

ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

---

CAMPUS DI CESENA  
SCUOLA DI INGEGNERIA E ARCHITETTURA  
Corso di Laurea Magistrale in  
Ingegneria Informatica

SISTEMI HANDS-FREE BASATI SU  
SMART-GLASS A SUPPORTO DI ATTIVITÀ DI  
CUSTOMER CARE DI UN'IMPRESA

Tesi di laurea in  
Programmazione Concorrente e Distribuita LM

*Relatore*

Prof. ALESSANDRO RICCI

*Presentata da*

ANDREA LEARDINI

*Correlatore*

Dott. Ing. ANDREA SANTI

---

SESSIONE II  
ANNO ACCADEMICO 2014–2015



# PAROLE CHIAVE

Mobile Augmented Reality

Sistemi hands-free

Smart-glass

Google Glass

Gruppo Loccioni



Ai miei genitori, a Laura, a Cecilia e a Roberto



# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Mobile Augmented Reality</b>	<b>7</b>
2.1	Definizioni . . . . .	7
2.2	Panoramica storica . . . . .	9
2.3	Piattaforma computazionale . . . . .	11
2.4	Tracciamento e registrazione . . . . .	11
2.4.1	Tracciamento sensor-based . . . . .	12
2.4.2	Tracciamento vision-based . . . . .	13
2.4.3	Tracciamento ibrido . . . . .	14
2.5	Display . . . . .	15
2.5.1	Optical see-through display . . . . .	15
2.5.2	Video see-through display . . . . .	16
2.5.3	Surface projection display . . . . .	17
2.5.4	Altre classificazioni . . . . .	18
2.6	User Interface e interazione . . . . .	19
2.6.1	Tangible UI e puntamento 3D . . . . .	20
2.6.2	Haptic UI e gesture recognition . . . . .	21
2.6.3	Eye tracking . . . . .	21
2.6.4	Riconoscimento vocale . . . . .	22
2.6.5	Input testuale . . . . .	22
2.6.6	UI ibride . . . . .	22
2.7	Reti wireless . . . . .	23
2.7.1	Wireless Wide Area Network (WWAN) . . . . .	23
2.7.2	Wireless Local Area Network (WLAN) . . . . .	24
2.7.3	Wireless Personal Area Network (WPAN) . . . . .	24
2.8	Software framework . . . . .	24

<b>3</b>	<b>Sistemi hands-free e smart-glass</b>	<b>27</b>
3.1	Sistemi hands-free . . . . .	27
3.1.1	Smart-glass . . . . .	28
3.2	Nuove opportunità e nuove sfide . . . . .	29
3.2.1	Nuove opportunità . . . . .	29
3.2.2	Nuove sfide . . . . .	30
3.3	Ambiti applicativi . . . . .	32
3.3.1	Medicina e healthcare . . . . .	32
3.3.2	Militare . . . . .	32
3.3.3	Soccorso . . . . .	33
3.3.4	Automotive . . . . .	34
3.3.5	Turismo . . . . .	34
3.3.6	Intrattenimento . . . . .	35
3.3.7	Industria ed enterprise . . . . .	35
3.3.8	Uno scenario a lungo termine: Mirror Worlds . . . . .	36
3.4	Limitazioni e problematiche attuali . . . . .	38
3.4.1	Limitazioni tecnologiche . . . . .	38
3.4.2	Privacy e accettazione sociale . . . . .	39
3.5	Sistemi rappresentativi . . . . .	41
3.5.1	Google Glass Explorer Edition . . . . .	41
3.5.2	Epson Moverio BT-200 . . . . .	42
3.5.3	Vuzix Wrap 1200DXAR . . . . .	43
3.5.4	Sony SmartEyeglass . . . . .	44
3.5.5	ODG R-7 . . . . .	45
3.5.6	Microsoft HoloLens . . . . .	45
3.5.7	DAQRI Smart Helmet . . . . .	46
3.5.8	Altri sistemi rappresentativi . . . . .	47
<b>4</b>	<b>Google Glass</b>	<b>49</b>
4.1	Caratteristiche tecniche . . . . .	50
4.2	User Experience . . . . .	52
4.3	User Interface . . . . .	54
4.3.1	Timeline . . . . .	54
4.3.2	Card . . . . .	55
4.3.3	Immersion . . . . .	57
4.3.4	Menu contestuale . . . . .	57
4.3.5	Input vocale . . . . .	57



4.3.6	Touchpad . . . . .	59
4.4	Approcci di sviluppo . . . . .	61
4.4.1	Glass Development Kit . . . . .	61
4.4.2	Mirror API . . . . .	61
4.4.3	Glassware ibridi . . . . .	62
4.5	Pattern di riferimento . . . . .	63
4.5.1	Ongoing task . . . . .	63
4.5.2	Immersion . . . . .	72
4.5.3	Periodic notification . . . . .	74
4.6	Potenzialità e limiti . . . . .	76
<b>5</b>	<b>Lozioni Augmented Service</b>	<b>79</b>
5.1	Ecosistema software BU Energy . . . . .	79
5.1.1	Energy Management System . . . . .	80
5.1.2	MyLeaf . . . . .	83
5.2	Requisiti . . . . .	84
5.2.1	Gestione del cliente . . . . .	84
5.2.2	Localizzazione dell'asset . . . . .	86
5.2.3	Gestione dell'asset . . . . .	86
5.2.4	Gestione dell'intervento . . . . .	87
5.3	Architettura . . . . .	88
5.3.1	Gestione del cliente . . . . .	90
5.3.2	Localizzazione dell'asset . . . . .	91
5.3.3	Gestione dell'asset . . . . .	93
5.3.4	Gestione dell'intervento . . . . .	95
5.4	Ingegnerizzazione . . . . .	98
5.4.1	Strumenti tecnologici di riferimento . . . . .	98
5.4.2	Struttura dell'applicazione . . . . .	101
5.4.3	Gestione del cliente . . . . .	103
5.4.4	Localizzazione dell'asset . . . . .	106
5.4.5	Gestione dell'asset . . . . .	109
5.4.6	Gestione dell'intervento . . . . .	112
<b>6</b>	<b>Collaudo del sistema</b>	<b>121</b>
6.1	Collaudo delle funzionalità . . . . .	121
6.1.1	Navigazione verso Leaf Water 1 . . . . .	123
6.1.2	Geolocalizzazione sul ponte del fiume . . . . .	123

6.1.3	Streaming video su rete locale . . . . .	124
6.1.4	Procedure guidate su Leaf Water 1 e Spray Mo- mentum . . . . .	124
6.1.5	Collaudo delle altre funzionalità . . . . .	128
6.2	Valutazione del sistema . . . . .	128
<b>7</b>	<b>Conclusioni</b>	<b>131</b>

# Capitolo 1

## Introduzione

Oggigiorno la quasi totalità dei settori enterprise è pervasa da sistemi computazionali in grado di supportare –e talvolta sostituire– l’attività lavorativa umana. Tali sistemi sono utilizzati sia per realizzare il software di controllo di impianti, macchinari e altri strumenti di lavoro, sia per rendere la gestione dei workflow aziendali più efficiente, più efficace e meno costosa. I workflow e i processi che possono beneficiare dell’utilizzo di sistemi e dispositivi tecnologicamente avanzati riguardano un ampio spettro di attività, tra cui quelle che richiedono un’elevata mobilità del lavoratore. In questo ambito, le piattaforme computazionali comunemente utilizzate includono dispositivi handheld tradizionali come smartphone e tablet: su tali sistemi è possibile installare applicazioni special purpose di cui il lavoratore può avvalersi durante lo svolgimento di task specifici.

Negli ultimi anni gli ambienti enterprise stanno divenendo terreno fertile per lo studio e la sperimentazione di sistemi di *Mobile Augmented Reality* (MAR): l’adozione di tecnologie di realtà aumentata potrebbe consentire ad un lavoratore di accedere agevolmente a preziose informazioni di contesto in grado di agevolare lo svolgimento delle sue attività. Sebbene attualmente esistano molteplici framework di realtà aumentata per dispositivi handheld, un numero crescente di ricercatori ritiene che, per esprimere appieno il loro potenziale, i sistemi MAR necessitino non soltanto di nuove piattaforme hardware, ma anche di nuove modalità di interazione con l’utente, nuove interfacce grafiche e nuove metodologie di sviluppo software. Una delle

principali motivazioni per cui i dispositivi handheld tradizionali non risultano adatti ad applicazioni MAR enterprise è che, siccome in alcuni contesti lavorativi l'utente potrebbe aver le mani impegnate con attrezzature e strumenti, l'interazione con il sistema MAR potrebbe risultare in questi casi disagiata. I sistemi *hands-free* sopperiscono a questo inconveniente supportando tecniche di interazione alternative, che consentono all'utente di utilizzare il dispositivo senza dover impegnare le proprie mani.

Nel panorama delle piattaforme hardware emergenti, gli *smart-glass* rappresentano una tipologia di dispositivo che, oltre ad esibire funzionalità hands-free, risulta particolarmente adatta ad applicazioni di realtà aumentata. Il display di uno smart-glass (o *i display*, in caso di stereoscopia) è disposto infatti all'interno del campo visivo dell'utente, consentendogli di percepire le informazioni aumentate unitamente agli oggetti reali che vede con i propri occhi. La progettazione di applicazioni MAR per sistemi hands-free basati su smart-glass rappresenta oggi un tema al fronte della ricerca, che coinvolge trasversalmente molteplici discipline tra cui ingegneria del software, computer vision, interaction design e scienze cognitive. Questa tematica sta destando inoltre un crescente interesse in ambito industriale: alcune importanti società come SAP, EasyJet, Beckhoff e Schlumberger stanno equipaggiando i propri operatori con smart-glass al fine di valutare i benefici apportati dalla sinergia tra questa nuova piattaforma computazionale e le tecnologie di realtà aumentata emergenti.

L'obiettivo di questo lavoro di tesi è stato lo studio e la realizzazione di una soluzione hands-free basata su smart-glass in grado di supportare e agevolare alcune attività di customer care del Gruppo Loccioni<sup>1</sup>, una società multinazionale che si occupa dello sviluppo di sistemi automatici di misura e controllo per migliorare la qualità, l'efficienza e la sostenibilità di prodotti, processi ed edifici.

Nel 1968 Enrico Loccioni, fondatore dell'impresa, ha iniziato il suo percorso imprenditoriale come elettricista di impianti industriali. Negli anni '70 un nuovo ramo dell'impresa si è specializzato nel collaudo di elettrodomestici e di componenti per auto, portando l'azienda a valutare le potenzialità di mercato del controllo qualità e dell'automazione dei processi. Negli anni '90 l'impresa inizia a sviluppare

---

<sup>1</sup><http://www.loccioni.com/>

competenze nel settore delle telecomunicazioni, dell'ICT e del monitoraggio ambientale e, nel decennio successivo, anche in ambito medico e healthcare.

Il Gruppo Loccioni è composto da cinque business unit (BU) che, pur avendo differenti competenze e occupandosi di ambiti settoriali distinti, lavorano in forte sinergia tra loro:

- **Energy:** progetta, realizza e gestisce soluzioni per la sostenibilità e l'efficienza energetica
- **Environment:** sviluppa e integra sistemi per il monitoraggio ambientale e di processo, capitalizzando le competenze di misura, controllo, automazione, gestione e valorizzazione dei dati
- **Industry:** progetta e sviluppa soluzioni nel settore di componenti industriali, elettrodomestici, vetro e alimenti
- **Humancare:** si occupa dello studio e dello sviluppo di sistemi di healthcare, applicando le conoscenze cliniche e il know-how tecnologico dell'impresa
- **Mobility:** progetta e realizza sistemi per il collaudo, l'assemblaggio e il controllo qualità di componenti per l'industria automotive.

Ciascuna BU è suddivisa a sua volta in più team, ciascuno dei quali si occupa di attività specifiche della BU collaborando e coordinandosi con i membri degli altri team. La BU Energy in particolare, quella presso cui si è svolto questo lavoro di tesi, è formata da molteplici team, tra cui *Ricerca e sviluppo* e *Service*. Il team di Ricerca e sviluppo elabora e progetta soluzioni innovative in grado di apportare un valore aggiunto all'intera business unit; i membri del team Service svolgono invece attività di customer care, occupandosi dell'installazione e della manutenzione di impianti energetici sostenibili –denominati *asset*– presso la sede delle imprese clienti della BU Energy.

Interagendo e collaborando con alcuni membri del team Service è stato possibile comprendere come avviene attualmente la gestione degli interventi di manutenzione e individuare i principali punti deboli e le inefficienze del processo. Ad oggi le attività di manutenzione

sono gestite inviando presso la sede del cliente un operatore specializzato del team Service che, in caso di necessità, si avvale di documentazione tecnica cartacea e del supporto telefonico di un operatore remoto che si trova presso la sede del Gruppo. La gestione degli interventi può talvolta risultare poco agevole poiché l'operatore presso il cliente, mentre ha le mani impegnate con strumenti e attrezzature, potrebbe aver necessità di consultare un manuale tecnico, oppure di contattare un tecnico per ottenere assistenza durante l'intervento. In particolare, la collaborazione remota tramite telefonata tradizionale risulta spesso problematica, in quanto l'operatore remoto per fornire assistenza efficacemente spesso avrebbe necessità di vedere l'impianto da mantenere in tempo reale. Instaurare una videoconferenza invece di una telefonata tradizionale risolverebbe il problema solamente in parte, poiché l'operatore presso il cliente sarebbe costretto ad impegnare almeno una mano per sorreggere il dispositivo handheld che abilita la videoconferenza, intralciando di conseguenza l'attività di manutenzione.

Una possibile soluzione alla problematica descritta è rappresentata dall'adozione di un sistema hands-free basato su smart-glass su cui sia installato un applicativo che supporti l'operatore durante l'intero workflow di manutenzione, dal momento della presa in carico della richiesta fino al termine dell'intervento. Nel contesto di questo lavoro di tesi si è dunque progettato e sviluppato un sistema software per dispositivi hands-free che supporti i membri del team Service durante il workflow di manutenzione e riparazione degli impianti, mostrando informazioni aumentate rilevanti e puntuali al fine di rendere l'intero processo più efficiente ed efficace. L'applicazione sviluppata, denominata *Lozioni Augmented Service*, è stata realizzata in collaborazione con i team di Ricerca e sviluppo e Service della BU Energy.

La trattazione di questo lavoro di tesi è organizzata nei seguenti capitoli:

- Nel secondo capitolo si introduce il concetto di Mobile Augmented Reality, presentando una breve panoramica storica sui sistemi di realtà aumentata e classificando i componenti che costituiscono tipicamente un sistema MAR.

- All'interno del terzo capitolo si descrivono le principali caratteristiche di sistemi hands-free e smart-glass, si evidenziano le opportunità e le sfide che queste tipologie di sistemi introducono e si illustrano i principali ambiti applicativi che ad oggi beneficiano di queste tecnologie. Infine, si descrivono le problematiche che attualmente affliggono i sistemi hands-free e si illustrano alcuni sistemi commerciali basati su smart-glass.
- Il quarto capitolo riguarda i Google Glass, il dispositivo che è stato utilizzato per lo sviluppo dell'applicazione Loccioni Augmented Service; si presentano le caratteristiche tecniche e di design adottate dallo smart-glass, se ne descrive il modello di programmazione e infine se ne discutono potenzialità e limiti.
- Nel quinto capitolo si descrivono le fasi di sviluppo del sistema Loccioni Augmented Service, illustrando dapprima l'ecosistema software all'interno del quale l'applicazione è stata collocata, e presentando in seguito i requisiti del sistema, la sua architettura e i principali aspetti di ingegnerizzazione.
- Il sesto capitolo riporta i risultati della fase di collaudo del sistema realizzato, illustrando gli obiettivi raggiunti e discutendo le problematiche che non sono state risolte nel contesto di questo lavoro di tesi.
- Infine, all'interno del settimo capitolo si riportano alcune considerazioni conclusive sul lavoro svolto e si presentano alcuni spunti per eventuali sviluppi futuri.





# Capitolo 2

## Mobile Augmented Reality

In questo capitolo si introducono i concetti di *Augmented Reality* (AR) e *Mobile Augmented Reality* (MAR), si presenta una breve panoramica storica della loro evoluzione e si descrivono i sottosistemi che tipicamente compongono un sistema MAR.

### 2.1 Definizioni

Il significato del termine *Augmented Reality* deriva dal concetto astratto di *virtuality continuum*, proposto da Milgram e Kishino nel 1994 [1]. Il *virtuality continuum* può essere schematizzato visivamente con una linea (Figura 2.1), i cui estremi rappresentano rispettivamente un ambiente completamente reale ed un ambiente completamente artificiale; quest'ultimo rappresenta un'incarnazione di realtà virtuale (*virtual reality*, VR). I livelli intermedi di fusione di ambienti reali e virtuali si trovano tra i due estremi del continuum, e caratterizzano nel loro insieme il concetto di *mixed reality*. Gli ambienti rappresentati nell'intervallo della *mixed reality* dunque presentano sia oggetti reali che virtuali.

In prossimità dei due estremi del continuum trovano definizione i concetti di *augmented reality* e *augmented virtuality*: il primo rappresenta un ambiente reale a cui sono aggiunti oggetti ed informazioni virtuali, mentre il secondo rappresenta un ambiente virtuale a cui sono aggiunti oggetti ed informazioni reali. Mentre la *virtual reality* e l'*augmented virtuality* hanno lo scopo di creare un ambiente artificiale

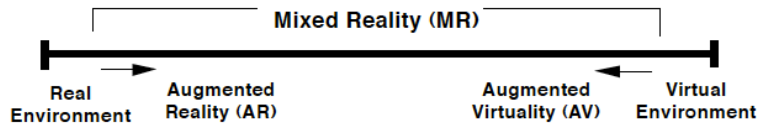


Figura 2.1: Il virtuality continuum.

che l'utente può percepire ed esplorare interattivamente, l'augmented reality ha invece l'obiettivo di supportare l'utente durante lo svolgimento di un task, arricchendo l'ambiente reale con informazioni utili e contenuti pertinenti.

In passato sono state proposte molteplici definizioni per il concetto di augmented reality; in questa trattazione si adottano quelle presentate in [2, 3], secondo le quali un sistema AR ha le seguenti caratteristiche:

- combina oggetti reali e virtuali in un ambiente reale
- correla e allinea visivamente le informazioni virtuali con gli oggetti reali, attraverso un processo denominato *registrazione*
- è interattivo, la sua esecuzione avviene in tempo reale.

Il valore apportato dall'impiego di tecnologie di realtà aumentata è dato dal potenziamento sensoriale sperimentato dall'utente. Un sistema AR infatti abilita nuove modalità di percezione e interazione con l'ambiente, fornendo informazioni che l'utente non potrebbe percepire attraverso i propri sensi al fine di supportare lo svolgimento di un'attività. Idealmente l'informazione aumentata dovrebbe coinvolgere tutti i sensi, non solamente la vista: i contenuti virtuali potrebbero essere sonori, tattili, persino olfattivi o gustativi.

Le recenti innovazioni apportate in ambito di tecnologia mobile, infrastrutture cloud e reti wireless hanno reso possibile la realizzazione di sistemi AR su dispositivi mobili, introducendo il campo di ricerca della *Mobile Augmented Reality* (MAR). La possibilità di poter utilizzare un sistema AR teoricamente ovunque ha suggerito innumerevoli scenari di utilizzo, abilitando lo studio e l'esplorazione di nuovi ambiti applicativi in precedenza impraticabili. D'altra parte, la portabilità di

un sistema AR introduce numerose sfide e problematiche, in quanto presuppone requisiti tecnologici molto più stringenti rispetto ad un sistema AR non portatile.

Un sistema MAR non deve necessariamente essere costituito da un unico device standalone che generi autonomamente tutte le informazioni da presentare all'utente; infatti possono essere presenti più livelli di *partecipazione ambientale* [4]. In un caso più *device-centric*, la totalità dell'informazione virtuale è generata e visualizzata su un unico dispositivo che l'utente indossa o porta con sé. In un caso intermedio, la funzionalità del sistema MAR è distribuita su più device; questo accade ad esempio qualora il sistema utilizzi tecnologie di connettività wireless come 802.11 (Wi-Fi) o Bluetooth. Infine, nel caso estremo in cui il dispositivo indossato dall'utente non disponga della potenza computazionale necessaria per generare e processare le informazioni da presentare, tale device (denominato *thin client*) potrebbe usufruire della potenza di calcolo di server remoti e servizi cloud, limitandosi a presentare le informazioni elaborate.

## 2.2 Panoramica storica

I primi prototipi di realtà aumentata furono sviluppati da Ivan Sutherland verso la fine degli anni '60 [5]. In questi sistemi si utilizzava un head-mounted display (HMD) per presentare immagini di grafica 3D. Nei decenni successivi la generazione di computer grafica divenne un importante ambito di ricerca, e iniziarono ad emergere le prime sperimentazioni riguardanti la realtà virtuale. Negli anni '70 e '80 alcune istituzioni proseguirono la ricerca su sistemi AR, tra cui lo U.S. Air Force Armstrong Laboratory, il NASA Ames Research Center, il Massachusetts Institute of Technology e la University of North Carolina at Chapel Hill. Nel 1986 Tom Furness sviluppò un HMD con sistema sonoro 3D integrato per piloti di aerei da combattimento, nell'ambito del progetto *Super Cockpit* dell'aeronautica militare statunitense [6]. Nello stesso periodo apparvero i primi dispositivi mobili, come il Sony Walkman e gli orologi da polso digitali. Negli anni '90 grazie alla continua miniaturizzazione dei componenti elettronici vennero intro-

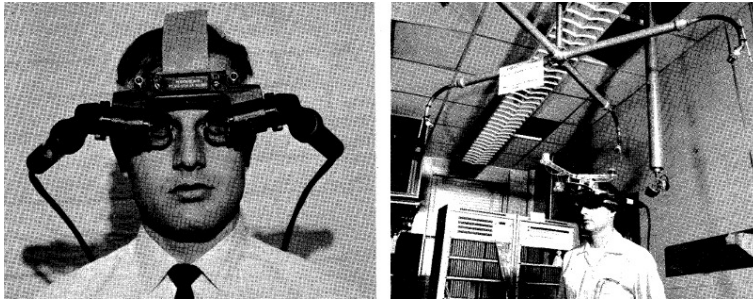


Figura 2.2: il primo prototipo di HMD, proposto da Ivan Sutherland.

dotti i primi computer palmari, segnando l'inizio dell'era del wearable computing [7].

