ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Scienze
Corso di Laurea in Ingegneria e Scienze Informatiche

STUDIO E APPLICAZIONE DELLA PIATTAFORMA ARCORE PER LO SVILUPPO DI SISTEMI DI REALTÀ AUMENTATA: ROCCA DELLE CAMINATE COME CASO DI STUDIO

Elaborato in
SISTEMI EMBEDDED E INTERNET OF THINGS

Relatore
Prof. ALESSANDRO RICCI

Co-relatore
Ing. ANGELO CROATTI

Presentata da
MATTEO ANDRUCCIOLI

Seconda Sessione di Laurea
Anno Accademico 2017 – 2018

PAROLE CHIAVE

Augmented Reality

Mixed Reality

Google ARCore

Unity

"Tonight we'll stand, get off our knees
Fight for what we've worked for all these years
And the battle was long, it's the fight of our lives
But we'll stand up champions tonight
It was the night things changed
Can you see it now?
These walls that they put up to hold us back fell down
This revolution, throw your hands up
'Cause we never gave in
And we'll sing hallelujah, we sang hallelujah"
-TS

Indice

Introduzione

La tesi si pone come obiettivo quello di indagare l'ambito della mixed reality per dispositivi mobili Android, concentrandosi sull'sdk fornito da Google ARCore per Unity come esempio di software che abilita allo sviluppo di applicazioni di mixed reality sfruttando il sistema di tracking SLAM.

Ciò che mi ha spinto a scegliere la mixed reality come tema per questa tesi è stata la mia curiosità verso questo campo di ricerca, suscitato anche dagli accenni fatti durante il corso di "Sistemi Embedded e Internet of Things" riguardo le possibili applicazioni che questa tecnologia ha nella vita di tutti i giorni.

La tesi si è articolata in due fasi principali. La prima parte è stata dedicata al lato teorico: inizialmente mi sono informato sulle differenze tra AR, VR, MR, e sui dispositivi che abilitano all'utilizzo di tali tecnologie. Successivamente sono passato allo studio delle tecniche che consentono il tracciamento degli elementi fisici e la registrazione di quelli virtuali in un contesto di mixed reality. Ho quindi deciso di approfondire Google ARCore, che sfrutta la tecnica SLAM per ottenere il mapping dell'ambiente fisico in tempo reale, ovvero contestualmente all'esplorazione dell'ambiente. Completato lo studio teorico, ho poi messo in pratica quanto appreso cimentandomi nello sviluppo di una applicazione di mixed reality.

La tesi ripercorre il lavoro svolto e riassume le nozioni apprese. In tutto abbiamo cinque capitoli:

- il primo capitolo è incentrato sulla spiegazione di cosa si intenda per augmented reality, mixed reality e virtual reality; una rapida presentazione dei principali dispositivi che permettono di visualizzare oggetti virtuali e infine una introduzione ai software incontrati durante lo studio condotto in preparazione alla tesi.

- Il secondo capitolo è incentrato sulle tecniche di tracking adottate nel contesto della mixed reality per il posizionamento degli oggetti nell'ambiente fisico.

- Il terzo capitolo contiene un approfondimento su ARCore, in particolare in riferimento alle funzionalità messe a disposizione da Google per lo sviluppo di applicazioni Android utilizzando Unity game engine.

- Dal quarto capitolo inizia la parte relativa al progetto. In particolare in questo capitolo viene inquadrato il contesto in cui si inserisce l'applicazione da sviluppare, viene presentata la richiesta del committente e c'è l'analisi dei requisiti.

- Nel quinto capitolo invece vengono prese in considerazione le fasi di progettazione e sviluppo della applicazione. E' poi presente una sezione di validazione in cui sono riportate le criticità presenti in quanto sviluppato.

La realizzazione di questo progetto non è servita tuttavia per la semplice realizzazione dell'applicazione, ma è stata utile per inquadrare quali funzionalità per la progettazione di applicazioni di realtà aumentata vengano fornite agli sviluppatori mobile da ARCore e quale livello di accuratezza esse abbiano.

Capitolo 1

Realtà aumentata

La realtà aumentata (AR) è un'area di ricerca che ha come obiettivo quello di arricchire un'esperienza nel mondo reale sovrapponendo a questo uno strato di dati virtuali. Nel paper "A Survey of Augmented Reality" del 1997, Azuma identifica tre caratteristiche fondamentali dei sistemi AR:

1. consentono l'inserimento di oggetti virtuali nel mondo fisico;
2. consentono memorizzazione e gestione dei dati relativi a elementi tridimensionali;
3. prevedono una forma di interazione in tempo reale.

Le prime dimostrazioni di realtà aumentata (AR) risalgono alla seconda metà del secolo scorso, tuttavia solo recentemente sono emerse tecnologie che hanno permesso la diffusione della fruizione di applicazioni di realtà aumentata da parte di un grande pubblico.

Figura 1.1: Esempio di Head Mounted Device. (tratto da [3])

Le prime applicazioni di AR necessitavano infatti del supporto fornito da computer desktop e richiedevano necessariamente l'utilizzo di HMDs (head mounted displays), una sorta di caschi che l'utente doveva indossare per potersi immergere nel mondo aumentato. Questo tipo di tecnologie ha permesso la nascita delle prime applicazioni di AR sviluppate in alcuni ambiti come l'assemblaggio industriale, il training in ambito chirurgico e il mondo video-ludico. In ogni caso queste applicazioni, sebbene fossero state realizzate con successo, si sono dovute scontrare con i costi della tecnologia abilitante e la necessità di esperienza tecnica per il corretto utilizzo, che ne hanno impedito la diffusione.

Dal punto di vista hardware un ruolo fondamentale nella diffusione della realtà aumentata è rivestito sicuramente dagli smartphones: cellulari equipaggiati di fotocamere ad alta risoluzione, sensori per geo-localizzazione e orientamento, potenza di calcolo significativa e notevoli prestazioni nell'elaborazione delle immagini. La diffusione degli smartphones è stata affiancata dall'evoluzione del web che, a partire dalla sua versione 2.0, fornisce una infrastruttura altamente scalabile per produrre, elaborare, distribuire e conservare dati geo-referenziati.

La combinazione della diffusione di smartphones e il progressivo potenziamento dell'infrastruttura fornita dal web in termini di prestazioni e funzioni ha permesso lo sviluppo di un nuovo tipo di piattaforma per realtà aumentata che può essere utilizzata su scala mondiale. Di riflesso la possibilità che sempre più persone possano approcciarsi a questo mondo sta portando gli sviluppatori alla creazione di software accessibili a un pubblico sempre più generalista, il che contribuirà alla diffusione di questa tecnologia.

1.1 Overview AR, MR, VR

1.1.1 Realtà Aumentata

La realtà aumentata (AR), che non deve essere confusa con la realtà virtuale (VR), sovrappone contenuti digitali (che possono essere di tipo testuale, immagini o oggetti tridimensionali eventualmente animati) al mondo reale percepito dall'utente.

L'Enciclopedia Britannica afferma che la Realtà Aumentata è un processo che combina video o immagini fotografiche a dati utili generati da un calcolatore.

La realtà aumentata è la visione di uno strato di informazioni sovrapposto a immagini del mondo reale aggiornate istante per istante. Tali informazioni possono essere generate sia da un processore e una sorgente dati locali, che da un database remoto e sono "aumentate" da input provenienti da sensori

che rilevano audio, video o posizione (intesa sia a livello di orientamento del dispositivo sia in termini di coordinate geografiche).

Un aspetto fondamentale della realtà aumentata è quindi proprio la componente del mondo reale, e questo la pone a metà strada tra il mondo reale che è quello che percepiamo con i nostri sensi e la realtà virtuale che rimpiazza completamente il mondo reale con uno simulato.

I requisiti tecnologici necessari alla realizzazione della realtà aumentata sono molto maggiori rispetto a quelli richiesti per la realtà virtuale e questo è il motivo per cui lo sviluppo della realtà aumentata ha chiesto molto piu tempo rispetto a quello della realtà virtuale.

1.1.2 Caratteristiche dispositivi per AR

Nonostante sia passata oltre una cinquantina di anni dalle prime sperimentazioni in questo campo, i componenti chiave per la realizzazione di sistemi di realtà aumentata sono rimasti gli stessi: displays, trackers, computer e software grafici sono essenziali per questo tipo di esperienze.

Il rendering di un mondo aumentato può sfruttare varie tecnologie tra cui sistemi di proiezione ottica, display, dispositivi mobile general purpose come smatphone e tablet e dispositivi special purpose indossabili (i cosidetti "wearable") ad esempio occhiali, caschi (HUDs), ecc.

Tutti questi dispositivi impiegano diverse tecnologie:

- Una GPU che gestisca i sistemi di visualizzazione;
- Display o sistema di proiezione per la rappresentazione delle immagini e dispositivi ottici per disporle correttamente all'interno del campo visivo dell'utente;
- Sensori:
  - Camere per la visione frontale ed elaborazione di immagini provenienti dal mondo fisico che determinano la posizione degli elementi del mondo reale per mapparlo in una rappresentazione 3D;
  - Sensori di movimento;
  - Sensori che determinano l'altezza dell'utente rispetto a terra, in modo da determinare correttamente il suo punto di vista;
  - Sensori per tracciare la direzione in cui l'utente sta guardando;
  - Ecc.
- Sistemi audio di input (ad esempio microfono) e output (speakers per la riproduzione dei suoni);

- Sistemi di riconoscimento e classificazione degli oggetti che permettano allo strumento di determinare superfici e oggetti reali posti di fronte al visore, così che possano essere sfruttati come basi di appoggio per oggetti aumentati;

- Un sistema operativo che controlli le immagini virtuali attraverso voce, occhi, mani e movimenti;

- Comunicazione wireless (o wired) con un dispositivo che funga da server.

La realtà aumentata consente a un qualsiasi tipo di informazione digitale (sia essa un video, una foto, un link o del testo, ...) di essere visualizzata in sovrimpressione agli oggetti del mondo reale quando vengano visti attraverso le lenti di un dispositivo mobile o di un wearable.

Proprio i wearable sono in effetti dispositivi che si sposano bene con una tecnologia quale la realtà aumentata. Essi sono leggeri, dotati di sensori per comunicare in modo costante con il mondo circostante e trarre informazioni da esso e spesso fanno parte dell'universo IoT essendo in connessione continua con Internet, di solito attraverso wifi. Questo li rende i dispositivi ideali per sfruttare la realtà aumentata: assieme infatti queste tecnologie possono diventare un'estensione dei nostri sensi e migliorare il modo in cui le persone comunemente interagiscono con il mondo.

1.1.3 Tipi di Realtà Aumentata

Spesso quando si parla di Augmented Reality si finisce necessariamente con il citare anche la Mixed Reality (MR) che è la diretta evoluzione della stessa.

Nella definizione più ampia di realtà aumentata infatti, i dati sovrapposti al mondo reale possono essere originati da una sorgente di ogni tipo: video, gioco, modello 3D o info catturate localmente e non devono essere necessariamente correlati logicamente oppure sovrapporsi in modo realistico al mondo reale in cui sono immersi.

Figura 1.2: AR alcuni oggetti virtuali sono sovrapposti a quelli reali ma senza integrarsi con essi

Una applicazione di Mixed Reality ha una specifica conoscenza del mondo reale e di dove gli oggetti siano collocati. Gli elementi virtuali che popolano il mondo aumentato hanno una profonda correlazione con il mondo reale in cui sono immersi e l'utente può interagire con essi come con un oggetto reale.

Figura 1.3: MR oggetti fisici e virtuali si compenetrano, il mondo virtuale arricchisce quello fisico espandendo la sua espressività. (tratto da [5])

Il paragrafo che segue scenderà maggiormente nel dettaglio di questa nozione.

1.1.4 Mixed Reality

Il concetto di Mixed Reality (MR), nel contesto del mondo aumentato, fa riferimento alla combinazione e alla fusione di un ambiente virtuale con uno reale al fine di ottenere un mondo-ibrido, dove entrambe le realtà coesistano.

Un utente immerso in una esperienza di mixed reality si muove e agisce contemporaneamente e con soluzione di continuità in entrambe le realtà relazionandosi con i loro oggetti. Gli oggetti virtuali sono posizionati accuratamente nello spazio del mondo reale e la percezione degli oggetti è la stessa: se ci avviciniamo a un oggetto virtuale questo diventerà piu grande man mano che procediamo nella sua direzione e viceversa, allo stesso modo girandoci attorno l'oggetto cambierà prospettiva e potremo vederlo da differenti angolature proprio come se fosse reale e la sensazione sarà la stessa di avere di fronte a noi un oggetto olografico.

Gli utenti avranno inoltre la possibilità di manipolare gli oggetti virtuali durante l'esperienza di MR nello stesso modo in cui manipolerebbero degli oggetti fisici.

Come la realtà aumentata, la mixed reality sfrutta molte tecnologie: per creare su sistema di mixed reality è necessario sostituire la vista del mondo reale, che si può ottenere dalla visione attraverso smartglasses e HMD, con una visione attraverso videocamera, dal momento che lo schermo del dispositivo (di solito uno smartphone) occlude la vista. Per percepire il mondo come lo vedremmo dai nostri occhi, sarà quindi necessario espandere il campo visivo della telecamera (altrimenti avremmo l'effetto di vedere attraverso un tubo) per mezzo di un opportuno sistema di lenti.

Un altro approccio alla mixed reality è quello che prevede la proiezione degli oggetti virtuali sull'ambiente che circonda l'osservatore. Questo metodo consente di evitare l'utilizzo della telecamera in quanto gli oggetti vengono messi a fuoco semplicemente dagli occhi dell'utente e permette contemporaneamente a più persone di vedere la superficie e gli oggetti aumentati.

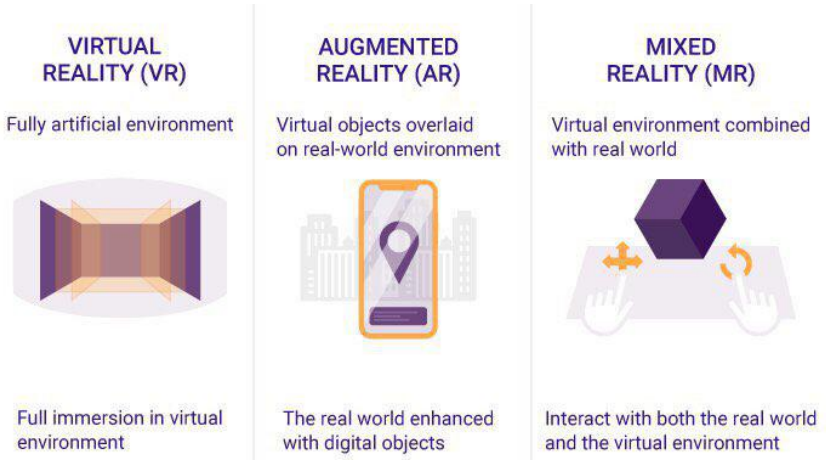


Figura 1.4: Confronto tra VR, AR, MR. (tratto da [6])

1.1.5 Differenze tra Virtual Reality e Augmented Reality

Spesso i termini Virtual Reality (VR) e Augmented Reality vengono usati come sinonimi. Questo accade erroneamente visto che non solo non sono la stessa cosa, ma nemmeno sono uno una sottofamiglia dell'altro. Capita che alcuni dispositivi possano consentire entrambe le esperienze, ma queste in se per se sono profondamente diverse.

