di Unity: la soluzione che si propone è quella di aggiornare l'engine a una versione 2019.x e generare una nuova licenza come mostrato in figura 3.18. Si ricordi che eliminare una chiave di licenza e crearne una nuova non implica l'eliminazione di tutti i target caricati.

Un altro vantaggio non indifferente nell'utilizzo di una versione aggiornata è legato alla semplicità nel setting: in particolare si evita allo sviluppatore di scaricare manualmente SDK e NDK per Android e OpenJDK di Java con il rischio di avere problemi di compatibilità.

## Add a free Development License Key

License Name \*  
License

You can change this later

### License Key

Develop  
Price: No Charge  
Reco Usage: 1,000 per month  
Cloud Targets: 1,000  
VuMark Templates: 1 Active  
VuMarks: 100

☒ By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Cancel Confirm

Figura 3.18: Creazione di una License Key

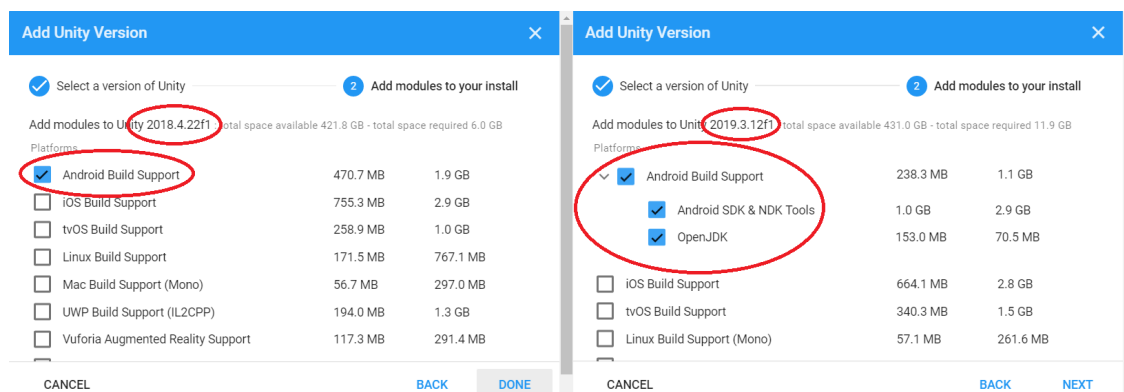


Figura 3.19: Differenza tra versione 2018 e 2019

Quanto detto vale sia per applicazioni AR con Vuforia che VisionLib. In entrambi i casi il deploy è identico: una volta selezionata l'opzione Android nella sezione Build Settings basterà procedere con la build. Tra i file che verranno prodotti quello di maggiore interesse è \*.apk che bisognerà caricare e installare nel device mobile. Si noti infine che il dispositivo dove si installa l'applicazione deve avere una versione pari o maggiore di quella selezionata in fase di sviluppo.

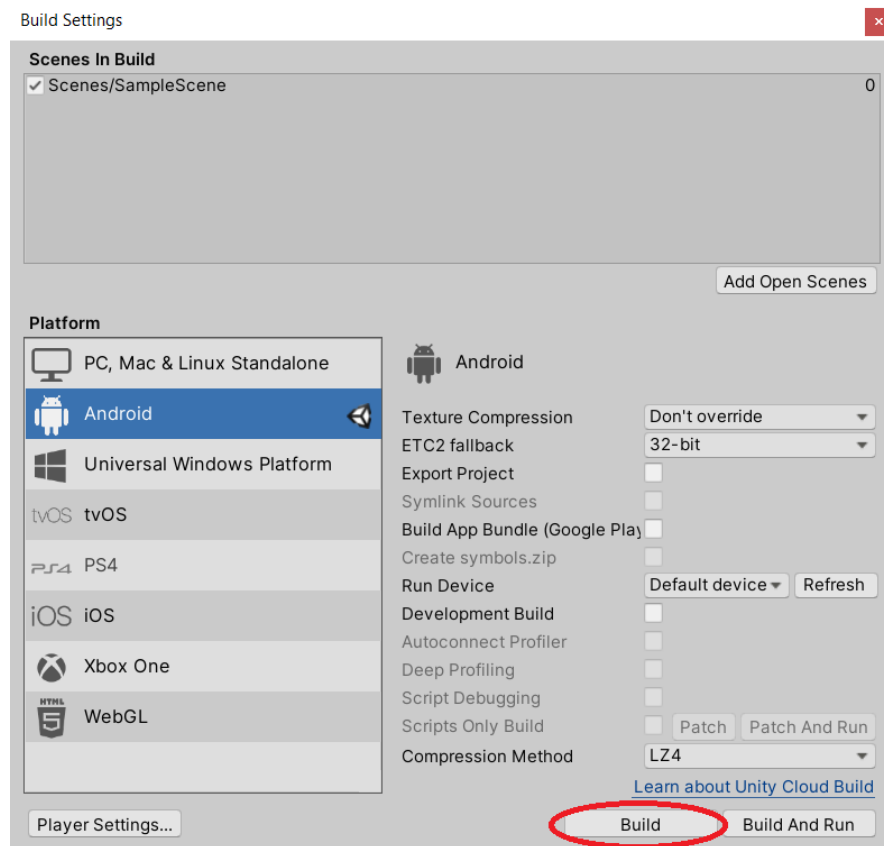


Figura 3.20: Esempio di build su Unity



# Capitolo 4

## Sviluppo di una soluzione con Hololens

### 4.1 Requisiti progetto

L'obiettivo del lavoro di tesi è quello di posizionare in maniera precisa un ologramma in un ambiente industriale, pieno di tubi che si estendono il lungo e largo. Bisogna anche immaginare diversi piani, ognuno con specifici tubi, manopole, manometri etc. Un ambiente molto rumoroso, caotico e a cielo aperto come mostrato in figura. Gli operatori dovranno tagliare porzioni di tubi esistenti e inserire un tie-in cioè un



Figura 4.1: Esempio di impianto industriale

collegamento con altri tubi o una valvola. Quello che bisogna fare sarà riconoscere, tra la selva di tubi, quello corretto e posizionare in maniera precisa un ologramma del tie-in che l'operatore dovrà montare. I vantaggi derivanti dall'impiego di un sistema AR in questo contesto sono:

- l'utente vedendo come dovrà essere il risultato finale della sua operazione eviterà di eseguire un montaggio errato

- non sarà necessario istruire ogni singolo utente su quale tubo scegliere e dove montarlo poiché l'applicazione mostrerà all'utente cosa fare
- avendo un supporto visivo l'utente potrà procedere in maniera più spedita.

I dispositivi che verranno impiegati sono smartphone, tablet o Hololens. Viene data una limitazione nell'impiego della tecnologia: non è consentito usare tecniche marker based cioè non sarà possibile aggiungere niente all'impianto (beacon o marker). Questa limitazione viene giustificata dal fatto che se per ogni montaggio si dovrebbe andare a posizionare un marker la soluzione non sarebbe scalabile. Anche la connessione non può essere garantita in ogni punto dell'area industriale quindi soluzioni che richiedano una connessione costante e un massiccio trasferimento di dati potrebbero non girare bene in base a dove verrebbe operata la riparazione. Una soluzione basata sull'intercettazione delle frequenze satellitari potrebbe non portare un buon risultato: andrebbe valutata la distanza dall'antenna di riferimento. In ogni caso una soluzione del genere richiederebbe uno sforzo economico, come visto nel capitolo precedente, e tale sforzo può essere giustificato solo da prestazioni molto superiori a quelle presentate. Ma come visto la precisione massima che si può raggiungere tramite queste strategie è a livello del centimetro: questi risultati possono essere raggiunti anche dalle soluzioni proposte. Viene messo a disposizione un CAD dell'impianto industriale. I parametri che interessano maggiormente sono l'usabilità (l'applicazione è automatica o l'utente deve intervenire in qualche modo, l'utente ha familiarità con il dispositivo o deve imparare a usarlo), la precisione (l'oggetto virtuale si trova esattamente dove voglio, le sue dimensioni sono corrette, la sua posizione nello spazio e la sua rotazione sono giuste) e le limitazioni (l'applicazione scala bene aumentando il numero di tubi da processare, presenza di banchi).

## 4.2 Sviluppo dell'applicazione

L'applicazione di tracking presentata è marker-less e serve a mostrare all'utente dove montare la valvola, una volta raggiunta l'area dove avverrà la riparazione. L'intervento dell'utente per poter posizionare in maniera precisa il tie-in è determinante. Verrà definito un sistema di riferimento rispetto al quale verrà posizionato in maniera precisa un oggetto. L'applicazione è sviluppata per Hololens di prima generazione usando unity e il Mixed Reality Toolkit messo a disposizione da Microsoft.

### 4.2.1 Definire un sistema di riferimento

Il sistema di riferimento cartesiano ha 3 assi di riferimento perpendicolari tra loro e con un preciso orientamento. Le considerazioni che verranno fatte da adesso in poi sono relative a Unity: usando 3 punti sarà possibile avere un sistema di riferimento. Un punto rappresenterà l'origine del sistema di riferimento, un altro andrà a determinare direzione e orientamento dell'asse  $z$  e infine l'ultimo servirà a definire direzione e orientamento dell'asse  $y$  (l'asse  $x$  sarà il prodotto vettoriale di quelli precedentemente definiti). L'utente avrà origine e verso del sistema di riferimento che l'utente andrà a settare. La precisione del sistema di riferimento dipende da quanto l'utente sarà abile e scrupoloso nel seguire i vari passi dell'applicazione. Inoltre sapendo che l'asse  $Y$  è perpendicolare al suolo e orientato verso l'alto saranno necessari solo 2 punti per piazzare il sistema di riferimento cartesiano.

### 4.2.2 Definire un interfaccia

L'applicazione si apre con il tutorial che spiega quali passi seguire. Per rendere più intuitivo e semplice il sistema si sono aggiunte delle immagini.

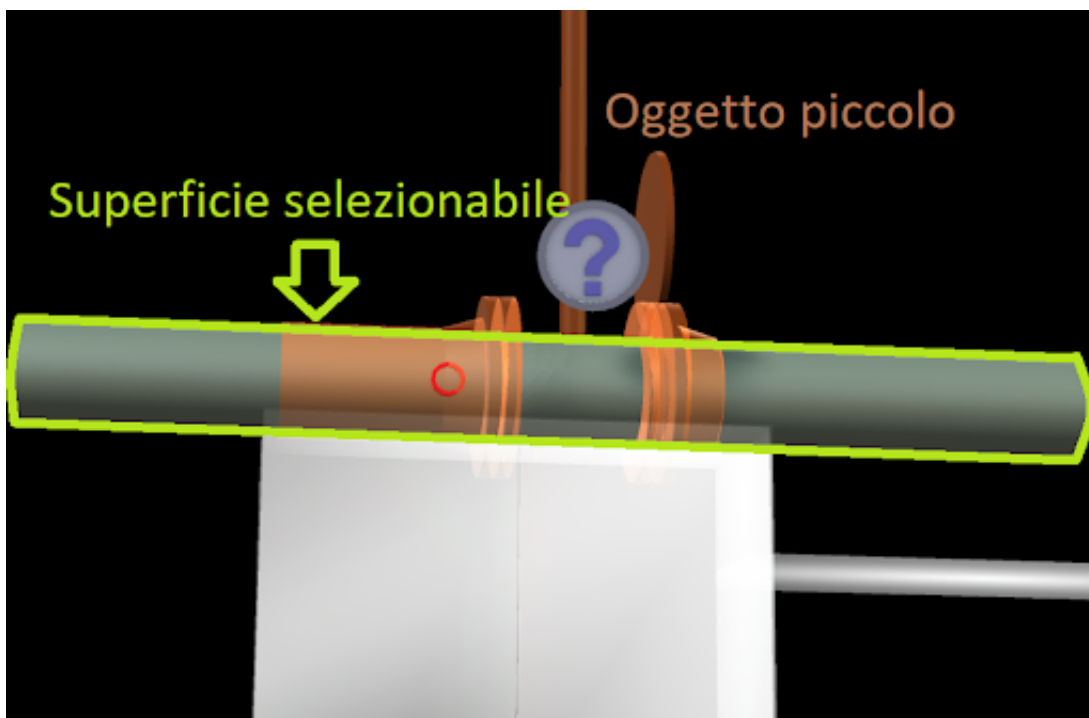


Figura 4.2: Immagini del Tutorial

Finito il tutorial l'utente avrà davanti il CAD in dimensioni ridotte dell'ambiente in svolgere la riparazione: tale oggetto sta sempre all'interno del cono visivo della camera ed è rivolto sempre verso l'osservatore. In caso l'utente dimentichi cosa andava fatto può tappare sull'icona "tutorial" per rivedere cosa fare.