Il termine *augmented reality* venne coniato solamente nei primi anni '90 da Caudell e Mizell [8], due scienziati della Boeing Corporation che svilupparono un sistema AR sperimentale per supportare i tecnici durante le operazioni di cablaggio di componenti degli aeromobili. Nella seconda metà degli anni '90 apparvero i primi survey riguardanti l'augmented reality, contenenti la definizione e la classificazione dei sistemi AR, le problematiche da risolvere e lo stato di avanzamento della ricerca, tra cui [1, 2, 3]. La realtà aumentata divenne un campo di ricerca a sé stante, numerose conferenze ebbero inizio e vennero istituite organizzazioni ad essa dedicate, tra cui l'*International Symposium on Mixed and Augmented Reality (ISMAR)*<sup>1</sup>.

Il primo prototipo di AR realizzato su un dispositivo mobile apparve nel 1997, quando Feiner et al. [9] svilupparono un sistema che sovrapponeva informazioni turistiche tridimensionali agli edifici e agli artefatti che i visitatori vedevano attorno a loro. Le più importanti innovazioni tecnologiche degli ultimi due decenni hanno preparato un terreno molto fertile per la ricerca su sistemi MAR: l'avvento di Internet, la costante miniaturizzazione dei componenti hardware e il successo della telefonia mobile hanno reso possibile lo studio e la prototipazione di applicazioni MAR su piattaforme moderne quali smartphone e tablet.

Nel 2013 Google ha rilasciato il primo prototipo di smart-glass a riscuotere un notevole impatto mediatico, i Google Glass. Attualmente,

<sup>1</sup><http://www.ismar-society.org/>

le tecnologie MAR stanno lentamente uscendo dai laboratori accademici e di ricerca per rivolgersi al mondo dell'industria, che sta dimostrando un interesse crescente verso soluzioni basate su smart-glass. Alcune importanti compagnie infatti si stanno dedicando alla progettazione di nuovi modelli di smart-glass, tra cui Vuzix, Epson, Sony e Microsoft, mentre altre stanno adottando questo tipo di dispositivi al fine di migliorare il flusso di processo aziendale [10, 11, 12].

## 2.3 Piattaforma computazionale

I prototipi MAR sviluppati nei primi anni 2000 tipicamente utilizzavano un notebook come piattaforma computazionale; l'eccessivo ingombro costituiva uno dei principali ostacoli per la diffusione di questo tipo di sistemi. Nello stesso periodo altre piattaforme utilizzate erano costituite da Personal Digital Assistant (PDA) e i primi tablet PC basati sul sistema operativo Windows. In seguito si iniziarono ad impiegare anche Ultra mobile PC (UMPC), telefoni cellulari, smartphone e sistemi hands-free, come ad esempio smart-glass. Questi ultimi sono diventati una delle piattaforme predominanti per lo sviluppo di sistemi MAR, grazie alla loro compattezza, alla presenza di molteplici sensori built-in e allo scarso ingombro.

## 2.4 Tracciamento e registrazione

Il processo di tracciamento e registrazione del sistema si occupa di valutare la posizione e l'orientamento dell'utente al fine di associare ed allineare visivamente i contenuti virtuali con gli oggetti fisici del mondo reale. Il processo può essere implementato attraverso tecnologie sensor-based, vision-based, o ibride. Nel primo caso si impiegano sensori inerziali, magnetici, elettromagnetici o ultrasonici. I metodi vision-based utilizzano feature visuali, impiegando dei marker o delle caratteristiche estratte dalle immagini acquisite dal sistema MAR. I metodi ibridi combinano differenti tecnologie al fine di ottenere un processo di tracciamento e registrazione più affidabile e robusto.

### 2.4.1 Tracciamento sensor-based

I metodi di tracciamento sensor-based compiono una elaborazione ad anello aperto: l'errore di tracciamento non può essere valutato ed utilizzato per correggere il processo. Inoltre, questa tipologia di tracciamento soffre di problematiche legate al rumore introdotto dai sensori e da eventuali interferenze. I metodi di tracciamento sensor-based si suddividono in inertial-based, magnetic-based, electromagnetic-based e ultrasonic-based, in funzione del tipo di sensore utilizzato. Molti sistemi MAR moderni combinano tipi di sensore differenti al fine di migliorare i risultati del processo di tracciamento.

#### **Inertial-based**

La maggior parte dei sensori inerziali misura il valore dell'accelerazione che, integrato due volte, fornisce informazioni sulla posizione e la direzione. La principale problematica di questo metodo è causata dalla propagazione dell'errore dovuta alla doppia integrazione e alla presenza di jitter di misurazione. In letteratura sono stati proposte alcune tecniche per migliorare l'accuratezza di questo metodo di tracciamento.

#### **Magnetic-based**

I sensori magnetici impiegano il campo magnetico terrestre per ottenere l'orientazione. Questo metodo è tipicamente combinato con un metodo di tracciamento della posizione per ottenere un tracciamento a sei gradi di libertà. I metodi magnetic-based sono affetti da problemi di interferenze in caso di campi magnetici ambientali. Questo può accadere in presenza di quantità importanti di metallo nell'ambiente circostante, ad esempio all'interno di un edificio in cemento armato.

#### **Electromagnetic-based**

I metodi electromagnetic-based tracciano la posizione basandosi sul *time of arrival* (TOA), sul *received signal strength indicator* (RSSI) o sulla differenza di fase di segnali elettromagnetici. In ambito MAR sono utilizzati molteplici metodi electromagnetic-based. La tecnologia

GPS (Global Positioning System) è ampiamente utilizzata in applicazioni outdoor, ma una delle sue problematiche principali è l'inevitabile schermatura di tutti gli ambienti indoor. Inoltre, la tecnologia GPS ha una precisione dell'ordine di alcuni metri, rendendola inadatta ad applicazioni in cui è richiesto un tracciamento accurato. Altri metodi electromagnetic-based proposti in letteratura impiegano tecnologie Wi-Fi, Ultra Wideband (UWB), Radio Frequency Identification (RFID), infrarossi e Bluetooth.

### **Ultrasonic-based**

I metodi ultrasonic-based sono in grado di stimare informazioni sia sulla posizione che sulla velocità. I sensori ultrasonici sono sensibili al rumore ambientale, alla temperatura e alle occlusioni. Si tratta di un metodo di tracciamento molto accurato, ma in applicazioni MAR moderne è raramente utilizzato, poiché è stato gradualmente sostituito da altre tipologie di sensori.

## **2.4.2 Tracciamento vision-based**

Per stimare informazioni sulla posa del sistema MAR, i metodi vision-based tracciano alcune feature dei fotogrammi acquisiti dalla videocamera del dispositivo. Si tratta di un'elaborazione ad anello chiuso: il risultato del tracciamento è utilizzato come feedback per correggere dinamicamente l'errore introdotto. A seconda della tipologia di feature tracciate, è possibile distinguere metodi marker-based, model-based e metodi basati su feature naturali.

### **Metodi marker-based**

I metodi di tracciamento marker-based impiegano fiducial marker come feature artificiali da tracciare. Un fiducial ha geometria e proprietà note a priori, che lo rendono facilmente identificabile. I metodi marker-based sono impiegati prevalentemente in applicazioni indoor, poiché spesso non è possibile disporli e mantenerli in ambienti outdoor ampi o sconosciuti.

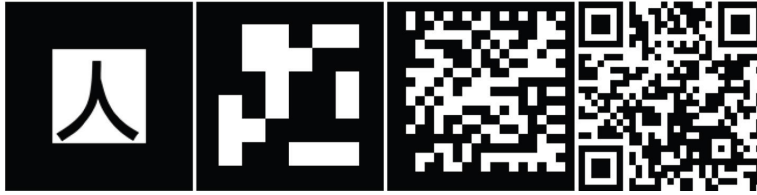


Figura 2.3: Alcune tipologie di fiducial markers. Da sinistra a destra: Template, BCH, Data Matrix, QR.

### Metodi model-based

Le tecniche di tracciamento model-based utilizzano un modello delle feature dell'oggetto da tracciare, come ad esempio un modello CAD o un template 2D. La corrispondenza tra il modello e la scena reale tipicamente viene elaborata usando linee e bordi come feature: gli algoritmi di edge-detection odierni sono computazionalmente efficienti e robusti a variazioni dell'illuminazione ambientale.

### Metodi basati su feature naturali

La posa della videocamera può essere determinata tracciando feature naturali quali punti, linee e texture contenute nella scena reale. Uno dei vantaggi di questo tipo di tecnica è che non richiede alcuna informazione a priori sull'ambiente. Le feature sono estratte ed elaborate in ogni frame acquisito: in questo modo è possibile correggere le incongruenze e gli errori di deriva di cui soffrono la maggior parte dei metodi sensor-based. Uno dei limiti dei metodi basati su feature naturali è la loro sensibilità alla distorsione dell'immagine acquisita, alle occlusioni della visuale e alle variazioni di illuminazione ambientale. Inoltre, questa tipologia di tracciamento è computazionalmente costosa: questa caratteristica rappresenta un problema rilevante per applicazioni MAR che richiedono prestazioni real-time su dispositivi mobili, spesso dotati di limitata potenza computazionale.

### 2.4.3 Tracciamento ibrido

Ciascuna metodologia di tracciamento presenta vantaggi e svantaggi. Combinare opportunamente metodi sensor-based e vision-based



può aiutare a risolvere le limitazioni di ciascuna tecnica, e quindi ad ottenere un processo di tracciamento più robusto ed accurato.

## 2.5 Display

Nel seguito si descrivono brevemente le principali tipologie di display che un sistema MAR può adottare, in accordo con la tassonomia proposta in [13].

### 2.5.1 Optical see-through display

Utilizzando un optical see-through display le informazioni virtuali sono proiettate su uno schermo semitrasparente e semiriflettente, in modo che l'utente sia in grado di vedere sia la scena reale che quella virtuale. Lo schermo in questione può essere solidale con il capo dell'utente, come nel caso di head-mounted display o di smart-glass, oppure può trovarsi ad una distanza sufficientemente ridotta dall'utente, come nel caso di un head-up display (HUD) installato sul parabrezza di un'automobile o nell'abitacolo di un aereo da combattimento.



Figura 2.4: Confronto tra HMD, smart-glass e HUD.

Un optical see-through display presenta i seguenti vantaggi:

- lascia intatta la risoluzione del mondo reale
- è più sicuro rispetto ad altre tipologie di display, poiché in caso di improvvisa interruzione dell'alimentazione l'utente continua a vedere la scena reale.

D'altra parte, gli svantaggi legati all'utilizzo di optical see-through display sono i seguenti:

- la scena reale e quella virtuale presentano talvolta un livello ridotto di luminosità, rendendo questo tipo di display meno adatto per l'utilizzo outdoor
- il campo di vista (*field of view*) della scena virtuale tipicamente è minore di quello della scena reale
- la gestione delle oclusioni tra oggetti reali e virtuali può risultare problematica.

### 2.5.2 Video see-through display

Questo tipo di display può essere utilizzato in due modalità. La prima consiste nell'impiegare un dispositivo HMD e sostituire l'occhio dell'utente con una o più videocamere per acquisire la scena del mondo reale. Il video acquisito viene miscelato con il contenuto virtuale ed inviato al display del HMD per la visualizzazione. La seconda modalità utilizza l'hardware di un dispositivo handheld, come ad esempio un tablet o uno smartphone; la videocamera built-in del dispositivo acquisisce il video che viene in seguito miscelato con l'informazione virtuale prima di essere visualizzato sul display.



Figura 2.5: Un esempio di video see-through display.

I vantaggi apportati dai video see-through display sono i seguenti:

- la miscelazione dei contenuti reali e virtuali avviene in maniera più semplice, così come la gestione delle occlusioni tra oggetti reali e virtuali
- la luminosità relativa delle due scene può essere gestita opportunamente
- la digitalizzazione della scena reale consente un miglior tracciamento dell'utente e una migliore registrazione del sistema.

Tuttavia, l'impiego di video see-through display comporta i seguenti svantaggi:

- la scena del mondo reale ha una risoluzione limitata
- il campo di vista è limitato
- l'utente potrebbe provare una sensazione di disorientamento a causa della distanza presente tra la videocamera e i suoi occhi.

### 2.5.3 Surface projection display

I recenti progressi in ambito di miniaturizzazione e consumo di potenza dei proiettori ha incentivato il loro impiego in applicazioni MAR. Questo tipo di dispositivi consente di proiettare contenuti virtuali sulla superficie di un oggetto reale, come ad esempio un muro, un foglio di carta o il palmo di una mano.

Un surface projection display presenta i seguenti vantaggi:

- gli utenti non devono indossare alcun dispositivo aggiuntivo
- per ottenere un ampio campo di vista è sufficiente proiettare le informazioni virtuali su una superficie ampia
- la superficie di proiezione può avere struttura, forma e colore arbitrari.

Gli svantaggi principali sono i seguenti:

- l'impiego di questo tipo di display è limitato ad applicazioni indoor a causa della bassa luminosità delle immagini proiettate

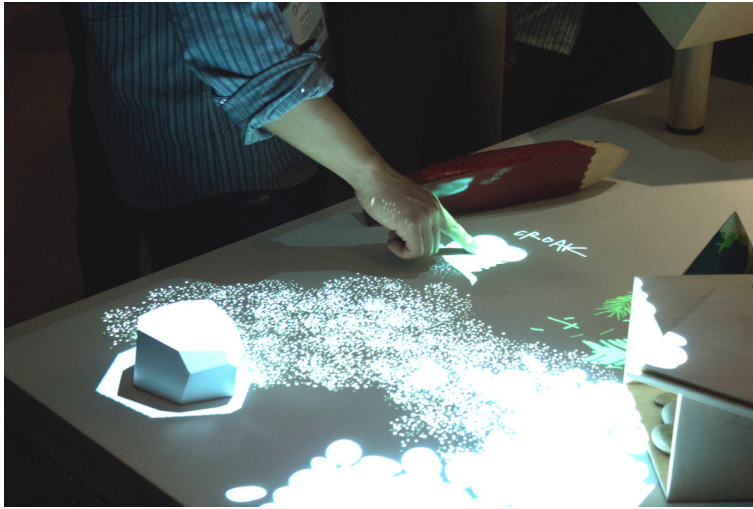


Figura 2.6: Un esempio di surface projection display.

- il proiettore deve essere ricalibrato ogniqualvolta cambi la superficie di proiezione o la distanza da esso
- i problemi di occlusione sono difficilmente risolvibili.

#### 2.5.4 Altre classificazioni

La tipologia di display di un sistema MAR può essere qualificata anche secondo criteri differenti da quello appena illustrato: nel seguito si presentano due ulteriori metodi di classificazione.

##### Display stereoscopico vs singolo

Un display stereoscopico è dotato di due schermi, ciascuno dei quali è visibile da un solo occhio dell'utente. Un sistema MAR munito di display stereoscopico è in grado di attribuire l'illusione di tridimensionalità ai contenuti virtuali che l'utente vede nel proprio campo visivo.

Un display singolo può essere condiviso da entrambi gli occhi dell'utente, oppure può essere visibile solamente da uno di essi. La presentazione di un'unica immagine ad entrambi gli occhi rende i contenuti

virtuali di un display singolo meno realistici rispetto a quelli generati da un display stereoscopico.

### Display immersivo vs non immersivo

Per chiarezza di esposizione, in questo lavoro di tesi un display è definito *immersivo* se consente un'associazione diretta tra oggetti reali e virtuali attraverso la loro sovrapposizione nel campo visivo. Un display immersivo dunque ha tipicamente un campo di vista sufficientemente ampio, ed è disposto centralmente rispetto al campo visivo dell'utente.

Un display non immersivo ha caratteristiche opposte: non è in grado di sovrapporre contenuti virtuali ad oggetti reali, solitamente è disposto nella periferia del campo visivo dell'utente ed è dotato di un campo di vista di ampiezza limitata.

## 2.6 User Interface e interazione

Al fine di consentire all'utente finale di interagire con i contenuti virtuali in maniera intuitiva ed efficace è necessario adottare appropriate tecniche di interazione studiate appositamente per applicazioni AR. La metafora *WIMP* (Windows, Icons, Menus, Pointing) bidimensionale utilizzata nelle UI per ambienti desktop convenzionali non è infatti adeguata per sistemi MAR, sia perché l'interazione con l'ambiente aumentato necessita di 6 gradi di libertà, sia perché l'ingombro di dispositivi tradizionali quali mouse e tastiera ridurrebbe notevolmente la user experience. E' necessario dunque un nuovo paradigma di interazione che consenta di compiere agevolmente molteplici tipi di azione, quali ad esempio:

- selezionare, posizionare e ruotare oggetti virtuali
- disegnare percorsi e traiettorie 3D
- assegnare valori quantitativi a proprietà di oggetti virtuali
- comporre e gestire input testuale.

Nel seguito si presentano i principali approcci proposti in ambito di UI e interazione con ambienti aumentati, considerando quanto presentato in [14, 15].

### 2.6.1 Tangible UI e puntamento 3D

Il termine *Tangible User Interface* (TUI) rappresenta una categoria di tecniche di interazione e dispositivi che consentono di interagire con contenuti digitali attraverso la manipolazione di oggetti fisici. Esempi di TUI per sistemi MAR comprendono dispositivi a forma di bacchetta o paletta, che realizzano un sistema di puntamento 3D analogo a quello di un mouse tradizionale. Utilizzando lo stesso approccio il sistema AR può associare un oggetto virtuale a qualunque oggetto fisico sia in grado di riconoscere: le azioni che l'utente compie sull'oggetto reale (ad esempio spostamento o rotazione) sono mappate sull'oggetto virtuale ad esso accoppiato. Alcune dispositivi TUI hanno marker o placeholder su di essi, in modo che il sistema AR possa sostituirli visivamente con oggetti virtuali.



Figura 2.7: Un esempio di tangible UI: *Reactable*<sup>2</sup>.

---

<sup>2</sup><http://www.reactable.com/>

### 2.6.2 Haptic UI e gesture recognition

Le TUI che consentono una comunicazione bidirezionale basata sul senso del tatto sono denominate *Haptic UI*. Per quanto riguarda l'interazione di input, un'interfaccia tattile può essere in grado di misurare posizioni, spostamenti e l'applicazione di forze. L'output di una haptic UI può simulare una resistenza o un impatto, ma anche comunicare una sensazione di ruvidezza, rigidità o una variazione di temperatura. Questa categoria di dispositivi comprende haptic device, data glove e joystick.



Figura 2.8: A sinistra: un haptic device. A destra: un data glove.

L'utilizzo di data glove può consentire, oltre al tracciamento della posizione della mano dell'utente, anche il riconoscimento di gesti, grazie ai sensori contenuti all'interno del dispositivo. Ad ogni modo, il movimento delle mani e il componimento di gesti possono essere tracciati anche visivamente, senza che l'utente debba indossare un data glove o un dispositivo simile. A questo scopo è possibile impiegare ad esempio una videocamera solidale con il capo dell'utente, montata sul colletto di un indumento oppure integrata in un bracciale.

### 2.6.3 Eye tracking

L'oculometria è una tecnologia che abilita interessanti modalità di interazione e, se sufficientemente precisa, può costituire un valido sistema di puntamento per il sistema MAR. La direzione dello sguardo può essere tracciata utilizzando delle videocamere che inquadrano le pupille dell'utente, opportunamente calibrate per filtrare eventuali movimenti oculari involontari.

### 2.6.4 Riconoscimento vocale

La possibilità di interagire con il sistema MAR attraverso un'interfaccia a comandi vocali costituisce un'interessante opzione soprattutto in contesti in cui l'utente ha necessità di rimanere concentrato sull'attività che sta svolgendo. In questi casi infatti l'utilizzo di interazioni basate su gesture recognition o su oculometria potrebbero costituire una soluzione non ottimale, poiché il compimento di una gesture impegnerebbe le mani dell'utente, mentre l'eye tracking lo costringerebbe a distogliere lo sguardo dall'attività che sta svolgendo. Una delle principali problematiche di questa tecnologia è che potrebbe risultare inefficace in ambienti rumorosi.

### 2.6.5 Input testuale

Attualmente è problematico implementare un meccanismo di input testuale rapido ed affidabile su un dispositivo mobile. Dispositivi come tastiere tradizionali, compatte, pieghevoli e a proiezione laser non sono adatte in ambito MAR poiché risultano ingombranti e disagiati. Ad oggi, né il riconoscimento di gesti glove-based, né quello vision-based offrono una sufficiente accuratezza per poter essere impiegati in contesti reali. Una delle proposte più promettenti è l'utilizzo di *chorded keyboard*: si tratta di tastiere compatte dotate di un numero limitato di pulsanti, che richiedono una combinazione di più tasti per inserire un singolo carattere (si veda Figura 2.9). Il riconoscimento vocale negli ultimi anni ha sperimentato notevoli miglioramenti e, se affiancato ad un sistema di input di fallback (ad esempio una *chorded keyboard*) potrebbe costituire in un prossimo futuro un valido metodo di input testuale.

### 2.6.6 UI ibride

Ciascuna modalità di interazione appena descritta presenta sia vantaggi che svantaggi: un sistema MAR di successo potrebbe impiegare una UI multimodale al fine di avvalersi dei pregi di molteplici approcci. Un'opportuna combinazione di differenti modalità di interazione potrebbe offrire all'utente una UI più naturale, robusta e predicibile.





Figura 2.9: Un esempio di chording keyboard: *Twiddler 3*<sup>3</sup>.

## 2.7 Reti wireless

Nell'ambito dei sistemi MAR le reti wireless sono utilizzate principalmente per la trasmissione di dati, per la connettività Internet e per stimare la posizione dell'utente. Nel seguito si descrivono le caratteristiche principali delle reti wireless maggiormente utilizzate; in Figura 2.10 si ha una comparazione delle tecnologie più rappresentative di ciascuna tipologia di rete [13].