La realtà virtuale trasporta l'utente in un mondo totalmente fittizio, generato al computer, e questi risulta essere isolato dall'ambiente reale circostante. Tale mondo tipicamente ha solo 3 gradi di libertà (i 3 assi cartesiani), mentre il mondo percepibile attraverso AR ne ha 6 (aggiunge la possibilità di muoversi avanti indietro e di girarsi lungo i propri assi) e include quello reale.

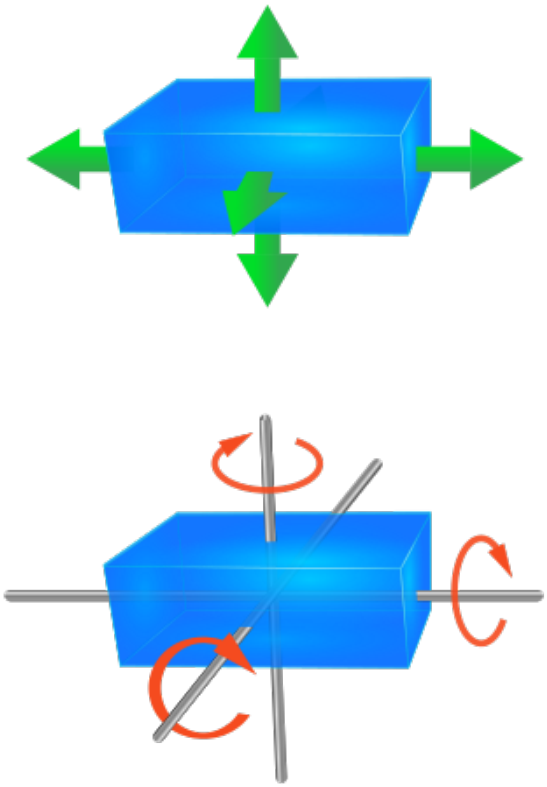


Figura 1.5: rappresentazione esplicativa dei 6 gradi di libertà (6DOF) nello spazio

Dal un punto di vista di un neofita, la realtà aumentata e la realtà virtuale si somigliano ad eccezione del fatto che la realtà aumentata consente la visione del mondo reale mentre la realtà virtuale lo impedisce; da qui il fatto che i dispositivi hardware abilitanti alle due tecnologie spesso hanno quasi lo stesso aspetto. La realtà aumentata presenta tuttavia maggiori requisiti dal punto di vista ottico e di tracking dal momento che deve interagire con gli oggetti del mondo reale.

Nonostante quindi sembrino simili si potrebbe dire che hanno un obiettivo diametralmente opposto infatti, mentre la realtà aumentata vuole arricchire la nostra percezione del mondo reale, la realtà virtuale sradica le sensazioni che l'utente ha del mondo fisico, isolandolo in uno fittizio. Entrambe le esperienze possono essere usate in campo tecnico e educativo per allenare l'utente ad azioni che dovrà poi compiere nel mondo reale tuttavia, mentre per quanto riguarda la realtà virtuale l'esperienza è limitata al test, la mixed reality può essere usata anche successivamente come aiuto all'utente nello svolgere le mansioni reali in quanto non ostacola la visione del mondo fisico.

Nella realtà aumentata, il computer usa sensori di localizzazione, movimento e orientamento e algoritmi per determinare posizionamento e orientamento della telecamera. La tecnologia dell'AR renderizza poi le grafiche 3D così come apparirebbero dal punto di vista della telecamera, sovrapponendole poi alle immagini del mondo reale che circonda l'utente. Mentre la realtà aumentata può sfruttare smartphones, caschi, tablet pc e occhiali, la realtà virtuale può usare solo un casco con uno schermo che non permetta alcuna visione diretta del mondo reale.

Poiché chi indossa un casco per la realtà aumentata vede anche il mondo reale, le informazioni riguardo il tracking degli oggetti fisici diventano più critiche: infatti una imprecisione in queste misure provocherà degli errori nella visualizzazione degli oggetti aumentati, che porterà a una sensazione di estraniamento dell'utente rispetto alla scena che sta osservando: verrà meno l'illusione che quegli oggetti siano effettivamente reali.

Tutto sommato quindi la realtà virtuale è più semplice da realizzare e ha costituito il primo passo nella direzione della più difficile ma anche più utile (in termini di applicazioni reali) realtà aumentata.

1.2 Dispositivi per AR

1.2.1 Caratteristiche e requisiti

Un sistema di AR ideale dovrebbe essere capace di creare degli oggetti aumentati 3D che popolino il modo fisico in modo convincente. C'è chi potrebbe scegliere di conferire agli oggetti aumentati un aspetto differente da quello degli elementi reali, ma certamente sarebbe apprezzabile aver la possibilità di creare contenuti virtuali che possano integrarsi con la realtà fisica in modo che l'utente non sia in grado di percepire la finzione.

Quindi la domanda da porsi sarebbe "di quali schermi abbiamo bisogno per ottenere un'esperienza di realtà aumentata convincente?" Hainich in "The End of Hardware: Augmented Reality and Beyond" (2009) ha affermato che per rimpiazzare ed emulare tutti gli schermi del mondo avremmo bisogno di un solo tipo di schermo personale near-eyes che sia: non-invasivo, confortevole, ad alta risoluzione, con un campo visivo largo, un range altamente dinamico e un sistema di tracciamento perfetto.

Naturalmente siamo lontani dal raggiungimento di questa tecnologia per il momento ma nel corso del tempo sono stati proposti vari tipi di dispositivi per la visione aumentata e ognuno di essi ha dei pro e dei contro, quindi per la scelta dello strumento da utilizzare è sempre necessario ricercare un compromesso.

Fra i tipi di visual display attualmente presenti possiamo identificare quattro famiglie: near-eye, handheld, stationary, projected.

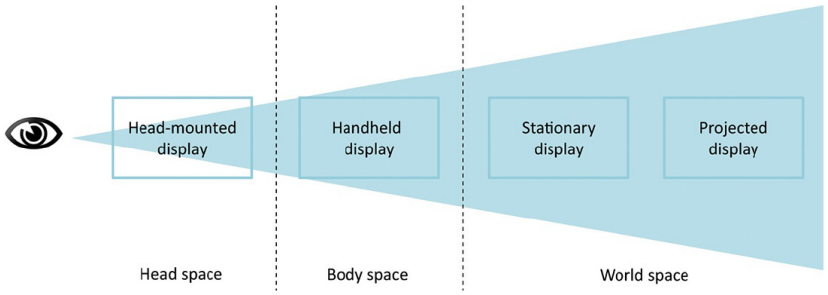


Figura 1.6: Display AR categorizzati in base alla distanza. (tratto da [1])

1.2.2 Near-eye display

La classe più nota di display per AR sono probabilmente gli Head Mounted Display (HMD). Questo tipo di dispositivi è pensato per essere indossato in testa, come un casco; non devono ostruire la vista ma anzi devono fornire la migliore visione possibile e devono essere confortevoli per l'utente. Un punto critico per i near-eye display è l'aspetto ergonomico dello schermo da indossare. Naturalmente, un HMD dovrebbe essere il piu possibile leggero affinché possa essere utilizzato anche per lunghi periodi di tempo senza stancare l'utente. Al di la dei componenti elettronici e ottici il peso del dispositivo è in gran parte determinato dal case. Gli HMD possono essere pesanti e ingombranti ma l'utente sarà maggiormente portato ad utilizzarli qualora la mansione da svolgere richieda comunque di indossare un casco (si pensi ai piloti o ai pompieri).

Esistono anche dispositivi che si agganciano agli occhiali estendendone la funzionalità, tuttavia il campo visivo della realtà aumentata è fortemente limitato rispetto a un HMD, per cui dispositivi che adottano questa soluzione (si vedano ad esempio i Google Glass) sono più orientati alla visualizzazione di informazioni di tipo testuale, che non a una esperienza di MR.

Un display incorporato all'interno di un visore è decisamente l'approccio migliore e piu potente. Questo genere di dispositivi richiede tuttavia una certa cura nella progettazione del case: a causa della posizione dello schermo posto davanti al viso dell'utente, il peso si concentrerà in questa zona del casco e sarà necessario prendere opportuni accorgimenti per rendere il tutto equilibrato e stabile. Il casco dovrà inoltre adeguarsi alla dimensione della testa dell'utente ed è necessario che consenta il passaggio di aria per respirare.

Un HMD deve essere in grado di combinare realtà aumentata e mondo fisico per mezzo dello schermo (o delle lenti) attraverso cui l'utente sta osservando il mondo reale, si parla quindi di see-through display.

Esistono due strade per ottenere questo risultato: optical see-through display (OST) o un video see-through display (VST), gli HMDs possono sfruttare entrambe.

Gli optical see-through display (OST), per combinare elementi virtuali e reali, fanno affidamento su un componente ottico che è in parte trasmissivo e in parte riflettente, una sorta di specchio semi-argentato. Tale specchio lascia passare una sufficiente quantità di luce proveniente dal mondo reale, così che la componente reale possa essere vista direttamente, e allo stesso tempo uno schermo che mostra gli oggetti virtuali è posto sopra o a fianco dello specchio in modo tale che questo rifletta le immagini virtuali che appaiono sovrapposte al mondo reale.

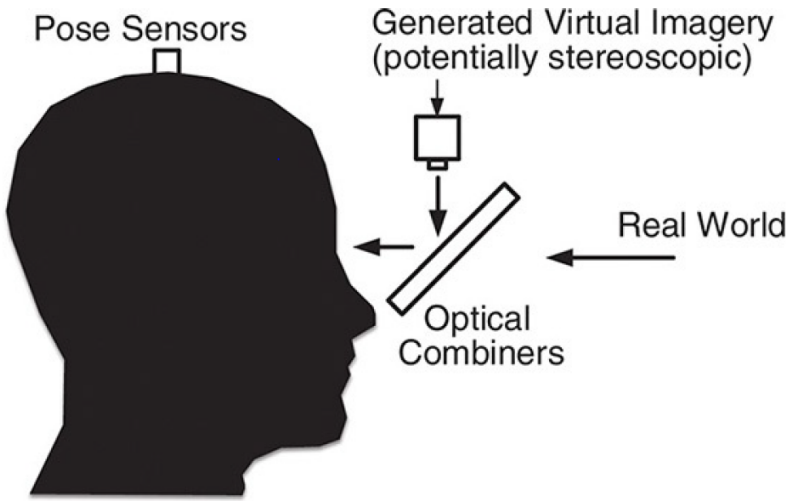


Figura 1.7: Uno schermo see-through ottico usa un elemento ottico per combinare la vista che l'utente ha del mondo reale con le immagini virtuali. (tratto da [1])

I video see-through display (VST) ottengono la combinazione del mondo reale e virtuale per via elettronica. Viene acquisita una immagine video del mondo reale e trasferita a un processore grafico, che combina tale immagine con quelle generate dal computer, spesso semplicemente impostando l'immagine del mondo reale come sfondo dell'immagine digitale contenente gli oggetti virtuali. Le immagini così combinate vengono poi mostrate all'utente attraverso un comune mezzo di visualizzazione.

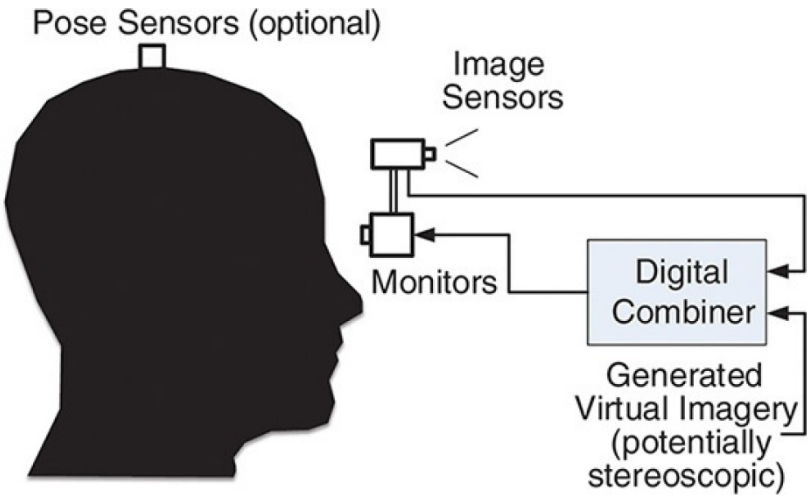


Figura 1.8: Uno schermo see-through video cattura le immagini del mondo reale attraverso una videocamera e poi tali immagini vengono modificate con tools grafici per restituire l'immagine aumentata. (tratto da [1])

1.2.3 Handheld display

Il rapido sviluppo della tecnologia mobile (smartphones e tablet) ha reso gli handheld display la piattaforma più popolare per la realtà aumentata (AR e MR) al giorno d'oggi. La videocamera posteriore è utilizzata per catturare uno streaming video per l'esperienza VST (video see-through). Poiché la videocamera punta direttamente fuori dal retro del dispositivo, è di solito necessario tenere il device almeno all'altezza delle spalle. Questo può essere un problema dal punto di vista dell'esperienza AR in quanto questa posizione tende a stancare l'utente in un periodo di tempo relativamente breve e questo può precludere la fruizione del monto aumentato in tutti i suoi dettagli.

Il fatto che il dispositivo possa essere facilmente riposto in tasca in caso di non utilizzo è una lama a doppio taglio. Da un lato permette di non doverlo indossare anche nel momento in cui non è necessario, come avviene invece per un HMD. Tuttavia, limita l'immediatezza dell'utilizzo: se infatti il dispositivo è stato riposto sarà necessario recuperarlo, riavviarlo e attendere il caricamento del programma.

Un handheld display ospita sia il display che una o più videocamere rigidamente montate nello stesso case. Il tracciamento della posizione del device nel mondo verrà effettuato attraverso la videocamera nella maggior parte dei casi, ma potranno essere utilizzati anche altri metodi di tracciamento (es.: GPS).