Figura 4.3: Icone principali

L'utente capisce su quale tubo operare tramite il modello ridotto: quanto capisce dove operare dovrà tappare l'oggetto per ingrandirlo (dimensioni reali). A questo punto l'utente dovrà selezionare (tramite la gesture tap) il primo punto e piazzarlo sull'oggetto reale. E la stessa cosa farà con l'altro punto. Se si sbaglia il posizionamento di uno dei punti è possibile ripetere l'operazione: il ciclo continua finché l'utente non è soddisfatto del risultato ottenuto. Finito il settaggio dei punti è possibile selezionare l'icona fix Position per nascondere tutto quello che non serve e mostrare solo l'oggetto interessato. Nel caso si voglia tornare indietro è possibile usare l'icona gear.



- Fix-position false: rendi visibili e selezionabili i punti e gli altri ologrammi

Per motivi di spazio alcuni stati sono stati duplicati. Questi stati sono da considerarsi identici tra di loro anche se non presentano le stesse connessioni. Gli stati in questione sono “2°pt tap” e “Anchor tap”. Il tutto inizia con lo stato tag-along true (oggetto piccolo che sta dentro il cono visivo degli HoloLens): l'unica azione che potrà far proseguire da questo stato sarà il tap sul tubo virtuale. L'applicazione (nello stato calibrazione) aiuta l'utente nell'individuazione dei punti da selezionare e inoltre fa vedere nel CAD dove dovranno essere posizionati nel mondo reale. All'avvio però verrà mostrato solo il primo punto e solo dopo averlo piazzato verrà mostrato il secondo. Si noti che una volta selezionato un punto (stato Anchor tap o 2°pt tap) parte lo spatial mapping che rimane abilitato finché l'utente non esegue il tap su un punto dello spazio. Una volta che si finisce nello stato Anchor tap o 2°pt tap si potrà solo andare nello stato set anchor o set 2°pt: questo perché l'unica interazione possibile è quella con lo spatial mapping. Di fatto vengono disabilitati tutti i collider tranne quello dello spatial mapping. Facendo ciò si evita il rischio che i collider degli oggetti virtuali coprano il punto che voglio selezionare. Un altro vantaggio è il risparmio di risorse: operazioni pesanti come lo spatial mapping devono essere sfruttate solo quando è necessario. L'icona del menu è sempre visibile tranne nel caso in cui lo spatial mapping è attiva e la stessa cosa vale per lo stato tag-along. Un commento va fatto anche allo stato fix position: disponibile solo dopo che l'utente abbia posizionato tutti i punti. Si è lasciata comunque la possibilità all'utente di tornare indietro in caso di tap errato e di posizionare meglio i punti (usando l'icona gear).

#### 4.2.4 Script principali

##### Tag along e Interpolator

Ecco i principali parametri dello script tag-along:

- TagAlong Distance: la distanza in metri che l'oggetto avrà dalla camera quando viene aggiornata la sua posizione
- Position Update Speed: la velocità con cui l'oggetto si muove quando viene aggiornata la sua posizione (metri al secondo)
- Smoothing factor: quanto dolcemente deve avvenire l'aggiornamento della posizione (valore da 0 a 1)
- Minimum Horizontal/vertical Overlap: il volume visibile dell'oggetto (in percentuale) prima di aggiornare la sua posizione (valore compreso tra 0 e 1)
- Minimum TagAlong Distance: la distanza minima in metri tra utente e oggetto sotto la quale quest'ultimo dovrà essere riposizionato

TagAlong va insieme allo script Interpolator che viene abilitato non appena l'oggetto inizia ad uscire dalla vista della camera.

### Billboard

In base all'orientamento dell'oggetto virtuale e a come voglio che l'utente lo veda dovrò scegliere l'asse di riferimento che dovrà essere rivolto verso l'utente.

### Anchor Manager

Si occupa di salvare e cancellare le ancore. Le operazioni principali sono la WordAnchor Store.Save(string name,Word Anchor anchor) e la WordAnchor Store.Delete(string name). Le ancore sono caratterizzate da un nome (univoco) e dal component WordAnchor. Per non appesantire l'applicazione il numero di ancore è stato ridotto al minimo: ogni qual volta si vuole riposizionare l'ologramma l'applicazione andrà a cercare l'ancora da spostare nella anchor Store (usando Word Anchor Store.GetAllIds che torna tutte i nomi delle ancore presenti nello store), la elimina se ne crea una nuova con lo stesso nome. Questo processo è inevitabile a meno di volere creare una nuova ancora ogni spostamento dell'oggetto: ricordando che un ancora una volta creata non può essere traslata,ruotata o scalata.

### Gesture Manager

Questo script si occupa di riconoscere le gesture e capire cosa fare in base all'oggetto che viene tappato appoggiandosi su altri script. Nel caso trattato la gesture tap viene riconosciuta: si utilizzano funzioni come GestureRecognizer() e SetRecognizableGestures(GestureSettings.Tap). La gestione di tale gesture viene affidata a un delegate: recognizer.Tapped += GestureRecognizer Tapped; Un altro aspetto molto importante di tale script è la presenza di un raycast che si occupa verificare quale oggetto è stato selezionato durante il tap Physics.Raycast(Vector3 origin,Vector3 direction, out RaycatHit). Sulla base dell'oggetto selezionato si chiama un metodo piuttosto che un altro.

### Game Manager

Il game manager è il direttore dei lavori. Ogni qual volta che si vorrà creare o aggiornare un ancora oppure riconoscere una gesture si dovrà fare riferimento al game manager.

### ISelectHandler

Ogni oggetto con cui si vuole interagire avrà un mesh collider e dovrà realizzare l'interfaccia ISelectHandler. Tale interfaccia riguarda tutti gli oggetti selezionabili e ha il metodo OnSelect() che, in questo caso, specificherà quali eventi o metodi devono essere chiamati.

### State Handler

Questo script si occupa di cambiare il materiale agli oggetti selezionati. Ogni oggetto che deve cambiare colore ha questo script: il colore rappresenta lo stato dell'oggetto (selezionato e non selezionato).

### WorldCursor

Questo script rende visibile il cursore solo quando il raycast collide con qualunque oggetto. Quando viene abilitato lo spatial mapping non è detto che tutte le superfici reali possano essere riconosciute: la presenza o meno di tale cursore aiuta a capire se si può interagire o meno con una superficie.



Figura 4.5: Cursore dell'applicazione

### 4.2.5 Oggetti principali

#### Tubo

Realizza l'interfaccia `ISelectHandler` quindi il metodo `OnSelect()`: in questo caso si va ad ingrandire l'oggetto (o ridurlo di scala).

#### Primo punto di riferimento

Il primo punto si riferisce all'origine del sistema di riferimento che bisogna posizionare. In questo caso lo script `Interaction Handler` (che sviluppa l'interfaccia `ISelectHandler`) si assicura che non esista un ancora associata a tale oggetto e in caso positivo la elimina: le operazioni di verifica ed eliminazione sono effettuate dall'Anchor Manager. Da adesso in poi il game Manager si occupa di abilitare e disabilitare lo spatial mapping, piazzare nello spazio il punto e di associare un ancora a tale punto.

#### Secondo punto di riferimento

Il secondo punto serve per orientare uno degli assi di riferimento. Il flusso è un pò più lungo di quello proposto sopra ma il ragionamento è uguale. La posizione di un ancora non può essere cambiata quindi l'unica possibilità è quella di cancellare quella che si ha, aggiustare l'orientamento dell'asse e salvarla come se fosse una nuova ancora.

#### Spatial mapping

Questo oggetto si riferisce a tutte le superfici rilevabili dal Hololens cioè a tutti gli oggetti reali. Quando selezionato dovrà passare al game manager la posizione tappata dall'utente. Tale punto è quello che determina le coordinate dell'ancora (nel caso sia stato selezionato il primo punto) o l'orientamento dell'asse z dell'ancora.

### Tutorial, check e gear

Non fanno altro che nascondere degli oggetti e mostrane altri. Mostrati in figura 4.3.

#### 4.2.6 Utilizzo di un terzo punto

Per semplicità si utilizzano solo due punti per la definizione del sistema di riferimento (dando per certo che l'asse Y sia sempre orientato correttamente). Su Unity se si volesse controllare anche la direzione dell'asse y bisognerebbe aggiungere il terzo punto. Come detto più volte il primo punto rappresenta l'origine, il secondo direzione e verso di un asse e lo stesso per il terzo. Su Unity viene data l'opportunità solo di orientare l'asse Z verso un punto specifico e non un asse qualsiasi. Quindi si dovrà fare qualche calcolo in più per arrivare al risultato desiderato. Nello specifico si farà quanto segue:

- Definire un punto che rappresenta l'origine degli assi di riferimento
- Uso della funzione di puntamento verso l'asse Z per determinare direzione e orientamento di Z
- Calcolare la rotazione che bisogna applicare all'asse z per far puntare Y verso tale punto

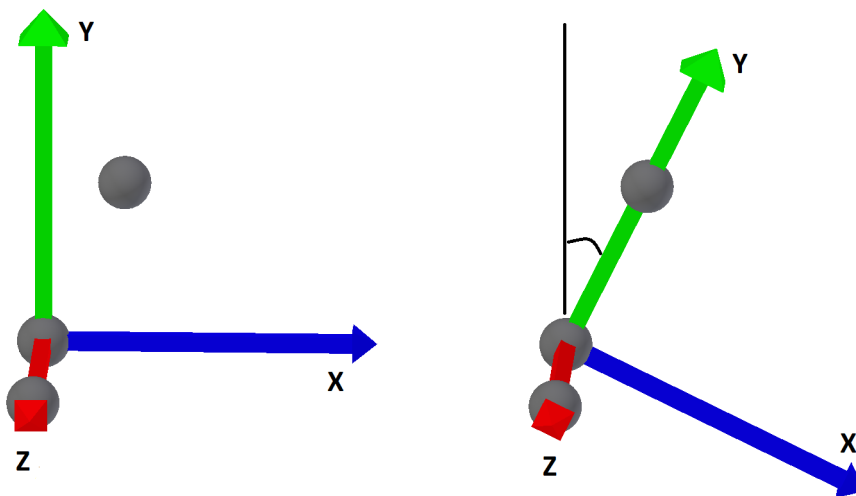


Figura 4.6: Impiego di un terzo punto

A questo punto l'unica difficoltà sarà capire di quanti gradi ruotare tale asse Z. Verrà utilizzato un punto di supporto. e questo avrà coordinate precise: (0 , 0 , z Point 3) in riferimento all'asse locale creato dall'ancora (origine del sistema di riferimento). Il valore di z Point 3 sarebbe la coordinata z del 3° punto. Quest'ultimo verrà agganciato da tale punto di supporto: l'angolo che si viene a formare tra "support" e Point 3 è lo stesso che andrà ad assegnare come rotazione all'asse z del sistema di riferimento.



# Capitolo 5

## Sviluppo di una soluzione con Vuforia e VisionLib

Oggi esistono diversi software che cercano di effettuare il tracking di precisione. Non si valuteranno soluzioni che riguardano l'utilizzo delle frequenze L1,L2 e la nuova L5 come Trimble Catalyst ma soluzioni riguardanti il riconoscimento degli oggetti: Vuforia e VisionLib. In particolare si descriveranno le applicazioni sviluppate e si metteranno a confronto pregi e difetti offerti dalle due tecnologie.

### 5.1 Vuforia

Nel terzo capitolo si è parlato delle potenzialità di Vuforia e in particolare di Model Target Generator (MTG), che rappresenta il cuore dell'applicazione realizzata. Prima di descrivere i dettagli tecnici si parlerà sinteticamente di cosa fa l'applicazione. Essa ha un tutorial, come mostrato in figura 5.1, in cui viene spiegato cosa fare e un interfaccia.



Figura 5.1: Tutorial Applicazione

L'obiettivo è inquadrare l'oggetto reale con la camera del dispositivo e l'applicazione mostrerà l'ologramma in verde: quest'ultimo rappresenta la valvola che l'operatore dovrà montare. Un esempio è mostrato in figura 5.2.