### 2.7.1 Wireless Wide Area Network (WWAN)

Le reti WWAN sono adatte per applicazioni ad elevata mobilità; le principali implementazioni sono basate su differenti tecnologie, tra cui 2G (GSM e CDMA), 2.5G (GPRS), 3G (UMTS) e 4G (LTE). Le reti di nuova generazione hanno una larghezza di banda maggiore ed una minore latenza. WWAN è la tipologia di rete wireless più utilizzata, in quanto è attualmente l'unica tecnologia disponibile nella quasi totalità degli ambienti pubblici.

---

<sup>3</sup><http://twiddler.tekgear.com/>

### 2.7.2 Wireless Local Area Network (WLAN)

Le reti WLAN offrono una larghezza di banda maggiore e una latenza minore rispetto alle tecnologie WWAN. Questo tipo di rete gode di ampia popolarità, ma disponendo di un'area di copertura ridotta, è impiegato quasi esclusivamente per applicazioni indoor.

### 2.7.3 Wireless Personal Area Network (WPAN)

Le reti WPAN sono state progettate per interconnettere tra loro dispositivi quali telefoni cellulari, PDA e computer. Le principali implementazioni WPAN sono Bluetooth, ZigBee e UWB. Questo tipo di rete ha un basso consumo energetico e un'elevata larghezza di banda, ma non è adatta ad applicazioni con elevata mobilità.

Type	Technology	Coverage	Bandwidth(bps)	Latency(ms)	Power(mw)
WWAN	GSM	~35km	60K	high	1000~2000
	CDMA	250km~350km	384K	high	200~100
	GPRS	~10km	56K~114K	high	~1000
	UMTS	1km~2km	2M	medium	~250
WLAN	Wi-Fi	~90m	11M~54M	low	~100
	MIMO	~100m	300M	medium	unknown
WPAN	UWB	10m~100m	20M~1G	low	20~1000
	Bluetooth	~10m	1M~3M	medium	1~2.5
	ZigBee	~75m	20K~250K	low	1~100

Figura 2.10: Comparazione delle principali tipologie di reti wireless per applicazioni MAR.

## 2.8 Software framework

La maggior parte dei sistemi MAR ha la necessità di implementare alcuni task di basso livello, come ad esempio il tracciamento dell'utente e la visualizzazione di contenuti virtuali. Queste funzionalità possono essere supportate da software framework indipendenti, che consentono agli sviluppatori di focalizzarsi sulle funzionalità di alto livello dell'applicazione MAR mettendo a loro disposizione un insieme di strumenti come API, SDK e librerie. A partire dalla fine degli anni '90 sono stati proposti molteplici software framework in ambito accademico e, negli ultimi anni, anche in ambito industriale. Tra i framework che hanno sperimentato una notevole diffusione vi sono ARToolKit [16],

StudierStube [17], DWARF [18], D'Fusion [19], Wikitude [20], Vuforia [21] e Metaio [22].

Esistono inoltre numerose applicazioni che consentono ad utenti senza abilità di programmazione di visualizzare e pubblicare contenuti di realtà aumentata, tra cui Layar [23], Blippar [24], Aurasma [25] e Junaio [26].



## Capitolo 3

# Sistemi hands-free e smart-glass

Le piattaforme computazionali maggiormente utilizzate negli ultimi anni per lo sviluppo di applicazioni MAR sono senza alcun dubbio smartphone e tablet. Tuttavia recentemente una nuova piattaforma sta destando un interesse sempre maggiore sia in ambito accademico che industriale: si tratta degli smart-glass, una particolare categoria dei sistemi hands-free che si indossano come un paio di occhiali da vista o da sole e offrono funzionalità di realtà aumentata analoghe a quelle degli head-mounted display.

In questo capitolo si presentano i vantaggi derivanti dall'adozione di dispositivi hands-free –in particolare smart-glass– e si descrivono sia le nuove opportunità di ricerca e di mercato che questa tipologia di dispositivo offre, sia le nuove sfide da affrontare per realizzare sistemi hands-free di successo. In seguito si analizzano gli ambiti applicativi di maggior interesse e si esaminano le principali problematiche legate all'adozione ed alla diffusione di questo tipo di sistemi. Infine, si presentano alcune soluzioni commerciali basate su smart-glass, riportandone le principali caratteristiche e il target di mercato.

### 3.1 Sistemi hands-free

L'evoluzione dei sistemi AR sta promuovendo la progettazione e lo sviluppo di dispositivi che supportano modalità di interazione non

convenzionali. I sistemi hands-free rappresentano attualmente lo stato dell'arte nel settore della realtà aumentata: la loro caratteristica distintiva è quella di consentire all'utente di interagire col dispositivo AR senza dover impegnare le proprie mani. Questa caratteristica implica alcune importanti conseguenze. In primo luogo, mentre l'utente fruisce le informazioni aumentate, può utilizzare le proprie mani per proseguire l'attività che sta svolgendo nel mondo reale invece che impegnarle per interagire con esso. Inoltre, il fatto che l'utente non debba maneggiare il dispositivo AR mentre lo utilizza significa che è meno distratto da esso; la sua attenzione rimane focalizzata sul task che sta compiendo.

### 3.1.1 Smart-glass

Le recenti innovazioni tecnologiche apportate in ambito di mobile computing e optical see-through display hanno reso possibile lo sviluppo di sistemi MAR hands-free sempre meno ingombranti ed intrusivi, al punto che oggi è ormai possibile indossarli come dei normali occhiali da sole. Gli smart-glass rappresentano la naturale evoluzione di un head-mounted display: si tratta di un dispositivo hands-free che consente all'utente di vedere le informazioni aumentate all'interno del proprio campo di visione. Questa caratteristica permette inoltre all'utente di mantenere il contatto visivo con l'attività che sta svolgendo, limitando notevolmente la distrazione derivante dall'utilizzo e la consultazione del dispositivo.

In tutti i contesti in cui la sicurezza e la minimizzazione dell'errore umano rappresentano requisiti di primaria importanza questa caratteristica assume un valore ancora maggiore: si pensi ad esempio ad un operatore che esegue una delicata procedura di manutenzione su un impianto industriale ed ha le mani impegnate con strumenti e attrezzature, oppure ad un medico che esegue un intervento chirurgico, o ancora ad un agente di commercio alla guida che, utilizzando un navigatore tradizionale, deve continuamente distogliere lo sguardo dalla strada per seguire le indicazioni stradali. In queste ed altre circostanze l'adozione di smart-glass potrebbe apportare notevoli benefici all'utente, consentendogli di rimanere concentrato sull'attività che sta svolgendo e permettergli al contempo di fruire di informazioni

aumentate puntuali, che possono aiutarlo a svolgere tale attività più efficacemente ed efficientemente.

## 3.2 Nuove opportunità e nuove sfide

La ricerca sui sistemi hands-free emergenti sta aprendo la strada ad importanti opportunità sia in ambito accademico che industriale. L'eventuale successo commerciale di questi sistemi, oltre a modificare drasticamente il mercato dell'elettronica di consumo, potrebbe avere inoltre importanti ripercussioni in ambito sociale e culturale. Al contempo la realizzazione di sistemi hands-free di successo impone il confronto con nuove sfide e nuove problematiche da risolvere.

### 3.2.1 Nuove opportunità

Attualmente, alcune importanti aziende tra cui Google, Epson, Vuzix, Sony e Microsoft stanno proponendo le prime soluzioni prototipali basate su smart-glass con target enterprise e consumer. Nonostante le tecnologie MAR attuali non siano ancora del tutto mature, è innegabile che si stia manifestando un interesse crescente verso questo tipo di dispositivi [10, 11, 27]. Secondo le previsioni del sito specializzato *AugmentedReality.Org* le vendite di smart-glass nel 2017 raggiungeranno i 50-100 milioni di unità vendute, per poi arrivare alla cifra di un miliardo entro la fine del decennio [28, 29] (si veda Figura 3.1). Inoltre *Digi-Capital*, una banca d'investimento per media e servizi digitali, stima che il mercato dei dispositivi di realtà aumentata genererà nel 2020 un fatturato di 120 miliardi di dollari [30] (si veda Figura 3.2). Dunque non è affatto improbabile che entro alcuni anni la tecnologia impiegata in sistemi MAR hands-free sarà sufficientemente matura da poter permettere lo sviluppo e la diffusione di soluzioni e applicazioni che siano in grado di apportare un valore significativo in molteplici ambiti applicativi. Inoltre, esiste la possibilità che la diffusione di questo tipo di dispositivi nel mercato di massa possa generare un enorme successo commerciale e mediatico, similmente a quanto avvenuto in passato con l'avvento dei Personal Computer, di Internet, e della telefonia mobile.

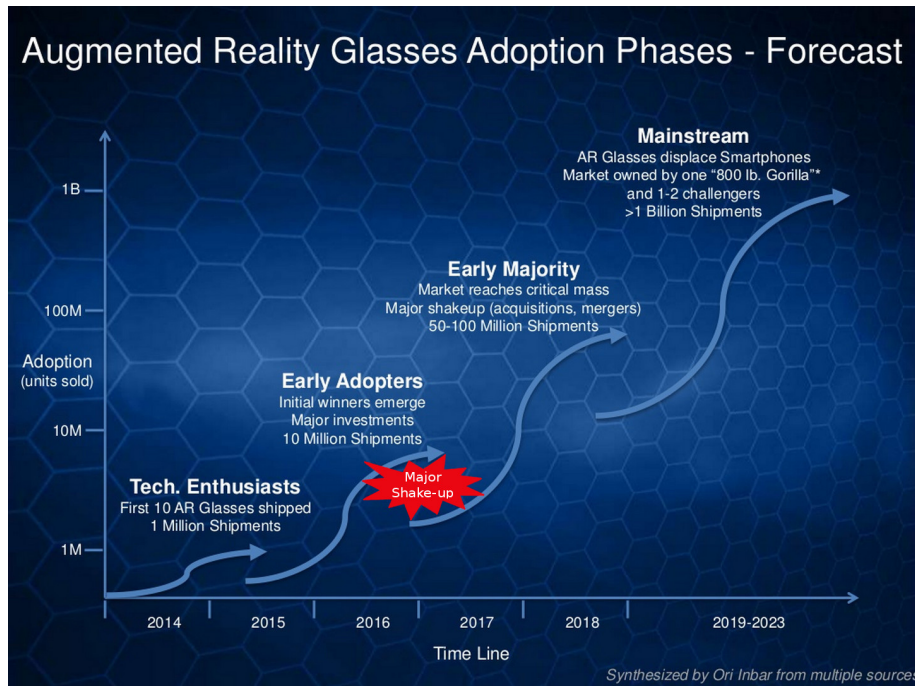


Figura 3.1: Previsione del numero di smart-glass venduti entro la fine del decennio.

### 3.2.2 Nuove sfide

Progettare un sistema hands-free o uno smart-glass che fornisca all'utente le informazioni di cui ha bisogno nel momento esatto in cui ne ha bisogno, ma che al tempo stesso non lo distraiga e non lo intralci non è semplice, in quanto richiede il soddisfacimento di molteplici requisiti. In primo luogo, è necessario che l'intrusività e l'ingombro del dispositivo siano minimi, al punto che, idealmente, l'utente non lo percepisca come uno strumento esterno, ma come parte di sé, proprio come accade con un normale indumento o un orologio da polso. Per quanto riguarda lo sviluppo del software installato sul sistema hands-free, è probabile che sia necessario rinnovare gli approcci e i paradigmi di programmazione impiegati attualmente in applicazioni per piattaforme tradizionali e hand-held, al fine di elaborare nuove astrazioni software che catturino opportunamente i requisiti e le caratteristiche peculiari delle applicazioni hands-free del prossimo futuro.

Un ulteriore importante requisito per un sistema hands-free di suc-



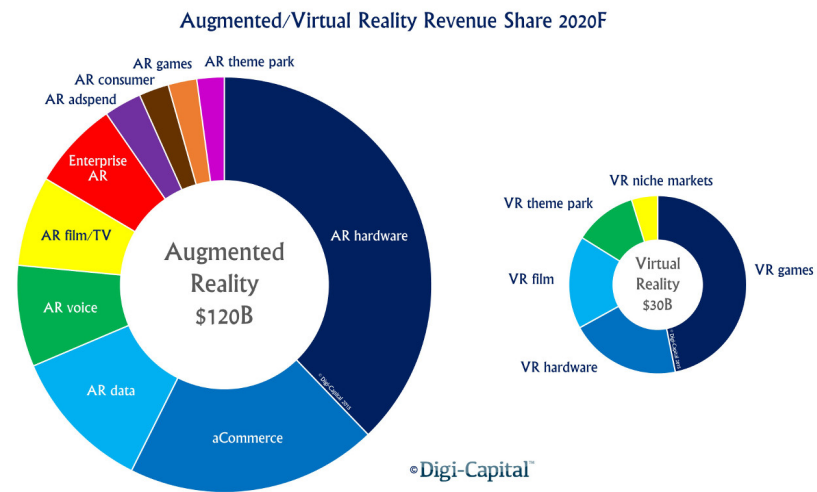


Figura 3.2: Previsione del fatturato generato dal mercato di dispositivi AR e VR nel 2020.

cesso è che i contenuti virtuali siano percepiti dall'utente nella maniera più naturale possibile, affinché la miscelazione delle informazioni reali e aumentate siano elaborate dal cervello umano come un unico flusso cognitivo coerente. Per raggiungere questo scopo è pressoché indispensabile il coinvolgimento di alcune discipline non appartenenti all'ambito dell'Information Technology –in primis le scienze cognitive– al fine di rendere l'informazione aumentata facilmente processabile dal sistema cognitivo umano.

La progettazione e la realizzazione di sistemi e applicazioni hands-free rappresenta dunque una sfida interdisciplinare che richiede, oltre allo studio e l'adozione di approcci completamente nuovi di progettazione, anche la definizione di paradigmi di programmazione appropriati per modellare e strutturare l'architettura, l'interazione e la GUI di tali applicazioni, al fine di rendere l'esperienza utente il più efficace e coinvolgente possibile. L'adempimento dei requisiti appena presentati può aiutare a rendere il dispositivo di realtà aumentata non solamente uno strumento utile, ma una vera e propria *estensione mentale* dell'utente, in grado di potenziare ed amplificare i suoi sensi e le sue facoltà psicofisiche. Utilizzando il sistema hands-free l'utente può infatti vedere e sentire cose che normalmente non potrebbe percepire, così come

disporre di memoria virtualmente illimitata.

### 3.3 Ambiti applicativi

Ricercatori e sviluppatori continuano ad indagare nuovi campi di applicazione che potrebbero beneficiare dell'utilizzo di sistemi di realtà aumentata di ultima generazione. L'innovazione tecnologica apportata a sistemi hands-free e smart-glass infatti crea continuamente nuove opportunità di impiego in molteplici ambiti applicativi.

#### 3.3.1 Medicina e healthcare

Medici, infermieri e altro personale medico potrebbero avvalersi di sistemi hands-free in molteplici circostanze. In letteratura sono stati proposti alcuni approcci prototipali per la sovrapposizione in tempo reale di radiografie e risonanze magnetiche sul corpo di un paziente [31]. Indossando uno smart-glass di ultima generazione, un chirurgo potrebbe vedere nel proprio campo visivo informazioni aumentate che gli consentono di sapere esattamente dove eseguire un incisione con bisturi, dove è situato un tumore o dove si trovano i vasi sanguigni che potrebbero causare emorragie. Applicazioni di questo tipo richiedono ovviamente un'elevata accuratezza di calibrazione e registrazione del sistema AR.

Recentemente sono state sviluppate alcune applicazioni per Google Glass volte a supportare persone affette da alcune patologie, quali autismo e Parkinson [32, 33]. In novembre 2014 un team di chirurghi di Firenze ha impiegato per la prima volta al mondo degli smart-glass durante un'operazione di prelievo di fegato [34]: il dispositivo è stato utilizzato per inviare in tempo reale alcune immagini dell'intervento a personale medico non presente in sala operatoria e per la creazione di video didattici per la formazione e l'addestramento di specializzandi.

#### 3.3.2 Militare

I primi sistemi di realtà aumentata di successo trovarono applicazione in aeronautica militare, con l'introduzione degli HUD utilizzati dai piloti di aerei da combattimento. Nel 2010 un progetto di ricerca

della DARPA ha avviato lo sviluppo di un sistema MAR che consente la visualizzazione di icone tattiche nel campo visivo di un soldato [35, 36]. Il display del dispositivo dunque segnala la presenza di alleati, postazioni nemiche e target da raggiungere nell'ambiente circostante. Il sistema prototipale realizzato è in grado di funzionare in ambienti outdoor sconosciuti, è immune a disturbi elettromagnetici e dispone di un accurato sistema di riconoscimento di comandi vocali e gestuali. L'ambito applicativo del sistema impone requisiti molto stringenti: il posizionamento di icone sul display ha un'accuratezza angolare di 10 milliradiani, mentre il sottosistema di gesture recognition produce meno dell'1% di falsi positivi.

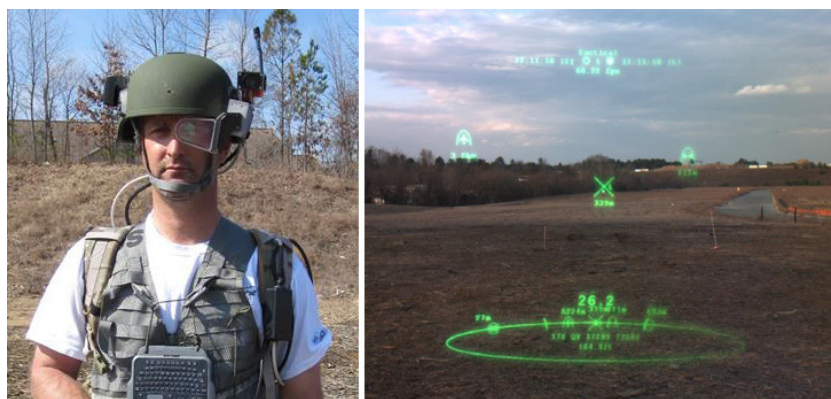


Figura 3.3: Prototipo del sistema MAR a scopo militare proposto in [35, 36].

I dispositivi hands-free emergenti risultano particolarmente adatti anche per la realizzazione di sistemi di addestramento militare AR, in cui il campo di battaglia viene aumentato con target virtuali da raggiungere, nemici e pericoli di varia natura. Un primo prototipo di questa tipologia di applicazione è stato proposto in [37].

### 3.3.3 Soccorso

Smart-glass e altri dispositivi hands-free potrebbero essere impiegati in un prossimo futuro in operazioni di soccorso e salvataggio: sul display dei soccorritori si potrebbero visualizzare informazioni aumentate che rendano gli interventi più rapidi ed efficienti. I dati e le indicazioni virtuali potrebbero essere generati da un centro operativo o

da singoli membri del team di soccorso: ad esempio, una squadra di Vigili del Fuoco potrebbe vedere su smart-glass l'esatta posizione di un civile da salvare che è stato avvistato dal supporto aereo, oppure ottenere informazioni in tempo reale sulla direzione di espansione di un incendio.

### 3.3.4 Automotive

In ambito automotive è di vitale importanza che durante l'utilizzo di sistema AR l'utente rimanga concentrato sulla guida del veicolo. L'impiego di un sistema basato su smart-glass può costituire una valida soluzione, in quanto consente all'utente di mantenere lo sguardo sulla strada. Le informazioni aumentate da visualizzare sul display durante la guida potrebbero includere indicazioni stradali per la destinazione che l'utente deve raggiungere, informazioni sul traffico, limiti di velocità e POI (*Point Of Interest*) di varia natura. Nel caso in cui il dispositivo sia in grado di supportare sofisticati algoritmi di visione artificiale, potrebbe inoltre segnalare alcuni pericoli imminenti che minacciano la sicurezza, come la presenza di ostacoli sull'asfalto e il mantenimento di una distanza di sicurezza insufficiente dal veicolo antistante.

La casa automobilistica *Mini*, in collaborazione con la società specializzata in smart-glass ODG<sup>1</sup>, ha recentemente proposto uno smart-glass in grado di supportare alcune di queste funzionalità [38], offrendo inoltre un'interessante sistema di visione "a raggi X" che, sfruttando alcune telecamere disposte all'esterno dell'auto, consente al conducente di vedere porzioni dell'ambiente esterno normalmente ostruite dall'abitacolo dell'auto.

### 3.3.5 Turismo

In passato sono state proposte alcune soluzioni MAR in grado di mostrare contenuti virtuali interattivi a visitatori di musei e siti archeologici. *Archeoguide* [39] è un sistema prototipale con lo scopo di mostrare ai turisti la ricostruzione virtuale di un sito archeologico e di ottenere informazioni aggiuntive sugli artefatti e gli edifici circostanti. Il

---

<sup>1</sup><http://www.osterhoutgroup.com/>

prototipo è stato testato nel 2002 nel sito archeologico di Olimpia in Grecia; le principali problematiche emerse erano legate all'ingombro del sistema e alla limitata potenza computazionale dell'hardware.

Un sistema di realtà aumentata odierno sarebbe sicuramente affetto da tali limitazioni in misura molto minore e, per questo tipo di applicazioni, potrebbe offrire un ottimo compromesso tra ingombro e prestazioni.

### 3.3.6 Intrattenimento

Ad oggi esistono numerosi giochi di realtà aumentata commerciali per smartphone e tablet. In questo tipo di applicazioni generalmente si utilizza la videocamera del dispositivo per inquadrare la scena reale, su cui vengono sovrapposti oggetti e informazioni virtuali; il display viene dunque utilizzato in modalità video see-through (si veda il paragrafo 2.5.2). L'impiego di smart-glass potrebbe rendere l'esperienza più immersiva ed interattiva, collocando l'utente al centro dell'ambiente di gioco e consentendogli di utilizzare meccanismi di interazione più coinvolgenti, come ad esempio il riconoscimento di gesti e di comandi vocali.

Smart-glass e altri sistemi hands-free potrebbero risultare adatti anche per altri tipi di intrattenimento domestico, come ad esempio la navigazione in rete, la visione di film e la fruizione di altri contenuti multimediali.