Alcune proposte prodotte di recente suggeriscono di ragionare dal punto di vista dell'utente piuttosto che da quello del dispositivo. Ovvero, invece di

mostrare una immagine aumentata puramente dalla prospettiva della videocamera, non tenendo in considerazione la posizione dell'utente, tenere in tracking anche l'utente. Il tracciamento dell'utente può essere realizzato attraverso l'utilizzo di una videocamera frontale, che ormai è presente in molti dispositivi. Da notare che questa configurazione è molto piu costosa in termini di risorse di quella che tiene conto della sola prospettiva del dispositivo. Infatti non solo sono necessari due sistemi di tracciamento, ma il dispositivo deve renderizzare due flussi video diversi contemporaneamente.

Con il sempre piu diffuso utilizzo di smartphones e tablet e con il fatto che questi rivestano sempre più spesso dei ruoli chiave in context-sensitive computing, non c'è da meravigliarsi che le piattaforme handheld stiano diventando la tecnologia abilitante leader nell'interfacciarsi con il mondo della realtà aumentata. Ciò non toglie che nel corso del tempo i dispositivi wearable tipo HMD possano sostituire gli attuali dispositivi mobile almeno in questo settore.

1.2.4 Stationary display

In precedenza, si è accennato al possibile display AR personale ideale e si è detto che dovrebbe essere in grado di emulare un qualsiasi altro display fisico ovunque, semplicemente generando un display virtuale corrispondente in AR. Anche presupponendo la realizzazione effettiva di questo scenario, i display fisici non potrebbero comunque sparire del tutto, anzi sarebbero necessari per abilitare una comunicazione con quel gruppo di persone che non utilizzano la realtà aumentata. Senza contare che il mondo è pieno di display fisici di vario tipo ed è bene considerare eventuali utilizzi per questo tipo di schermi in ambito AR.

1.2.5 Projected display

Col tempo i proiettori sono diventati sempre piu potenti e accessibili. Questo li sta portando ad essere utilizzati per scopi che vanno oltre la proiezione di film o video in pubblico, ad esempio possono essere utilizzati per ricreare effetti speciali ad eventi pubblici o proiezioni interattive in luoghi outdoor.

Le ultime applicazioni, cui di solito si fa riferimento come digital projection mapping, esemplificano il concetto di spatial AR.

1.3 Tools per applicazioni AR

Molte compagnie produttrici di software stanno cominciando ad offrire tool che aiutino gli sviluppatori a creare applicazioni che lavorino nel contesto di un mondo aumentato. I tool sono degli insiemi di strumenti utilizzati per

lo sviluppo e il mantenimento di applicazioni per un certo tipo di device. Le SDK sono usate per arricchire le applicazioni con funzionalità avanzate, avvisi, notifiche push ecc.

Le SDK vengono spesso fornite agli sviluppatori sotto forma di plugin utilizzabili in un certo ambiente di sviluppo che supporta le funzionalità di base. Di seguito riporterò alcuni esempi di strumenti con cui sono venuto in contatto durante l'elaborazione di questa tesi.

1.3.1 Vuforia

È un game engine cross-platform che fornisce un SDK per lo sviluppo di applicazioni di realtà virtuale e realtà aumentata. Vuforia permette di collocare contenuti 3D in un ambiente fisico. Il fulcro dell'offerta è il Vuforia Engine, che fornisce funzionalità di computer vision che permettono riconoscimento di oggetti e la ricorstruzione dell'ambiente reale.

Una delle funzioni di riconoscimento fornite da Vuforia è basata su VuMark, un codice visivo personalizzabile che può essere posizionato su un qualsiasi prodotto o macchina. È pensato per indicare visivamente ad un utente che una esperienza di realtà aumentata è disponibile e correlata a quell'oggetto.

Un'altra funzione di riconoscimento si basa su image target, questa funzione permette ai programmatori di riconoscere e tenere traccia di immagini stampate su una superficie piana. Vuforia riesce a memorizzare, riconoscere e tracciare fino a mille immagini salvate in locale sul dispositivo o milioni salvate in cloud. Tali immagini vengono utilizzate solitamente per posizionare contenuti su prodotti come libri, giornali, imballaggi.

Attraverso il proprio stack di ricostruzione 3D, Smart Terrain, Vuforia fornisce agli sviluppatori l'accesso alle superfici e agli oggetti scoperti nell'ambiente: questo costituisce un ottimo punto di partenza per quegli sviluppatori che progettano app che prevedono una interazione tra mondo fisico e virtuale.

1.3.2 ARCore

È una piattaforma per costruire esperienze di realtà aumentata. Attraverso molteplici APIs, ARCore permette agli smartphone di percepire l'ambiente circostante, comprendere il mondo fisico ed interagire con esso. Alcune API sono disponibili sia per sistemi Android che per iOS.

ARCore fonda l'esperienza di realtà aumentata su tre capacità chiave che permettono di integrare i contenuti virtuali nel mondo reale così come viene visto dalla videocamera:

1. Motion tracking che permette al dispositivo di comprendere la propria posizione in relazione al mondo e di tenerne traccia;

2. Environmental understanding che permette al dispositivo di comprendere le dimensioni e la posizione dei vari tipi di piano (orizzontale e verticale e inclinate rispetto al terreno);

3. Light extimation che permette al dispositivo di determinare le condizioni di luminosità correnti.

Per quanto riguarda il funzionamento, ARCore si occupa prevalentemente di tracciare la posizione del device nel mondo reale, determinare i piani e tenere traccia degli oggetti aumentati che l'utente pone nell'ambiente.

Come vuforia permette di riconoscere delle immagini marker che possono costituire un punto di partenza per l'esperienza di AR.

Non tutti i device android sono supportati da ARCore. Essi devono infatti passare attraverso un processo di certificazione. La certificazione è importante ai fini di mantenere una certa qualità nell'esperienza di realtà aumentata.

La possibilità di ottenere tale certificazione è fortemente correlata alla componente sensoristica di cui il dispositivo è dotato. Le funzionalità di ARCore sono infatti fortemente legate ad un accurata rilevazione dei movimenti, dell'orientamento del dispositivo, alla precisione del gps e alla qualità della camera. Il dispositivo deve essere dotato di una CPU dotata di potere computazionale sufficiente ad avere buone prestazioni nell'elaborazione real time delle immagini e dello streaming data acquisiti.

In ambito Android è necessario che il sistema operativo sia aggiornato alla versione 7.0 o superiore o, in alternativa, un dispositivo iOS che supporti ARKit.

1.3.3 ARKit

È una tecnologia equivalente a ARCore studiata dal brand Apple. Fornisce quindi supporto alla realtà aumentata e virtuale: tiene traccia dello spostamento del dispositivo, effettua riconoscimento dei piani e permette il posizionamento degli oggetti virtuali. Implementa le stesse funzioni indicate per ARCore.

1.3.4 Unity

Unity è uno strumento di authoring integrato multipiattaforma che mette a disposizione un motore grafico e un'interfaccia utente che permette agli sviluppatori di creare giochi sia in 2D che in 3D. Esso supporta la programmazione in C# e permette la creazione di applicazioni Android e IOS. Le componenti fondamentali di un programma unity sono:

- Assets: insiemi di elementi che possono essere utilizzati in un progetto. Di solito gli elementi che costituiscono un asset mirano a fornire determinate funzionalità. Un asset può essere importato in piu di un progetto contribuendo così al riutilizzo del codice.

- Scenes: ogni programma ne contiene almeno una e costituisce uno scenario della applicazione.

- GameObject: sono gli elementi fondamentali che popolano la scena. Essi costituiscono le entità che agiscono all'interno di quella scena e possono rappresentare degli elementi virtuali o fisici (ad esempio per gestire la videocamera di norma viene creato un elemento camera, stessa cosa per la gestione dell'illuminazione o per le rappresentazioni virtuali di oggetti reali) . Ogni GameObject è caratterizzato da una posizione e un orientamento nel mondo e il suo comportamento può essere definito agganciandogli alcuni componenti.

- Components: sono degli elementi che determinano il comportamento degli oggetti cui sono agganciati, essi permettono di legare ad un oggetto degli script con cui possiamo attribuire determinate caratteristiche o associare specifiche funzioni.

- Prefab: sono un template per la creazione di GameObject che abbiano determinate caratteristiche e comportamenti di base. Possiamo usare un prefab per istanziare più elementi simili tra loro a runtime, questo è utile perché non sarà necessario specificare ogni volta come è costruito l'elemento che vogliamo venga generato, inoltre nel caso volessimo apportare una modifica sarà necessario aggiornare soltanto il prefab.

Unity si è rivelato essere uno strumento veramente utile poiché per questo editor esistono plug-in per l'utilizzo delle SDK di Vuforia, Mapbox, ARCore e ARKit.

1.3.5 Mapbox

È una piattaforma per lo sviluppo di applicazioni che utilizzano informazioni riguardanti mappe, geolocalizzazione, ambienti outdoor sia urbani che non. Gli strumenti forniti da mapbox supportano ogni aspetto della creazione di mappe che possono necessitare di dettagli come la conformazione del terreno o la distribuzione e l'aspetto degli edifici così come appaiono nella realtà.

In particolare Mapbox permette di:

1. Creare mappe che abbiano uno stile custom ad alto livello di dettaglio;

2. Sviluppare applicazioni web e mobile con tutte le caratteristiche necessarie;

3. Estendere le funzionalità di una applicazione con servizi web di geocoding, direzioni, analisi dello spazio, ecc.;

4. Creare mappe statiche;

5. Ottenere informazioni riguardo lo stato del traffico, delle strade, degli edificie degli elementi naturali di uno specifico paesaggio aggiornati attraverso informazioni ottenute da satellite;

6. Fornisce immagini ottenute via satellite;

7. Fornisce le basi per esperienze di realtà aumentata di tipo TableTop o WorldScale.

Capitolo 2

Registrazione

In questo capitolo analizzeremo le tecniche che consentono il tracking degli elementi che compongono l'ambiente fisico e la registrazione degli elementi virtuali al fine di dare vita a un mondo aumentato. Proprio il concetto di registrazione è un aspetto cruciale per la realizzazione di applicazioni di realtà aumentata. In generale, la registrazione potrebbe essere interpretata come l'accuratezza nel posizionamento di un oggetto virtuale nel mondo reale. Essa dovrebbe preoccuparsi di aggiornare posizione e orientamento degli oggetti in relazione ai cambiamenti di posizione e di prospettiva degli utenti.

2.1 Marker vs Markerless

L'accurata rilevazione del movimento, e la (geo)localizzazione sono due dei principali problemi da affrontare nel campo della realtà aumentata. Per gestire con successo queste funzioni è necessario utilizzare piu sensori: dispositivi meccanici, basati su ultrasuoni, sensori che rilevano il campo magnetico, dispositivi per il controllo dell'inerzia, GPS, bussola e ovviamente sensori ottici. Non esiste una singola soluzione per risolvere tale problema, ma tutte passano attraverso una miniaturizzazione dei sensori della videocamera, una loro migliore accuratezza e la scoperta di nuove tecniche di visione.

Le applicazioni di realtà aumentata si fondano sull'accurata elaborazione della posizione e della rotazione della videocamera in uno spazio tridimensionale con 6 gradi di libertà (6DOF).

Il termine "pose" fa riferimento sia alla posizione che all'orientamento della videocamera ed è necessario per determinare dove la telecamera si trovi in relazione all'origine del mondo e in che direzione sia puntata. 6DOF fa riferimento alla pose della videocamera nello spazio (avanti/indietro, su/giu, sinistra/destra) unita alle possibili rotazioni lungo gli assi (imbardata, beccheggio, rollio). La pose può essere determinata attraverso l'utilizzo di sensori di

inerzia (acellerometro, giroscopio), di localizzazione (bussola, magnetometro, GPS, barometro nel caso dell'altezza) e/o tecniche di radio-localizzazione usate dai dispositivi mobile che permettano di determinare la pose della camera con una cera precisione.

La tecnica piu semplice per consentire di determinare la posizione di un dispositivo è l'utilizzo di marker di riferimento (fiducial marker).

Le tecniche di tracciamento visive sono suddivise in metodi che richiedono una conoscenza preliminare (model-based tracking) e metodi ad hoc(feature-tracking). Questi ultimi sono ulteriormente suddivisi in base al modo in cui la mappa dell'ambiente viene generata: tracking-only method, simultaneous localization and mapping method (SLAM) e extensible method. E' possibile suddividerli anche in metodi di simple marker tracking, e dynamic maker fields tracking.

Un sistema di tracciamento può memorizzare una mappa creata su misura e usarla la volta successiva come una informazione conosciuta a priori, quindi le tecniche di tracking possono essere usate in combinazione l'una con l'altra, non sono esclusive.

2.2 Markers e Fiducials

Figura 2.1: Grazie al riconoscimento del marker una applicazione AR ha generato l'oggetto virtuale e lo ha posizionato correttamente. (tratto da [7])

Nella realtà aumentata marker based il modo più semplice per determinare la posizione della videocamera è basato su un approccio visivo con un marker di riferimento (di solito un quadrato con un pattern bianco e nero che codifica informazioni riguardo a come vada "aumentato" graficamente). La posizione del marker assieme alla calibrazione della fotocamera permettono di posizionare l'oggetto aumentato nella posizione corretta sullo schermo: le funzioni di tracking possono inoltre utilizzare queste informazioni per determinare la pose del dispositivo.

I marker possono essere considerati " un pezzo di informazione che è rilevante per risolvere delle funzioni di elaborazione relative a una certa applicazione ".

Grazie alla loro forma e al pattern che li caratterizza, i marker sono facilmente individuabili vicino o sull'oggetto di interesse e quindi costituiscono un mezzo di identificazione rapido, 4 marker point correttamente posizionati inoltre permettono un calcolo preciso della pose della videocamera. I marker possono anche essere usati per migliorare altri tipi di tracking in condizione di scarsa luminosità o in altre situazioni che possono mettere in difficoltà la telecamera.

Solitamente si usano immagini bidimensionali come marker, tuttavia anche un oggetto tridimensionale può rivestire questo ruolo, sebbene riconoscere un oggetto come marker sia computazionalmente più difficile e richieda condizioni di luce ottimali.

In ogni caso il tracking della pose della videocamera in un ambiente sconosciuto può essere una sfida ardua. Proprio per risolvere le difficoltà che si possono incontrare cercando di creare un esperienza AR su mobile in ambiente sconosciuto è nata una tecnologia detta SLAM (Simultaneoous Localization and Mapping).

2.3 Natural Feature Tracking using Markers

Natural Feature Tracking (NFT) è l'idea di riconoscere e tener traccia di una scena senza l'utilizzo dei marker. NFT può usare elementi presenti nella scena per migliorare il tracking di punti e aree al suo interno. Il risultato sembra avvicinarsi a un tipo di tracking markerless poiché l'utente non riconosce gli elementi di cui il sistema tiene traccia come marker in quando facenti parte di quell'ambiente per loro natura.