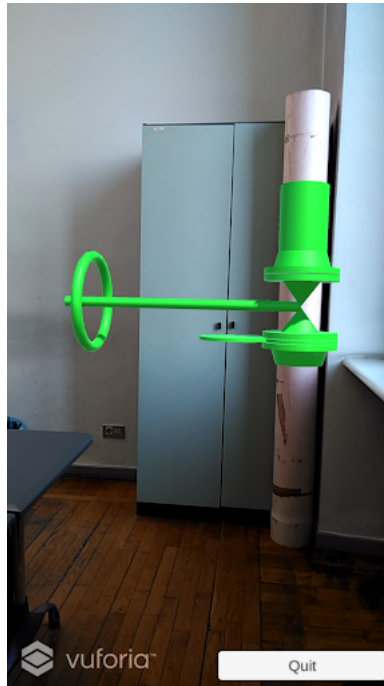


Figura 5.2: Esempio di utilizzo

La figura 5.2 presenta un armadio e un tubo che possono essere assimilati ad una cabina, che controlla la pressione, e il tubo.

### 5.1.1 Architettura e dettagli tecnici dell'applicazione

Nella figura 5.3 viene mostrata l'architettura dell'applicazione realizzata che ha come obiettivo il riconoscimento di oggetti reali.

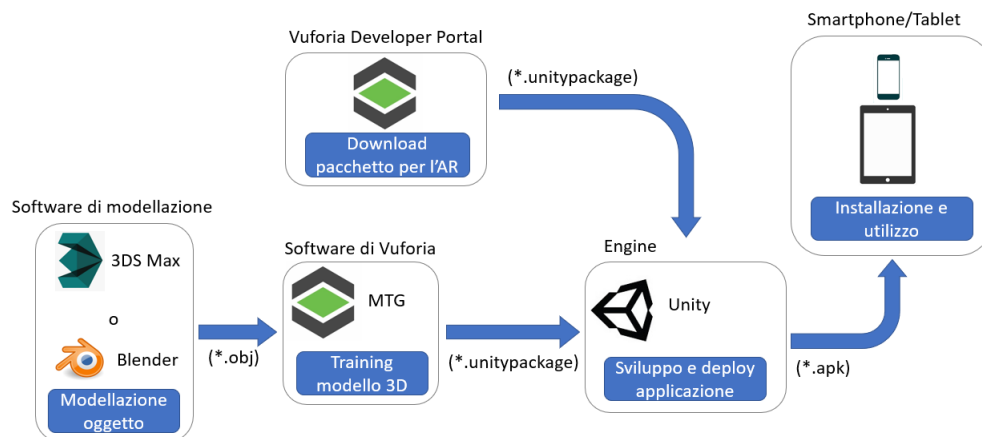


Figura 5.3: Architettura dell'applicazione

Partendo da sinistra verso destra si ha come primo step la creazione di un modello e il training dello stesso. Per la modellazione si sono prese le misure precise di

ogni elemento: altezza e raggio del tubo, altezza, larghezza e spessore delle porte dell'armadio e così via. Fatto questo si è passato alla realizzazione del modello 3D: come visto nel terzo capitolo, questo deve avere dimensioni reali e colori diversi per ogni parte differente dell'oggetto come mostrato in figura 5.4: porte dell'armadio (marrone), tubo (rosso), maniglie dell'armadio (verde) e i lati (blu).

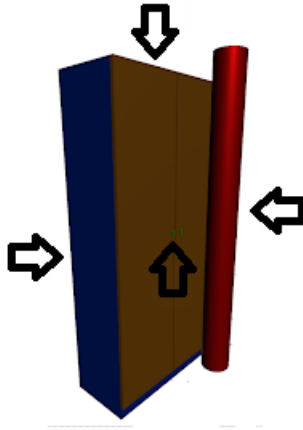


Figura 5.4: Colorazione Modello

Si noti che MTG è in grado di fare il training di diversi formati e che un qualunque software di modellazione, in grado di produrre formati validi, va bene: Blender e 3DS Max sono degli esempi. Come visto nel terzo capitolo per MTG mette a disposizione diverse modalità di training: quella selezionata per l'applicazione è "Advanced 360". Questa ci consente di riconoscere un solo oggetto da qualunque vista e ha richiesto circa due ore per il training. Fatto questo bisognerà importare i file necessari su Unity, scaricabili dal sito ufficiale di Vuforia, e inserire la licenza posseduta come mostrato in figura 5.5.

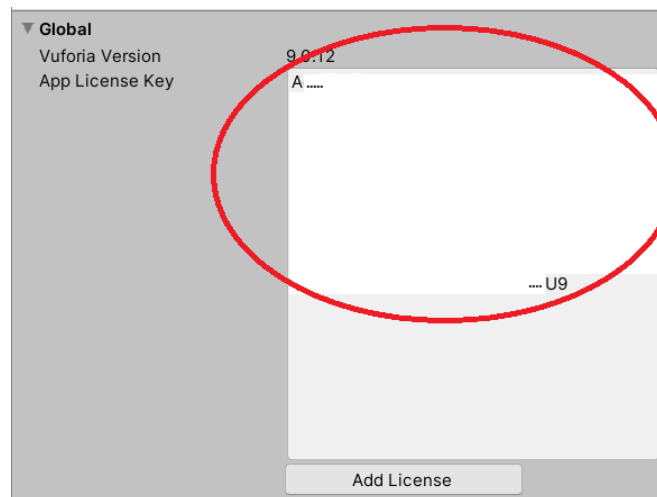


Figura 5.5: Licenza di utilizzo

Ora si ha tutto il necessario per poter sviluppare l'applicazione: basterà importare il modello di cui si è fatto il training e aggiungere vari script di controllo per aprire e chiudere la finestra del tutorial e uscire dall'applicazione. Nello specifico il database importato da MTG è stato nominato New360DB e il Model Target NewArmadio: i parametri specifici sono stati mostrati in figura 5.6.

Database	New360DB
Model Target	NewArmadio
Physical Length (m)	0.654613
Physical Width (m)	2.065
Physical Height (m)	0.9
Show Bounding Box	<input type="checkbox"/>
Preserve Child Size	<input type="checkbox"/>
Motion Hint	STATIC

Figura 5.6: Dettagli Model Target

L'ultimo step mostrato dall'architettura è il deploy dell'applicazione. Per quanto riguarda le versioni dei vari software: 2019.3.12 di Unity, la 8.6.7 di MTG, 9.0.12 di Vuforia e 6.8 di Android. Si vedrà nei prossimi paragrafi che la semplicità del modello considerato nasconde dei limiti.

### 5.1.2 Model Target Generator: limitazioni

Di sicuro una soluzione del genere è molto usabile poiché l'utente non dovrà preoccuparsi di essere preciso nel piazzare dei punti, a differenza della soluzione del quarto capitolo, ma dovrà solo inquadrare l'armadio o più in generale un target reale, al resto ci pensa l'applicazione. L'accuratezza raggiunge i centimetri, non si hanno limiti sulle dimensioni dell'oggetto da riconoscere e basta avere il cad per sviluppare l'applicazione. Quest'ultima considerazione si adatta perfettamente al caso di studio in esame poiché si ha già il cad del modello reale. In generale però non è facile creare modelli molto fedeli alla realtà soprattutto se si tratta di replicare strutture esistenti di grandi dimensioni. Le limitazioni non mancano e queste si concentrano, oltre che nella realizzazione del cad, nei tempi di training del modello e nella complessità intrinseca dell'oggetto da riconoscere. Nel terzo capitolo si è visto come dovrebbe essere un modello facilmente riconoscibile: le geometrie semplici rappresentano un problema e i tubi non hanno geometrie complesse, inoltre non è detto che il modello realizzato abbia esattamente le stesse dimensioni del modello reale. Un errore di misurazione dell'oggetto reale si propaga nel cad e a sua volta un errore nel cad si propaga nel training che si fa dell'oggetto portando ad imprecisioni nel posizionamento finale. Un altro problema si ha con i falsi positivi cioè oggetti reali simili all'oggetto target che vengono riconosciuti erroneamente come target (questo aspetto lo si vedrà meglio con le immagini del paragrafo sui risultati).

## 5.2 VisionLib

Il rivale di Vuforia è proprio VisionLib. Si tratteranno in questo paragrafo, oltre i dettagli tecnici dell'applicazione sviluppata, le differenze che intercorrono tra le due tecnologie e se una delle due si adatta meglio al caso di studio. L'applicazione realizzata è identica a livello estetico e funzionale: esattamente come l'altra ha un

tutorial introduttivo e l'obiettivo è quello di inquadrare, con il proprio dispositivo, il target.

### 5.2.1 Architettura e dettagli tecnici dell'applicazione

Come fatto prima viene mostrata l'architettura: figura 5.7.

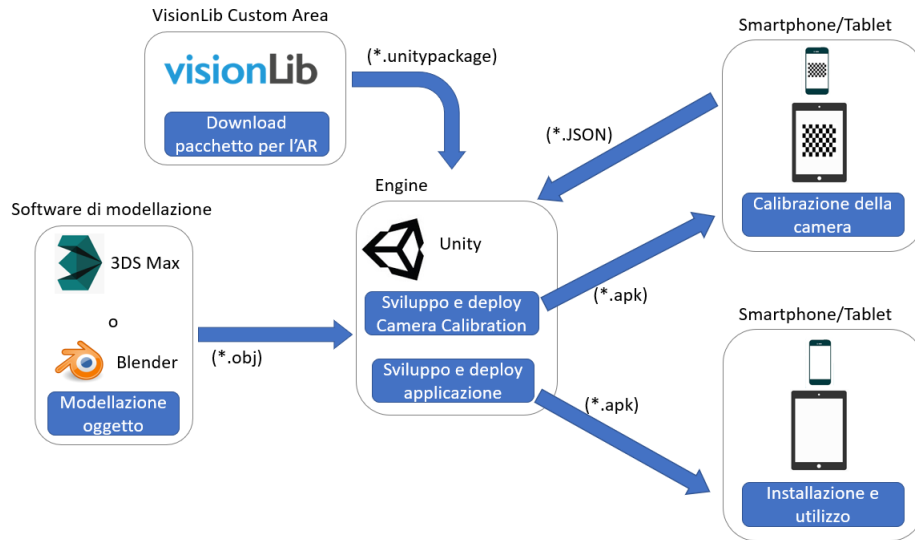


Figura 5.7: Architettura dell'applicazione

In questo caso è tutto centralizzato: non si hanno software esterni e tutto quello che serve è il pacchetto scaricabile dal sito ufficiale di VisionLib. Quest'ultimo al suo interno contiene tutti i prefab, script ed esempi che serviranno per lo sviluppo dell'applicazione. Dopo aver importato il pacchetto per l'AR e il modello su Unity bisognerà effettuare la calibrazione della camera del device. Per fare ciò basta andare ad aprire la scena "SimpleCameraCalibration" che si trova nella cartella Camera Calibration del pacchetto scaricato ed effettuare il deploy. Dopo aver installato nel proprio device il file \*.apk e seguito i passi indicati verrà prodotto un file \*.json. A questo punto si ha tutto quello che serve per sviluppare l'applicazione: basterà aggiungere il file prodotto e la licenza alla VLCamera come mostrato in figura 5.8. Si

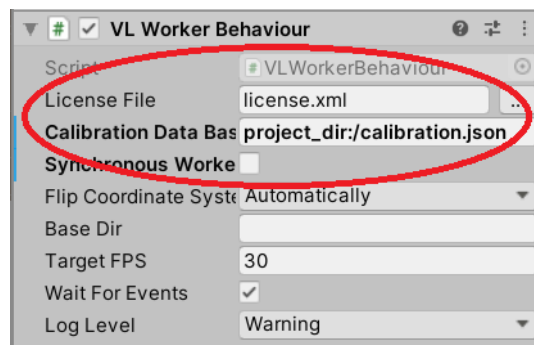


Figura 5.8: Dettagli VLCamera

dovranno infine inserire i parametri dell'oggetto da riconoscere nel file \*.vl e creare uno script che consenta a VLCamera di avere il path per andare a trovare tale file.

Si fa notare che, se non si è acquistata la licenza, esiste un limite agli utilizzi possibili che è cinque. Non resta che passare al deploy e all'installazione dell'applicazione su tablet o smartphone.