### 3.3.7 Industria ed enterprise

Le tecnologie MAR e hands-free risultano particolarmente utili per rendere più efficienti i workflow aziendali e per gestire e mantenere impianti ed apparecchiature industriali. Nel 2001 Klinker et al. [40] hanno sviluppato un sistema MAR per la manutenzione di centrali nucleari, in grado di fornire al personale tecnico alcune procedure guidate interattive per lo svolgimento di interventi di manutenzione e riparazione. Goose et al. [41] nel 2004 hanno realizzato un sistema per verificare e controllare lo stato di apparecchiature e strumenti: dopo aver identificato l'apparecchiatura inquadrata utilizzando dei marker visivi, il sistema inviava i dati identificativi della strumentazione ad un

software, che restituiva le informazioni correlate all'apparecchiatura (ad esempio pressione e temperatura) e le visualizzava sul dispositivo dell'operatore.

Negli ultimi anni sempre più imprese ed industrie stanno mostrando interesse verso i più recenti sistemi hands-free basati su smart-glass. Schlumberger, la più grande società di servizi petroliferi al mondo, ha equipaggiato i tecnici che operano nelle aree petrolifere con Google Glass [10]; la società ritiene che le informazioni relative all'ambiente e al workflow che gli smart-glass forniscono possano incrementare l'efficienza e la produttività dei lavoratori. Anche Beckhoff, una società tedesca che si occupa di automazione industriale, ha mostrato interesse verso i Google Glass [11]: il personale tecnico della società li impiega, ad esempio, per visualizzare le proprietà e lo stato corrente di sensori e attuatori, tramite la scansione di codici QR posti sulle apparecchiature di interesse. La compagnia aerea EasyJet sta impiegando gli Epson Moverio BT-200 per semplificare la gestione dei processi di manutenzione e riparazione, con l'obiettivo di migliorare ulteriormente il livello di sicurezza dei propri sistemi e di minimizzare i ritardi dei voli aerei [12]. Vuzix, un'azienda produttrice di soluzioni hands-free e smart-glass, ha recentemente avviato collaborazioni con Ubimax [42], Evolar [43] e SAP [27] al fine di realizzare applicazioni che rendano la gestione di magazzino e la preparazione degli ordini più efficiente.

Alcune aziende come APX Labs<sup>2</sup>, Joinpad<sup>3</sup> e NGRAIN<sup>4</sup> hanno incentrato il loro core business su sistemi di realtà aumentata industriali, specializzandosi nello sviluppo di soluzioni software per molteplici dispositivi e piattaforme.

### 3.3.8 Uno scenario a lungo termine: Mirror Worlds

Gli ambiti applicativi appena presentati lasciano scorgere alcuni trend e scenari che potrebbero realizzarsi in un prossimo futuro. In tal senso, l'integrazione di *ubiquitous computing* e *smart space* costituisce una delle visioni più condivise in letteratura. Come si è detto nel paragrafo 2.1, l'architettura di un sistema MAR può comprendere un'opportuna

---

<sup>2</sup><http://www.apx-labs.com/>

<sup>3</sup><http://www.joinpad.net/>

<sup>4</sup><http://www.ngrain.com/>

infrastruttura ambientale in grado di ridurre il carico di lavoro localizzato sul wearable device (*thin client*), oppure di estendere le potenzialità e le funzionalità del dispositivo. In questo secondo caso, si può considerare che l'*intelligenza* del sistema AR non sia concentrata sul wearable device, ma distribuita nell'ambiente.

Il concetto di *Ambient Intelligence* (AmI) è definito dalla convergenza tra ubiquitous computing, intelligenza artificiale e tecnologie di reti di sensori. Un ambiente AmI è caratterizzato "dall'utilizzo pervasivo di dispositivi computazionali completamente integrati nel tessuto della vita quotidiana, al punto da divenire indistinguibili da esso" [44]. La maturità tecnologica che stanno raggiungendo le discipline coinvolte nei sistemi AmI consentirà in un prossimo futuro la progettazione e la realizzazione di smart environment in cui la realtà fisica può essere aumentata con funzionalità e servizi virtuali. L'ambiente reale in cui si trova l'utente verrebbe infatti arricchito con un layer digitale contenente informazioni virtuali situate e servizi che possono essere fruiti tramite wearable device. Le tecnologie MAR odierne dunque influenzano pesantemente la progettazione di ambienti AmI, consentendo all'utente di interagire con l'ambiente aumentato utilizzando smart-glass o altri dispositivi hands-free.

Nel contesto di ambienti AmI, un *mirror world* rappresenta una controparte digitale del mondo fisico, in grado di estenderlo con un layer computazionale aperto [45, 46]. Un mirror world è strettamente accoppiato con l'ambiente reale che estende, ed è strutturato ed organizzato come una città digitale i cui abitanti sono agenti software (si veda Figura 3.4). Gli oggetti presenti nell'ambiente fisico possono avere una controparte digitale nel mirror world; l'interazione di un utente con un oggetto fisico lascia una traccia (ovvero un effetto) nel mondo digitale. Tale traccia può influenzare il comportamento degli agenti software, poiché essi sono in grado di percepirla e quindi di reagire ad essa. Inoltre, essendo delle entità proattive, gli agenti software possono a loro volta interagire con gli oggetti virtuali presenti nel mirror world: l'effetto che tale interazione produce può essere percepito da un utente del mondo fisico direttamente (attraverso i cinque sensi), oppure tramite un wearable device, come ad esempio uno smart-glass o uno smartphone.

Un mirror world consente dunque di aumentare lo spazio di proget-

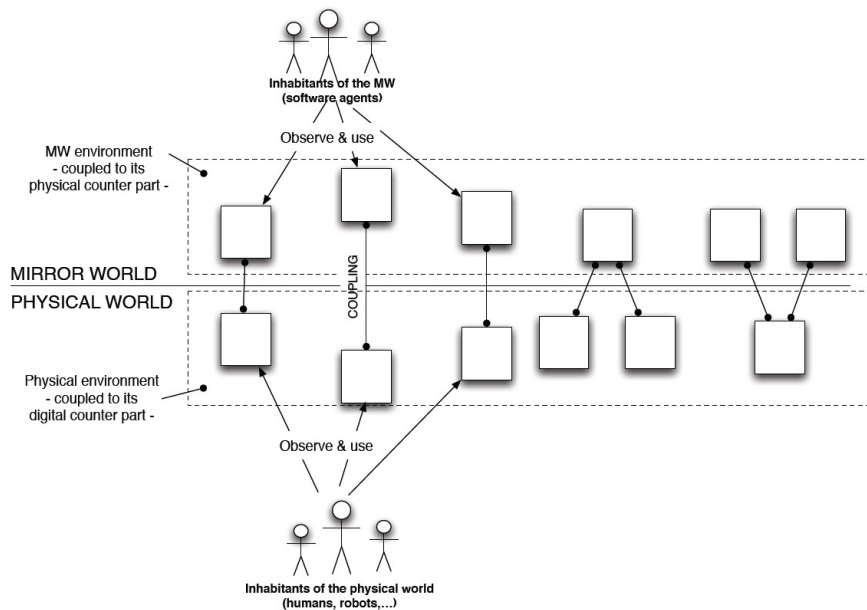


Figura 3.4: Struttura ad alto livello di un mirror world.

tazione di un sistema MAR, supportando e promuovendo meccanismi di interazione e comunicazione avanzati. Uno dei principali obiettivi dello studio dei mirror worlds è l'esplorazione di aspetti interdisciplinari riguardanti smart space e mixed-reality, al fine di modellare ambienti aperti e le loro interazioni con gli utenti.

## 3.4 Limitazioni e problematiche attuali

In questa sezione si presentano le principali problematiche di cui sono attualmente affetti i sistemi hands-free basati su smart-glass, includendo le cause che potrebbero ostacolare la loro adozione e diffusione [13, 14].

### 3.4.1 Limitazioni tecnologiche

Le problematiche tecnologiche più rilevanti che presentano smart-glass e altri sistemi hands-free riguardano l'autonomia di alimentazione, la



potenza computazionale e il campo di vista (FOV, *field of view*) del display.

Negli ultimi decenni le tecnologie impiegate per la costruzione di batterie hanno sperimentato un progresso meno repentino rispetto a quello avvenuto a circuiti integrati e componenti elettronici. Il fatto che gli smart-glass proposti fino ad ora spesso non dispongano di un'autonomia di alimentazione sufficiente è probabilmente dovuto a questa causa, rendendo in alcuni casi impossibile utilizzare il dispositivo per più di qualche ora. Inoltre, le funzionalità maggiormente utilizzate in applicazioni AR –quali GPS, connessione di rete e videocamera– hanno un consumo di energia elevato, dunque tendono a scaricare il livello della batteria più rapidamente.

Oltre ad essere onerose da un punto di vista energetico, alcune funzionalità richiedono una potenza di calcolo superiore a quella disponibile nel processore installato sul dispositivo. Ad esempio, le tecniche di tracciamento più accurate proposte nel campo della computer vision spesso non possono essere direttamente impiegate su smart-glass e altri dispositivi hands-free a causa della loro limitata potenza computazionale.

Per quanto riguarda le tecnologie adottate per la costruzione dei display, le limitazioni derivanti dall'utilizzo di optical see-through display sono state presentate nel paragrafo 2.5.1. In particolare, le attuali tecnologie vincolano il campo di vista della scena virtuale ad un'ampiezza massima di circa  $40^\circ$ , una dimensione molto minore rispetto al campo di vista dell'occhio umano [29].

### 3.4.2 Privacy e accettazione sociale

I fattori che influenzano maggiormente l'accettazione sociale di un sistema hands-free comprendono questioni relative a privacy e sicurezza, all'ingombro del dispositivo e al suo appeal. Dal punto di vista dell'intrusività, gli smart-glass commercializzati recentemente hanno mostrato progressi molto significativi rispetto ai dispositivi HMD proposti in passato. Tuttavia, ad oggi non è ancora possibile costruire uno smart-glass che abbia dimensioni ed ingombro comparabili con quelle di un normale occhiale da vista o da sole: questo fattore potrebbe costituire un ostacolo per la diffusione del prodotto nel mercato

di massa, in quanto la maggior parte dei consumatori prima dell'acquisto di un accessorio da indossare valuta, oltre alla sua utilità, anche il suo aspetto estetico. Figura 3.5 illustra la correlazione esistente tra l'ampiezza del campo di vista del display e l'ingombro del dispositivo, riportando i dati relativi agli smart-glass odierni.

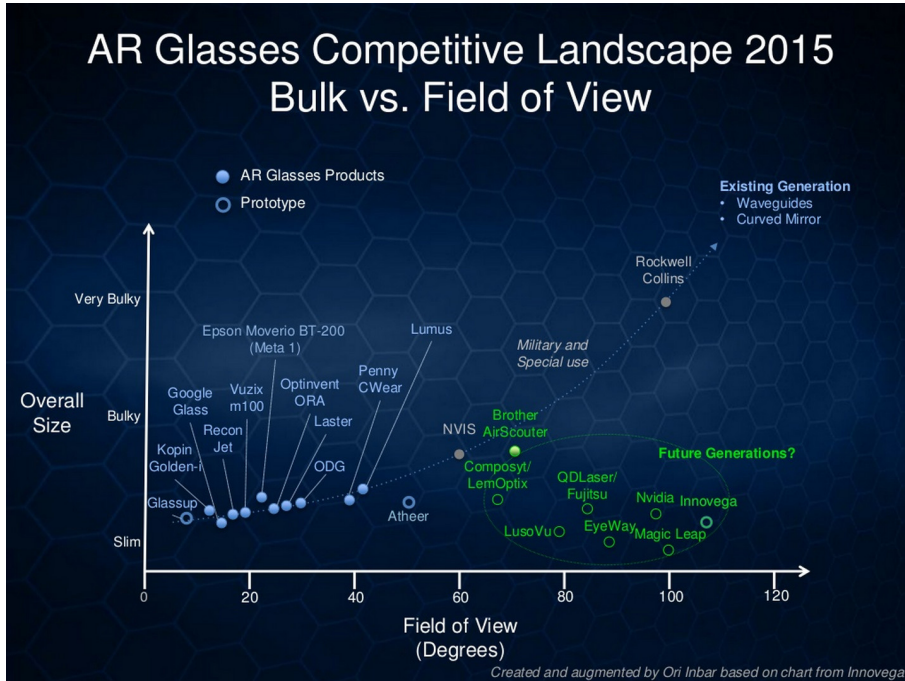


Figura 3.5: Correlazione tra l'ampiezza del campo di vista del display e l'ingombro dello smart-glass.

Privacy e sicurezza rappresentano un ulteriore ostacolo all'accettazione sociale di questo tipo di dispositivi: si tratta di ambiti che sollevano problematiche relative all'identificazione personale, al trattamento di dati sensibili e alla profilazione degli utenti. Ad esempio, molti servizi e applicazioni necessitano di dati relativi alla posizione corrente dell'utente per funzionare correttamente: tali informazioni potrebbero essere trasmesse su server di terze parti ed utilizzate per tracciare la sua attività. Alcune persone ritengono che l'utilizzo di smart-glass in talune circostanze sia irrispettoso per la privacy altrui: alcuni luoghi pubblici vietano addirittura l'ingresso a chi li indossa

[47]. La diffidenza verso questo tipo di dispositivi è motivata principalmente dal fatto che l'utente che li utilizza potrebbe scattare foto o registrare video senza che le persone inquadrature ne siano consapevoli.

### 3.5 Sistemi rappresentativi

In questa sezione si presentano alcuni dei dispositivi hands-free di maggior notorietà. Figura 3.6 riporta una panoramica sullo stato di produzione e la popolarità degli smart-glass proposti sino ad ora.

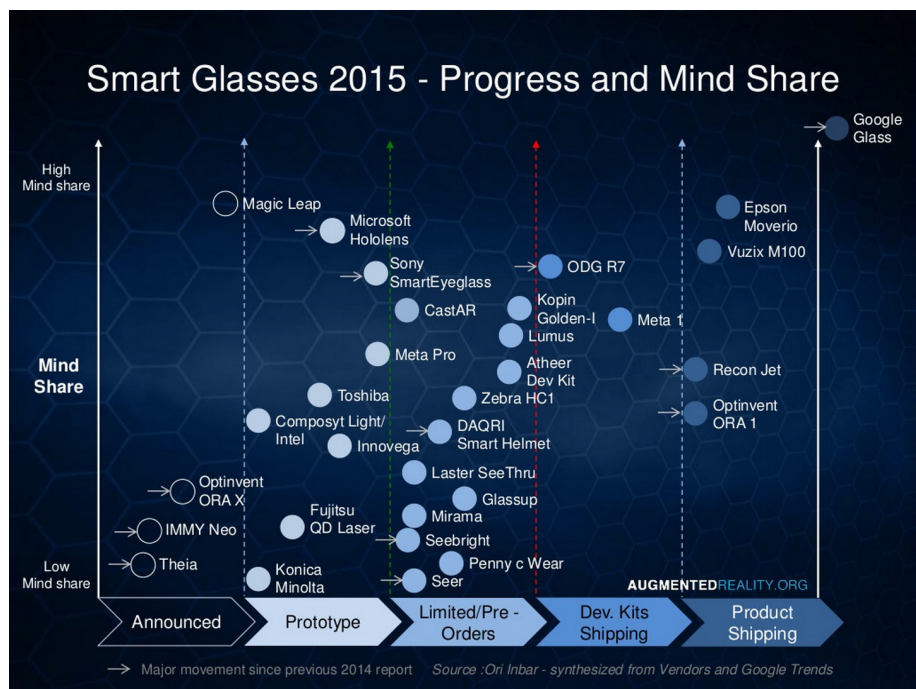


Figura 3.6: Stato di produzione e popolarità degli smart-glass odierni.

#### 3.5.1 Google Glass Explorer Edition

I Google Glass<sup>5</sup> sono stati il primo prototipo di smart-glass ad ottenere un importante impatto mediatico. In Aprile 2013 è stata rilasciata

<sup>5</sup><http://www.google.com/glass/start/>



Figura 3.7: Google Glass Explorer Edition.

una prima versione per sviluppatori, e a partire da Maggio 2014 il dispositivo è divenuto disponibile al mercato statunitense. In Gennaio 2015 Google ha deciso di ritirare il prodotto dal mercato, e alcuni mesi più tardi ha avviato collaborazioni con aziende partner per progettare una nuova versione del dispositivo.

### **Caratteristiche**

- Display: 14° FOV (Field Of View), singolo, non immersivo (si veda il paragrafo 2.5.4)
- Acquistabile: no
- Prezzo: 1500 \$
- Target di mercato: consumer ed enterprise

### **3.5.2 Epson Moverio BT-200**

Gli Epson Moverio BT-200<sup>6</sup> godono di ottima notorietà poiché sono stati tra i primi smart-glass ad offrire un display immersivo ad un prezzo di mercato competitivo.

<sup>6</sup><http://www.epson.it/it/viewcon/corporatesite/products/mainunits/overview/12411>



Figura 3.8: Epson Moverio BT-200.

### Caratteristiche

- Display: 23° FOV, stereoscopico, immersivo
- Acquistabile: sì
- Prezzo: 699 € IVA inclusa
- Target di mercato: consumer ed enterprise

### 3.5.3 Vuzix Wrap 1200DXAR



Figura 3.9: Vuzix Wrap 1200DXAR.

Vuzix è un'azienda leader nel settore di display wearable e smart-glass. I Vuzix Wrap 1200DXAR<sup>7</sup> sono pensati per applicazioni di

<sup>7</sup>[http://vuzix.eu/augmented-reality/products\\_wrap1200dxar/](http://vuzix.eu/augmented-reality/products_wrap1200dxar/)

realtà aumentata immersiva: dispongono infatti di una fotocamera stereoscopica e di un display binoculare dotato di un buon campo di vista.

### **Caratteristiche**

- Display: 35° FOV, stereoscopico, immersivo
- Acquistabile: sì
- Prezzo: 1499,99 € IVA inclusa
- Target di mercato: enterprise

### **3.5.4 Sony SmartEyeglass**



Figura 3.10: Sony SmartEyeglass.

In Marzo 2015 Sony ha rilasciato sul mercato SmartEyeglass<sup>8</sup>, un dispositivo di realtà aumentata orientato sia al mercato di massa che al mondo dell'industria.

### **Caratteristiche**

- Display: 20° FOV, stereoscopico, immersivo
- Acquistabile: sì
- Prezzo: 817,40 € IVA inclusa
- Target di mercato: consumer ed enterprise

---

<sup>8</sup><http://developer.sonymobile.com/products/smarteyeglass/>

### 3.5.5 ODG R-7



Figura 3.11: ODG R-7.

ODG è una società specializzata in sistemi wearable e smart-glass che negli ultimi anni ha progettato soluzioni di realtà aumentata per il governo statunitense. Gli ODG R-7<sup>9</sup> sono smart-glass con target enterprise; la società prevede tuttavia di rilasciare anche un modello per il mercato di massa entro il 2016.

#### Caratteristiche

- Display: 30° FOV, stereoscopico, immersivo
- Acquistabile: su prenotazione
- Prezzo: non disponibile
- Target di mercato: enterprise

### 3.5.6 Microsoft HoloLens

HoloLens è un head-mounted display il cui target è sia il mercato di massa dell'home entertainment, sia il mondo del lavoro e dell'industria. Il dispositivo non è ancora presente sul mercato, ma potrebbe essere rilasciato tra la fine del 2015 e il 2016.

<sup>9</sup><http://www.osterhoutgroup.com/products-r7-glasses.php>



Figura 3.12: Microsoft HoloLens.

### **Caratteristiche**

- Display: stereoscopico, immersivo
- Acquistabile: no
- Prezzo: non disponibile
- Target di mercato: consumer ed enterprise

### **3.5.7 DAQRI Smart Helmet**



Figura 3.13: DAQRI Smart Helmet.

DAQRI Smart Helmet è un head-mounted display progettato specificatamente come soluzione enterprise. Il dispositivo è equipaggiato con una videocamera 3D ad alta risoluzione e un set di videocamere che supportano visione e navigazione a 360°.



### Caratteristiche

- Display: 80° FOV, stereoscopico, immersivo
- Acquistabile: su ordinazione
- Prezzo: non disponibile
- Target di mercato: enterprise

### 3.5.8 Altri sistemi rappresentativi

L'elevato numero di dispositivi che stanno facendo il loro ingresso nel mercato dimostra l'ampio interesse che stanno destando smart-glass e sistemi hands-free in generale. Altri dispositivi che godono di ampia popolarità includono: Meta Pro<sup>10</sup>, Epson Moverio BT-2000<sup>11</sup>, Vuzix M100<sup>12</sup>, Kopin Golden-i<sup>13</sup>, Recon Jet<sup>14</sup>, Atheer AiR<sup>15</sup>, Magic Leap<sup>16</sup>, Laster SeeThru<sup>17</sup>, RideOn<sup>18</sup>, Optinvent ORA-1<sup>19</sup>, GlassUp<sup>20</sup>, Caputer SEER<sup>21</sup>, Lumus<sup>22</sup>, e castAR<sup>23</sup>.

<sup>10</sup><https://www.getameta.com/products>

<sup>11</sup><http://www.epson.it/it/it/viewcon/corporatesite/products/mainunits/overview/16149>

<sup>12</sup>[http://vuzix.eu/consumer/products\\_m100/](http://vuzix.eu/consumer/products_m100/)

<sup>13</sup><http://www.mygoldeni.com/>

<sup>14</sup><http://www.reconinstruments.com/products/jet/>

<sup>15</sup><https://www.atheerlabs.com/>

<sup>16</sup><http://www.magicleap.com/>

<sup>17</sup><http://www.laster.fr/products/seethru/>

<sup>18</sup><http://www.rideonvision.com/>

<sup>19</sup><http://optinvent.com/see-through-glasses-ORA>

<sup>20</sup><http://www.glassup.net/>

<sup>21</sup><http://www.caputer.com/>

<sup>22</sup>[http://www.lumus-optical.com/index.php?option=com\\_content&task=view&id=8&Itemid=13](http://www.lumus-optical.com/index.php?option=com_content&task=view&id=8&Itemid=13)

<sup>23</sup><https://technical-illusions.myshopify.com/products/castar>



## Capitolo 4

# Google Glass

Il 15 Aprile 2013 Google ha rilasciato il primo prototipo di smart-glass ideato e progettato per il mercato di massa: i *Google Glass Explorer Edition*. Il dispositivo è stato concepito nell'ambito del *Project Glass* e sviluppato presso i laboratori Google X, ed è stato venduto negli Stati Uniti al prezzo di 1500 \$ ad un ristretto gruppo di sviluppatori selezionati, per poi divenire disponibile il 15 Maggio 2014 allo stesso prezzo all'intero mercato statunitense. Il 15 Gennaio 2015 Google ha annunciato l'intenzione di chiudere il programma Explorer, cessando le vendite del dispositivo prototipale e avviando la progettazione di una nuova versione dello smart-glass. Differentemente dal primo prototipo, il nuovo modello probabilmente non sarà destinato al mercato di massa, bensì sarà orientato unicamente ad un target business e industriale [48]. I Google Glass Explorer sono stati il primo smart-glass a riscuotere un notevole impatto mediatico e, nonostante siano state sollevate alcune critiche nei loro confronti per problematiche legate a privacy e accettazione sociale (si veda il paragrafo 3.4.2), il clamore da essi suscitato ha probabilmente influenzato e incentivato il proponimento di numerosi altri modelli di smart-glass, alcuni dei quali sono stati presentati nel paragrafo 3.5.