Usare questa tecnologia al posto della SLAM può essere computazionalmente meno esigente in termini di risorse e piu pratico per dispositivi su cui queste risorse sono limitate come può essere nel caso degli smartphones. Tuttavia al crescere del numero di immagini da tracciare il gap di risorse computaziona-

li richieste può abbassarsi, e oltre una certa quantità di elementi risulta più conveniente adottare la tecnologia SLAM.

2.4 SLAM-Markerless Location

Un requisito cruciale che una applicazione di AR deve soddisfare è sapere sempre dove si trova l'utente (e la camera) e cosa lo circonda. Come ampiamente discusso nel paper "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age" ([13]), la SLAM, simultaneous localization and mapping, è una delle tecnologie che consente di conoscere tali informazioni, in particolare è un processo attraverso cui il dispositivo crea una mappa che rappresenta gli elementi fisici che si trovano nelle vicinanze, questo gli permette di orientarsi all'interno dei confini di ciò che il dispositivo percepisce in tempo reale.

In partenza SLAM si trova in un ambiente sconosciuto a partire dal quale il dispositivo cerca di generare una mappa su cui tiene traccia della propria posizione attraverso una serie di calcoli e algoritmi che utilizzano i dati provenienti dai sensori IMU (Inertial Measurement Unit).

Nel caso di alcune applicazioni outdoor la necessità dell'utilizzo di SLAM viene del tutto meno grazie all'alta precisione dei sensori GPS.

Affinché SLAM possa funzionare il sistema deve creare, in una fase preliminare, una mappa del mondo circostante e orientarsi in essa, raffinando la mappa stessa.

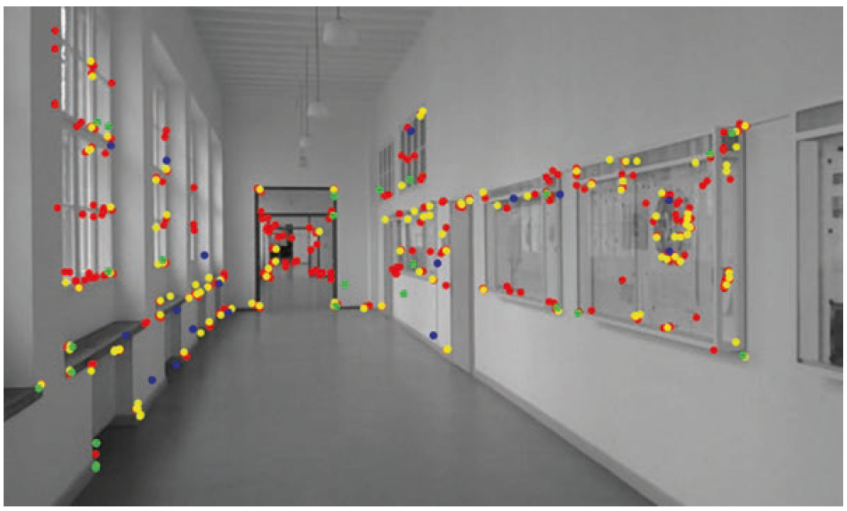


Figura 2.2: Visione dell'ambiente da parte di un sistema AR che sfrutta la tecnologia SLAM.(tratto da [2])

Durante gli spostamenti veloci all'interno di un ambiente di AR è molto facile perdere temporaneamente il tracking e conseguentemente avere dei failure nell'applicazione. Nell'ottica di riprendersi da queste situazioni di errore, il sistema necessita di una routine di recupero della posizione che possa calcolare rapidamente la posa (eventualmente con una certa approssimazione) della videocamera nel caso in cui le immagini inquadrate siano sfocate o corrotte.

Il sistema ha anche il compito di generare una mesh 3D dell'ambiente che possa essere usata dall'applicazione per produrre una esperienza aumentata in cui gli elementi virtuali siano adeguatamente mescolati a quelli reali.

Il tracciamento senza marker basato su dati visivi è solitamente combinato al tracciamento del giroscopio. Nella AR markerless, il problema di determinare la pose della videocamera richiede una capacità di calcolo significativa e l'impiego di algoritmi per l'elaborazione delle immagini più complessi come la ricerca di punti caratteristici, classificazione degli oggetti e computazione real-time.

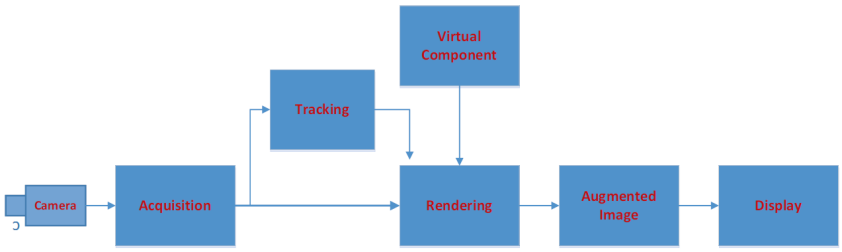


Figura 2.3: Diagramma degli stati di un sistema di realtà aumentata con tracking. (tratto da [2])

La figura precedente esemplifica il funzionamento di un semplice sistema di AR.

Il modulo Acquisition cattura le immagini dal sensore della camera.

Il modulo Tracking calcola il corretto posizionamento e orientamento del device da usare poi per collocare lo strato virtuale su quello fisico. Nel contesto di un dispositivo che non viene indossato come un casco, questo è il vero cuore del sistema: infatti il calcolo della pose della videocamera è critico in un sistema di realtà aumentata che fa elaborazione video continua.

Il modulo di Rendering combina l'immagine originale con la componente virtuale usando la posa calcolata, e poi renderizza l'immagine sullo schermo.

Un esempio di software che sfrutta la tecnologia SLAM è costituito da Google ARCore, cui è dedicato il prossimo capitolo.

Capitolo 3

Google ARCore

ARCore è una piattaforma sviluppata da Google con l'intento di mettere a disposizione dei programmatori degli strumenti per lo sviluppo di esperienze in realtà aumentata. ARCore fornisce diverse API che permettono allo smartphone di percepire l'ambiente circostante e gli oggetti fisici che lo popolano e interagire con essi. Vengono fornite API sia dedicate al mondo Android che al mondo IOS, l'obiettivo degli sviluppatori è infatti quello di rendere possibili delle esperienze di realtà aumentata condivise e che astraggano dal tipo di dispositivo usato per fruirne. Al fine di dare vita ad esperienze di realtà aumentata, tale piattaforma utilizza la tecnologia SLAM che, come già illustrato nel capitolo precedente, consente un mapping graduale dell'ambiente fisico che viene portato avanti dal programma durante tutta la fase di vita dell'applicazione. Mentre il dispositivo esplora l'ambiente fisico esso raccoglie dati che vengono elaborati per consentire una comprensione sempre aggiornata dell'ambiente reale in cui dovranno essere immersi gli oggetti virtuali.

3.1 Funzionalità di ARCore come applicazione SLAM

ARCore fornisce principalmente due funzionalità:

- Traccia la posizione del dispositivo mobile mentre questo si muove;
- Costruisce una rappresentazione di come percepisce il mondo.

Il metodo usato da ARCore per determinare gli spostamenti sfrutta la videocamera del dispositivo per identificare i punti di interesse (detti features o feature points) per tracciare come questi si muovano nel tempo. Incrociando

gli spostamenti di tali feature points con i dati rilevati dai sensori di movimento, ARCore è in grado di determinare sia la posizione che l'orientamento del cellulare (o tablet) nello spazio e come esso si muove.

Oltre all'identificazione di questi punti chiave, ARCore è in grado di rilevare la presenza di superfici sia orizzontali (come tavoli o pavimenti) che verticali (come i muri) ed è in grado di stimare il livello di luminosità a cui l'ambiente è esposto. Tutte queste capacità permettono ad ARCore di ottenere una propria comprensione dell'ambiente in cui si trova il device.

Questa comprensione del mondo fisico permette all'utente di posizionare oggetti, annotazioni, o altre informazioni in modo che si integrino alla perfezione con il mondo reale. È possibile per esempio piazzare un tavolo virtuale su un pavimento o agganciare una lampada a un muro e, grazie alla tecnologia di motion tracking, potremo muoverci attorno a questi oggetti e vederli da ogni prospettiva; oppure potremo lasciare la stanza per poi ritornare e scoprire che gli oggetti posizionati precedentemente sono ancora nello stesso punto in cui li avevamo lasciati.

Di seguito vengono analizzati piu nel dettaglio i concetti su cui ARCore basa il proprio funzionamento.

- Motion Tracking: dal momento che lo smartphone si muove nel mondo, ARCore utilizza un processo chiamato "cuncurent odometry and mapping" anche detto COM per comprendere quale sia la posizione istante per istante del dispositivo nel mondo. ARCore trova delle caratteristiche visivamente percepibili dette feature points, presenti nell'immagine catturata dalla videocamera, e usa questi punti per calcolare i cambi di posizione. L'informazione visiva è combinata con le misure provenienti dai sensori IMU che rilevano il movimento, per stimare la pose della videocamera in relazione al cambiamento del mondo nel tempo. Allineando la pose della videocamera virtuale che renderizza i contenuti 3D con quella del del sensore ottico del dispositivo fornita da ARCore, gli sviluppatori possono renderizzare i contenuti virtuali dal giusto punto di vista. Tali immagini virtuali possono essere sovrapposte alle immagini ottenute della telecamera del dispositivo, dando la sensazione che i contenuti virtuali facciano parte del mondo reale. Il continuo aggiornamento della percezione del mondo da parte di ARCore ha costituito un problema in principio, dal momento che cambiando il mondo cambiava anche la percezione della posizione degli elementi aumentati nello spazio. Un esempio è il fatto che se avviamo un'esperienza di AR outdoor e posizioniamo un elemento virtuale in una certa posizione del mondo semplicemente indicando le coordinate rispetto all'origine e poi ci spostiamo fino a perderlo di vista, quando torniamo al punto di partenza l'oggetto

non ci apparirà più nella stessa posizione nella quale lo avevamo lasciato, ma potrà essere spostato in una qualche direzione di un certo offset. Per compensare a questo problema sono nate le ancore che verranno trattate in seguito.

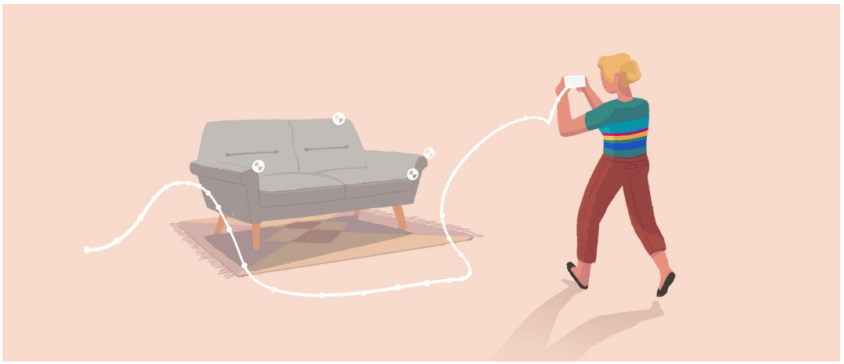


Figura 3.1: Mentre l'utente si muove nello spazio, l'applicazione ARCore individua i feature points che gli permetteranno di riconoscere l'ambiente e tracciare gli elementi fisici.(tratto da [8])

- Environmental Understanding: ARCore tenta di migliorare costantemente la propria comprensione del mondo reale, trovando nuovi punti caratteristici e piani. ARCore cerca dei cluster di feature point che sembrino appartenere allo stesso piano (sia che esso sia verticale o orizzontale) e rende disponibili queste superfici all'applicazione sotto forma di oggetti "planes". ARCore cerca di determinare anche i "confini" del piano e fornisce tale informazione allo sviluppatore, in modo tale che sia possibile determinare se un oggetto è posto su una determinata superficie o meno. Poiché ARCore usa i feature points per determinare i piani, superfici lisce, riflettenti o senza alcun tipo di decorazioni che le rendano distinguibili (si pensi ad esempio ad un muro bianco) potrebbero non essere identificate correttamente.

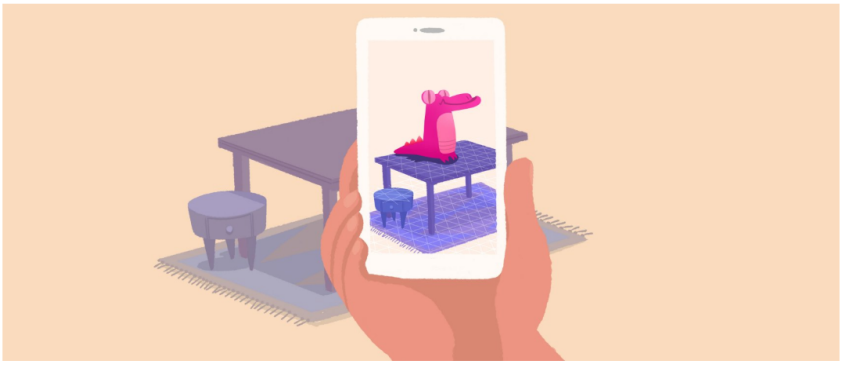


Figura 3.2: Nell'immagine l'applicazione ha riconosciuto i piani del tappeto e del tavolo, che vengono mostrati attraverso un reticolato. Il piano del tavolo in particolare è stato utilizzato per il posizionamento di un oggetto virtuale.(tratto da [8])

- Light estimation: ARCore è in grado di determinare la luminosità media dell'ambiente in cui è stata scattata un'immagine e permette di applicare la color correction su di essa. Questo permette di immergere gli oggetti virtuali nelle stesse condizioni di luce in cui si trovano gli oggetti del mondo fisico, aumentando così la sensazione di realismo percepita dall'utente.

Figura 3.3: L'immagine mostra una applicazione che riconosce una differenza di luminosità nella scena inquadrata dal dispositivo, i colori degli oggetti aumentati vengono modificati opportunamente per aumentare il realismo. (tratto da [8])

- User interaction: ARCore utilizza gli "hit test" per estrapolare una coordinata (x,y) corrispondente a un punto dello schermo dello smartphone (ricavato da un tocco sul display o da qualsiasi altro tipo di interazione

con l'utente supportata dall'applicazione) e proietta un raggio sulla vista che la camera ha del mondo, restituendo un qualsiasi piano o feature point che intercetta quel raggio, assieme alla posisione e orientamento di quell'intersezione nel mondo. Questo permette agli utenti di selezionare o comunque interagire con gli oggetti dell'ambiente.