### 5.2.2 VisionLib: differenze con Vuforia

In questo paragrafo verranno trattati i vantaggi di VisionLib in relazione a Vuforia. Tra i vantaggi si hanno:

- Assenza di software esterni (con Vuforia si aveva Model Target Generator) quindi si poteva fare tutto su Unity.
- Si risparmia tempo: non bisogna fare il training del modello 3D

Ma ci sono anche degli svantaggi:

- Bisogna perdere un po' di tempo nella calibrazione del dispositivo ma poi posso inserire tutti i modelli che voglio senza dover eseguire altre calibrazioni.
- La precisione che si ottiene è paragonabile rispetto quella ottenuta con Vuforia però sembra che VisionLib sia più sensibile agli spostamenti dell'utente con la camera.
- Il corrispettivo della modalità 360° che si aveva con Vuforia è la Multiview, di cui esiste un beta e non una versione definitiva.
- Il settaggio della dimensione: prima veniva fornita una misura precisa (0.63 m x 2.1 m x 0.9 m) ora con VisionLib si può intervenire definendo l'ordine di grandezza (metri,centimetri, millimetri etc) e la scala dell'oggetto (meno intuitivo). Questo può portare a fare più tentativi prima di trovare una dimensione corretta per ottenere un riconoscimento dell'oggetto.

Vuforia ottiene risultati leggermente migliori a scapito della complessità e del tempo richiesto per l'elaborazione. Si noti che sia Vuforia che VisionLib nella versione gratuita pongono limiti di utilizzo, come mostra la tabella riassuntiva.

	Vuforia	VisionLib
Vantaggi	L'accuratezza raggiunge i centimetri	L'accuratezza raggiunge i centimetri
	Intuitivo	Intuitivo
	Modalità Advanced360	Non richiede il training del modello
Svantaggi	Mani occupate	Mani occupate
	Falsi positivi	Falsi positivi
	Richiede tempo per il training del modello	Richiede tempo nella calibrazione del dispositivo
	Limite sulla creazione dei Model Target (solo per la versione gratuita)	Limite sul numero di utilizzi dell'applicazione (solo per la versione gratuita)
		Beta per il riconoscimento multiplo di oggetti
		Sensibile agli spostamenti della camera

## 5.3 Risultati

Si mostrano ora alcune immagini che mettono in evidenza alcune limitazioni e pregi legate alle tecnologie proposte. I test effettuati consistono nel riconoscimento dell'armadio e del tubo come mostrato in figura 5.9.

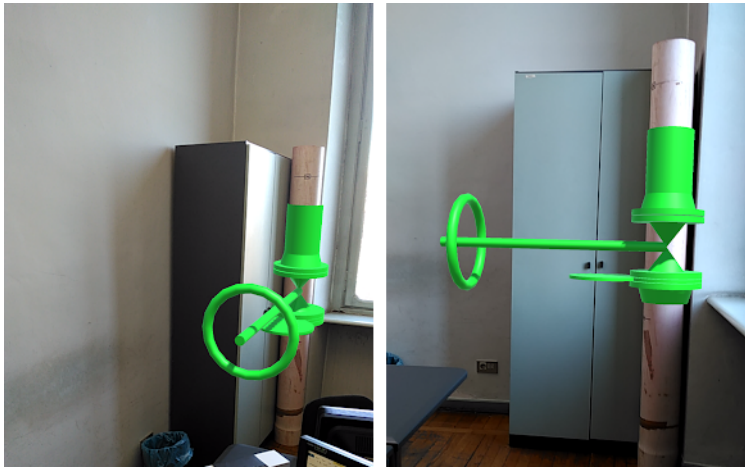


Figura 5.9: Risultati ottenuti

Bisogna immaginare che l'armadio sia una cabina che regoli la pressione dei tubi. Come si vede la precisione è molto buona: circa 1 centimetro di errore e utilizzando la modalità Advanced360 non si ha il vincolo di avere una vista predefinita. Da distanze ravvicinate però il tracking perde precisione e dopo una certa soglia viene perso: se ci si avvicina più di mezzo metro al target in questione si perde il tracking e si iniziano ad avere ologrammi instabili. Si è parlato delle limitazioni derivanti dall'analisi di modelli relativamente semplici e adesso si mostreranno i limiti come la generazione di falsi positivi. Si consideri il modello presentato prima avente l'opzione 1 view: ciò comporta una notevole riduzione dei tempi di training ma presenta

dei problemi. Come si vede dalle immagini 5.10, 5.11 e 5.12 l'algoritmo di training riconosce come target lo stesso armadio con il tubo spostato oppure l'armadio senza tubo e infine oggetti reali che lontanamente ricordano un armadio ma che non sono un armadio (finestra di dimensioni simili). Le soluzioni a questi problemi possono essere molteplici: aumentare la complessità del modello includendo oggetti reali, aggiungere dettagli agli oggetti reali esistenti oppure utilizzare la modalità di training più selettiva cioè la Advanced360. Purtroppo non è così semplice adottare le prime due soluzioni in quanto non è consentito andare a modificare la struttura dell'impianto e aggiungendo oggetti si rischia di avere un modello enorme per dimensioni fisiche: farne il training è possibile ma si rischia di non avere lo spazio per inquadrare l'intero target. Di fatto la soluzione migliore resta la modalità 360° con cui si evitano i casi di falsi positivi visti. Sarà cura dello sviluppatore quindi scegliere, ove possibile, modelli abbastanza diversi gli uni dagli altri per evitare di avere questo problema e se necessario utilizzare la modalità 360°. In questo caso si ha un modello molto semplice quindi è opportuno adottare tale strategia ma in casi in cui il modello sia molto complesso utilizzare la modalità Advanced360 potrebbe portare ad avere falsi negativi.

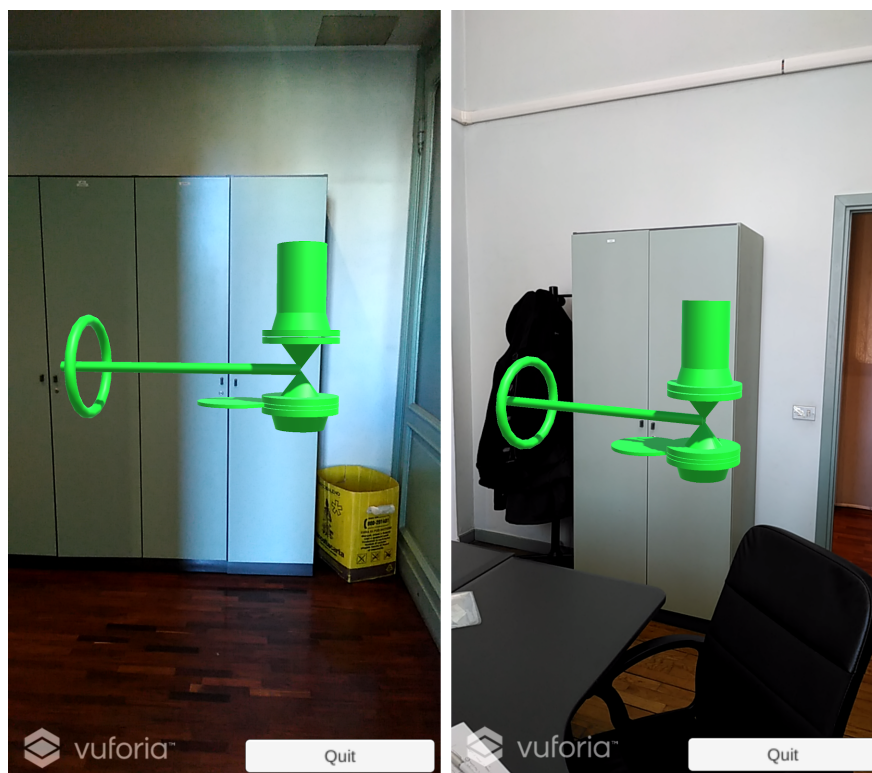


Figura 5.10: Esempi di falsi positivi



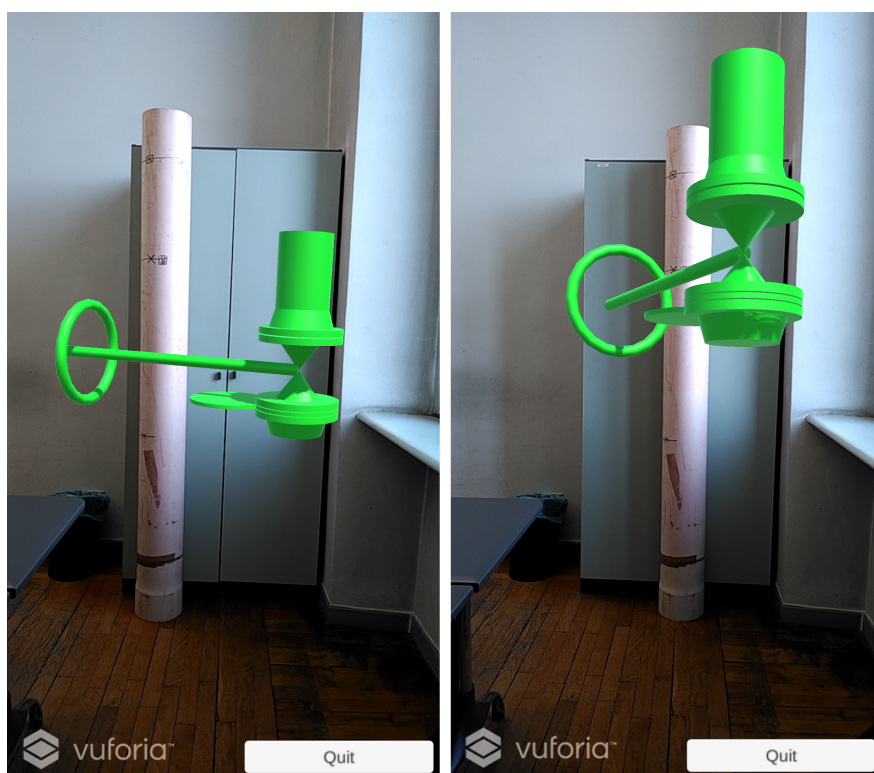


Figura 5.11: Esempi di falsi positivi

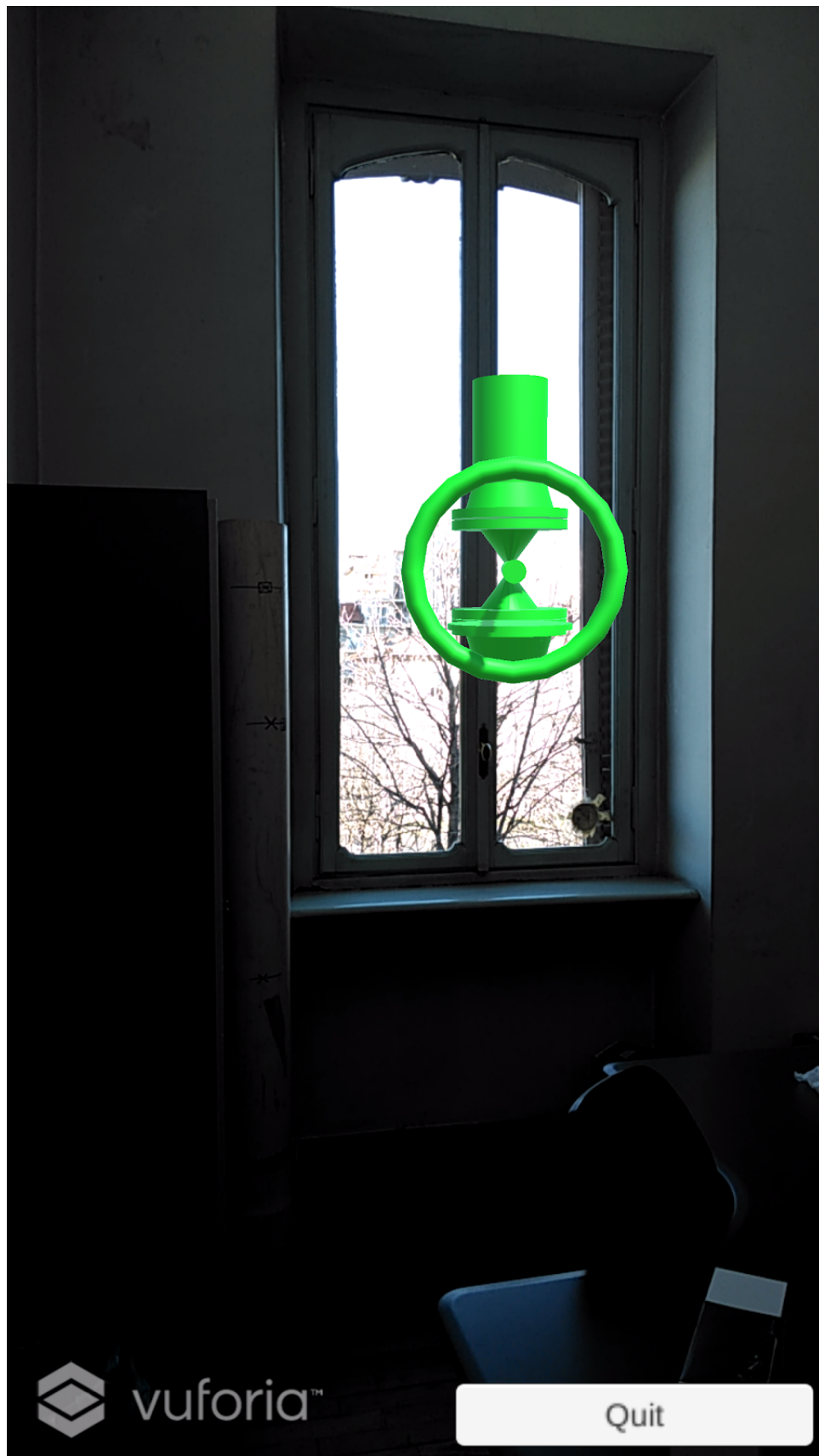


Figura 5.12: Esempio di falso positivo

# Capitolo 6

## Risultati

Per verificare la validità di un sistema è opportuno sottoporre utenti esterni all'utilizzo dell'applicazione realizzata. La prima parte di questo capitolo riguarda i test oggettivi cioè l'accuratezza raggiunta dalle varie soluzioni e la seconda riguarda i test soggettivi.