In questo capitolo si presentano le specifiche tecniche del dispositivo, i principi di User Experience adottati e i costrutti principali della User Interface; in seguito si illustrano i possibili approcci di sviluppo e i pattern di riferimento per la realizzazione di glassware. Infine, si riassumono le potenzialità e le limitazioni dello smart-glass.

## 4.1 Caratteristiche tecniche

I Google Glass Explorer esibiscono le seguenti specifiche tecniche:

- Sistema operativo: Android KitKat 4.4
- Display: 640x360 Himax HX7309 LCoS
- Videocamera: foto con risoluzione di 5 Megapixel, registrazione video a 720p
- Connettività: Wi-Fi 802.11b/g, Bluetooth
- Storage: 16 GB totali di cui 12 GB disponibili
- Processore: Texas Instruments OMAP 4430 SoC 1.2Ghz Dual (ARMv7)
- Memoria RAM: 1 GB
- Sensori: giroscopio a 3 assi, accelerometro a 3 assi, magnetometro (bussola) a 3 assi, sensore di prossimità, sensore di luce ambientale
- Audio: trasduttore a conduzione ossea

Lo smart-glass non include un sensore GPS e non dispone di uno slot per scheda SIM, ma può essere accoppiato con uno smartphone stabilendo una connessione Bluetooth. Il *pairing* dei due dispositivi consente ai Google Glass di accedere ai dati del sensore GPS installato sullo smartphone, di connettere lo smart-glass ad Internet tramite tethering della rete cellulare e di utilizzare il microfono e l'altoparlante dei Google Glass per effettuare o rispondere ad una telefonata hands-free.

Il display LCoS (*Liquid Crystal on Silicon*) utilizza un divisore di raggio a specchio parzialmente riflettente (*Polarizing Beam Splitter*) per proiettare l'immagine dello schermo nel campo visivo dell'utente. Lo schema dell'unità ottica dello smart-glass è riportato in Figura 4.1; la traiettoria dei raggi luminosi ha origine sul display LED e termina sull'occhio dell'utente, raffigurato nella parte inferiore della figura. La

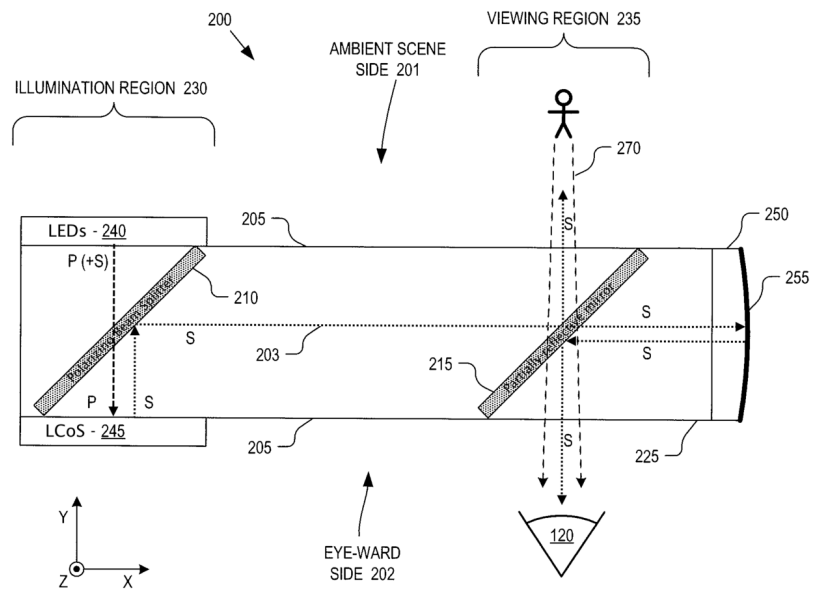


Figura 4.1: Schema dell'unità ottica dei Google Glass Explorer.

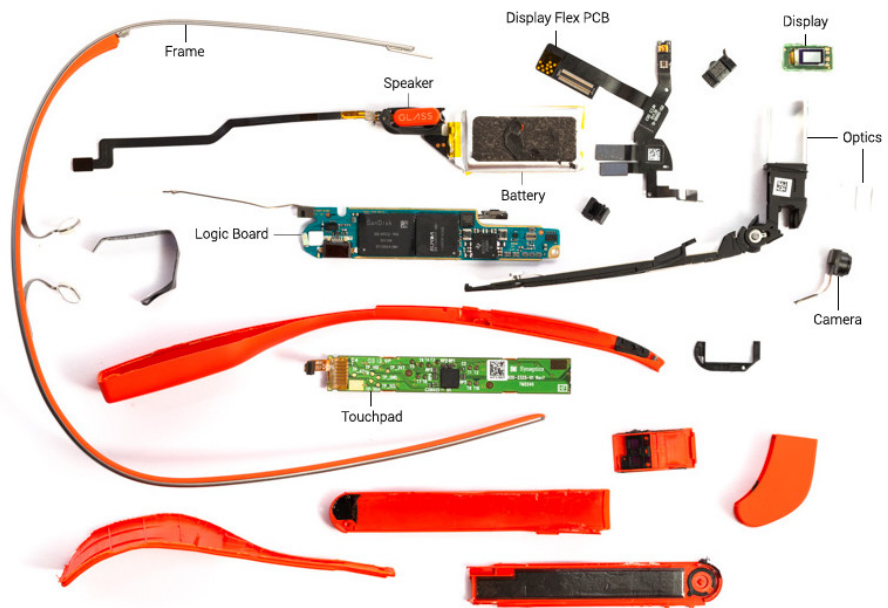


Figura 4.2: Componenti hardware interni dei Google Glass Explorer.

percezione della dimensione del display virtuale è equivalente a quella di uno schermo ad alta definizione da 25 pollici osservato da due metri di distanza. Il display dei Google Glass non è immersivo (la definizione di immersività è stata presentata nel paragrafo 2.5.4) poiché, essendo disposto nella periferia del campo visivo dell'utente, non consente di associare ed allineare tra loro oggetti reali e virtuali.

## 4.2 User Experience

L'efficacia di un dispositivo tecnologico è correlata al ritardo temporale esistente tra l'intenzione dell'utente di utilizzare una certa feature e l'accesso effettivo a tale funzionalità. Citando Larry Page, amministratore delegato di Google, l'obiettivo principale del Project Glass è proprio "abbreviare l'intervallo temporale tra intenzione e azione"; se il tempo di accesso diventa sufficientemente ridotto, l'utente tende a percepire il dispositivo tecnologico non come un oggetto esterno, ma come un'estensione di sé stesso [49]. L'intento è dunque fornire un insieme di meccanismi che consentano all'utente di gestire e concludere il più rapidamente possibile le interazioni con il dispositivo, rinominate a questo scopo in *microinterazioni*. Un esempio di microinterazione di successo è controllare l'ora su un orologio da polso: si tratta di un'azione di durata molto breve, così naturale e semplice da divenire completamente automatica per l'utente. Inoltre, la praticità e la durata esigua dell'interazione fa sì che l'attenzione richiesta all'utente sia minima.

Gli smartphone odierni non sono così efficaci dal punto di vista dell'interazione: recuperare il dispositivo (ad esempio estraendolo da una tasca), sbloccarlo e navigare la UI per trovare la funzionalità desiderata potrebbe richiedere anche più di 15-20 secondi. Dunque risulta chiaro che difficilmente uno smartphone può essere percepito come un'estensione di sé, vista la durata e l'attenzione richieste tipicamente per interagire con il dispositivo. L'obiettivo del Project Glass è dunque progettare e realizzare un'esperienza utente che renda l'accesso alle funzionalità dello smart-glass agevole come controllare l'ora su un orologio da polso. La modalità di interazione principale supportata dai Google Glass si basa su un'interfaccia di riconoscimento vocale:

l'utente può utilizzare la propria voce per accedere rapidamente alle funzionalità desiderate e comporre input testuale in linguaggio naturale. L'interazione col dispositivo può avvenire inoltre tramite un touchpad disposto su un'asticella dello smart-glass: questa modalità risulta particolarmente utile in ambienti molto rumorosi o in contesti in cui non sia opportuno parlare ad alta voce. Ulteriori informazioni sull'interfaccia a comandi vocali e sul touchpad verranno fornite nei paragrafi 4.3.5 e 4.3.6, dopo che si saranno presentate le caratteristiche principali della User Interface adottata dallo smart-glass.

Google propone alcuni principi di design da seguire in fase di progettazione di un software per Google Glass –denominato *glassware*– al fine di offrire all'utente finale una User Experience consistente ed efficace.

- *Design for Glass*: un sistema basato su smart-glass è radicalmente differente da altri tipi di dispositivi mobili quali smartphone, tablet e laptop, dunque è necessario offrire un'esperienza che sfrutti le caratteristiche e gli elementi peculiari di questa nuova tipologia di dispositivo.
- *Don't get in the way*: il glassware dovrebbe offrire le proprie funzionalità senza richiedere l'attenzione dell'utente per un periodo di tempo prolungato, evitando di distrarlo e di intralciare l'attività che sta svolgendo nel mondo reale.
- *Keep it relevant*: il glassware dovrebbe esibire comportamenti sensibili al contesto, fornendo informazioni rilevanti nel momento e nel luogo più opportuni.
- *Avoid the unexpected*: essendo il display dello smart-glass sempre all'interno del campo visivo dell'utente, è importante che il glassware non esibisca comportamenti inattesi o imprevedibili, e che non proponga notifiche e contenuti troppo frequentemente o in momenti inopportuni.
- *Build for people*: immagini, icone e foto sono fruibili più facilmente rispetto a contenuti testuali, la GUI del glassware dovrebbe dunque privilegiare elementi grafici ed animazioni. I coman-

di vocali supportati dal glassware dovrebbero essere semplici e colloquiali.

## 4.3 User Interface

Nel seguito si presentano le principali astrazioni adottate per realizzare la User Interface offerta dai Google Glass e si descrivono le modalità con cui l'utente può interagirvi.

### 4.3.1 Timeline

La timeline è il principale componente della UI: è composta da un insieme di schermate, denominate *card*, che presentano contenuti di rilievo di varia natura in ordine cronologico. La timeline è organizzata in più sezioni che possono essere navigate utilizzando il touchpad del dispositivo: l'utente può eseguire una gesture di *swipe* verso sinistra o destra per visualizzare rispettivamente la card precedente o successiva della timeline. Al centro della timeline vi è la *home card*, una schermata che mostra l'ora corrente e che consente di avviare i glassware installati sul dispositivo pronunciando un comando vocale o utilizzando il touchpad (si veda Figura 4.3). A destra della home card vi sono gli elementi relativi al passato, come le foto scattate recentemente, lo storico delle conversazioni di messaggistica e le card relative ad eventi passati. A sinistra della home card vi sono invece gli elementi relativi al presente e al futuro –ad esempio le informazioni sul meteo o sul prossimo volo da prendere– e la card per accedere alle impostazioni dello smart-glass. Le sezioni in cui è organizzata la timeline sono illustrate in Figura 4.4.

Il software installato sullo smart-glass implementa un meccanismo di sincronizzazione che, qualora sia attivo il pairing con uno smartphone Android, consente di visualizzare automaticamente sulla timeline dei Google Glass le notifiche ricevute sul dispositivo handheld; quando l'utente rimuove una notifica su uno dei due dispositivi, il sistema provvede ad eliminarla anche sull'altro.



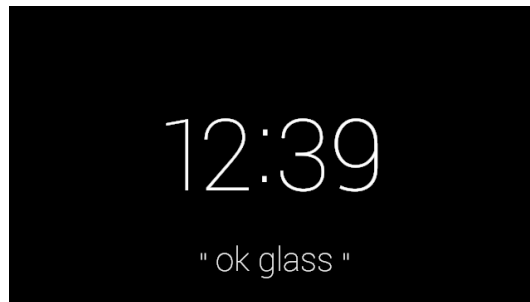


Figura 4.3: La home card della timeline.

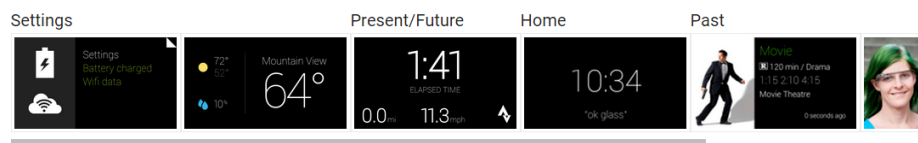


Figura 4.4: La timeline e le sezioni in cui è organizzata.

### 4.3.2 Card

Le card rappresentano l'astrazione fondante su cui si basa la UI dei Google Glass e costituiscono il componente grafico standard per presentare le informazioni sul display del dispositivo. La piattaforma software dello smart-glass le distingue in due tipologie: le *live card* e le *static card*.

#### Live card

Questo tipo di card è utilizzato per presentare informazioni e dati in tempo reale su una UI costantemente aggiornata. La frequenza di aggiornamento di una live card consente di gestire e visualizzare elementi grafici custom, che possono spaziare da semplici widget 2D a complessi contenuti tridimensionali. Le live card sono disposte nella sezione presente/futuro della timeline e rimangono in esecuzione mentre l'utente interagisce con altre card, offrendo all'utente funzionalità multitasking. Le live card hanno inoltre accesso ai dati dei sensori di basso livello come GPS, accelerometro e bussola. In Figura 4.5 si riporta un esempio di live card.

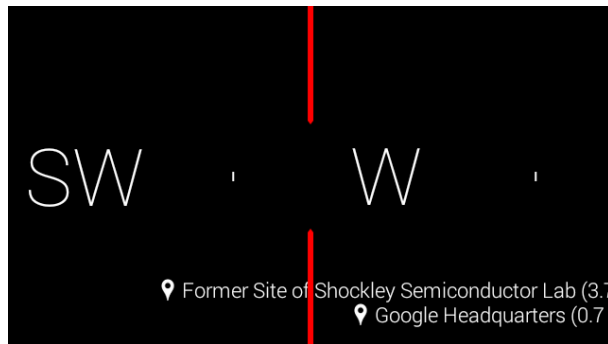


Figura 4.5: Il glassware *Compass* utilizza i dati del magnetometro per visualizzare su una live card l'orientamento dell'utente in tempo reale.

### Static card

Le static card consentono di mostrare contenuti multimediali che non necessitano di essere aggiornati frequentemente come testo, immagini, video e codice HTML. La loro natura le rende adatte a realizzare notifiche relative ad attività o eventi sia passati che futuri: una static card può essere disposta infatti sia nella sezione del passato della timeline, sia in quella del presente/futuro. A differenza delle live card, le static card non hanno accesso ai dati dei sensori hardware dello smart-glass. Il contenuto di una static card può essere condiviso con altre persone o con altri glassware installati sul dispositivo. In Figura 4.6 si riportano due esempi di static card.

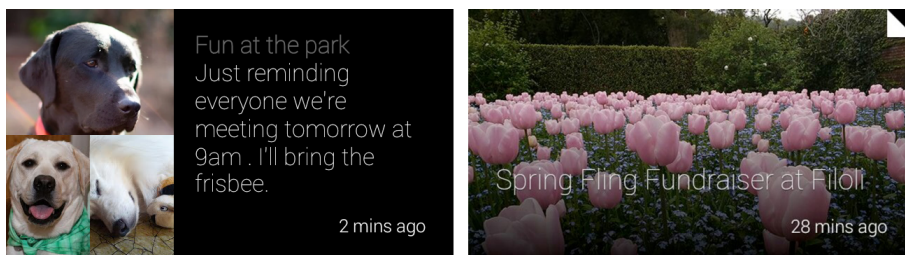


Figura 4.6: Esempi di static card.

### 4.3.3 Immersion

Un'*immersion* è un'astrazione che consente ad un glassware di offrire un'esperienza interattiva personalizzata al di fuori del contesto della timeline. La UI offerta dalla timeline infatti vincola alcuni aspetti dell'interazione con l'utente, come ad esempio il comportamento del touchpad e dell'interfaccia a comandi vocali. L'esecuzione di un'*immersion* permette invece di ottenere il pieno controllo della User Experience, consentendo al glassware di gestire l'input utente, la UI e di accedere all'hardware del dispositivo. L'utente può chiudere l'*immersion* e tornare alla timeline eseguendo uno swipe verso il basso sul touchpad dello smart-glass.

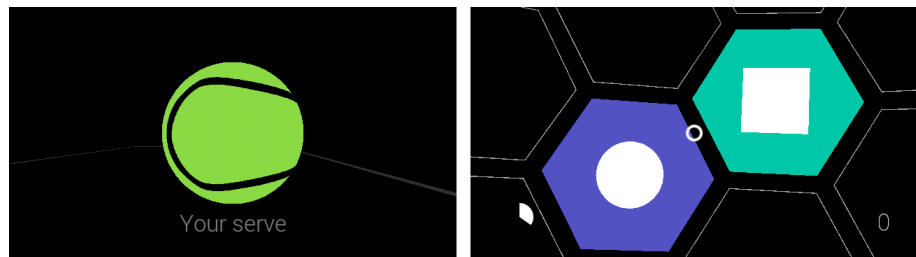


Figura 4.7: Le immersion sono particolarmente adatte per applicazioni interattive e giochi.

### 4.3.4 Menu contestuale

Le card e le immersion possono offrire un menu contestuale accessibile tramite comando vocale oppure eseguendo una gesture di *tap* sul touchpad. Le voci di menu sono utilizzate per accedere alle funzionalità del glassware oppure per eseguire un'azione inerente ai contenuti della card attualmente visualizzata sul display. Ad esempio, accedendo al menu di una card relativa ad una foto scattata recentemente può essere possibile condividere l'immagine oppure eliminarla.

### 4.3.5 Input vocale

Il sistema di input vocale dei Google Glass consente all'utente di accedere alla maggior parte delle funzionalità dello smart-glass tramite

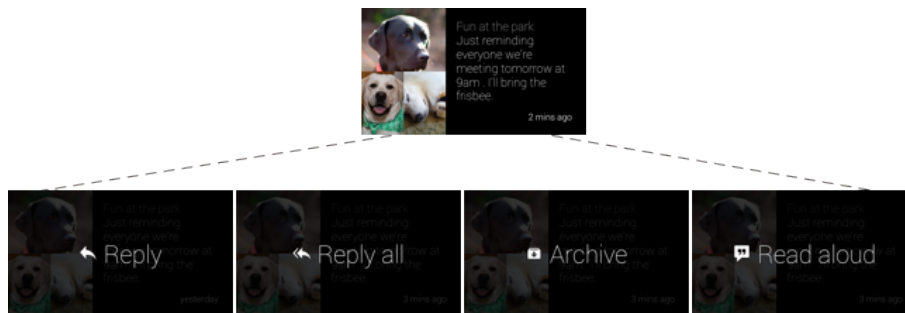


Figura 4.8: Esempio di menu contestuale.

un'interfaccia hands-free, in modo che possa utilizzare il dispositivo mentre svolge altre attività. La piattaforma software dello smart-glass supporta tre modalità di input vocale:

- comandi vocali principali
- comandi vocali contestuali
- riconoscimento vocale.

### Comandi vocali principali

Questa modalità di input vocale è utilizzata per avviare un glassware dalla home card. Per impartire un comando vocale, l'utente pronuncia il trigger "Ok Glass" seguito dall'attività che desidera intraprendere; la lista delle attività supportate compare sul display non appena lo smart-glass rileva il trigger vocale. Il dispositivo è in grado di riconoscere solamente comandi in lingua inglese, dunque la descrizione delle attività è riportata in tale lingua. Ad esempio, con riferimento a Figura 4.9, qualora l'utente intenda scrutare il cielo notturno per esplorare le stelle e vedere il loro posizionamento, può pronunciare "Ok Glass, explore the stars". Dopodiché lo smart-glass presenta la lista dei glassware con cui è possibile intraprendere tale attività; l'utente può a questo punto pronunciare "Sky Map" oppure "Star Chart" per avviare il glassware che preferisce utilizzare.

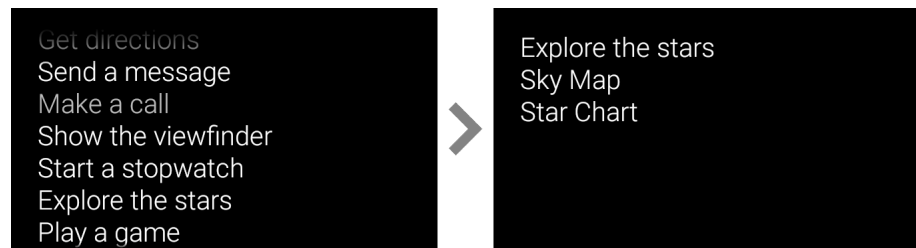


Figura 4.9: Avvio di un glassware tramite comando vocale.

### Comandi vocali contestuali

Le immersion che implementano un menu contestuale possono supportare l'attivazione di una voce di menu tramite comando vocale. Il funzionamento è analogo a quello dei comandi vocali principali: mentre il display visualizza la UI dell'immersion, l'utente può pronunciare "Ok Glass" per attivare il menu vocale. A questo punto il display mostra la lista delle voci di menu disponibili e l'utente può pronunciare il titolo della voce di menu che desidera attivare.