- Punti orientati: i punti orientati possono essere usati per posizionare oggetti virtuali su superfici inclinate. Quando viene effettuato un hit test, che ritorna un feature point; ARCore cerca i feature points vicini e li usa per tentare di stimare l'angolo della superficie nel punto esatto del feature point da restituire. ARCore renderà poi disponibile la pose che tenga conto di tale inclinazione. Poiché ARCore utilizza dei cluster di feature points per determinare l'angolo di inclinazione della superficie, una superficie difficile da determinare (come detto nella parte di environmental understanding) potrebbe non fornire abbastanza informazioni per un calcolo accurato dell'inclinazione.

- Ancore e Trackable: il valore delle pose acquisite può cambiare nel corso del tempo, man mano che ARCore migliora la propria percezione del mondo e della propria posizione. Quando si vuole posizionare un oggetto virtuale è necessario definire un ancora (anchor) per assicurarsi che ARCore tracci la posizione dell'oggetto nel tempo. Capita spesso di creare un ancora basandosi su una posa ottenuta da un hit test. Il fatto che le pose possano cambiare significa che ARCore potrebbe aggiornare anche la posizione degli oggetti che costituiscono l'ambiente come i piani o i feature points. Entrambi questi tipi di elementi fanno parte di una famiglia di oggetti piu grande detta trackable, che esattamente come suggerisce il termine, sono oggetti di cui ARCore traccia la posizione nel corso del tempo. È possibile ancorare oggetti virtuali a specifici trackable per assicurarsi che la relazione tra gli oggetti reali e virtuali rimanga stabile nonostante il dispositivo si muova nello spazio. Questo significa che quando piazzo un vaso virtuale su un tavolo (ancorandolo quindi al piano del tavolo rilevato da ARCore), se ARCore corregge la posizione del piano associato al tavolo, il vaso continuerà a rimanere su quel piano e quindi sul tavolo nel mondo aumentato. Bisogna notare che alla stessa ancora è possibile agganciare piu oggetti e questo permette di migliorare le performance, tuttavia gli oggetti dovrebbero, per logica (e anche ai fini di una buona resa), essere posti a una distanza non eccessiva dall'ancora.

- Augmented Images: Le immagini aumentate permettono di creare applicazioni di AR che possano reagire a immagini 2D specifiche, come immagini sull'imballaggio dei prodotti o i poster di un film. Gli utenti

possono inoltre dare avvio all'esperienza in realtà aumentata puntando la telecamera dello smartphone verso una specifica immagine.

Le immagini aumentate funzionano in 3 passi:

1. È necessario indicare ad ARCore a che immagine si è interessati. Ci sono 2 modi:
   - Si può dire ad ARCore di individuare un certo tipo di immagini a tempo di esecuzione: quindi è possibile scaricare una immagine da un server e chiedere ad ARCore di caricare quell'immagine a runtime, capire come trovarla nella scena e avvisare quando questo avviene.
   - Oppure si può fare la stessa cosa offline: nel momento in cui si crea il programma è possibile caricare fino a 1000 immagini, creando un database. Quando il programma sarà avviato ARCore tenterà di trovare tali immagini.
2. A questo punto è possibile avviare l'applicazione e muoversi con il dispositivo nell'ambiente che contiene le immagini caricate. ARCore cercherà costantemente di trovare una corrispondenza tra le texture che identifica e le immagini bersaglio che deve trovare, e quando riesce nel suo intento restituisce informazioni all'applicazione come spiegato nel passo 3.
3. ARCore restituisce il trackable corrispondente all'elemento fisico che ha fatto match con uno degli elementi da tracciare. Viene inoltre indicata la pose del trackable e un riferimento all'immagine che ha fatto match con tale trackable.

Quindi le augmented image possono essere usate esattamente come si usa un altro piano o punto e possono dare vita ad ancore cui è possibile agganciare oggetti.

- Condivisione: Esistono anche ancore in cloud (cloud anchors) per un'esperienza aumentata multiutente che prescinda dall'utilizo di un device Android o IOS. Un utente, che assume il ruolo di host, posiziona un ancora e invia a un cloud le informazioni riguardanti i feature point ad essa circostanti. Il cloud salverà queste informazioni e dopo averle processate creerà una cloud anchor che verrà restituita all'host. Nella cloud-anchor c'è un attributo chiamato cloud-anchor ID che identifica univocamente quell'ancora e può essere utilizzato per fare riferimento a quell'oggetto specifico. L'host potrà quindi condividere la cloud-anchor ID con i device guest, i quali potranno collegarsi al cloud e ottenere le informazioni

necessarie per partecipare all'esperienza di realtà aumentata. L'ultimo passo richiesto per iniziare tale esperienza è che i guest inquadrino l'area corrispondente a quella individuata dall'ancora.

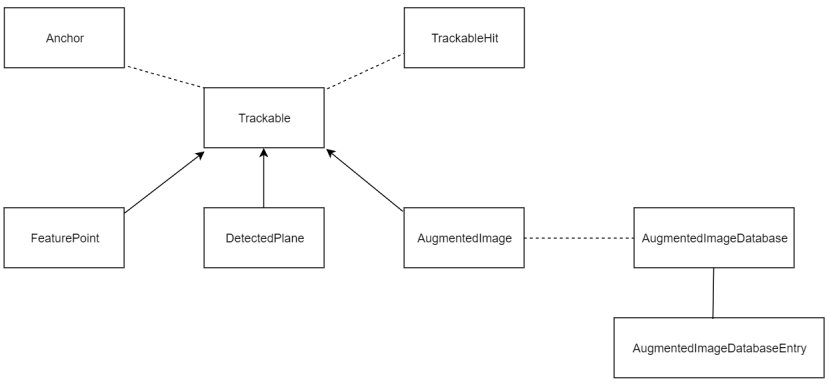
3.2 ARCore per Unity

Nonostante le API fornite da Google per le varie piattaforme contengano differenze di implementazione, si può individuare un percorso logico di base per utilizzare le funzionalità messe a disposizione da ARCore. Di seguito un breve elenco con gli step principali da seguire:

1. Creazione di una sessione
2. A ogni update possiamo richiedere alla sessione un frame, questo contiene riferimenti a:
   - L'oggetto rappresentante la videocamera
   - I metadati relativi alla videocamera
3. I frame ci forniscono le rappresentazioni dei trackable individuati e il loro stato
4. Possiamo usare i trackable per posizionare delle ancore
5. Possiamo usare le ancore per agganciare agli oggetti reali degli oggetti virtuali

3.2.1 Guida all'SDK messo a disposizione per Unity

Nota: non verranno mostrati tutti i metodi di tutte le classi messe a disposizione ma solo quelli fondamentali, cercando di mettere in risalto i collegamenti logici che esistenti tra i vari elementi messi a disposizione da ARCore.



Trackable:

- ha 3 sottoclassi: `FeaturePoint, DetectedPlane e AugmentedImage`;
- per ottenere la lista di uno dei 3 tipi di trackable è necessario richiamare il metodo `GetTrackable<T>` della classe `Session` dove T è il tipo di trackable che stiamo cercando. Avremo quindi rispettivamente: `Session.GetTrackables<AugmentedImage>(AugmentedImageList)`,

  `Session.GetTrackables<FeaturePoint>(FeaturePointList)`,

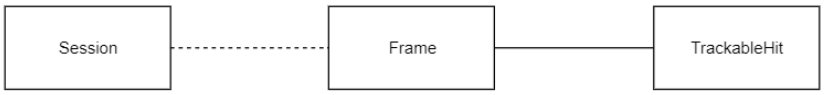
  `Session.GetTrackables<DetectedPlane>(PlanesList)`;
- Ha la proprietà `TrackingState` che ci permette di sapere se l'oggetto è attualmente tracciato o meno;
- Permette di accedere a tutte le ancore collegate ad esso attraverso il metodo `GetAllAnchors(List<Anchor> anchors)`, i riferimenti alle ancore verranno inseriti nella lista passata al metodo;
- Permette la creazione di una nuova ancora associata al trackable attraverso la funzione `CreateAnchor(Pose pose)`, il metodo richiede la posa dell'ancora, e restituisce l'oggetto ancora
- La struttura `TrackableHit`, che viene creata quando un hit test ha successo, contiene un riferimento al trackable che ha intercettato il raggio. Questa struct verrà analizzata piu approfonditamente in seguito;
- `AugmentedImage, FeaturePoint, DetectedPlane` contengono tutte un campo `Pose` che indica posizione e orientamento dell'oggetto;
- `AugmentedImage, DetectedPlane` contengono indicazioni relative alle dimensioni;
- `AugmentedImage` contiene i riferimenti all'immagine vera cui il trackable è assegnato

AugmentedImageDatabase: Contiene riferimenti alle varie immagini le cui informazioni sono memorizzate sotto forma di strutture `AugmentedImageDatabaseEntry`;

Anchor:

- Mantiene informazioni riguardo allo stato del tracking;

- Internamente permette di istanziare nuove ancore attraverso un metodo factory, ma il tutto è nascosto al programmatore che chiama il metodo opportuno della classe `Trackable`.



Session:

- Rappresenta la sessione ARCore, è una delle classi principali per fruire dei servizi messi a disposizione da ARCore toolkit;
- Contiene il campo `Status` che tiene traccia dello stato attuale della sessione che può essere non inizializzata, in fase di inizializzazione, in tracking, può aver perso il tracking, oppure non aver niente da tracciare o essere in una delle condizioni di errore;
- Permette di controllare se ARCore è installato sul dispositivo attraverso il metodo `CheckApkAvailability()` e, nel caso risultasse non installato, può richiederne l'installazione attraverso il metodo `RequestApkInstallation()`;
- Permette la creazione di ancore data una posa e un trackable attraverso il metodo `CreateAnchor(Pose pose, Trackable trackable)` che restituisce l'ancora creata. Da notare che questo metodo internamente chiama il metodo `CreateAnchor(Pose pose)` sull'oggetto trackable passato;
- Contiene il metodo `GetTrackables<T>(List<T> trackables, TrackableQueryFilter filter)` che fa quanto spiegato nella classe Trackable.

  `TrackableQueryFilter` può assumere i valori `ALL, NEW, UPDATED`

Frame:

- Fornisce uno snapshot dello stato di ARCore all'esatto momento in cui viene richiesto.
- Ha come campi:
  - `Pose` che indica posizione e orientamento della videocamera

  - `LightEstimate`, una struttura che contiene riferimenti a luminosità stimata dell'ambiente e color correction

- Ha 4 metodi che effettuano raycast ovvero che "sparano" un raggio in una certa direzione e valutano se questo raggio interseca dei trackable:

  ```
  Raycast(float x, float y, TrackableHitFlags filter,
  out TrackableHit hitResult)
  ```

  ```
  Raycast(Vector3 origin, Vector3 direction,
  out TrackableHit hitResult, float maxDistance,
  TrackableHitFlags filter)
  ```

  ```
  RaycastAll(float x, float y, TrackableHitFlags filter,
  List<TrackableHit> hitResults)
  ```

  ```
  RaycastAll(Vector3 origin, Vector3 direction,
  List<TrackableHit> hitResults, float maxDistance,
  TrackableHitFlags filter)
  ```

  I primi due metodi restituiscono solo il `TrackableHit` corrispondente alla prima collisione, mentre gli ultimi due quelli corrispondenti a tutte le collisioni;

  Nel primo e terzo metodo `x,y` fanno riferimento al punto del display toccato per dare origine al raggio, la direzione del terzo vettore è perpendicolare al piano della camera e uscente; nel secondo e nel quarto metodo invece `origin` indica il punto di origine del `raycast` e `direction` la direzione nelle coordinate del mondo;

  Il valore `filter` permette di indicare di quali trackable tenere conto nella ricerca delle collisioni;

TrackableHit: È una struttura che mantiene informazioni sull'esito del raycast, in particolare ha i campi:

- `Distance`: distanza dall'origine al punto di collisione;
- `Flags`: indica il tipo di trackable con cui il raggio ha avuto una collisione;
- `Pose`: indica la posa del punto in cui c'è stata la collisione;
- `Trackable`: è un riferimento al trackable con cui è avvenuta la collisione.

Capitolo 4

Caso di studio

4.1 Realtà Aumentata e beni culturali

Indipendentemente dal contesto in cui è inserita, per una applicazione di realtà aumentata, gli aspetti e le tecnologie abilitanti fondamentali sono: metodi di tracciamento e di registrazione; modellazione dell'ambiente virtuale; computer, schermi, dispositivi di input e tracciamento, e interfacce per le interazioni.

Le applicazioni di realtà aumentata nell'ambito del patrimonio culturale possono essere marker-based, markerless o usare un approccio ibrido. HMD ottici see-through, dispositivi handheld e schermi proiettati sono scelte comuni per la visione aumentata, mentre interfacce concrete e collaborative sono usate più spesso per interagire con informazioni virtuali, nonostante esistano alternative. I sistemi AR per il patrimonio culturale sono solitamente pensati per l'outdoor piuttosto che per l'indoor. Grazie alle recenti innovazioni tecnologiche è stato possibile combinare i due usi.

4.1.1 Applicazioni per il patrimonio culturale

I siti che costituiscono il patrimonio culturale acquisiscono un notevole valore aggiunto se integrati con mezzi digitali. Ciononostante, molti esperti d'arte reputano che l'utilizzo della tecnologia porti le opere ad essere relegate a semplice sfondo. Questo approccio spesso è determinato dal loro retaggio culturale e generazionale. Infatti c'è un forte scetticismo (tra coloro che non sono a loro agio con le nuove tecnologie) verso i benefici che queste possono apportare. Inoltre ci sono interrogativi riguardo a come la tecnologia vada usata. La tendenza, per quanto riguarda le applicazioni multimediali, è mostrare e affascinare gli utenti attraverso l'innovazione tecnologica piuttosto che concentrarsi nella risoluzione di specifici problemi computazionali.

In ogni caso è opinione comune che strumenti ottici, adatti a persone non esperte nell'utilizzo di dispositivi multimediali, possano essere utili nella diffusione dell'interesse verso la conoscenza di beni di importanza storico-artistica, soprattutto quando ci si rivolge a un pubblico particolarmente giovane. A conferma di ciò, si può notare che spesso le installazioni che non introducono nuove tecnologie a supporto dell'esperienza sono percepite dai visitatori come meno interessanti e attraggono una quantità di pubblico mediamente inferiore. Le esperienze di apprendimento che si avvalgono solo di etichette e descrizioni possono essere informative ma non sono interattive. Un ambiente intelligente, che reagisce alla presenza del visitatore adattandosi dinamicamente poiché supportato da tecnologie mobile, rende l'esperienza di visita più attraente, aprendo nuove strade sia verso il patrimonio culturale tangibile sia non tangibile.

4.1.2 Patrimonio culturale tangibile e non tangibile

Il "patrimonio culturale tangibile" si riferisce ad artefatti fisici della società, incluse creazioni artistiche, edifici storici, altri prodotti fisici pregni di una qualsiasi forma di valenza culturale.