### 6.1 Test in fase di progettazione

Si sono visti nei capitoli precedenti degli indici di accuratezza delle tecnologie presentate. Si vuole definire in maniera precisa il metodo utilizzato per la misurazione e a cosa si riferiscono i valori visti nei capitoli precedenti. L'accuratezza di un dispositivo nello spazio tridimensionale, nel nostro caso specifico, può essere vista come la distanza tra posizione che un punto dovrebbe assumere e la posizione che effettivamente assume. Un esempio si ha nella figura 6.1: l'armadio si trova a una distanza  $d$  dalla posizione desiderata (punto A). In questo caso non si hanno informazioni

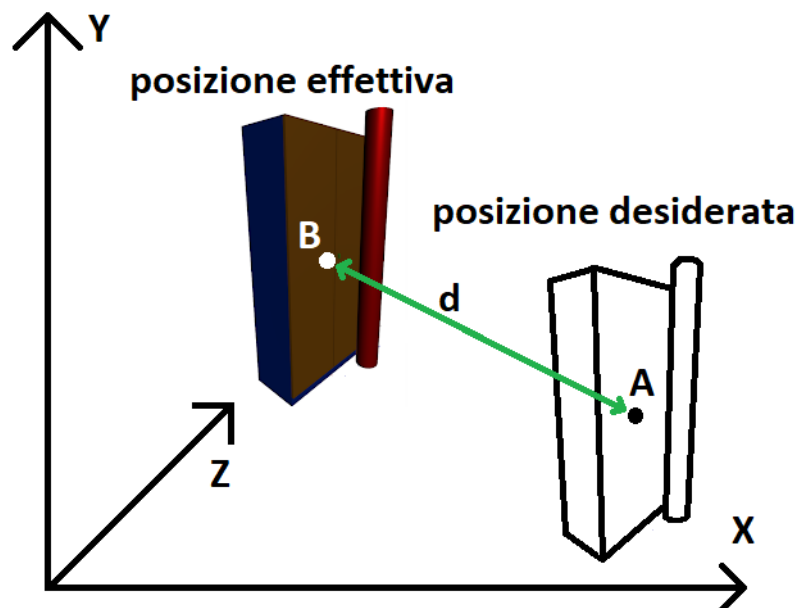


Figura 6.1: Calcolo della distanza

circa l'errore sul singolo asse. Il calcolo dell'accuratezza che si presenta però non

si riferisce alla distanza  $d$  ma si riferisce all'asse parallelo al tubo: nel caso degli Hololens il tubo è in posizione orizzontale invece nel caso di Vuforia e VisionLib è in posizione verticale. Nello specifico si sono fatti dei segni sul tubo nei punti in cui ci si aspetta appaia l'ologramma: la distanza è stata calcolata sulla base di questi punti. D'ora in avanti quando si parla di accuratezza dei dispositivi ci si riferisce a quella sopracitata. Va inoltre ricordato che i valori riportati hanno un errore massimo di un centimetro circa e questo è dovuto al fatto che non è possibile avvicinarsi troppo all'ologramma senza introdurre ulteriori spostamenti dello stesso. Non sono stati registrati però spostamenti visibilmente significativi negli altri assi. La prima tecnologia presentata è Hololens. In questo caso dare dei valori esatti è molto difficile poiché il dato dipende fortemente dalla capacità dell'utente di piazzare i punti nell'ambiente reale e dall'accuratezza dello spatial mapping. In ogni caso dopo vari tentativi e considerando un ambiente con illuminazione controllata l'accuratezza massima che si è raggiunta è di circa 1 centimetro. Per un utente esperto sono necessari 2 o 3 tentativi per raggiungere questo risultato. Infine va ricordato che l'ologramma della valvola che viene aggiunto resta stabile se l'utente non si avvicina troppo ad esso: tra mezzo metro e un metro si possono avere problemi di spostamento (2-4 centimetri). Per quanto riguarda Vuforia e VisionLib si hanno i seguenti risultati: l'accuratezza raggiunta è di 1,5 centimetri. Questo valore è causato da due fattori: l'errore che si commette quando si prendono le misure su oggetti reali e quella legata al dispositivo utilizzato (ad esempio versione del sistema operativo, risoluzione della fotocamera etc). Per quanto riguarda invece la mobilità dell'utente all'interno della scena si ha che tra un metro e mezzo metro l'ologramma inizia a tremare e si sposta dai 4 ai 20 centimetri. Se si ci si avvicina più di mezzo metro verso il target questo non viene più visualizzato. Il tutto è riassunto nella figura 6.2: nella parte sinistra si ha la distanza tra utente e target (in metri) e nella tabella a destra si ha l'accuratezza raggiunta dalle varie soluzioni (in centimetri). Va fatto notare che in questo paragrafo non si vuole fare il confronto tra Hololens (dispositivo hardware) e Vuforia o VisionLib (software): il confronto verrà fatto nel prossimo paragrafo tra Hololens e smatphone/tablet.

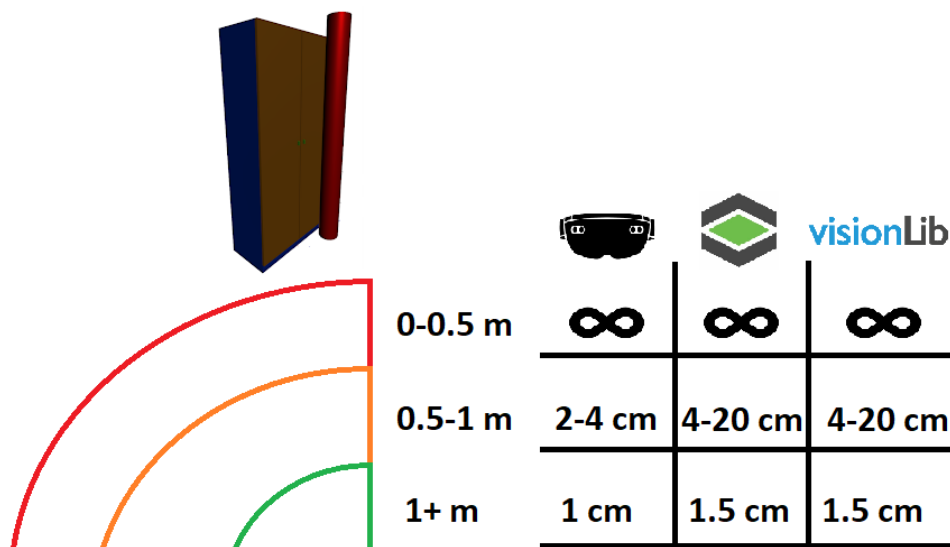


Figura 6.2: Risultati sull'accuratezza

## 6.2 Test soggettivi

Questo paragrafo riguarda i test soggettivi e si divide in 3 parti: la prima è dedicata ai test di usabilità, la seconda ai test SSQ e la terza si concentra su test qualitativi. I test di usabilità, come suggerisce la stessa parola, servono a valutare l'usabilità di una soluzione AR (cioè dispositivo e applicazione). Dato l'utilizzo di Hololens è opportuno verificare se l'utente accusa problemi fisici o psicofisici dopo l'utilizzo: per verificare ciò si utilizza il questionario SSQ. Infine vengono analizzate caratteristiche specifiche dell'applicazione come la facilità di utilizzare le gesture oppure la qualità delle immagini.

### 6.2.1 Scala di usabilità del sistema

Il SUS è un questionario che misura l'usabilità di un sistema e conta 10 affermazioni: per ognuna delle quali sarà possibile dare un voto (da 1 a 5). Un voto alto rappresenta quanto l'utente è d'accordo con la frase:

- 1: fortemente in disaccordo
- 2: disaccordo
- 3: non molto d'accordo
- 4: d'accordo
- 5: fortemente d'accordo

Di seguito vengono elencate le affermazioni:

1. Pensi che userai il sistema frequentemente
2. Ho trovato il sistema esageratamente complesso
3. Penso il sistema sia semplice da usare
4. Penso che sarà necessario il supporto di un tecnico per poter usare il sistema
5. Ho trovato le varie funzioni ben integrate nel sistema
6. Penso ci siano troppe inconsistenze nel sistema
7. Presumo che la maggior parte degli utenti imparerà facilmente il sistema
8. Trovo il sistema molto ingombrante
9. Mi sento molto sicuro nell'utilizzo del sistema
10. Dovrò imparare diverse cose prima di usare il sistema

Il punteggio per ogni dichiarazione viene calcolato come segue: le numero 1,3,5,7,9 danno un punteggio p-1 mentre le numero 2,4,6,8,10 danno punteggio 5-p. Ad esempio se all'affermazione 1 si ottiene p=3 il punteggio sarà  $(3 - 1 = 2)$  due invece se nella 2 si ottiene p=4 il punteggio sarà  $(5 - 4 = 1)$  uno. Il massimo punteggio ottenibile è 40, questo risultato verrà moltiplicato per 2,5 (il massimo è dunque su scala

100). Se il punteggio è maggiore di 68 allora il sistema risulta accettabile. Ogni utente ha provato sia l'applicazione per Hololens che per smartphone e per ognuna ha compilato il questionario SUS. L'ordine delle prove è stato alternato: se il primo utente aveva l'ordine Hololens e tablet il secondo aveva tablet e Hololens e così via. Nelle tabelle 6.3 e 6.4 vengono mostrate le risposte anonime che ogni utente ha dato. Nelle ultime righe si trova la media, il punteggio e il totale secondo le regole viste in questo paragrafo. Gli stessi dati possono essere visti nei grafici 6.5 e 6.6. La stessa identica cosa può essere fatta per Hololens: 6.7, 6.8, 6.9 e 6.10. Si nota che il punteggio totale dei due sistemi è sopra la soglia di usabilità ma tra i due c'è un distacco di 10 punti: lo smartphone/tablet sembra essere molto più usabile di un Hololens. Bisogna porre maggiore attenzione a:

- Presumo che la maggior parte degli utenti imparerà facilmente il sistema
- Trovo il sistema molto ingombrante
- Mi sento molto sicuro nell'utilizzo del sistema

In questi casi la differenza di punteggio è compresa tra 0,5 e 1 pt. In una scala da 1 a 5 questa differenza è significativa. Da notare anche le dichiarazioni simili a queste:

- Penso il sistema sia semplice da usare
- Penso che sarà necessario il supporto di un tecnico per poter usare il sistema