### Riconoscimento vocale

Questa modalità di input vocale consente di fornire input testuale all'applicazione in esecuzione sullo smart-glass. Invocando questa funzionalità il glassware ha infatti la facoltà di convertire in formato testo una frase in linguaggio naturale pronunciata dall'utente. Ad esempio, qualora l'utente desideri comporre un'email, grazie a questa feature può specificarne l'oggetto e il corpo. Anche in questo caso il sistema è in grado di riconoscere solamente frasi in lingua inglese.

#### 4.3.6 Touchpad

Il touchpad dello smart-glass è utilizzato come sistema di input complementare ai comandi vocali. Infatti, per svolgere alcuni tipi di interazione col dispositivo –come ad esempio navigare la timeline o una qualunque lista di card– è indispensabile il suo utilizzo. L'esperienza offerta dai Google Glass dunque non può essere considerata totalmente hands-free. D'altra parte, in linea di principio il touchpad potrebbe

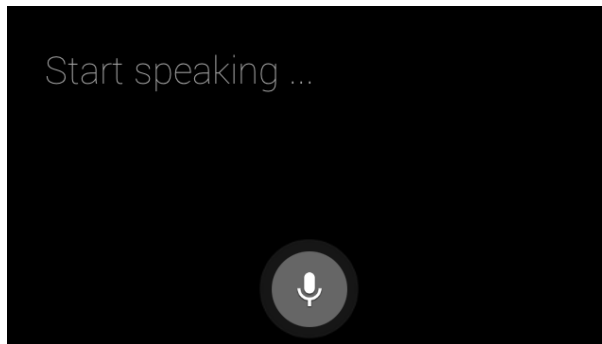


Figura 4.10: La UI della card di riconoscimento vocale.

anche essere utilizzato come sistema di input principale, poiché consente di accedere a tutte le feature del dispositivo tranne l'input testuale (funzionalità realizzabile unicamente tramite riconoscimento vocale). Le gesture principali per interagire con il touchpad sono il *tap* e lo *swipe*: per eseguire la prima gesture l'utente tocca la superficie del touchpad con un dito per un istante, mentre per compiere la seconda fa scorrere un dito sul touchpad verso sinistra, verso destra oppure verso il basso. Per avviare un glassware tramite il touchpad l'utente può eseguire un tap sulla home card della timeline, scorrere le attività disponibili compiendo più swipe verso sinistra o verso destra, selezionare l'attività che intende intraprendere eseguendo un tap, scorrere i glassware disponibili per tale attività tramite swipe e infine compiere un tap per avviare il glassware desiderato.

Nonostante le immersion abbiano la facoltà di ridefinire le azioni intraprese quando l'utente interagisce con il touchpad, i comportamenti predefiniti delle gesture principali sono i seguenti:

- **tap**: seleziona la card corrente, apre il menu contestuale, attiva la voce di menu corrente
- **swipe verso sinistra/destra**: naviga la timeline, scorre le voci del menu contestuale, scorre una lista di card
- **swipe verso il basso**: chiude un'immersion e torna alla timeline, chiude il menu contestuale, torna al livello di navigazione superiore, spegne lo schermo dello smart-glass.

## 4.4 Approcci di sviluppo

I nuovi costrutti che la piattaforma dello smart-glass propone –la timeline, le card e il sistema a comandi vocali in primis– hanno un duplice scopo: da una parte forniscono un’esperienza utente ad hoc, specifica per sistemi computazionali basati su smart-glass; dall’altra suggeriscono nuove astrazioni e nuovi meccanismi di interazione con cui progettare applicazioni e sistemi software per questa nuova tipologia di piattaforma. Nel seguito si presentano gli approcci di sviluppo che è possibile adottare per la realizzazione di applicazioni per Google Glass.

### 4.4.1 Glass Development Kit

Lo strumento di sviluppo standard per realizzare glassware è il *Glass Development Kit* (GDK): si tratta di un add-on dell’Android SDK che consente di sviluppare applicazioni native per Google Glass. Utilizzando il GDK è possibile realizzare glassware in un ambiente Android-based che permette di accedere all’hardware di basso livello del dispositivo, di implementare funzionalità che non richiedono connettività di rete e di supportare interazioni real-time con l’utente. Inoltre, il GDK consente di avvalersi della vasta gamma di API Android e di tutti gli strumenti di sviluppo esistenti per tale piattaforma, come ad esempio ADB (*Android Debug Bridge*).



Figura 4.11: Funzionalità e API implementate dall’Android SDK e dal GDK.

### 4.4.2 Mirror API

Questa collezione di API RESTful consente di sviluppare glassware tramite l’implementazione di servizi cloud-based che interagiscono con

il dispositivo. La principale funzionalità supportata dalle Mirror API è l'inserimento periodico di static card nella timeline: ad esempio, un glassware realizzato con questo approccio di sviluppo potrebbe continuamente selezionare le news di rilievo e presentarle ogni ora in una card di notifica. Le Mirror API non richiedono l'esecuzione di codice sullo smart-glass, e possono essere utilizzate con i seguenti linguaggi e piattaforme: Go, Java, .NET, PHP, Python e Ruby. Le applicazioni realizzate con questo approccio non hanno accesso all'hardware dello smart-glass e, essendo le Mirror API una tecnologia web-based, richiedono necessariamente la connettività di rete.

Questo approccio di sviluppo non è progettato per rispondere alle richieste dell'utente in tempo reale o con bassa latenza. Ad esempio, non sarebbe opportuno realizzare un glassware per cercare ricette di cucina con questa tecnologia poiché l'utente, dopo aver pronunciato il nome della ricetta da ricercare, si aspetterebbe che i risultati della ricerca vengano mostrati immediatamente. Le Mirror API invece inserirebbero nella timeline la card con tali risultati con un certo ritardo temporale, offrendo un'esperienza utente non ottimale. D'altra parte, questa tecnologia risulta adatta per interazioni di tipo *fire-and-forget* nelle quali l'utente non si aspetta una risposta dal glassware dopo aver invocato la funzionalità desiderata; un esempio di questo tipo di interazione può essere la creazione di una nota tramite il comando "Ok Glass, take a note...".

### 4.4.3 Glassware ibridi

Il Glassware Development Kit e le Mirror API non costituiscono due strumenti di sviluppo mutualmente esclusivi: è possibile adottare un approccio di progettazione ibrido che si avvalga di entrambe le tecnologie. Le Mirror API infatti offrono la possibilità di inserire nella timeline static card che, attraverso l'invocazione di una voce di menu, consentono di avviare live card e immersioni sviluppate con il GDK. Questo tipo di approccio consente di utilizzare le static card come notifica, e impiega le live card e le immersioni per offrire un'esperienza interattiva.



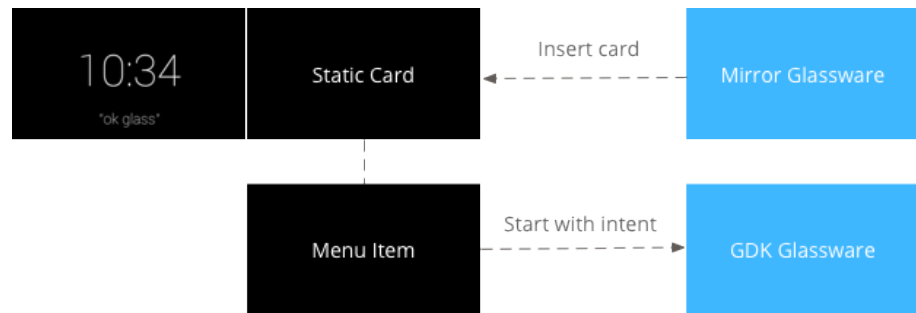


Figura 4.12: Schema di interazione di un glassware ibrido.

## 4.5 Pattern di riferimento

L'utilizzo di pattern di interazione collaudati e condivisi da tutti i glassware aiuta ad offrire all'utente una User Experience uniforme, consistente e predicibile. Nel seguito si illustrano i pattern di riferimento che Google propone al fine di impiegare efficacemente le astrazioni della UI e i meccanismi di interazione supportati dallo smart-glass.

### 4.5.1 Ongoing task

Il pattern *ongoing task* è adatto per attività di lunga durata che richiedono periodicamente l'attenzione dell'utente: il glassware inserisce nella timeline una live card che l'utente torna a consultare di tanto in tanto mentre svolge altre attività nel mondo reale o con lo smart-glass. Ad esempio, un glassware che implementa un cronometro potrebbe adottare questo pattern, consentendo all'utente di controllare periodicamente il proprio tempo durante una seduta di jogging. Le live card richiedono un contesto di esecuzione di lunga durata per poter svolgere la loro attività, dunque è necessario gestire il loro rendering all'interno di un servizio Android di background. La card può essere pubblicata sulla timeline non appena il servizio che la gestisce viene avviato, oppure all'occorrenza di un evento che il servizio monitora. Il rendering della live card può avvenire a bassa frequenza –ad esempio ogni 10 secondi– oppure ad alta frequenza, fino ad un massimo valore consentito dal dispositivo.

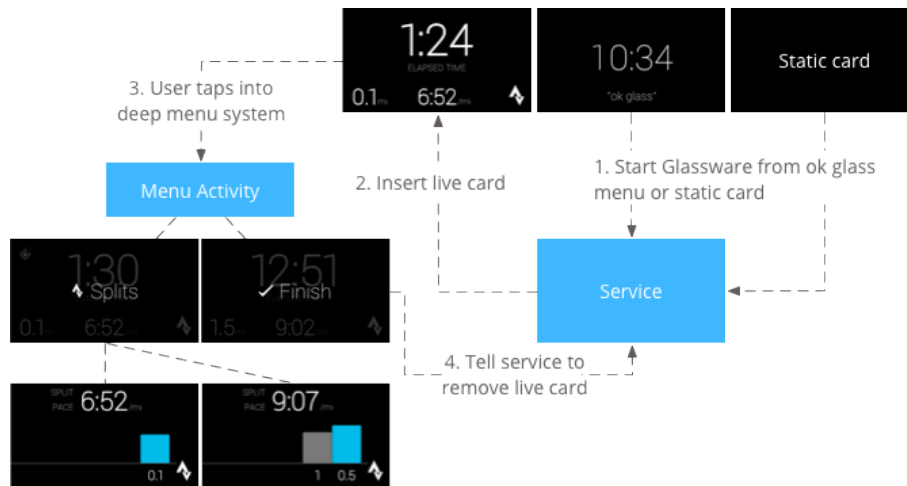


Figura 4.13: Schema di esempio di un glassware che adotta il pattern ongoing task.

## Rendering a bassa frequenza

Questa modalità è adatta per creare live card che presentano contenuti semplici e che non richiedono un rendering costante o aggiornamenti frequenti della UI. Il rendering a bassa frequenza è supportato da un ristretto insieme di elementi grafici Android, poiché richiede una UI supportata dalla classe `RemoteViews`, che rappresenta una gerarchia di elementi grafici che possono essere visualizzati su un processo differente. I widget e i layout Android supportati sono i seguenti: `FrameLayout`, `LinearLayout`, `RelativeLayout`, `GridLayout`, `AdapterViewFlipper`, `AnalogClock`, `Button`, `Chronometer`, `GridView`, `ImageButton`, `ImageView`, `ListView`, `ProgressBar`, `StackView`, `TextView` e `ViewFlipper`.

Per creare una live card a bassa frequenza è necessario definire il layout o il widget che si intende renderizzare e implementare il servizio Android che gestisce la card. In Listing 4.1 si riporta uno stralcio implementativo di un servizio di esempio che renderizza la live card di una partita di basket fittizia.

```

1 public class BasketService extends Service {
2     private LiveCard mLiveCard;
3     private RemoteViews mLiveCardView;
4     private int homeScore, awayScore;
5     private Random mPointsGenerator;
6     private final Handler mHandler = new Handler();

```

```
7     private final UpdateLiveCardRunnable
          mUpdateLiveCardRunnable = new
              UpdateLiveCardRunnable();
8     private static final long DELAY_MILLIS = 30000;
9     private static final String LIVE_CARD_TAG = "
          BasketLiveCard";
10
11     @Override
12     public int onStartCommand(Intent intent, int
          flags, int startId) {
13         if (mLiveCard == null) {
14             // Get an instance of a live card
15             mLiveCard = new LiveCard(this, LIVE_CARD_TAG)
                ;
16             // Inflate a layout into a remote view
17             mLiveCardView = new RemoteViews(
                getPackageName(), R.layout.main_layout);
18             // Set up initial RemoteViews values
19             homeScore = 0;
20             awayScore = 0;
21             mLiveCardView.setTextViewText(R.id.
                home_team_name_text_view, "Lakers");
22             mLiveCardView.setTextViewText(R.id.
                away_team_name_text_view, "Celtics");
23             mLiveCardView.setTextViewText(R.id.
                footer_text, "4th quarter");
24             // Set up the live card's action with a
                pending intent to show a menu when tapped
25             Intent menuIntent = new Intent(this,
                MenuActivity.class);
26             menuIntent.addFlags(Intent.
                FLAG_ACTIVITY_NEW_TASK | Intent.
                FLAG_ACTIVITY_CLEAR_TASK);
27             mLiveCard.setAction(PendingIntent.getActivity
                (this, 0, menuIntent, 0));
28             // Publish the live card on the timeline
29             mLiveCard.publish(PublishMode.REVEAL);
30             // Queue the update text runnable
31             mHandler.post(mUpdateLiveCardRunnable);
32         }
33         return START_STICKY;
34     }
35
36     @Override
37     public void onDestroy() {
```

```
38     if (mLiveCard != null && mLiveCard.isPublished
39         ()) {
40         //Stop the handler from queuing more Runnable
41         jobs
42         mUpdateLiveCardRunnable.setStop(true);
43         mLiveCard.unpublish();
44         mLiveCard = null;
45     }
46     super.onDestroy();
47 }
48 /**
49  * Runnable that updates live card contents
50  */
51 private class UpdateLiveCardRunnable implements
52     Runnable {
53     private boolean mIsStopped = false;
54     /**
55      * Updates the card with a fake score every 30
56      seconds as a demonstration
57      */
58     public void run(){
59         if(!isStopped()){
60             // Generate fake points
61             homeScore += mPointsGenerator.nextInt(3);
62             awayScore += mPointsGenerator.nextInt(3);
63             // Update the remote view with the new
64             scores
65             mLiveCardView.setTextViewText(R.id.
66                 home_score_text_view, String.valueOf(
67                 homeScore));
68             mLiveCardView.setTextViewText(R.id.
69                 away_score_text_view, String.valueOf(
70                 awayScore));
71             // Always call setViews() to update the
72             live card's RemoteViews
73             mLiveCard.setViews(mLiveCardView);
74             // Queue another score update in 30 seconds
75             .
76             mHandler.postDelayed(
77                 mUpdateLiveCardRunnable, DELAY_MILLIS);
78         }
79     }
80 }
```

```
71     ...
72     }
73
74     ...
75     }
```

Listing 4.1: Stralcio di un servizio Android che gestisce una live card che visualizza il punteggio di una partita di basket



Figura 4.14: Screenshot della live card che mostra il punteggio della partita di basket.

### Rendering ad alta frequenza

L'alta flessibilità di questa modalità di rendering consente di gestire sia layout e widget 2D sia complessi contenuti OpenGL 3D, interagendo direttamente con la superficie di disegno associata alla live card. Per ottenere una card contenente widget e grafica non OpenGL è necessario realizzare una classe che implementi l'interfaccia `DirectRenderingCallback` e creare un servizio che imposti un oggetto di tale classe come callback del `SurfaceHolder` della live card. Per visualizzare invece contenuti OpenGL è necessario realizzare una classe che implementi `GLRenderer` e creare un servizio che imposti un oggetto di tale classe come renderer della live card.

In Listing 4.2 si riporta la struttura di una classe che si occupa del rendering ad alta frequenza di una live card tramite la modalità `DirectRenderingCallback`, mentre in Listing 4.3 si riporta uno stralcio implementativo di un servizio di esempio che gestisce la live card e imposta opportunamente la callback del suo `SurfaceHolder` (linea 11).

```
1 public class LiveCardRenderer implements
    DirectRenderingCallback {
2     // About 30 FPS
3     private static final long FRAME_TIME_MILLIS = 33;
4     private SurfaceHolder mHolder;
5     private boolean mPaused;
6     private RenderThread mRenderThread;
7
8     @Override
9     public void surfaceChanged(SurfaceHolder holder,
    int format, int width, int height) {
10        // Update your views accordingly
11    }
12
13    @Override
14    public void surfaceCreated(SurfaceHolder holder)
    {
15        mPaused = false;
16        mHolder = holder;
17        updateRendering();
18    }
19
20    @Override
21    public void surfaceDestroyed(SurfaceHolder holder
    ) {
22        mHolder = null;
23        updateRendering();
24    }
25
26    @Override
27    public void renderingPaused(SurfaceHolder holder,
    boolean paused) {
28        mPaused = paused;
29        updateRendering();
30    }
31
32    // Start or stop rendering according to the
    timeline state
33    private void updateRendering() {
34        boolean shouldRender = (mHolder != null) && !
    mPaused;
35        boolean rendering = mRenderThread != null;
36
37        if (shouldRender != rendering) {
38            if (shouldRender) {
```

```
39         mRenderThread = new RenderThread();
40         mRenderThread.start();
41     } else {
42         mRenderThread.quit();
43         mRenderThread = null;
44     }
45 }
46 }
47
48 // Draws the view in the SurfaceHolder's canvas
49 private void draw() {
50     Canvas canvas;
51     try {
52         canvas = mHolder.lockCanvas();
53     } catch (Exception e) {
54         return;
55     }
56     if (canvas != null) {
57         // Draw on the canvas
58         mHolder.unlockCanvasAndPost(canvas);
59     }
60 }
61
62 // Redraws in the background
63 private class RenderThread extends Thread {
64     private boolean mShouldRun;
65
66     // Initializes the background rendering thread
67     public RenderThread() {
68         mShouldRun = true;
69     }
70
71     // Returns true if the rendering thread should
72     // continue to run
73     private synchronized boolean shouldRun() {
74         return mShouldRun;
75     }
76
77     // Requests that the rendering thread exit at
78     // the next opportunity
79     public synchronized void quit() {
80         mShouldRun = false;
81     }
82
83     @Override
```

```

82     public void run() {
83         while (shouldRun()) {
84             draw();
85             SystemClock.sleep(FRAME_TIME_MILLIS);
86         }
87     }
88 }
89 }

```

Listing 4.2: Struttura di una classe che realizza `DirectRenderingCallback`

```

1  public class LiveCardService extends Service {
2      private LiveCard mLiveCard;
3      private static final String LIVE_CARD_TAG = "
        LiveCardDemo";
4
5      @Override
6      public int onStartCommand(Intent intent, int
        flags, int startId) {
7          if (mLiveCard == null) {
8              mLiveCard = new LiveCard(this, LIVE_CARD_TAG)
9                  ;
10             // Enable direct rendering
11             mLiveCard.setDirectRenderingEnabled(true);
12             mLiveCard.getSurfaceHolder().addCallback(new
13                 LiveCardRenderer());
14             Intent intent = new Intent(context,
15                 MenuActivity.class);
16             mLiveCard.setAction(PendingIntent.getActivity
17                 (context, 0, intent, 0));
18             mLiveCard.publish(LiveCard.PublishMode.SILENT
19                 );
20         } else {
21             mLiveCard.navigate();
22         }
23         return START_STICKY;
24     }
25
26     @Override
27     public void onDestroy() {
28         if (mLiveCard != null && mLiveCard.isPublished
29             ()) {
30             mLiveCard.unpublish();
31             mLiveCard = null;
32         }
33     }
34 }

```



```

27     super.onDestroy();
28 }
29
30 ...
31 }

```

Listing 4.3: Stralcio di un servizio che gestisce una live card tramite `DirectRenderingCallback`

### Menu contestuale

Siccome le live card utilizzano il sistema di menu standard di Android, il pattern ongoing task prevede che per implementare il menu contestuale di una card sia necessario creare un'activity che lo gestisca e lo visualizzi sul display. Il menu contestuale di una live card può contenere voci per avviare un'immersion ed accedere alle funzionalità del glassware; inoltre, dovrebbe sempre essere presente una voce di menu per rimuovere la card dalla timeline e terminare il glassware. Ulteriori dettagli implementativi sul menu contestuale sono riportati nel paragrafo 4.5.2.

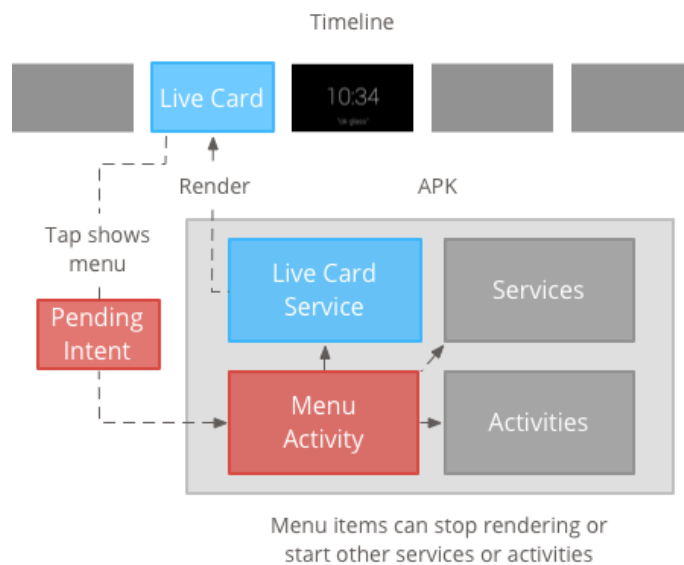


Figura 4.15: Una live card necessita di un'activity Android per realizzare il menu contestuale.