Per contro, il "patrimonio culturale non tangibile" identifica pratiche, espressioni, conoscenze e abilità non fisiche che sono riconosciute come componenti essenziali del patrimonio culturale.

I sistemi di realtà virtuale immersivi hanno dato prova di essere una possibile soluzione quando si vuole fornire ai visitatori esperienze personalizzate mettendo a loro disposizione specifiche istruzioni rispetto ai percorsi da seguire e differenti interazioni in base al tipo di persona che si approccia all'opera. Nell'archeologia ad esempio, il problema della diffusione del patrimonio è spesso legato a quello di mostrare beni che possono essere seriamente danneggiati o del tutto persi. Le nuove tecnologie possono essere usate come una sorta di Raggi-X e mostrare ciò che è celato dal terreno o possono aumentare un ambiente ricostruendo virtualmente ciò che è andato perduto.

Considerando i possibili campi di applicazione di AR, MR, VR in un contesto di valorizzazione dei beni culturali, si possono identificare alcune classi che possono suddividere le applicazioni in base allo scopo per cui sono create:

- **Educazione:** mira all'apprendimento da parte degli utenti degli aspetti storici del patrimonio culturale reale e virtuale;
- **Potenziamento dell'esposizione:** mira a migliorare l'esperienza dei visitatori del museo fisico o del sito archeologico, di solito attraverso una visita guidata;
- **Esplorazione:** supporta gli utenti nella visualizzazione e nell'esplorazione del patrimonio culturale sia per come appare attualmente che per

come doveva apparire nei tempi passati, aiuta inoltre nell'interpretazione di ciò che viene visto così da ottenere una comprensione e conoscenza più approfondite;

- **Ricostruzione:** permette agli utenti di visualizzare e interagire con ricostruzioni storiche. Si differenzia dall'esplorazione per due caratteristiche fondamentali: non è pensato solo per esperti e l'interazione non è necessariamente orientata alla scoperta di nuove conoscenze;

- **Musei virtuali:** creano riproduzioni digitali di opere d'arte (tangibile e non) e le espongono al pubblico.

Alcune di queste categorie si sovrappongono. Per esempio una applicazione di ricostruzione digitale può essere utile all'utente anche per apprendere la storia degli elementi riprodotti, in questo modo educazione e ricostruzione coesistono. Allo stesso modo, una esibizione di museo virtuale potrebbe benissimo essere ospitata all'interno di un museo fisico e potenziarne l'offerta. Nonostante questo ogni applicazione avrà un aspetto che predomina sugli altri e che ne definisce la categoria principale.

4.1.3 Benefici delle applicazioni AR e MR in contesti culturali

Due delle potenzialità più importanti offerte da AR, MR e VR sono quelle che riguardano il modo in cui l'esperienza culturale viene strutturata e la conseguente possibilità di estendere la stessa a un pubblico più ampio. Finora infatti la fruizione dell'arte prevedeva che l'esperienza partisse dal visitatore che, spinto dal proprio interesse, cercasse di comprendere le opere mentre queste erano semplicemente esposte, eventualmente corredate da alcune informazioni descrittive. Tuttavia, negli ultimi anni ci si sta rendendo conto che questo tipo di fruizione dell'arte può non essere adatto a tutti, dal momento che presuppone che il visitatore abbia già di suo la curiosità nei confronti del mondo artistico e tutti gli strumenti per comprendere ed apprezzare ciò che viene esibito. E' tuttavia evidente quanto spesso questa assunzione non sia vera. Tendenzialmente le persone visitano musei e siti di interesse culturale in gruppi e non tutti sono necessariamente interessati all'esperienza in se per se: magari per via della loro età o perché non hanno avuto modo di approcciarsi correttamente al mondo artistico; è quindi importante che luoghi come musei e siti di interesse storico e culturale stimolino la curiosità in questi soggetti. Il modo per fare ciò è costruire la visita non mettendo al centro dell'esperienza le opere esposte, ma la categoria di utente che si vuole raggiungere. Per fare un esempio concreto: un bambino, un conoscitore dell'arte e un adulto non

esperto della materia non potranno mai apprezzare lo stesso tipo di esperienza e, mentre l'esperto sarà in grado di effettuare una visita piacevole, è probabile che gli altri due utenti ne rimangano delusi. Per i visitatori più giovani è infatti necessario costruire un percorso che unisca la scoperta dei beni artistici a una componente ludica, semplice, che permetta una forte interazione con quanto viene esposto, ma senza che il tutto venga appesantito da nozioni troppo complesse. Un visitatore adulto senza conoscenze pregresse specifiche invece difficilmente apprezzerebbe la stessa esperienza proposta ai ragazzi, probabilmente sarebbe interessato ad avere una quantità di nozioni maggiori e legata a quelle che possono essere le proprie esperienze quotidiane. Un esperto infine vorrebbe probabilmente poter accedere a informazioni specifiche su ciò che vede, con collegamenti di approfondimento apprezzabili solo da chi è in possesso delle opportune conoscenze di background sull'argomento. In tutto ciò inoltre non abbiamo tenuto conto di categorie speciali di utenti che potrebbero beneficiare di percorsi specifici, come persone affette da disabilità o persone anziane che sono poco avvezze all'uso delle nuove tecnologie: per entrambe le categorie è necessaria una particolare attenzione rispetto alle interfacce utente, in modo tale che le applicazioni siano accessibili e che questi visitatori possano fruire della visita e apprezzarla quanto gli altri. Proprio l'apprezzamento delle esperienze culturali è fondamentale, in primis per l'utente stesso che sarà stimolato a compiere nuove visite o cercare esperienze simili; e in secondo luogo per la diffusione fra la popolazione della fruizione del patrimonio artistico come pratica comune. I pareri positivi dei visitatori sono infatti un efficace stimolo per i loro conoscenti a provare le stesse esperienze, e questo è importantissimo in un paese come l'Italia che vive di turismo ed è ricco di beni culturali da conoscere ed esplorare.

4.2 Descrizione del caso di studio

L'applicazione richiesta si inserisce perfettamente nel contesto descritto in precedenza, viene infatti richiesto di creare un software per arricchire l'esperienza offerta ai visitatori di Rocca delle Caminate attraverso la visualizzazione di oggetti virtuali.

L'applicazione è commissionata da Ser.In.Ar., una società impegnata in ambito culturale e che si occupa, nei territori della provincia di Forlì-Cesena, della gestione di siti di particolare interesse storico come la rocca. Proprio tale edificio sarà l'ambiente fisico con cui interagirà l'applicazione.

4.2.1 Presentazione del contesto fisico

Rocca delle Caminate è un fortilizio completamente circondato da un parco: un ipotetico visitatore che intenda accedervi ha come punto di partenza della propria visita la parte più bassa del parco. Superata l'entrata, dovrebbe seguire un primo percorso pavimentato a ghiaia fino a raggiungere la base della rampa leggermente inclinata e con la pavimentazione in pietra che permette di raggiungere il portale di accesso al corpo della rocca vero e proprio.

Figura 4.1: Rampa di accesso alla corte interna del fortilizio. (tratto da [9])

Una volta oltrepassato il portale di accesso, il visitatore si trova nella corte interna che occupa buona parte dello spazio difeso dalle mura.

Figura 4.2: Ampia corte interna alle mura. (tratto da [10])

Dalla parte opposta, rispetto a quella usata per accedere al cortile interno, si trova poi la porta che permette di accedere al corpo principale dell'edificio. Il

piano terra e il primo piano sono attualmente suddivisi in più ambienti dedicati a sale riunioni, ambienti di ricerca e studi privati per cui di scarso interesse per il visitatore che si dirigerà verso la torre, che permette di raggiungere una grande terrazza panoramica.

Figura 4.3: La foto mostra la terrazza panoramica della rocca e il panorama circostante. (tratto da [11])

4.2.2 Requisiti

Come accennato in precedenza, la richiesta del committente è quella di produrre una applicazione che permetta di arricchire l'ambiente fisico con elementi virtuali sia statici che in movimento. Gli elementi dovranno essere visibili attraverso lo schermo di smartphone e tablet che avranno in esecuzione una applicazione in grado di riconoscere il luogo e fungere da finestra verso il mondo aumentato. L'esperienza dovrà essere multiutente per cui più utenti puntando i propri dispositivi nella stessa direzione dovranno essere in grado di visualizzare gli stessi elementi virtuali, orientati opportunamente in base alla loro direzione. Per quanto riguarda l'ambiente della terrazza panoramica il committente vorrebbe poi realizzare delle postazioni fisse, dotate di opportuna strumentazione che permettano di ammirare il paesaggio e contemporaneamente ricevere informazioni su ciò che l'osservatore sta vedendo.

4.2.3 Elaborazione della richiesta

In accordo con il committente si è deciso di sviluppare questo progetto in maniera incrementale. Da subito si sono evidenziati quattro passi fondamentali

per la produzione di tale applicazione:

- il primo prevede la creazione di una applicazione di mixed reality, world-scale, multiutente che permetta di visualizzare oggetti virtuali fissi in posizioni determinate in fase di design;
- il secondo passo porterà alla gestione di elementi virtuali che si muoveranno nell'ambiente fisico;
- il terzo passo permetterà l'interazione dei visitatori con tali elementi;
- si penserà poi alla realizzazione della visione panoramica aumentata da realizzare sul terrazzo.

In particolare, per quanto riguarda il primo passo, visto l'avvicinarsi del periodo natalizio, si è deciso di porsi come obiettivo quello di addobbare, con elementi virtuali a tema natalizio, tre aree della rocca corrispondenti alla rampa di accesso al cortile interno, la corte, il terrazzo panoramico. Riuscire a portare a termine questo primo obiettivo produrrà delle basi solide su cui costruire il resto dell'applicazione.

4.3 Analisi dei requisiti

Il progetto portato avanti nello svolgimento della tesi ha come obiettivo quello di creare una applicazione di mixed reality per dispositivi android in grado di collocare elementi virtuali statici in punti precisi del fortilizio Rocca delle Caminate.

Segue una analisi dettagliata di quelli che sono i requisiti necessari alla realizzazione di una applicazione di questo tipo.

4.3.1 Requisiti funzionali

L'applicazione che si vuole produrre rientra chiaramente nel campo della mixed-reality world-scale. E' necessario infatti creare oggetti virtuali che non si sovrappongano semplicemente al mondo fisico ma si inseriscano nel suo contesto, esattamente come farebbero gli oggetti fisici ad essi corrispondenti, dando quindi la sensazione di appartenervi. Al fine di ottenere questa resa visiva dobbiamo soddisfare i seguenti requisiti:

- In primo luogo abbiamo bisogno di costruire un mondo fittizio da sovrapporre a quello reale. Conoscendo la posizione reale corrispondente

a quella dell'origine del mondo virtuale potremo comprendere come posizionare gli elementi virtuali per ottenere la resa visiva desiderata. Modellando correttamente il mondo virtuale potremmo inoltre avere una rappresentazione degli oggetti fisici e questo ci permetterà in un secondo momento di gestire le interazioni dell'utente e degli oggetti virtuali con gli oggetti fisici. Avere una rappresentazione virtuale dello spazio fisico in cui ci muoviamo permetterà inoltre al sistema di tenere traccia della posizione dell'utente, il che è fondamentale per determinare cosa mostrare al visitatore.

- E' inoltre necessario riuscire a determinare con precisione le misure degli oggetti posizionati nel mondo reale, in modo che siano realistici se confrontati in dimensione agli oggetti fisici circostanti. La dimensione di tali oggetti deve inoltre essere coerente con la distanza da cui l'osservatore li sta guardando: se un utente si avvicina all'oggetto virtuale questo dovrà apparire sempre più grande fino ad essere troppo grande per comparire sullo schermo, allo stesso modo allontanandoci dall'oggetto esso deve rimpicciolirsi sempre piu fino a sparire. La precisione nel calcolo delle misure è fondamentale anche per determinare correttamente le distanza tra oggetti e quelle tra oggetti e utente.

- Per ottenere la sensazione di realismo ricercata è necessario mettere in campo accorgimenti come:

  - gestione di luci e ombre: le ombre sono fondamentali in quanto conferiscono tridimensionalità all'oggetto che le proietta, tuttavia vanno gestite con accortezza. In base alla posizione dell'ombra di un oggetto infatti siamo in grado di determinare la posizione della sorgente di luce che proietta quell'ombra e, quando parliamo di oggetti virtuali in un contesto di mixed reality, è necessario che la posizione dell'ombra dell'oggetto virtuale sia coerente con quella proiettata dagli oggetti fisici in modo da non indurre nell'osservatore un senso di straniamento.
  - gli oggetti virtuali devono sembrare inoltre illuminati allo stesso modo di quelli fisici. Un problema spesso evidente nelle applicazioni di realtà aumentata meno recenti era il fatto che gli oggetti sembravano eccessivamente luminosi / lucidi.
  - è inoltre necessario gestire correttamente le occlusioni. E se questa è una cosa semplice fra due oggetti virtuali o nel caso un oggetto virtuale occluda la vista di uno reale, non è altrettanto semplice quando accade l'opposto ovvero un oggetto virtuale deve essere coperto da uno fisico.

4.3.2 Requisiti non funzionali

Ai fini di rendere l'esperienza il più possibile piacevole per l'utente sarebbe importante:

- evitare di perdere la registrazione degli oggetti virtuali;
- rendere la posizione degli oggetti virtuali effettivamente stabile nello spazio;
- far si che l'applicazione sia rapida nel determinare la posizione dell'utente e quindi quella degli oggetti virtuali che lo circondano;
- evitare la necessità di una fase di calibrazione e rendere l'applicazione reattiva fin dai primi istanti;
- permettere all'applicazione di distinguere dinamicamente ciò che si trova fra lo schermo e l'oggetto virtuale al fine di renderlo invisibile quando questo viene coperto ai nostri occhi: pensiamo ad esempio a una persona che si sposta davanti a noi occludendoci momentaneamente la vista;
- poter gestire facilmente la posa degli oggetti ovvero la localizzazione degli stessi nel mondo e il loro orientamento.
- potere modificare facilmente gli oggetti che si decide di caricare, nonché la loro posa nel mondo, in modo da rendere questa applicazione facilmente riutilizzabile in altri contesti.

Capitolo 5

Progettazione e sviluppo

5.1 Scelte architetturali

5.1.1 Componenti fondamentali

Una volta determinati i requisiti si può procedere verso una prima fase di progettazione, in cui determineremo quali sono le funzionalità principali che l'applicazione dovrà supportare ai fini di soddisfare la richiesta. Questo ci porterà anche alla definizione di alcune componenti che si occuperanno di gestire una o più funzionalità e di scambiare informazioni con le altre parti di cui il programma è costituito.