Di fatto la difficoltà legata all'utilizzo Hololens e il suo ingombro hanno portato a questa differenza di punteggio totale. I motivi legati a questi score sono molteplici: come visto nel capitolo 3 Hololens ha un FOV ristretto e questo costringe gli utenti che non hanno mai usato tale dispositivo a muovere costantemente la testa per poter inquadrare il target e piazzare i punti. A questo problema va aggiunto anche il sistema di input di Hololens: le gesture non sono affatto intuitive e vengono ampiamente utilizzate nell'applicazione. Una scarsa padronanza delle gesture di Hololens porta a ripetere più volte il processo di piazzamento dei punti. Dover ripetere più volte la stessa procedura oltre ad essere fastidioso può risultare anche faticoso. Nel paragrafo successivo verranno mostrati i risultati legati al questionario SSQ, da cui si evince un leggero incremento della stanchezza: a conferma di quanto detto. Nel caso di smartphone/tablet invece il sistema di input risulta molto intuitivo: l'utente dovrà inquadrare con la fotocamera il target e al resto pensa l'applicazione. Questa procedura è risultata particolarmente facile da capire e svolgere se paragonata all'applicazione Hololens. I tempi per testare le applicazioni risultano minori per tablet/smartphone. Gli score legati alla semplicità di utilizzo hanno favorito lo smartphone/tablet (come ad esempio la numero 9. "Mi sento molto sicuro nell'utilizzo del sistema" oppure la numero 3. "Penso il sistema sia semplice da usare"). Altri aspetti che avrebbero potuto influenzare il punteggio sono gli occhiali: solo uno degli utenti sottoposti al test li aveva. Per smartphone e tablet il portare occhiali non risulta un problema. La durata della batteria non è stata presa in considerazione in quanto la durata del test non supera l'autonomia dei device. Un aspetto da non sottovalutare è l'età media degli utenti sottoposti al test: circa 30 anni. Essere giovani può influenzare la valutazione in senso positivo: un ragazzo riesce ad assimilare più velocemente l'utilizzo di nuove tecnologie. Se l'età media fosse stata più alta probabilmente la differenza sarebbe stata maggiore. Va ricordato che

il test è stato sottoposto anche a utenti che hanno familiarità con Hololens quindi il punteggio relativo alla semplicità legata all'utilizzo potrebbe essere diverso se si avessero solo utenti che non hanno mai usato tale tecnologia. Diverso è il caso per tablet/smartphone: è veramente difficile trovare qualcuno che non abbia mai usato tale dispositivo. Lo svantaggio principale del tablet/smartphone è quello di avere le mani occupate: tale problema però è stato compensato da altrettanti vantaggi che, come detto in questo paragrafo, hanno portato i risultati visti. Per concludere va fatto notare che la preferenza di usabilità verso la soluzione smartphone/tablet è anche dovuta al fatto che l'attività da svolgere non richiedeva l'impegno delle mani da parte dell'utente. Nel caso fosse stato necessario operare con entrambi le mani, probabilmente il giudizio sugli Hololens sarebbe stato migliore.

	Penso che userò il sistema frequentemente	Ho trovato il sistema esageratamente complesso	Penso il sistema sia semplice da usare	Penso che sarà necessario il supporto di un tecnico per poter usare il sistema	Ho trovato le varie funzioni ben integrate nel sistema
<b>Persona 1</b>	3	1	5	1	5
<b>Persona 2</b>	3	1	4	2	3
<b>Persona 3</b>	4	1	5	1	5
<b>Persona 4</b>	4	1	5	1	5
<b>Persona 5</b>	3	2	4	2	4
<b>Persona 6</b>	3	2	4	1	3
<b>Persona 7</b>	4	1	5	1	5
<b>Persona 8</b>	5	1	5	1	4
<b>Persona 9</b>	5	1	5	1	5
<b>Persona 10</b>	3	2	4	2	3
<b>Persona 11</b>	4	1	5	1	4
<b>Persona 12</b>	3	2	5	2	4
<b>Media</b>	<b>3,667</b>	<b>1,333</b>	<b>4,667</b>	<b>1,333</b>	<b>4,167</b>
<b>Punteggio</b>	<b>2,667</b>	<b>3,667</b>	<b>3,667</b>	<b>3,667</b>	<b>3,167</b>

Figura 6.3: Score affermazioni 1 - 5 questionario SUS per Tablet

## 6.2. TEST SOGGETTIVI

	Penso ci siano troppe inconsistenze nel sistema	Presumo che la maggior parte degli utenti imparerà facilmente il sistema	Trovo il sistema molto ingombrante	Mi sento molto sicuro nell'utilizzo del sistema	Dovrò imparare diverse cose prima di usare il sistema
Persona 1	1	4	1	4	2
Persona 2	3	4	1	4	1
Persona 3	4	3	2	4	2
Persona 4	1	5	1	5	1
Persona 5	2	5	1	5	1
Persona 6	3	4	2	4	2
Persona 7	1	5	1	5	1
Persona 8	2	4	1	5	2
Persona 9	1	5	1	4	1
Persona 10	2	4	1	5	1
Persona 11	2	5	1	5	1
Persona 12	2	5	1	5	1
Media	2	4,417	1,167	4,583	1,333
Punteggio	3	3,417	3,833	3,583	3,667
Totale	85,8375				

Figura 6.4: Score affermazioni 6 - 10 questionario SUS e score finale per Tablet

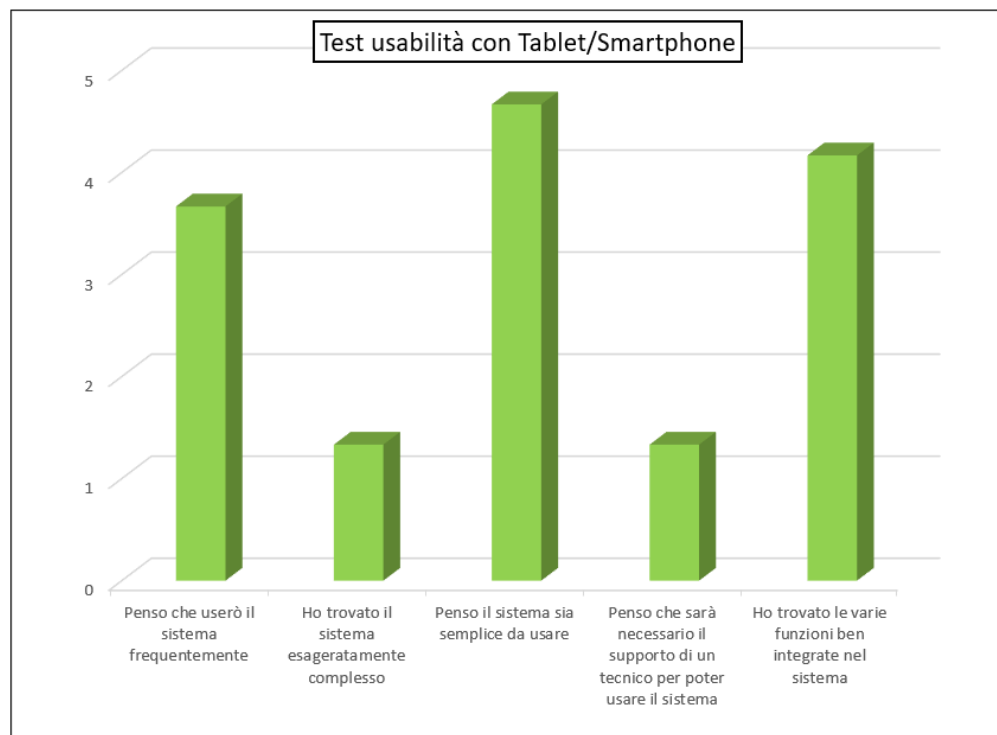


Figura 6.5: Grafico 1 con i risultati del test di usabilità per Tablet



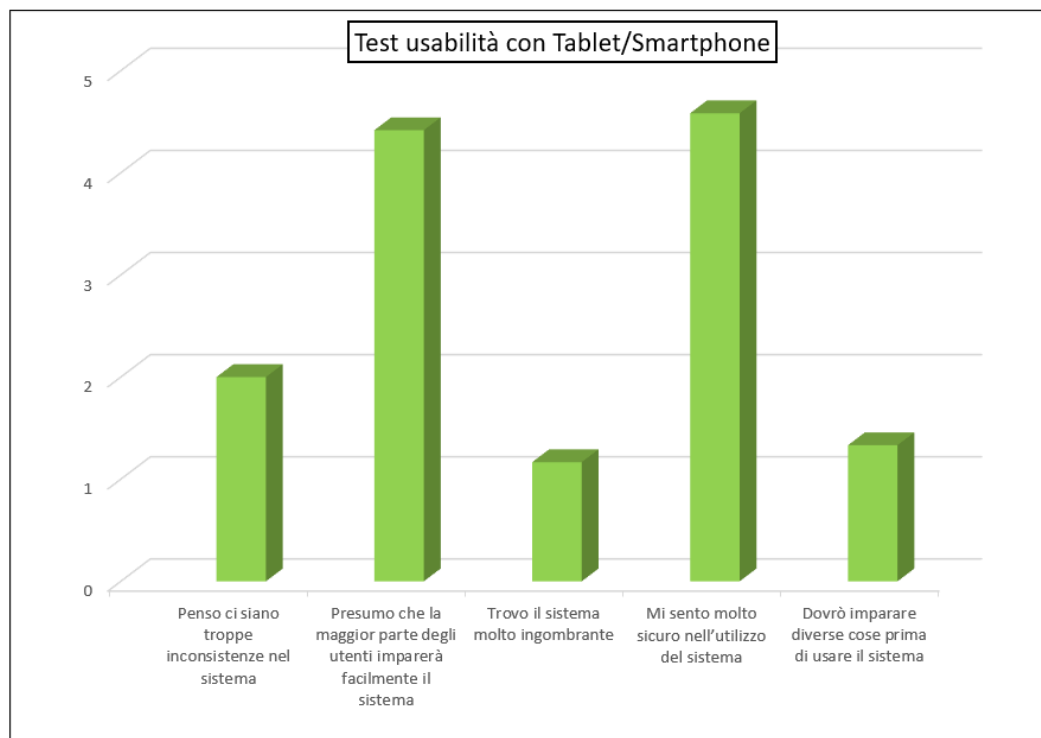


Figura 6.6: Grafico 2 con i risultati del test di usabilità per Tablet

	Penso che userò il sistema frequentemente	Ho trovato il sistema esageratamente complesso	Penso il sistema sia semplice da usare	Penso che sarà necessario il supporto di un tecnico per poter usare il sistema	Ho trovato le varie funzioni ben integrate nel sistema
<b>Persona 1</b>	4	2	4	4	4
<b>Persona 2</b>	3	2	4	2	4
<b>Persona 3</b>	4	1	5	1	5
<b>Persona 4</b>	4	1	4	1	5
<b>Persona 5</b>	3	2	4	1	4
<b>Persona 6</b>	4	1	4	1	5
<b>Persona 7</b>	1	1	5	1	3
<b>Persona 8</b>	4	1	3	1	5
<b>Persona 9</b>	4	2	4	1	4
<b>Persona 10</b>	3	1	4	3	5
<b>Persona 11</b>	2	1	4	4	4
<b>Persona 12</b>	5	2	5	2	4
<b>Media</b>	<b>3,417</b>	<b>1,417</b>	<b>4,167</b>	<b>1,833</b>	<b>4,333</b>
<b>Punteggio</b>	<b>2,417</b>	<b>3,583</b>	<b>3,167</b>	<b>3,167</b>	<b>3,333</b>

Figura 6.7: Score affermazioni 1 - 5 questionario SUS per Hololens

## 6.2. TEST SOGGETTIVI

	Penso ci siano troppe inconsistenze nel sistema	Presumo che la maggior parte degli utenti imparerà facilmente il sistema	Trovo il sistema molto ingombrante	Mi sento molto sicuro nell'utilizzo del sistema	Dovrò imparare diverse cose prima di usare il sistema
Persona 1	2	3	3	3	1
Persona 2	2	3	2	5	1
Persona 3	1	4	3	4	2
Persona 4	1	4	2	4	1
Persona 5	1	5	2	3	3
Persona 6	1	5	3	3	3
Persona 7	3	4	3	5	1
Persona 8	1	5	2	5	1
Persona 9	1	4	3	4	2
Persona 10	2	3	3	3	1
Persona 11	1	4	1	4	3
Persona 12	2	2	4	2	1
Media	1,5	3,833	2,583	3,75	1,667
Punteggio	3,5	2,833	2,417	2,75	3,333
Totale	76,25				

Figura 6.8: Score affermazioni 6 - 10 questionario SUS e score finale per Hololens