### 4.5.2 Immersion

Un'immersion è implementata da una o più activity Android che consentono di uscire momentaneamente dal contesto della timeline per offrire un'esperienza utente custom, in cui sia possibile personalizzare la UI del glassware e l'interazione con l'utente. Il pattern immersion descrive la creazione e la visualizzazione di activity in grado di gestire questo tipo di esperienza. Le immersion risultano particolarmente

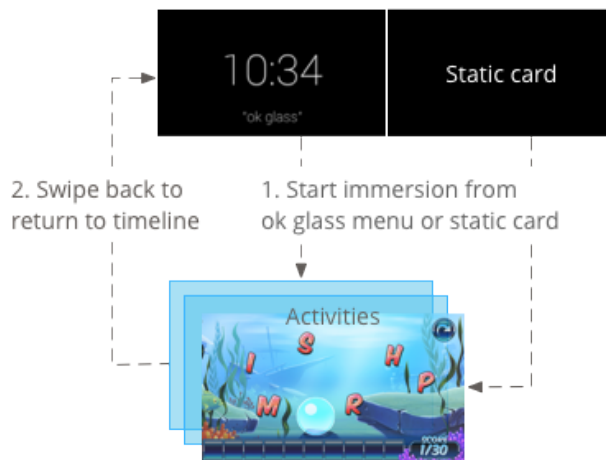


Figura 4.16: Schema di un glassware che adotta il pattern immersion.

adatte per realizzare glassware che richiedono l'attenzione dell'utente per un tempo prolungato o necessitano di sovrascrivere alcune funzionalità offerte dalla timeline, come ad esempio il comportamento delle gesture compiute sul touchpad e l'intervallo di inattività prima che si verifichi lo spegnimento display.

Per realizzare il pattern immersion si sviluppano i componenti del glassware utilizzando i costrutti della piattaforma Android, come activity, layout e widget; in seguito ci si avvale delle API del Glass Development Kit per offrire un'esperienza utente adatta al dispositivo, integrando la funzionalità dei comandi vocali, adottando uno stile grafico coerente e seguendo le best practice proposte dalla piattaforma. Il comportamento della gesture di swipe verso il basso non dovrebbe essere ridefinito; tale gesture è riservata per tornare al livello precedente

dello stack di activity della immersion, fino alla chiusura del glassware e il conseguente ritorno alla timeline.

### Menu contestuale

Proprio come accade per le live card, anche il menu contestuale di un'immersion è gestita da un'activity Android, che in questo caso coincide con quella impiegata per implementare l'immersion stessa. Utilizzando le API standard del sistema di menu della piattaforma Android il glassware adotta automaticamente lo stile grafico dello smart-glass. Per implementare il menu contestuale è sufficiente creare le risorse del menu e visualizzarle quando richiesto dall'utente, tipicamente all'occorrenza di una gesture di tap. Il processo di creazione delle risorse del menu per il glassware è lo stesso di quello per la piattaforma Android, ma è opportuno seguire alcune linee guida: il testo di ciascuna voce di menu dovrebbe essere breve e descrivere l'azione associata (ad esempio *Share* o *Reply all*); l'icona di ciascuna voce di menu dovrebbe avere dimensioni 50x50 pixel ed essere di colore bianco su sfondo trasparente.

L'activity della immersion contiene le callback da implementare per gestire la creazione, la visualizzazione e la chiusura del menu: `onCreateOptionsMenu()` carica il layout XML del menu, `onPrepareOptionsMenu()` può all'occorrenza mostrare o nascondere alcune voci di menu, infine `onOptionsItemSelected()` gestisce l'input dell'utente in base alla voce di menu che ha selezionato. Tipicamente, il menu contestuale dell'activity è mostrato quando l'utente esegue una gesture di tap sul touchpad: la callback associata a questo evento è denominata `onKeyDown()`. L'effettiva visualizzazione del menu avviene tramite l'invocazione del metodo `openOptionsMenu()`. In Listing 4.4 si riporta uno stralcio di codice che mostra un'implementazione di esempio delle callback appena descritte.

```
1 public class StopwatchActivity extends Activity {
2     @Override
3     public boolean onCreateOptionsMenu(Menu menu) {
4         MenuInflater inflater = getMenuInflater();
5         inflater.inflate(R.menu.stopwatch, menu);
6         return true;
7     }
```

```
8
9  @Override
10 public boolean onPrepareOptionsMenu(Menu menu) {
11     // Implement if needed
12 }
13
14 @Override
15 public boolean onOptionsItemSelected(MenuItem
16     item) {
17     // Handle item selection. Menu items typically
18     // start another activity, start a service, or
19     // broadcast another intent.
20     switch (item.getItemId()) {
21         case R.id.stop:
22             startActivity(new Intent(this,
23                 StopStopwatchActivity.class));
24             return true;
25         default:
26             return super.onOptionsItemSelected(item);
27     }
28 }
29
30 @Override
31 public boolean onKeyDown(int keyCode, KeyEvent
32     event) {
33     if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER) {
34         openOptionsMenu();
35         return true;
36     }
37     return super.onKeyDown(keyCode, event);
38 }
39
40 ...
41 }
```

Listing 4.4: Implementazione di esempio delle callback di gestione del menu contestuale

### 4.5.3 Periodic notification

Questo pattern prevede l'inserimento periodico nella timeline di static card di notifica, senza che l'utente compia alcuna azione. Per realizzare questo tipo di funzionalità è possibile utilizzare le Mirror API

oppure un servizio Android di background. Per inserire una static card avvalendosi delle Mirror API è necessario inoltrare all'endpoint REST una richiesta POST contenente la rappresentazione JSON della card. Tale richiesta può essere costruita specificando manualmente il contenuto del relativo messaggio HTTP oppure utilizzando le API fornite per uno dei linguaggi di programmazione supportati. In Listing 4.5 si riporta il contenuto di un messaggio HTTP che richiede l'inserimento di una static card contenente il testo *Hello world*. Qualora la richiesta sia elaborata con successo, si riceve il codice di risposta 201 Created e una copia della card creata. Il messaggio HTTP di risposta relativo alla richiesta di Listing 4.5 è riportato in Listing 4.6.

```
1 POST /mirror/v1/timeline HTTP/1.1
2 Host: www.googleapis.com
3 Authorization: Bearer {auth token}
4 Content-Type: application/json
5 Content-Length: 26
6
7 { "text": "Hello world" }
```

Listing 4.5: Messaggio HTTP di esempio per richiedere l'inserimento di una card nella timeline

```
1 HTTP/1.1 201 Created
2 Date: Tue, 25 Sep 2012 23:30:11 GMT
3 Content-Type: application/json
4 Content-Length: 303
5
6 {
7   "kind": "glass#timelineItem",
8   "id": "1234567890",
9   "selfLink": "https://www.googleapis.com/mirror/v1/
10     timeline/1234567890",
11   "created": "2012-09-25T23:28:43.192Z",
12   "updated": "2012-09-25T23:28:43.192Z",
13   "etag": "\"G5BIORWvj-0jWdBrdWrPZV7xPKw/
14     t25selcGS3uDEVT6FB09hAG-QQ\"",
15   "text": "Hello world"
16 }
```

Listing 4.6: Messaggio HTTP di risposta che comunica l'avvenuto inserimento di una card nella timeline

## 4.6 Potenzialità e limiti

I Google Glass Explorer Edition sono stati i primi smart-glass commercializzati per il mercato di massa statunitense. Essendo un modello prototipale risulta evidente che non si tratti di un prodotto perfettamente rifinito ed ottimizzato. Soprassedendo sulle problematiche di accettazione sociale e privacy di cui si è già parlato nel paragrafo 3.4.2, si ritiene che le principali limitazioni tecnologiche del dispositivo riguardino l'autonomia di alimentazione e l'eccessivo surriscaldamento del processore. Infatti, qualora si utilizzi lo smart-glass per svolgere attività computazionalmente dispendiose la batteria potrebbe scaricarsi completamente in meno di due ore e, nelle stesse circostanze, il sistema di raffreddamento passivo di cui è dotato lo smart-glass non è spesso in grado di dissipare efficacemente il calore, rendendo l'area del touchpad eccessivamente calda al tocco. Un'ulteriore limite tecnologico è dato dal fatto che il display dei Google Glass non risulta adatto per applicazioni di realtà aumentata immersiva. Non si tratta di una vera e propria limitazione in realtà, ma piuttosto di una decisione che riflette una scelta di design: come si è detto nel paragrafo 4.2, il dispositivo è stato progettato per interazioni brevi e puntuali, e non per utilizzi prolungati che richiedono un'eccessiva attenzione da parte dell'utente.

Per le ragioni appena descritte, si ritiene che lo smart-glass potrebbe non essere adatto ad ambienti di lavoro che richiedono strumenti altamente affidabili ed efficienti. D'altra parte, in contesti lavorativi che presentano requisiti meno stringenti i Google Glass potrebbero rappresentare una valida soluzione hands-free, in grado di apportare benefici in molteplici settori aziendali e industriali. Inoltre, la non immersività del display del dispositivo non costituisce necessariamente uno svantaggio: non sempre è necessario avere un'associazione visiva tra oggetti reali e virtuali, e in alcuni casi un'esperienza non immersiva potrebbe essere addirittura preferibile, poiché consente all'utente di vedere più chiaramente la scena reale. Per quanto riguarda il design e l'aspetto estetico, sebbene il dispositivo non passi di certo inosservato è indubbio che il suo ingombro limitato lo renda uno degli smart-glass più discreti proposti finora.

Una nuova versione del dispositivo orientata al mercato enterprise

è attualmente in fase di progettazione e, secondo alcune indiscrezioni [48], disporrà di un processore più potente, un display più ampio e offrirà una maggiore autonomia di alimentazione. La nuova versione del dispositivo potrebbe dunque risolvere alcune delle problematiche che affliggono la versione Explorer, e in tal caso potrebbe essere accolta favorevolmente dal mercato.





# Capitolo 5

## Lozioni Augmented Service

Come si è detto nel capitolo introduttivo, l'obiettivo di questo lavoro di tesi è stato lo studio e la progettazione di un sistema basato su smart-glass in grado di agevolare le attività di customer care del team Service del Gruppo Lozioni. Il sistema software da installare sullo smart-glass non dovrebbe essere progettato come un'applicazione standalone, ma come un nuovo componente dell'ecosistema software della business unit Energy; in questo modo l'applicativo può beneficiare dei servizi e della conoscenza in esso contenuti, ma anche apportare un valore aggiunto all'infrastruttura informatica dell'impresa.

In questo capitolo si presentano i principali sistemi che compongono l'ecosistema software della BU Energy e si illustrano le fasi di sviluppo della soluzione realizzata descrivendone i requisiti, l'architettura e l'ingegnerizzazione.

### 5.1 Ecosistema software BU Energy

L'infrastruttura informatica dell'impresa comprende un insieme coeso di servizi, applicazioni e database che gestiscono e supportano molteplici attività aziendali. Nel seguito si presentano i principali software di riferimento utilizzati dalla BU Energy e dalle sue imprese clienti.

### 5.1.1 Energy Management System

L'*Energy Management System* (EMS) è un sistema software sviluppato dal team R&D Energy del Gruppo Luccioni che si occupa della gestione di basso livello e dell'ottimizzazione delle fonti energetiche sostenibili e dei consumi dei clienti della BU Energy. La prima versione dell'EMS è stata sviluppata nel 2008: inizialmente era usato come software di supporto per la gestione di impianti fotovoltaici, ma nel corso degli anni è stato continuamente esteso e migliorato al fine di realizzare sofisticate logiche di controllo sugli asset e supportare molteplici fonti energetiche, tra cui idroelettrica, di storage e di cogenerazione. Tra le attività dell'EMS vi è l'acquisizione dei dati grezzi rilevati dai sensori installati sugli asset (punti di misura) del cliente e la loro storicizzazione: i dati misurati sul campo sono ricevuti dall'EMS, che provvede a trasmetterli al data center Luccioni tramite un sistema a code implementato dal software RabbitMQ<sup>1</sup>. Le macchine del data center infine storicizzano i dati salvandoli sul big data aziendale, realizzato da un vasto database SQL Server<sup>2</sup>.

L'EMS si occupa inoltre delle politiche di funzionamento e delle logiche di controllo che gestiscono in maniera automatizzata la produzione di energia degli impianti sostenibili, consentendo di ottimizzare lo scambio di potenza con la rete di distribuzione massimizzando l'autonomia energetica e minimizzando il costo energetico del cliente. Il sistema software supporta l'invio di comandi di controllo di basso livello (come ad esempio lo spegnimento o il riavvio di un asset) e gestisce lo stato di funzionamento degli asset, indicando la presenza di eventuali segnali di warning o di allarme. In caso di anomalia di funzionamento da parte di un asset l'EMS è in grado di notificare gli operatori tecnici responsabili dell'impianto, ad esempio inviando automaticamente un'e-mail o un SMS.

L'architettura dell'EMS è strutturata su molteplici layer gerarchici, e comprende i seguenti elementi (si veda Figura 5.1):

- *Managers*: driver per l'acquisizione di segnali e l'attuazione di comandi nel rispetto dei protocolli supportati (tra cui CAN,

---

<sup>1</sup><https://www.rabbitmq.com/>

<sup>2</sup><http://www.microsoft.com/sql/>

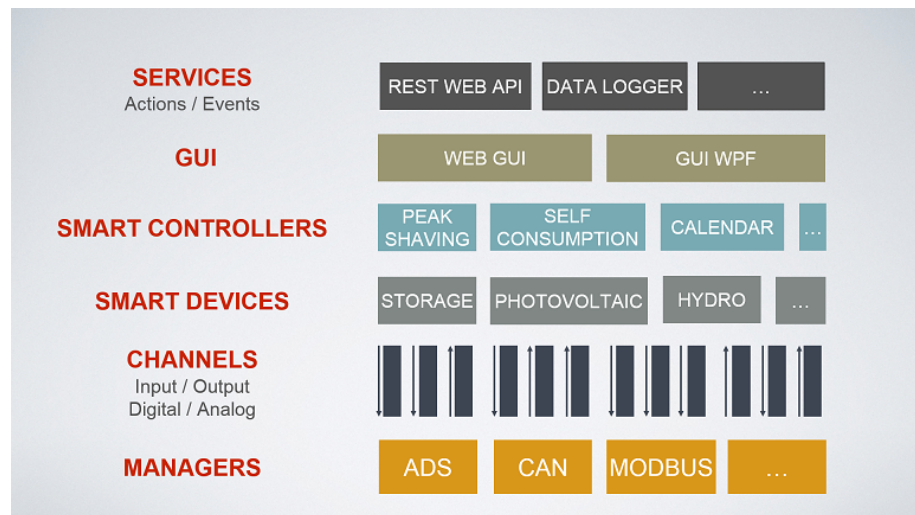


Figura 5.1: Architettura dell'EMS.

Modbus e IEC61850). Rappresentano il livello più basso dell'architettura e costituiscono l'interfaccia di connessione con il campo

- *Channels*: astrazione logica di basso livello utilizzata per mappare l'input/output di valori analogici e digitali sul campo. I channel implementano varie funzionalità di uso comune, come ad esempio il buffering dei dati
- *SmartDevices*: astrazione che rappresenta un generico dispositivo o asset che è possibile gestire con l'EMS. Uno SmartDevice è caratterizzato da una serie di proprietà o parametri di configurazione, un insieme di canali di input/output con i quali interagire con il campo e una logica di controllo che ne determina il funzionamento. Ad esempio, tale logica può consentire o proibire la carica o la scarica di un impianto di storage se determinate condizioni non sono soddisfatte
- *SmartControllers*: astrazione utilizzata per rappresentare una generica logica di controllo dell'EMS. Esempi di SmartController sono il *Peak shaving* (che ha l'obiettivo di tagliare i picchi

di consumo di un cliente) e il *Self consumption* (che si occupa dell'ottimizzazione dell'energia autoconsumata)

- *GUI*: interfacce grafiche per la visualizzazione e la configurazione degli asset gestiti dall'EMS (ovvero gli SmartDevices) e per la gestione delle logiche di controllo che li governano. Ad oggi sono state implementate una interfaccia web e una GUI WPF (Windows Presentation Foundation)
- *Services*: componenti software che implementano una o più funzionalità a livello applicativo. Esempi di servizi sono le Web API REST (consentono l'interazione con Channels, SmartDevices e SmartControllers), il *Data Logger* (logging dei dati) e il *Bind Engine* (triggering e gestione eventi).

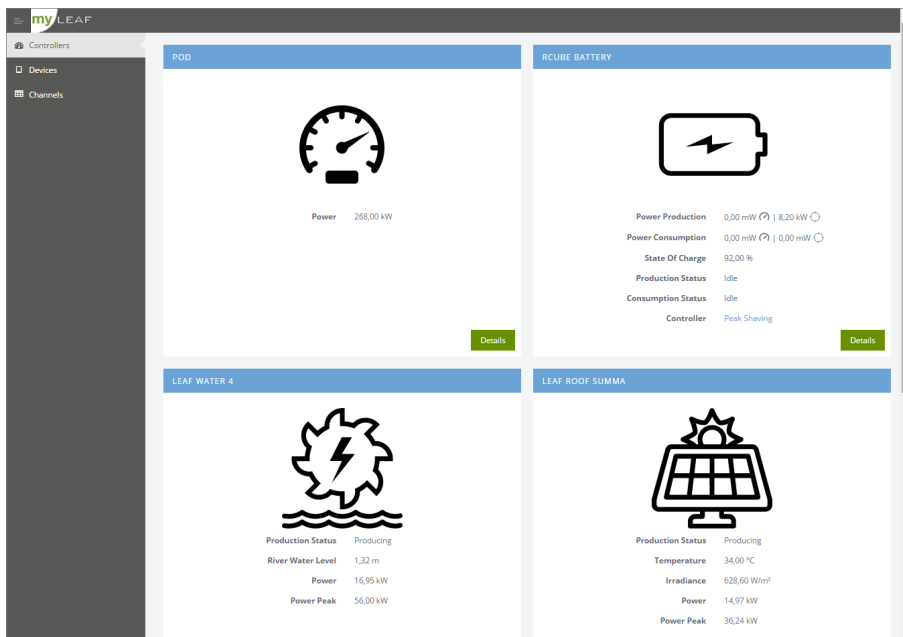


Figura 5.2: L'interfaccia web dell'EMS.

### 5.1.2 MyLeaf

*MyLeaf* è il sistema software di riferimento per manager, imprenditori clienti e collaboratori della BU Energy. Si tratta di un portale web-based modulare, che comprende un insieme di strumenti di analisi di alto livello e di reportistica in grado di presentare rilevanti informazioni di sintesi. Il portale è hostato sul cloud interno del data center Loccioni, ed è in esecuzione su un'infrastruttura di virtualizzazione VMWare<sup>3</sup> composta da un pool di macchine virtuali. Il software MyLeaf realizza le proprie funzionalità selezionando, elaborando ed aggregando la mole di dati di basso livello immagazzinata all'interno del big data Loccioni.

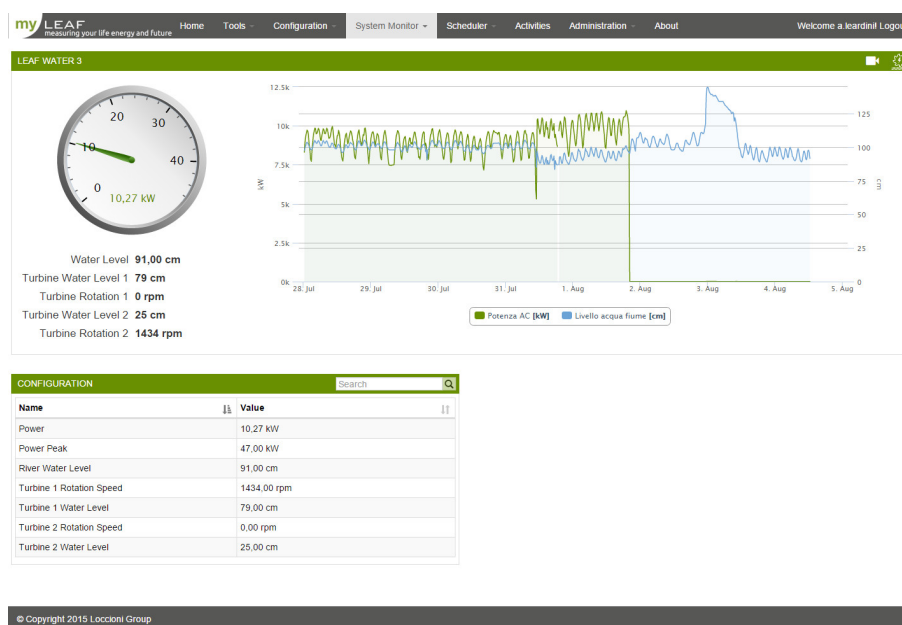


Figura 5.3: Lo strumento Monitor del software MyLeaf.

I principali strumenti di analisi che compongono MyLeaf includono:

- **Monitor:** dashboard configurabile che mostra informazioni di dettaglio sullo stato di funzionamento attuale e recente degli asset di una microrete (si veda Figura 5.3). Ciascun utente

<sup>3</sup><http://www.vmware.com/>

del portale può scegliere quali asset visualizzare sulla propria dashboard

- **Analysis:** consente di creare query personalizzate al fine di plot-tare, comparare ed aggregare dati storicizzati e statistiche su singoli punti di misura degli asset
- **Geoleaf:** geolocalizza gli impianti Loccioni presenti sul terri-torio e permette la consultazione di informazioni riassuntive e di dettaglio ad essi pertinenti, presentando una vista basata su mappa geografica. Consente inoltre di accedere alle immagini acquisite dalle webcam installate sul campo.

## 5.2 Requisiti

L'analisi delle criticità presenti nel processo di manutenzione e l'in-terazione con i membri del team Service ha portato alla definizione e alla validazione dei requisiti che l'applicazione Loccioni Augmented Service deve soddisfare. Le funzionalità del sistema software sono illu-strate nel diagramma dei casi d'uso riportato in Figura 5.4. Dopo aver descritto i requisiti dell'applicazione nel contesto di uno scenario di uti-lizzo reale, in Figura 5.5 si illustra il nuovo workflow di manutenzione abilitato da Loccioni Augmented Service.

### 5.2.1 Gestione del cliente

L'operatore, dopo aver preso in carico un intervento di manutenzio-ne, può avviare l'applicazione e selezionare l'impresa cliente che ha effettuato la richiesta; in questo modo il software carica tutte le in-formazioni a lei pertinenti di cui l'applicazione necessita. L'operatore dovrebbe poter selezionare l'impresa cliente attraverso una delle se-guenti modalità: pronunciando il suo nome, inquadrando un QR code che la identifichi, oppure scegliendola da una lista di nomi.