Considerando i requisiti elencati in precedenza è semplice distinguere quattro tipi di funzionalità di cui sicuramente l'applicazione avrà bisogno:

- funzionalità che gestisca il caricamento delle caratteristiche che definiscono gli elementi virtuali da renderizzare;
- funzionalità che gestisca la memorizzazione di tutte le informazioni che rimangono "stabili" nel tempo, ad esempio come sono costruiti gli oggetti virtuali, le relazioni tra loro e i riferimenti agli oggetti istanziati;
- funzionalità che determini la posizione dell'utente nel mondo e gli spostamenti effettuati, anche in relazione agli elementi virtuali;
- funzionalità che permetta di renderizzare gli elementi virtuali in modo coerente al punto di vista dell'utente.

Individuate tali funzionalità possiamo passare alla definizione dei componenti che se ne occupano:

- avremo un componente "Loader" che si occupa del caricamento delle informazioni, le quali potrebbero essere specificate su un file o trasmesse all'applicazione in altra maniera;
- necessiteremo poi di un componente che memorizzi tali informazioni in strutture dati che ne consentano la tracciabilità. Il gestore delle strutture dati dovrà anche fornire opportuni mezzi per accedere e modificare le informazioni memorizzate, si pensi al caso di oggetti che si spostano nel tempo o anche all'eventualità in cui nuovi oggetti virtuali debbano essere istanziati;
- avremo poi un componente che si occuperà della gestione degli elementi virtuali e del modo in cui vengono visualizzati, quindi anche delle condizioni di luce o della loro grandezza affinché abbiano dimensioni coerenti con quelle degli oggetti reali;
- ci sarà poi un componente che si occupa di tenere traccia della posizione dell'utente.
- infine un "Controller" gestirà le varie fasi di vita dell'applicazione e il passaggio delle informazioni tra le parti di questo piccolo "sistema".

5.1.2 Scelta delle tecnologie

Una volta determinati i requisiti, il passo successivo è stato determinare quali tecnologie utilizzare per soddisfarli tutti nella maniera più semplice ma allo stesso tempo accurata possibile.

Dal momento che esistono tantissimi ambienti di sviluppo che consentono di creare mondi virtuali tridimensionali in cui riprodurre elementi reali si è deciso di rimandare la scelta di tale tecnologia a un secondo momento e di concentrarsi invece sul risolvere altri due problemi fondamentali:

- come determinare accuratamente la posizione degli oggetti virtuali e dell'utente nel mondo reale;
- come accoppiare mondo virtuale e mondo reale in modo che i due si sovrappongano perfettamente.

Per quanto riguarda la questione della localizzazione il primo pensiero è stato utilizzare il GPS, considerando il contesto outdoor in cui avrà luogo l'esperienza di mixed reality (tutti e tre gli ambienti in cui si vogliono posizionare gli elementi aumentati, sebbene siano all'interno dei confini della rocca, sono all'aperto). Sono stati quindi effettuati i primi esperimenti, ma ci si è presto resi conto che sebbene il GPS come tecnologia sia molto precisa, dispositivi

come i cellulari sono dotati di un hardware che non consente una localizzazione abbastanza accurata da descrivere con esattezza i piccoli movimenti che una persona può compiere all'interno del cortile di un palazzo. Spesso infatti il segnale non risulta accurato, gli spostamenti non vengono rilevati con precisione e di conseguenza è stato chiaro fin da subito che questa non è una tecnologia utilizzabile per il nostro caso di studi.

Continuando la ricerca di una tecnologia che supportasse tali funzioni per dispositivi Android mi sono poi imbattuto in ARCore. Come più approfonditamente descritto nel capitolo dedicato a questa tecnologia, ARCore è una piattaforma messa a disposizione da Google per lo sviluppo di applicazioni AR e MR. ARCore mette a disposizione funzionalità che consentono:

- motion tracking, per tracciare la posizione dell'utente nell'ambiente;
- environmental understanding, per determinare la presenza di piani e per il riconoscimento di marker;
- light estimation per stimare le condizioni di illuminazione dell'ambiente che circonda l'utente e sulla base di questo determinare come applicare la color correction sugli elementi virtuali in modo da renderli realistici.

Permettendo il riconoscimento dei piani, ARCore fornisce una rappresentazione virtuale di quelli che sono oggetti reali, così che lo sviluppatore possa usare tali oggetti per creare il proprio mondo su quello fisico. ARCore è risultato quindi essere uno strumento essenziale per mappare la nostra rappresentazione virtuale del mondo fisico sul mondo reale.

A questo punto è stato necessario trovare uno strumento che permettesse di tenere traccia della posizione relativa degli oggetti virtuali e fisici.

Google mette a disposizione sdk di ARCore per vari ambienti di sviluppo tra cui Unity. Proprio questo è stato scelto per lo sviluppo dell'applicazione in quanto fornisce strumenti necessari a lavorare agevolmente su un mondo virtuale:

- è possibile vedere una rappresentazione grafica di quanto modellato nell'ide;
- fornisce funzionalità per gestire in maniera intuitiva le lunghezze e quindi le dimensioni degli oggetti;
- consente una manipolazione e memorizzazione efficace della posizione e dell'orientamento degli oggetti sia rispetto all'origine, che rispetto ad altri oggetti presenti nel mondo virtuale.

ARCore e Unity collaborano inoltre per fornire una gestione accurata della posizione occupata dall'utente nel mondo virtuale: all'avvio dell'applicazione infatti il software stabilisce l'origine del mondo virtuale in base al posizionamento, all'orientamento e all'altezza del dispositivo e posiziona gli oggetti virtuali di conseguenza.

5.2 Sviluppo del prototipo

5.2.1 Posizionamento oggetti

Una volta stabilite le tecnologie si è potuti passare alla fase di sviluppo. Considerando il fatto che la funzione principale che deve assolvere l'applicazione sarà quella di generare oggetti virtuali in posizioni specifiche del mondo reale, il primo problema da risolvere è stato come attuare l'accoppiamento fra i due mondi. Se da un lato è infatti vero che ARCore fornisce funzionalità per determinare la presenza di superfici con cui lo sviluppatore può interagire, l'applicazione non ha modo di comprendere che tipo di superfici siano e quindi ad esempio di distinguere il piano di un tavolo da quello di un pavimento. L'utilizzo dei piani riconosciuti dall'applicazione è quindi semplice nel caso sia prevista l'interazione diretta dell'utente con l'applicazione nella scelta delle superfici su cui posizionare gli oggetti, ma non è così triviale quando gli oggetti devono essere posizionati "automaticamente" dal programma. Sarebbe infatti necessario determinare un modo per riconoscere la superficie del terreno in modo da accoppiarla con il piano virtuale corrispondente, ma ARCore non fornisce una funzionalità simile al momento attuale. Cercando di trovare una soluzione a questo problema, è stato chiaro fin da subito che in realtà ciò di cui si necessita per accoppiare i due mondi non è un intero piano quanto più un punto nello spazio tridimensionale, che possa essere accoppiato con un punto reale. Va detto che quando pensiamo di accoppiare un punto reale con uno virtuale non stiamo parlando solo di sapere a che posizione del mondo reale corrisponda un punto descritto dalle coordinate (x, y, z) del mondo fittizio: vogliamo sapere anche l'orientamento di tale punto, ovvero immaginandoci di posizionare l'osservatore proprio in tale punto vogliamo sapere in che direzione sia rivolto e stia osservando l'ambiente, in altre parole vogliamo un accoppiamento che tenga in considerazione i 6 gradi di libertà. Il primo punto del nostro mondo virtuale che ci viene in mente di provare ad accoppiare con un punto del mondo reale è naturalmente l'origine. Potendo posizionare l'origine in una posizione specifica del mondo reale, o quanto meno sapendo che essa corrisponde a un punto preciso dell'ambiente fisico in cui lavoriamo, sarebbe semplice posizionare tutti gli altri oggetti di conseguenza. Ci sono purtroppo due problemi fondamentali in questa strategia:

- ARCore posiziona automaticamente l'origine, definendo quindi il verso degli assi e l'altezza al momento del lancio del programma, memorizzando la posizione e l'orientamento (posa) che il dispositivo ha in un preciso istante;

- ARCore non permette di riposizionare l'origine del mondo: non c'è un modo per dire al programma "d'ora in poi considera la posizione e l'orientamento attuali come origine del mondo".

Quindi, considerando il fatto che l'applicazione è pensata per utenti non consapevoli, questo approccio non è attuabile,soprattutto perché tutti i punti del mondo virtuale dipendono dalla posizione dell'origine e quindi lo stesso discorso fatto per l'origine vale per gli altri punti. Dal momento che non possiamo far corrispondere un punto virtuale a uno reale stabilendo la posizione del primo, proviamo a fare il ragionamento opposto: cerchiamo di determinare la posizione e l'orientamento di un oggetto fisico stabile secondo il sistema del mondo virtuale. Questo è fattibile attraverso un approccio marker-based. Google mette a disposizione un sistema di gestione dei marker, che nel contesto ARCore vengono definite AugmentedImage: in fase di costruzione dell'applicazione si caricano le immagini che dovranno fungere da marker nel progetto e si crea, a partire da esse, un database che verrà poi usato dall'applicazione per determinare quando queste vengono rintracciate nell'ambiente circostante. Una volta determinate tali immagini, ARCore creerà un opportuno oggetto virtuale corrispondente e lo metterà a disposizione dell'utente che potrà usarlo come punto di riferimento per posizionare gli oggetti virtuale. Bisogna notare che ARCore tenta in continuazione di migliorare la propria percezione del mondo aumentato, quindi l'oggetto marker potrà subire degli assestamenti nel tempo a livello di posizione, questo tuttavia non comprometterà gli oggetti virtuali la cui posizione relativa rimane costante. L'oggetto virtuale collegato a un marker, e preso come punto di riferimento per la collocazione degli oggetti virtuali, è detto "ancora", poiché come detto la sua funzione è quella di "ancorare" un oggetto virtuale a una posizione reale.

5.2.2 Metodi di caricamento

Per quanto riguarda il caricamento degli oggetti si è deciso di procedere come indicato in seguito.

Considerando il fatto che, in questa prima fase di sviluppo, l'applicazione funzionerà interamente in locale, si è deciso di specificare quali oggetti l'applicazione debba caricare attraverso un file json. In particolare, in una prima parte del file verranno definiti alcuni oggetti contenenti una stringa che riporta

il nome associato alle immagini nel database contenente i marker. Questi oggetti sono memorizzati in un array, che è valore di una chiave precedentemente determinata: all'avvio l'applicazione aprirà il file json e richiederà di caricare gli oggetti specificando tale chiave.

```
"markers": [
  {
    "name": "000"
  },
  {
    "name": "007"
  }
],
```

Figura 5.1: Gli elementi di tipo ancora contengono una sola scritta che ne specifica il nome.

Nella seconda parte del file ci saranno altri array, uno per ogni marker, in cui verranno riportati degli oggetti contenenti le informazioni principali sui GameObject da istanziare. Per ogni GameObject verrà indicato almeno posizione, rotazione, scalefactor e prefab di cui sarà istanza.

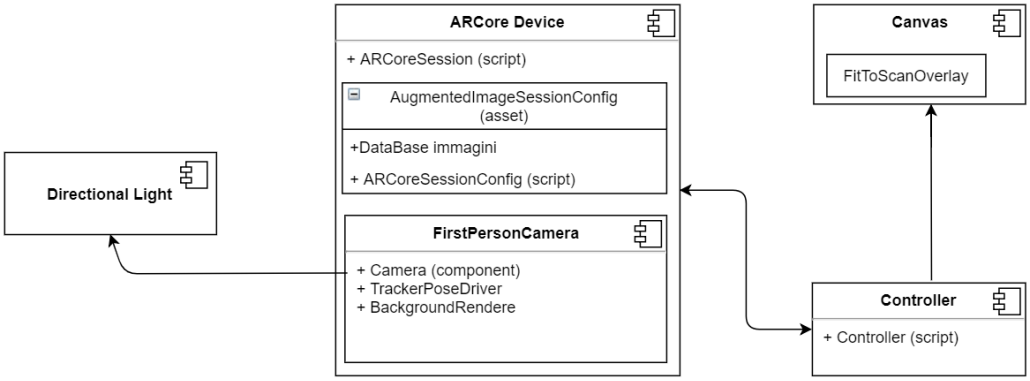
```
"007": [
  {
    "position": [ 0.0, 0.0, 0.0 ],
    "rotation": [ 0.0, 0.0, 0.0 ],
    "scaleFactor": [ 0.1, 0.1, 0.1 ],
    "prefabName": "elem1"
  },
  {
    "position": [ 1.5, 2.0, -1.0 ],
    "rotation": [ 45.0, 45.0, 45.0 ],
    "scaleFactor": [ 0.1, 0.1, 0.1 ],
    "prefabName": "elem2"
  }
]
```

Figura 5.2: Per ogni oggetto virtuale deve essere specificata: posizione, orientamento, sala rispetto al padre, nome del prefab da istanziare.

I prefab saranno contenuti nella cartella Asset/Resources ed è importante che per ogni oggetto la stringa valore del campo prefab corrisponda al nome di un prefab effettivamente esistente nella cartella Resources.

5.2.3 Componenti scena

Definito il principio utilizzato per posizionare gli oggetti possiamo passare a descrivere i componenti principali della scena Unity che caratterizza l'applicazione.