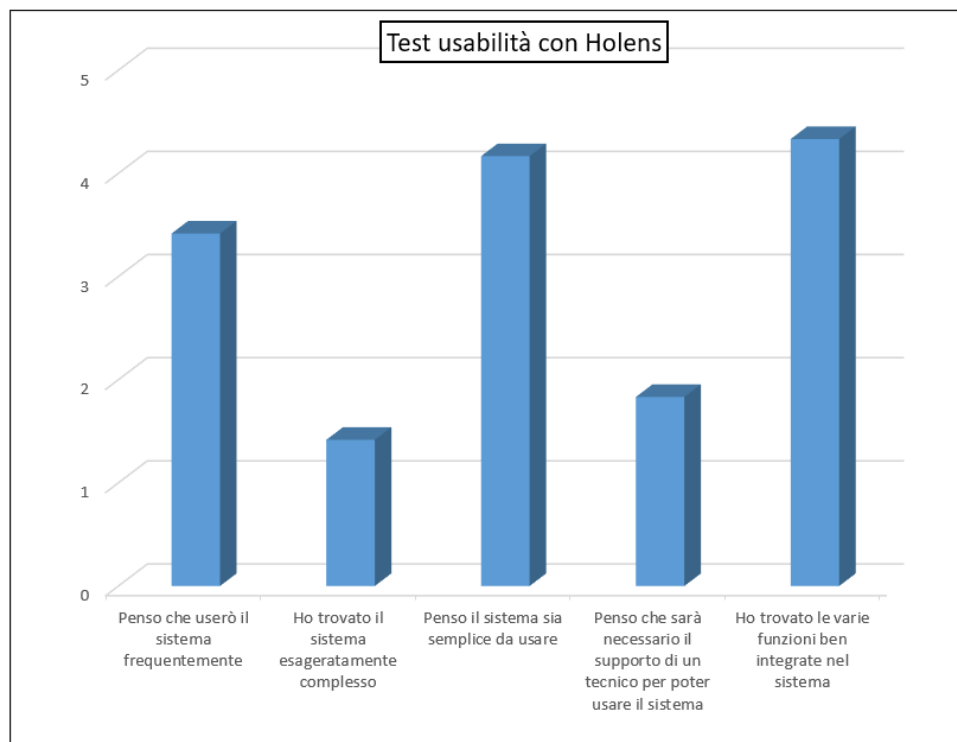


Figura 6.9: Grafico 1 con i risultati del test di usabilità per Hololens

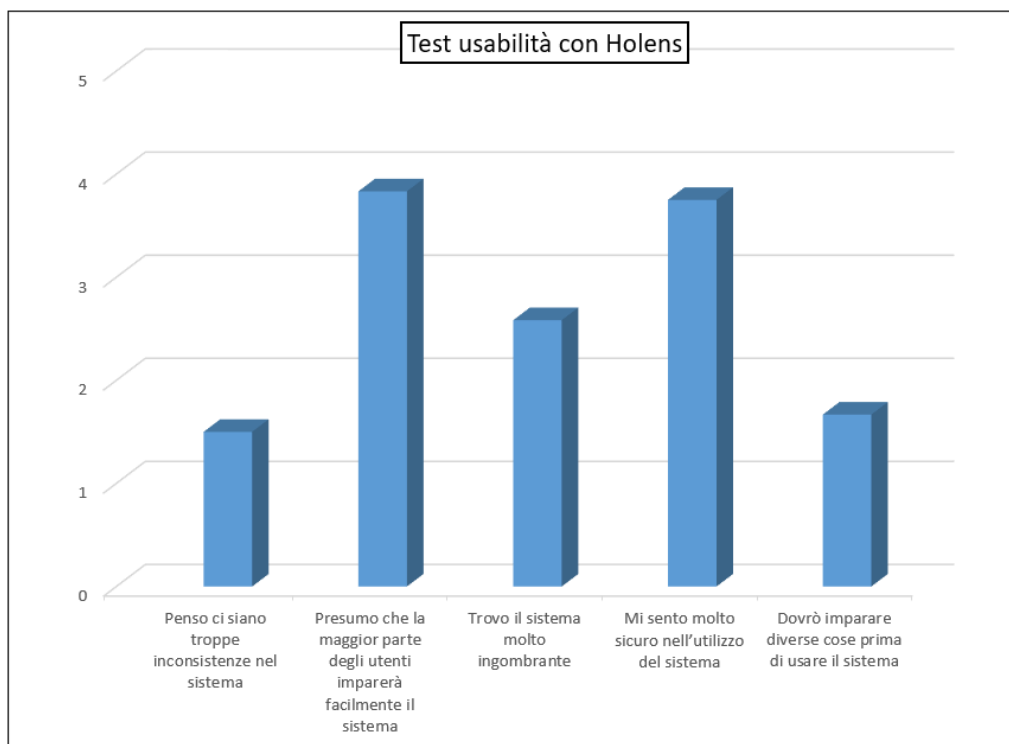


Figura 6.10: Grafico 2 con i risultati del test di usabilità per Hololens

### 6.2.2 SSQ

Un altro questionario molto importante da fare è l'SSQ per misurare la “cyber sickness” cioè le condizioni fisiche e soprattutto psichiche del soggetto. Non ha senso farlo per un tablet o smartphone (dato che ormai la stragrande maggioranza li utilizza abitualmente senza accusare forti problemi) ma ha senso per dispositivi come Hololens:

- Sconforto generico
- Stanchezza
- Mal di testa
- Affaticamento degli occhi
- Difficoltà di messa a fuoco
- Eccessiva salivazione
- Sudorazione
- Nausea
- Difficoltà a concentrarsi
- Visione offuscata
- Vertigini con occhi aperti
- Vertigini con occhi chiusi
- Mal di stomaco

Tale test verrà proposto prima e dopo l'utilizzo dell'applicazione: per assicurarsi che un ipotetico dolore dipenda effettivamente dal dispositivo e non da un malore già esistente. Come si nota non si hanno grossi problemi eccetto per il senso di stanchezza avvertito.

	Sconforto generico	Stanchezza	Mal di testa	Affaticamento degli occhi	Difficoltà di messa a fuoco	Eccessiva salivazione	Sudorazione
Persona 1	0	0	0	0	0	0	0
Persona 2	0	0	1	1	0	0	0
Persona 3	0	1	0	1	0	0	0
Persona 4	0	2	0	1	1	0	0
Persona 5	0	0	0	1	1	0	0
Persona 6	0	1	0	0	0	0	0
Persona 7	0	0	0	1	1	0	0
Persona 8	0	1	0	1	0	0	0
Persona 9	0	1	1	0	0	0	0
Persona 10	0	0	1	0	0	0	0
Persona 11	0	1	0	1	0	0	0
Persona 12	0	0	0	1	0	0	0
Media	0	0,583	0,25	0,667	0,25	0	0

Figura 6.11: Pre test 1 SSQ

## 6.2. TEST SOGGETTIVI

	Nausea	Difficoltà a concentrarsi	Visione offuscata	Vertigini con occhi aperti	Vertigini con occhi chiusi	Mal di stomaco
Persona 1	0	0	0	0	0	0
Persona 2	0	0	0	0	0	0
Persona 3	0	0	0	0	0	0
Persona 4	0	1	0	0	0	0
Persona 5	0	0	1	0	0	0
Persona 6	0	0	0	0	0	0
Persona 7	0	0	0	0	0	0
Persona 8	0	0	0	0	0	0
Persona 9	0	1	0	0	0	0
Persona 10	0	1	0	0	0	0
Persona 11	0	0	0	0	0	0
Persona 12	0	1	0	0	0	0
Media	0	0,333	0,083	0	0	0

Figura 6.12: Pre test 2 SSQ

	Sconforto generico	Stanchezza	Mal di testa	Affaticamento degli occhi	Difficoltà di messa a fuoco	Eccessiva salivazione	Sudorazione
Persona 1	0	1	0	0	1	0	0
Persona 2	1	1	1	1	0	0	1
Persona 3	0	1	0	0	0	0	0
Persona 4	0	1	0	1	1	0	0
Persona 5	0	0	0	1	0	0	1
Persona 6	0	1	0	1	0	0	0
Persona 7	0	1	0	2	1	0	0
Persona 8	0	2	1	1	1	0	0
Persona 9	0	2	2	0	0	0	0
Persona 10	1	0	1	1	1	0	0
Persona 11	0	2	0	1	0	0	1
Persona 12	0	1	1	2	0	0	0
Media	0,167	1,083	0,5	0,917	0,417	0	0,25

Figura 6.13: Test 1 SSQ

	Nausea	Difficoltà a concentrarsi	Visione offuscata	Vertigini con occhi aperti	Vertigini con occhi chiusi	Mal di stomaco
Persona 1	0	1	0	0	0	0
Persona 2	0	0	0	0	0	0
Persona 3	0	0	0	0	0	0
Persona 4	0	1	0	0	0	0
Persona 5	0	0	0	0	0	0
Persona 6	0	0	0	0	0	0
Persona 7	0	1	1	0	0	0
Persona 8	0	0	1	1	1	0
Persona 9	0	1	1	0	0	0
Persona 10	0	1	0	1	1	0
Persona 11	0	1	0	0	0	0
Persona 12	0	1	0	1	0	0
Media	0	0,583	0,25	0,25	0,167	0

Figura 6.14: Pre Test 2 SSQ

### 6.2.3 VRUSE e HAR

Alle affermazioni viste nei paragrafi precedenti è opportuno aggiungere qualcosa. Si è preso spunto dai questionari VRUSE e HAR:

- L'effetto di flickering è molto presente
- Le parole o simboli mostrati nel display sono facili da leggere
- Ho trovato il dispositivo di input facile da usare (gesture e bottoni)
- Ho trovato facile piazzare gli oggetti nell'ambiente (Hololens)
- Il campo visivo (FOV) di visualizzazione era appropriato per le procedure
- Quando spostavo la testa la qualità dell'immagine era accettabile

Anche in questo caso si ha un punteggio da 1 a 5 e valgono le regole dette prima: 1 per indicare forte disaccordo e 5 per indicare che si è fortemente in accordo. I dati confermano ed estendono i risultati del test di usabilità: in particolare anche gli utenti esperti hanno ritenuto la funzionalità di piazzamento dei punti non molto facile. Inoltre il FOV di Hololens sembra non essere adeguato. Va comunque ricordato che le prestazioni di Hololens sembrano essere superiori per quanto riguarda la stabilità dell'immagine in caso di movimento dell'utente. Questo pregio però non compensa i difetti: se per le prestazioni si ha un punteggio di mezzo punto a favore di Hololens si ha un punteggio di un punto a favore del sistema con tablet/smartphone per quanto riguarda la semplicità. Di seguito vengono mostrati rispettivamente le tabelle 6.15 e 6.16 con i risultati del test qualitativo per Hololens e tablet/smatphone. A seguire si hanno i grafici 6.17 e 6.18.

	L'effetto di flickering è molto presente	Le parole o simboli mostrati nel display sono facili da leggere	Ho trovato il dispositivo di input facile da usare (gesture o bottoni)	Ho trovato facile piazzare gli oggetti nell'ambiente	Il campo visivo (FOV) di visualizzazione è appropriato per le procedure	Quando mi sposto la qualità dell'immagine è accettabile
Persona 1	2	3	3	3	5	4
Persona 2	2	3	4	3	3	4
Persona 3	2	4	5	3	4	4
Persona 4	1	5	4	4	3	4
Persona 5	3	5	5	3	3	4
Persona 6	3	5	5	5	5	5
Persona 7	2	5	5	5	2	4
Persona 8	1	5	5	2	2	4
Persona 9	2	5	5	2	2	4
Persona 10	2	4	4	3	3	5
Persona 11	3	3	3	4	4	4
Persona 12	2	3	3	5	5	5
Media	2,083	4,167	4,25	3,5	3,417	4,25

Figura 6.15: Tabella con i risultati del test qualitativo per Hololens

## 6.2. TEST SOGGETTIVI

	L'effetto di flickering è molto presente	Le parole o simboli mostrati nel display sono facili da leggere	Ho trovato il dispositivo di input facile da usare (gesture o bottoni)	Ho trovato facile piazzare gli oggetti nell'ambiente	Il campo visivo (FOV) di visualizzazione è appropriato per le procedure	Quando mi sposto la qualità dell'immagine è accettabile
Persona 1	2	5	5	5	4	5
Persona 2	2	5	5	5	4	4
Persona 3	3	5	5	4	4	5
Persona 4	3	5	5	5	5	5
Persona 5	2	4	4	4	4	4
Persona 6	2	4	4	5	4	3
Persona 7	2	5	5	4	3	3
Persona 8	3	4	5	5	4	3
Persona 9	2	5	4	4	3	3
Persona 10	3	5	5	5	4	4
Persona 11	2	5	5	5	4	4
Persona 12	3	4	4	4	4	3
Media	2,417	4,667	4,667	4,583	3,917	3,833

Figura 6.16: Tabella con i risultati del test qualitativo per tablet/smartphone

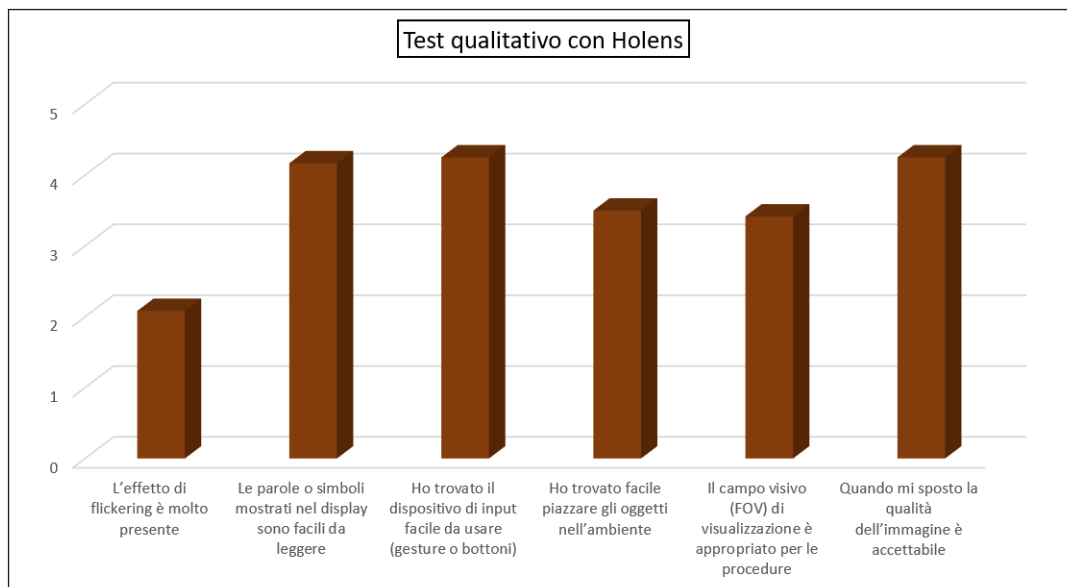


Figura 6.17: Grafico con i risultati del test qualitativo per Hololens

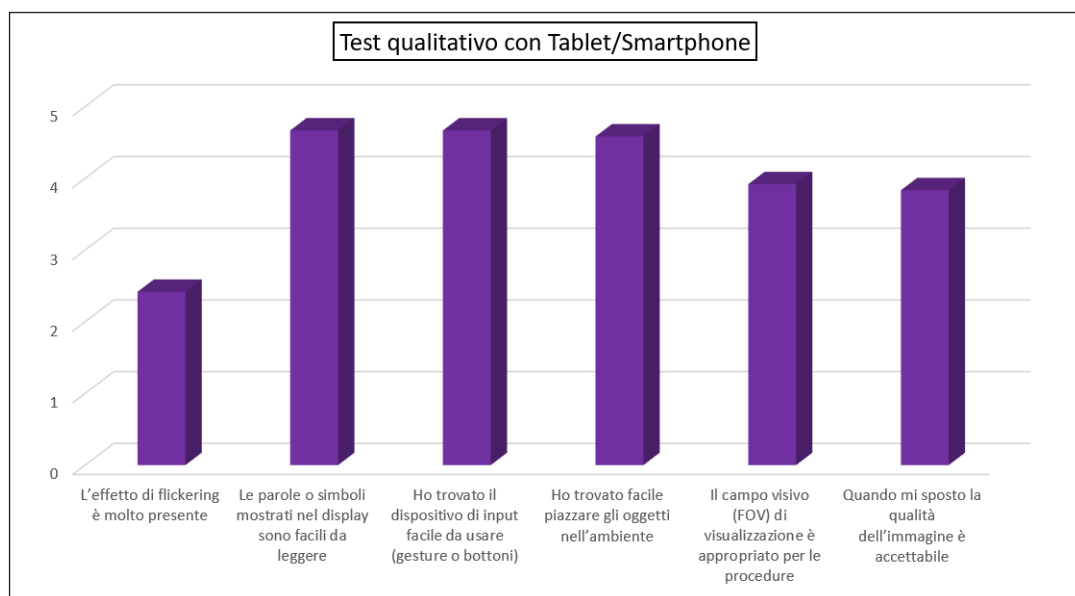


Figura 6.18: Grafico con i risultati del test qualitativo per tablet/smartphone



# Capitolo 7

## Conclusioni e sviluppi futuri

Tutto il lavoro svolto si basa principalmente su soluzioni che prevedono l'utilizzo di cellulari e tablet android e Hololens e sull'analisi di vantaggi e svantaggi delle soluzioni sopra citate. Per quanto riguarda gli Hololens un problema riguarda l'usabilità, utenti che non hanno molta esperienza difficilmente riusciranno a ottenere risultati ottimali se non dopo un paio di tentativi: ma questo potrebbe non essere vero in futuro se gli Hololens diventeranno dispositivi di uso comune (come è stato per gli smartphone). L'utilizzo invece di un'applicazione basata su tablet o cellulare richiede che l'utente debba inquadrare un oggetto: ciò risulta essere facile e intuitivo. I risultati ottenuti in termini di precisione però, se si presuppone un utente che sa usare gli Hololens, sono a favore di quest'ultima soluzione. La precisione assoluta in ogni caso è impossibile da raggiungere in entrambi i casi, almeno con gli strumenti attuali. Un altro aspetto da non sottovalutare è il confort o meglio l'ingombro del dispositivo: il cellulare è molto leggero da tenere in una mano o in una tasca, gli Hololens invece potrebbero risultare più fastidiosi e pesanti[4]. Shaharam Eivazi, Thiago Santini, Thomas C. Kübler e Enkelejd Kasneci dell'università di Tübingen hanno proposto una soluzione innovativa per quanto riguarda il confort degli occhialini (head mounted eye trackers). Puntato molto sulla leggerezza del dispositivo e anche su un costo accessibile: hanno infatti realizzato varianti dello stesso modello utilizzando hardware con costi differenti cercando di offrire un buon trade off tra prestazioni e prezzo. Si stanno conducendo studi sul tracking oculare: il vantaggio si traduce in una riduzione di movimenti della testa dell'utente [1][7][9]. Un problema che viene sollevato in questa pubblicazione riguarda il basso FOV, cioè il campo visivo del dispositivo: problema che sicuramente può essere fastidioso e può costringere l'utente a muovere continuamente la testa per rivolgersi verso il target nello spazio. La stessa cosa riguarda gli Hololens: servirebbe un FOV maggiore per dare all'operatore un'esperienza migliore. L'accuratezza è un problema per gli Hololens e in particolare il processo di spatial mapping. Si è visto come determinati oggetti non vengano completamente visti e altri (come ad esempio gli oggetti con molti dettagli) invece subiscono approssimazioni troppo grossolane. Il FOV basso e la scarsa accuratezza dello spatial mapping degli Hololens di prima generazione possono essere superati da quelli di seconda generazione: si potrebbe sviluppare la stessa applicazione per Hololens di seconda generazione e vedere se effettivamente si ha un miglioramento significativo. In questo senso si dovrebbe pensare a un sistema di misurazione della precisione migliore. Come visto nel precedente capitolo non è facile calcolare il posizionamento di un oggetto nella scena rispetto i 3 assi: una

---

possibile espansione potrebbe riguardare proprio questo aspetto. Questo vale per tutte e tre le tecnologie. Le sfide però non mancano per gli smatphone e tablet: si è visto come la precisione che si raggiunge nei dispositivi cellulari e tablet ha diversi vincoli e determinati ambienti risultano altamente critici. Fare training di oggetti poco complessi può portare ad avere falsi positivi. Un altro fattore riguarda le viste e VisionLib ha solo una beta per il momento che consente di sviluppare applicazioni con più viste: uno sviluppo futuro potrebbe essere quello di ampliare l'applicazione VisionLib con questa funzionalità. Anche la creazione dei modelli è fondamentale e in particolare nella fase di misurazione degli oggetti reali e di modellazione: utilizzare strumenti più precisi rispetto il classico metro potrebbe aiutare a rendere più robusta l'intera applicazione. Un fattore da non sottovalutare però è la possibilità di avere le mani libere durante il proprio task e in questo purtroppo non si potrà trovare una soluzione amenochè si ripensi all'intera struttura del dispositivo. Va ricordato che richiedere capacità computazionali maggiori e aumentare il comfort sono strategie che vanno in direzioni opposte: l'obiettivo è quindi raggiungere un trade-off più conveniente di quello attuale. Nonostante le sfide aperte e le limitazioni viste l'impiego della realtà aumentata a supporto dell'industria ha portato vantaggi.

# Bibliografia

- [1] Haofei Wang et al. *SLAM-based Localization of 3D Gaze using a Mobile Eye Tracker*. 2018. URL: <https://dl.acm.org/doi/10.1145/3204493.3204584>.
- [2] Nadia Zenati-Henda et al. *Augmented reality system based on hand gestures for remote maintenance*. 2014. URL: <https://ieeexplore.ieee.org/document/6911258>.
- [3] Sebastian Pimminger et al. *Low-Cost Tracking of Assembly Tasks in Industrial Environments*. 2019. URL: <https://dl.acm.org/doi/10.1145/3316782.3321526>.
- [4] Shaharam Eivazi et al. *An Inconspicuous and Modular Head-Mounted Eye Tracker*. 2018. URL: <https://dl.acm.org/doi/10.1145/3204493.3208345>.
- [5] Susanna Aromaa et al. *Use of Wearable and Augmented Reality Technologies in Industrial Maintenance Work*. 2016. URL: <https://dl.acm.org/doi/10.1145/2994310.2994321>.
- [6] Thiago Santini et al. *The Art of Pervasive Eye Tracking*. 2018. URL: <https://dl.acm.org/doi/10.1145/3208031.3208032>.
- [7] Frank H. Borsato e Carlos H. Morimoto. *Episcleral surface tracking: challenges and possibilities for using mice sensors for wearable eye tracking*. 2016. URL: <https://dl.acm.org/doi/10.1145/2857491.2857496>.
- [8] T. Caudell e D. Mizell. *Augmented reality: An application of heads-up display technology to manual manufacturing processes*. 1992. URL: <http://ieeexplore.ieee.org/document/183317/>.
- [9] Aayush K. Chaudhary. *Motion Tracking of Iris Features for Eye Tracking*. 2019. URL: <https://dl.acm.org/doi/10.1145/3314111.3322872>.
- [10] I-Chen Lin e Ming-Han Tsai Li-Chen Wu. *Augmented Reality Instruction for Object Assembly based on Markerless Tracking*. 2016. URL: <https://dl.acm.org/doi/10.1145/2856400.2856416>.
- [11] Marc Erich Latoschik e Klaus Schilling Doris Aschenbrenner. *Industrial Maintenance with Augmented Reality: Two Case Studies*. 2016. URL: <https://dl.acm.org/doi/10.1145/2993369.2996305>.
- [12] Sheng Zhang e He Xu Mengyun Yi. *Real-Time Online Multi-Object Tracking: A Joint Detection and Tracking Framework*. 2019. URL: <https://dl.acm.org/doi/10.1145/3374587.3374628>.
- [13] B. MacIntyre e D. Seligmann S. Feiner. *Knowledge-based augmented reality*. 1993. URL: <https://dl.acm.org/citation.cfm?id=159587>.

- [14] Oduor Andrew Orlando e Sandor Markon. *THE USE OF AUGMENTED REALITY IN THE MAINTENANCE OF MECHANICAL OBJECTS*. 2017. URL: <https://ieeexplore-ieee-org.ezproxy.biblio.polito.it/document/7988565>.
- [15] Steven J. Henderson e Steven Feiner. *Evaluating the Benefits of Augmented Reality for Task Localization in Maintenance of an Armored Personnel Carrier Turret*. 2009. URL: <http://ieeexplore.ieee.org/document/5336486/>.