Come mostrato dalla figura, la scena si compone di quattro GameObject principali:

- `Canvas`: rappresenta uno schermo trasparente posto fra l'utente e il mondo della mixed reality. Permette di agganciare elementi a tale schermo in caso di necessità. Contiene un GameObject figlio `FitToScanOverlay` che è la rappresentazione di un mirino che costituisce una call to action per l'utente, al quale è richiesto di inquadrare un marker per iniziare l'esperienza;

- `Controller`: è il componente che contiene lo script che gestisce le fasi di vita dell'applicazione e stabilisce il suo comportamento principale. E' ad esempio il controller che si occupa di attivare o disattivare la visualizzazione del GameObject `FitToScanOverlay`, interagisce inoltre con `ARCore Device` da cui ottiene informazioni e immagini;

- `ARCore Device`: è forse il GameObject piu corposo. Contiene infatti:

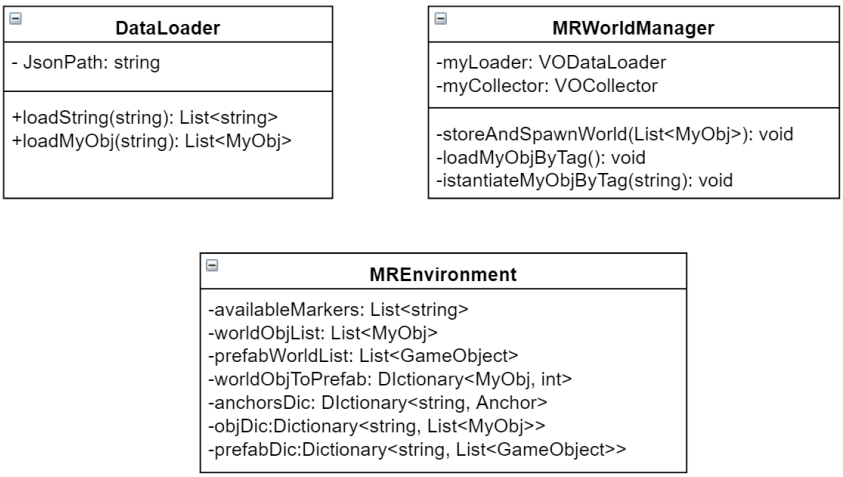
  - gli script che gestiscono la sessione;
  - l'asset `AugmentedImageSessionConfig` che gestisce il database delle immagini-marker;
  - il GameObject `FirstPersonCamera` che si occupa di recuperare le immagini (contiene infatti un componente che gestisce la videocamera del device), tracciare la posizione del dispositivo, renderiz-

zare il background. Dovendosi occupare fondamentalmente delle immagini è in comunicazione con `Directional Light`

- `Directional Light`: si occupa della gestione della luminosità ambientale.

5.2.4 Script

Il comportamento del sistema è determinato da tre classi principali:



DataLoader:

- si occupa del caricamento delle informazioni dal file json sfruttando l'asset UnityLitJson;
- al caricamento richiede che venga passata la stringa con il nome del file json da caricare. Notare che il file json deve essere collocato nella directory Assets/Resources .
- Il file mette a disposizione due metodi pubblici:
  - `loadString` permette di caricare la lista di oggetti contenenti una stringa specificando la chiave con cui cercare i valori nel file json. Viene usato per caricare i marker;
  - `loadMyObj` permette di caricare una lista di oggetti contenenti le informazioni sugli oggetti da istanziare. La stringa passata come argomento deve corrispondere al nome di un marker: gli oggetti caricati saranno figli di tale marker.

MREnvironment:

- gestisce il salvataggio delle informazioni relative ai GameObject

- permette di memorizzare l'elenco di marker disponibili (che devono essere passati all'oggetto MREnvironment al momento della creazione), l'elenco delle informazioni relative agli oggetti che vengono istanziati sia specificando le coordinate in base all'origine sia in base agli eventuali marker, tiene inoltre riferimenti per i GameObject corrispondenti

- Sebbene non specificati nell'immagine la classe mette a risposizione metodi per ottenere informazioni riguardo ai vari campi e per memorizzare nuovi dati nelle opportune collection (non sono stati specificati in quanto semplici metodi di set e get)

MRWorldManager:

- è l'elemento che definisce il comportamento vero e proprio della applicazione;

- contiene riferimenti a una istanza di DataLoader e una di MREnvironment.

- il comportamento dell'applicazione è descritto da due metodi.

  start(): Il MRWorldManager crea il DataLoader fornendogli il nome del file json da leggere; richiede poi al DataLoader di generare la lista contenente i marker, che passa all'oggetto MREnvironment al momento della crazione dello stesso; a questo punto, per ogni marker caricato, richiede al DataLoader la lettura della lista di oggetti da spawnare collegati al marker, oggetti che istanzierà e farà memorizzare all'oggetto MREnvironment.

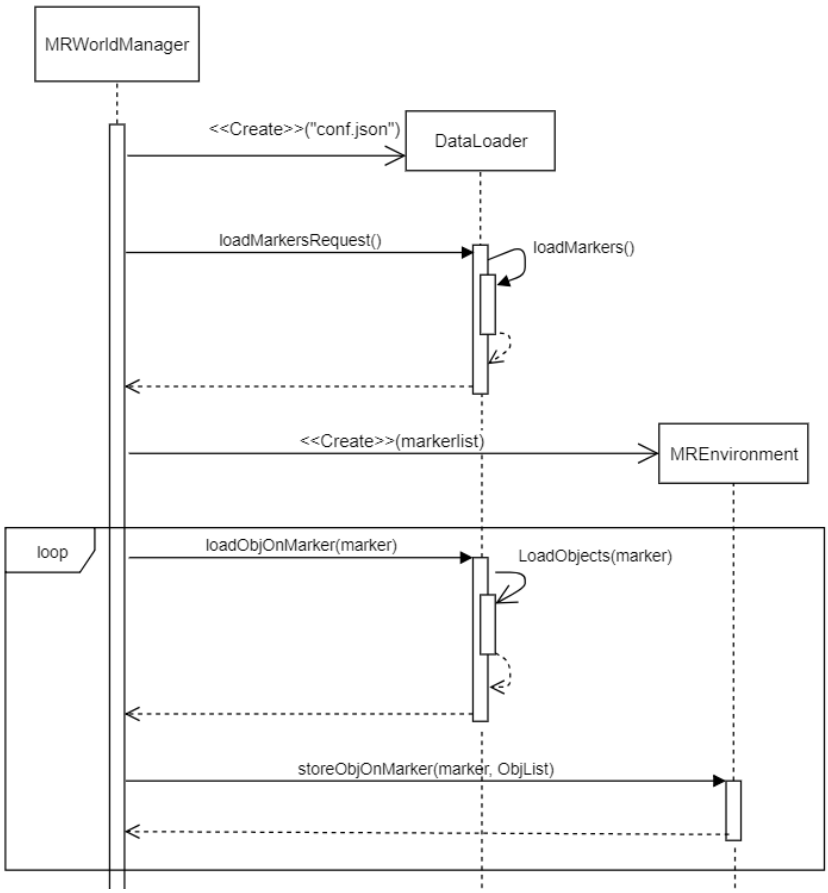


Figura 5.3: Illustrazione della fase di Start() dell'applicazione.

Update(): Ad ogni iterazione il MRWorldManager chiede all'istanza della sessione corrente di caricare le immagini usate come marker che sono presenti nella lista di tracking. A quel punto vengono passate in rassegna tutte le immagini trovate, nel caso ne sia stata individuata una che non era ancora stata scoperta la si utilizza per creare una nuova ancora.

```
// Update is called once per frame
void Update()
{
    if (!myMREnvironment.areAllAnchorsSet())
    {
        Session.GetTrackables<AugmentedImage>
            (m_TempAugmentedImages,
               TrackableQueryFilter.Updated);
```

```
        foreach (var image in m_TempAugmentedImages)
        {
            if (image.TrackingState == TrackingState.Tracking
                && myMREnvironment.getAvailableMarkerSafe()
                    .Contains(image.Name)
                && !new
                    List<string>(myMREnvironment.getAnchorDicKey())
                    .Contains(image.Name))
            {
                Anchor tempAnchor =
                    image.CreateAnchor(image.CenterPose);
                myMREnvironment.getAnchorDic().Add(image.Name,
                    tempAnchor);
                istantiateMyObjByTag(image.Name);
            }
        }
    }
}
```

Listato 5.1: Estratto del metodo `Update()` della classe MRWorldManager

5.3 Validazione del sistema

Il risultato raggiunto può dirsi soddisfacente in quanto rispecchia quelli che erano gli obiettivi da conseguire nella prima fase del progetto.

In condizioni di funzionamento ordinario infatti l'applicazione si comporta come segue:

- All'accensione inizia a determinare i piani che caratterizzano l'ambiente fisico circostante, individuando i relativi feature points;
- l'applicazione comincia poi la ricerca dei marker memorizzati nel database, e, quando li scopre, istanzia gli oggetti virtuali ad essi correlati;
- l'applicazione continua a tenere traccia della posizione degli oggetti virtuali e di quelli fisici mentre l'utente esplora il mondo aumentato, mostrando solo quelli che si trovano nel campo visivo dell'utente.

Non mancano tuttavia delle problematiche, ecco quelle fondamentali:

Perdita degli elementi tracciati: Per mantenere gli oggetti virtuali posizionati correttamente nel mondo virtuale il programma ha bisogno di tenere

continuamente traccia dei piani e dei feature points riconosciuti. Esistono tuttavia alcune condizioni ambientali che impediscono che ciò accada:

- parti di un ambiente o ambienti scarsamente illuminati;
- occlusione del campo visivo della videocamera (se ad esempio la scena viene coperta completamente da persone che si frappongono fra essa e il device);
- superfici completamente bianche o riflettenti o colorate in modo uniforme e lisce;

In tutti questi casi il programma non ha più modo di determinare la posizione degli oggetti fisici che aveva individuato e di conseguenza non sa cosa mostrare all'utente, per cui impedisce la visualizzazione degli elementi virtuali. Per ripristinare la vista della realtà aumentata sarà necessario inquadrare nuovamente i marker cui gli oggetti virtuali sono collegati, o eventualmente i piani a cui gli oggetti sono ancorati.

Occlusioni con elementi non tracciabili: Un altra condizione di criticità è quella in cui un oggetto fisico non tracciato dal programma, come può essere una persona, si pone in una posizione per cui dovrebbe essere parzialmente sovrapposto all'immagine virtuale. L'effetto ottenuto non sarà infatti quello desiderato dal momento che non vedremo parte dell'oggetto virtuale coperta dalla persona bensì avremo l'oggetto virtuale che copre la persona, nonostante esso in teoria sia a una distanza maggiore dalla videocamera. Questo fenomeno si verifica perché gli oggetti del mondo fisico vengono sempre posti sullo sfondo e non viene fatta distinzione fra oggetti che fanno parte del "paesaggio" e oggetti che si muovono in esso.

Accuratezza e stabilità: Attraverso prove empiriche effettuate si è notato che gli oggetti virtuali, sebbene posizionati attraverso le ancore, non siano del tutto immuni a un effetto, seppur temporaneo e limitato, di drifting. Ovvero capita che spostandosi attorno all'oggetto virtuale questo possa leggermente scivolare nella stessa direzione in cui ci stiamo spostando, per poi riassestarsi. L'entità di questo problema è maggiore in condizioni ambientali sfavorevoli, ovvero se l'ancora cui è legato l'oggetto virtuale si trova lontana da esso, e se la quantità di feature points individuata dal programma è scarsa ad esempio a causa di una debole illuminazione o per via di elementi che ostruiscono la corretta rilevazione dei piani. In ogni caso il fenomeno non è eccessivamente fastidioso per l'utente.

Figura 5.4: Un esempio dell'effetto visivo risultante

Purtroppo tutti questi problemi rilevati non sono originati dal modo in cui è stata progettata e sviluppata l'applicazione, ma sono dovuti alle difficoltà di riconoscimento dell'ambiente esterno presentata dall'sdk di Google ARCore, quindi non sono facilmente risolvibili. Nonostante questo nulla vieta, nel contesto di implementazioni future, di cercare di affinare le funzionalità messe a disposizione dall'sdk.

Conclusioni

Lo sviluppo di questa tesi ha portato all'elaborazione di una introduzione ad Google ARCore come tecnologia per lo sviluppo di applicazioni di mixed reality, con un approfondimento particolare sulle api fornite dall'sdk per lo sviluppo in ambiente Unity. Per quanto riguarda la parte di progetto l'esperienza può dirsi conclusa con successo dal momento che come risultato abbiamo ottenuto una applicazione che sfrutta l'sdk di Google ARCore per dare vita a una esperienza di mixed reality, come richiesto dal committente. Chiaramente tale applicazione dovrà essere ampliata nelle sue funzionalità per poter soddisfare tutte le richieste del committente, tuttavia già in questa prima fase sono presenti le funzionalità di base per l'integrazione di mondo reale e virtuale. Tra le funzioni che si potranno aggiungere abbiamo:

- gestione di elementi virtuali in movimento;
- aggiungere la possibilità di una interazione tra utente e elementi virtuali;
- riconoscimento di elementi fisici che non fanno parte dello sfondo, in modo da gestire opportunamente le occlusioni;
- rendere l'applicazione consapevole di come è composto l'ambiente fisico circostante caricando una rappresentazione 3D dello stesso, opportunamente posizionata.

Potrebbero inoltre essere migliorate le prestazioni per quanto riguarda aspetti come:

- tracking degli oggetti virtuali;
- stabilità degli oggetti virtuali;
- riconoscimento featurepoints in un contesto di scarsa luminosità;

In ogni caso reputo questa esperienza positiva, in quanto mi ha permesso di entrare a contatto con un mondo, quello della realtà aumentata, che mi ha sempre suscitato interesse. Inoltre ho potuto apprezzare quali sono le problematiche da affrontare quando ci si trova a lavorare con tecnologie nuove e quindi non del tutto consolidate.

Ringraziamenti

Non posso concludere questo percorso senza ringraziare chi mi ha sostenuto durante tutto il viaggio.

Le prime persone sono indubbiamente i miei genitori Elso e Brunella, che hanno subito pazientemente le mie crisi, incoraggiandomi e spronandomi ad andare avanti fino a gioire con me ogni qual volta abbia ottenuto un risultato. Insieme a loro non posso evitare di ringraziare mia nonna Pia che fornisce un supporto fisico indiscutibile alla famiglia.

Devo poi ringraziare i ragazzi con cui ho condiviso questi tre anni di corso.

Infine, ultimi ma non ultimi, tutti quegli amici che sono sempre presenti sia quando c'è bisogno di uno scherzo che di una parola, non sarei riuscito ad affrontare tutto questo senza di voi.

Bibliografia

[1] Dieter Schmalstieg, Tobias Höllerer, *Augmented Reality - Principles and Practice*, Addison Wesley, 2016.

[2] Jon Peddie *Augmented Reality - Where We Will All Live*, Springer, 2017.

[3] `https://buy.metavision.com/`

[4] `https://engt.co/2RlItlr`

[5] `https://bit.ly/2Fi5dxa`

[6] `https://rubygarage.org/blog/difference-between-ar-vr-mr`

[7] `https://techli.com/daqri-using-qr-codes-to-create-experiences/6749/`

[8] `https://developers.google.com/ar/discover/concepts`

[9] `http://www.roccadellecaminate.com/flyer.pdf`

[10] `https://bit.ly/2TUDWI7`

[11] `http://www.serinar.unibo.it/camminate-tra-note-parole-e-cultura/`

[12] `https://developers.google.com/ar/develop/unity/quickstart-android`

[13] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid, John J. Leonard *Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age*, IEEE TRANSACTIONS ON ROBOTICS, VOL. 32, NO. 6, DECEMBER 2016.

Nota: Sebbene non siano stati citati testualmente, lo studio che ha portato alla produzione dei capitoli uno e due si è basato principalmente sulle risorse [1] e [2].

Per quanto riguarda il terzo capitolo ci si è basati sul materiale fornito da Google sui siti [8] [12].