Lo smart-glass dovrebbe inoltre consentire all'operatore di effet-tuare una chiamata telefonica hands-free all'impresa cliente che ha richiesto l'intervento, in modo che possa acquisire ulteriori dettagli sull'impianto da mantenere.

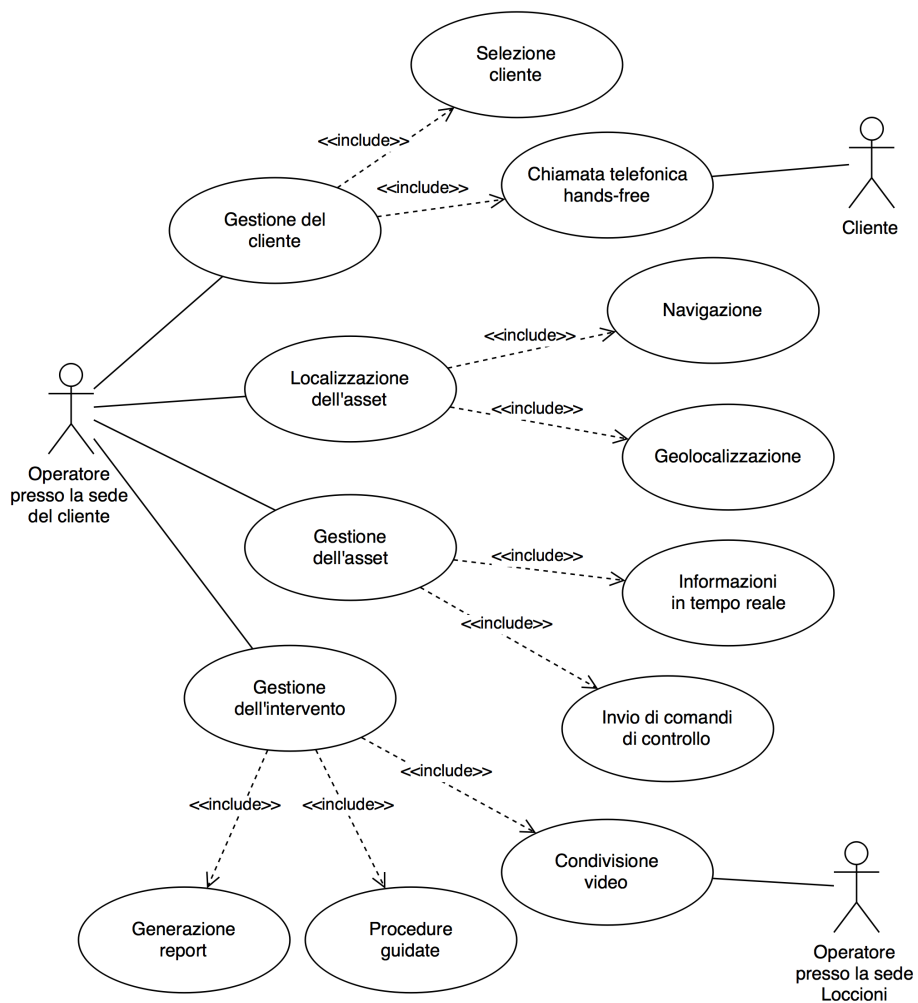


Figura 5.4: Diagramma dei casi d’uso dell’applicazione Loccioni Augmented Service.

### 5.2.2 Localizzazione dell'asset

L'operatore potrebbe necessitare di indicazioni stradali per raggiungere la sede dell'impresa cliente; il sistema di realtà aumentata dovrebbe quindi essere in grado di fornire tali indicazioni in maniera del tutto simile a quanto avviene con un navigatore tradizionale o uno smartphone.

Una volta raggiunta la sede del cliente, lo smart-glass dovrebbe fornire ulteriori indicazioni per raggiungere l'asset che necessita di manutenzione: il cliente potrebbe infatti possedere molteplici impianti energetici, oppure l'asset di interesse potrebbe trovarsi su una strada privata non raggiungibile tramite navigazione GPS. In queste circostanze è opportuno che lo smart-glass sia in grado di geolocalizzare gli asset dell'impresa cliente, mostrando all'interno del campo visivo dell'operatore un'icona in corrispondenza della posizione di ciascun impianto.

### 5.2.3 Gestione dell'asset

Dopo aver raggiunto l'impianto di interesse, l'operatore potrebbe necessitare di un resoconto aggiornato delle proprietà e dei parametri di funzionamento dell'asset. L'applicazione del sistema hands-free dovrebbe dunque essere in grado di mostrare in tempo reale informazioni di sintesi e di dettaglio sull'impianto che forniscano all'operatore un ragguglio sul suo stato attuale di funzionamento. Ad esempio, nel caso in cui l'asset da mantenere sia un impianto idroelettrico, il display del dispositivo potrebbe indicare la potenza istantanea di produzione, la velocità attuale delle turbine e l'eventuale presenza di segnali di allarme o di malfunzionamento.

Durante l'intervento di manutenzione, l'operatore potrebbe dover inviare dei comandi di input all'unità di controllo dell'asset, ad esempio per spegnere l'impianto, riavviarlo o eseguire altre procedure di diagnostica. Lo smart-glass dunque dovrebbe essere in grado di mostrare all'operatore la lista di comandi supportati dall'asset, richiedere la loro esecuzione e comunicare l'esito di tale richiesta.



### 5.2.4 Gestione dell'intervento

L'operatore presso la sede del cliente potrebbe necessitare dell'assistenza di un operatore remoto che si trova presso la sede Loccioni. Per facilitare la comunicazione tra i due tecnici, lo smart-glass dovrebbe supportare la trasmissione del video acquisito dalla videocamera in esso incorporata, consentendo all'operatore remoto di vedere in tempo reale ciò che sta vedendo il tecnico presso il cliente: la collaborazione remota diverrebbe in questo modo molto più puntuale ed efficace.

L'ingombro di manuali e documentazione cartacea potrebbe essere eliminato mostrando tali informazioni tecniche sul display del sistema di realtà aumentata. La maggioranza delle procedure di manutenzione eseguita dai tecnici Loccioni è standardizzata, dunque l'insieme di passi necessari per il loro completamento è solitamente noto a priori: questo significa che lo smart-glass potrebbe guidare l'operatore passo dopo passo durante l'intervento di manutenzione, suggerendo di volta in volta la prossima azione da compiere sull'impianto. La consultazione interattiva di procedure guidate potrebbe costituire un valido strumento per minimizzare l'errore umano e rendere il processi di manutenzione più efficienti e predicibili.

Il contenuto informativo di ciascun passo (*step*) di una procedura guidata dovrebbe poter essere presentato nella modalità più opportuna, ad esempio tramite una descrizione testuale, un'immagine che mostri il componente dell'asset su cui è necessario operare, una clip audio in voce sintetizzata o un breve video che illustri la prossima azione da compiere. Un'ulteriore possibilità, abilitata dalle capacità di realtà aumentata del sistema hands-free, dovrebbe consentire all'operatore di vedere indicazioni ed informazioni aumentate visivamente sovrapposte all'impianto su cui sta lavorando. Questa funzionalità consentirebbe al tecnico di vedere nel proprio campo visivo, ad esempio, un'animazione tridimensionale della prossima azione da compiere per procedere nell'intervento di manutenzione. Un bullone da rimuovere potrebbe essere segnalato dall'animazione di un cacciavite virtuale; allo stesso modo si potrebbe evidenziare un pulsante da premere o qualunque altro componente dell'asset con cui l'operatore dovrebbe interagire nello step corrente della procedura di manutenzione.

Oltre alla consultazione di una procedura guidata preesistente, è

importante che lo smart-glass consenta inoltre la creazione di materiale per nuove procedure guidate, al fine di favorire l'alimentazione della base di conoscenza delle attività di manutenzione. La possibilità di consultare procedure guidate su smart-glass e di ricevere assistenza remota tramite condivisione video consentirebbero ad operatori con limitata esperienza di eseguire interventi di manutenzione che altrimenti non sarebbero in grado di intraprendere: l'adozione di un dispositivo hands-free potrebbe dunque favorire un incremento delle competenze dei tecnici meno esperti e la riduzione dei costi di formazione.

Ad oggi al termine dell'attività di manutenzione l'operatore compila un report cartaceo contenente la descrizione e l'esito delle operazioni eseguite durante l'intervento. Il contenuto del report cartaceo viene successivamente riportato su un documento elettronico ed inviato per e-mail all'impresa cliente, al responsabile del team Service ed eventualmente ad altri collaboratori della BU Energy. Lo smart-glass indossato dal tecnico dovrebbe essere in grado di automatizzare tali operazioni, consentendogli di registrare l'esito di ciascun passo della procedura, di generare automaticamente un report elettronico al termine dell'intervento e di inviarlo ai destinatari desiderati.

### 5.3 Architettura

Dopo aver analizzato i requisiti del sistema Loccioni Augmented Service si è deciso che l'applicativo dovrebbe interagire principalmente con le web API REST dell'EMS, al fine di ottenere le informazioni in tempo reale sul funzionamento degli asset e di inviare i comandi di controllo da essi supportati. Inoltre, l'implementazione di alcune funzionalità –quali la consultazione di procedure guidate e la selezione del cliente– richiede la realizzazione di nuovi servizi web da aggiungere all'ecosistema software della business unit. Nel seguito si descrive l'architettura di alto livello adottata da ciascuna funzionalità di Loccioni Augmented Service, astraendo dalle scelte tecnologiche e dagli strumenti software che l'effettiva realizzazione dell'applicazione richiede.

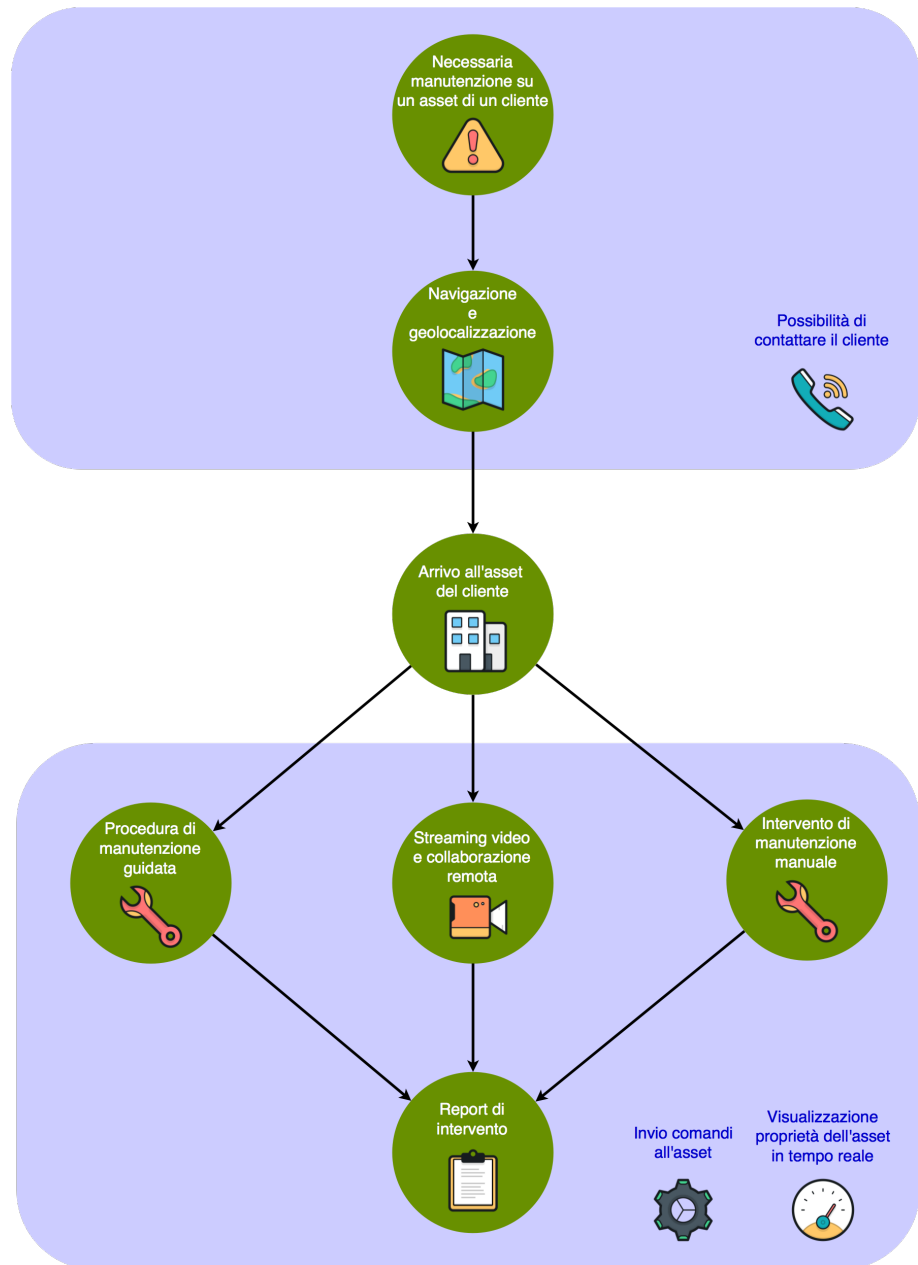


Figura 5.5: Flusso del processo di manutenzione abilitato da Loccioni Augmented Service.

### 5.3.1 Gestione del cliente

La gestione del cliente comprende la funzionalità di selezione dell'impresa cliente e la possibilità di effettuare una chiamata telefonica hands-free per contattarla.

#### Selezione del cliente

La prima attività che l'operatore intraprende dopo aver preso in carico un intervento di manutenzione è la selezione dell'impresa: questa funzione richiede che Loccioni Augmented Service interagisca con un servizio che contenga i dati riguardanti i clienti. L'applicazione dovrebbe infatti contattare un servizio web di anagrafica disponibile sulla VPN aziendale, fornirgli il nome del cliente che l'operatore intende selezionare –pronunciato vocalmente, letto da un QR code oppure selezionato da una lista– e ottenere come risposta tutti i dati di cui l'applicazione necessita, come ad esempio il recapito telefonico del cliente, i dati sugli asset di sua proprietà e le informazioni di contatto degli operatori assegnati al cliente.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.6.

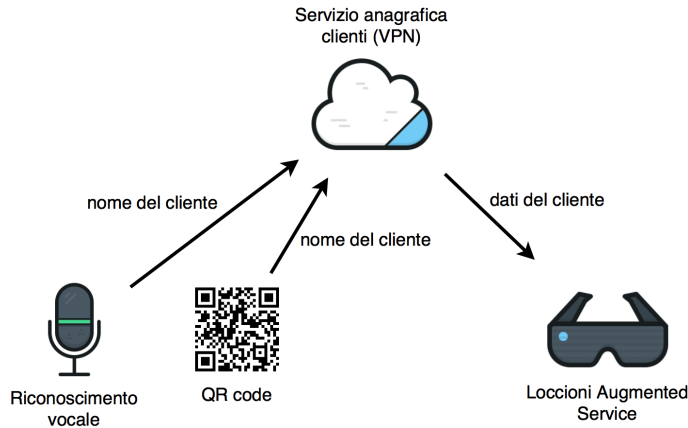


Figura 5.6: Architettura della funzionalità di selezione del cliente.

### Chiamata telefonica hands-free

Dopo che l'applicazione ha reperito i dati del cliente, l'operatore dovrebbe avere la possibilità di effettuare chiamate telefoniche hands-free, accedendo ad una rubrica integrata nell'applicazione che consenta di contattare agevolmente il cliente e gli operatori a lui assegnati. Lo smart-glass adottato dovrà ovviamente supportare le chiamate telefoniche, accedendo direttamente alla rete cellulare oppure stabilendo una connessione con lo smartphone dell'operatore.

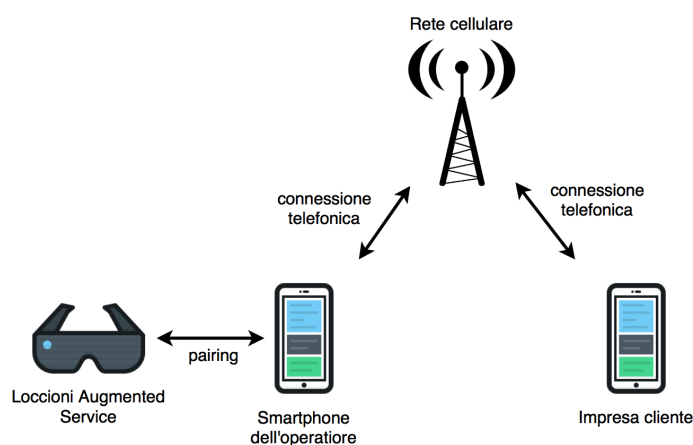


Figura 5.7: Architettura della funzionalità di chiamata telefonica hands-free.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.7.

### 5.3.2 Localizzazione dell'asset

La localizzazione dell'asset comprende le funzionalità di navigazione per raggiungere la sede del cliente e di geolocalizzazione per individuare l'asset che necessita di manutenzione.

#### Navigazione

Per supportare la navigazione verso la sede del cliente è indispensabile che lo smart-glass indossato dall'operatore sia dotato di sensore GPS, giroscopico e bussola. Poiché l'implementazione ex novo di un software

di navigazione sarebbe complessa e dispendiosa, per realizzare tale funzionalità all'interno dell'applicazione Loccioni Augmented Service si è deciso di impiegare un'applicazione o un framework di navigazione di terze parti.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.8.

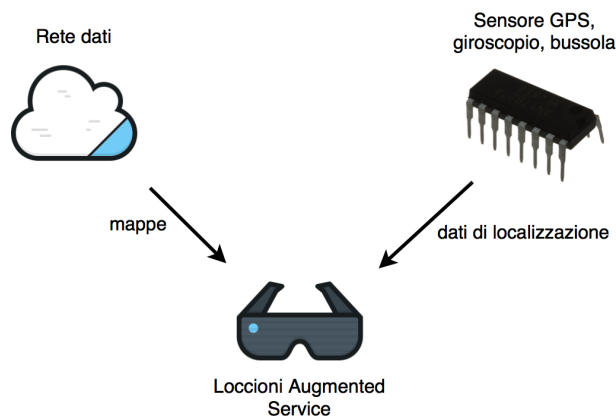


Figura 5.8: Architettura della funzionalità di navigazione.

### Geolocalizzazione

Anche la feature di geolocalizzazione richiede che lo smart-glass sia dotato di sensori di localizzazione. Il display degli smart-glass potrebbe mostrare –oltre alle icone che indicano la posizione degli asset rispetto alla visuale dell'operatore– anche il loro stato attuale, indicando se stanno funzionando correttamente o se hanno attivato un segnale di allarme o di warning. Come si è detto nel paragrafo 5.1.1, le informazioni sullo stato degli asset sono pubblicate su un URL della VPN Loccioni dal servizio Web API REST dell'EMS. Ad oggi le informazioni pubblicate dall'EMS non comprendono le coordinate GPS degli asset: questo significa che per implementare la funzionalità di geolocalizzazione è necessario modificare il servizio web dell'EMS affinché includa tali informazioni, oppure creare un nuovo servizio ad-hoc che, prendendo in ingresso le informazioni identificative di un cliente, fornisca le coordinate GPS degli asset di sua proprietà.

L'implementazione del sistema di tracciamento di icone all'interno del campo visivo dell'utente potrebbe richiedere una fase preliminare di scouting delle tecnologie e dei framework di realtà aumentata odierni al fine di determinare l'approccio migliore per realizzare questa funzionalità.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.9.

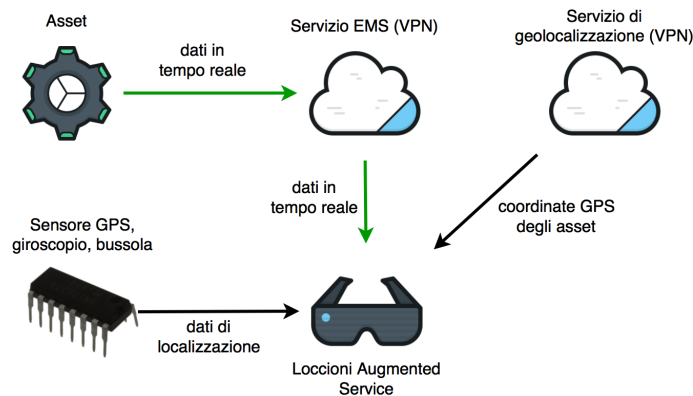


Figura 5.9: Architettura della funzionalità di navigazione.

### 5.3.3 Gestione dell'asset

La gestione dell'asset comprende la visualizzazione di informazioni sul suo funzionamento in tempo reale e l'invio di comandi di controllo e di diagnostica.

#### Informazioni in tempo reale

Le web API REST dell'EMS forniscono i dati aggiornati sugli asset del cliente in formato JSON: tali informazioni dovrebbero essere interpretate e presentate opportunamente sullo smart-glass affinché forniscano all'operatore un feedback immediato sullo stato attuale dell'asset e sui suoi parametri di funzionamento. L'applicazione Loccioni Augmented Service dovrebbe contattare periodicamente il servizio web dell'EMS al fine di mostrare all'operatore informazioni sempre aggiornate.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.10.

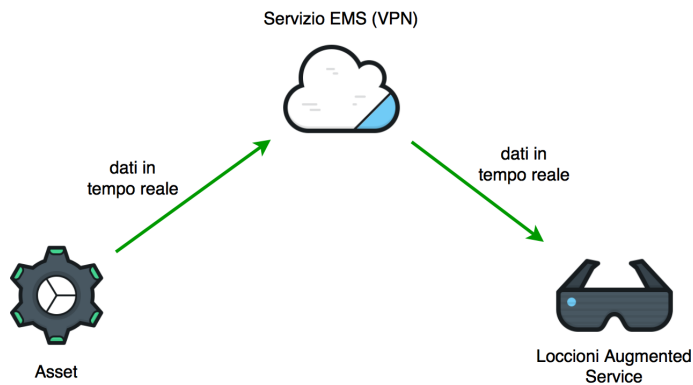


Figura 5.10: Architettura della funzionalità di visualizzazione di informazioni sugli asset in tempo reale.

### Invio di comandi di controllo

L'invio di comandi di controllo è un'altra feature che dovrebbe interagire con il software di basso livello dell'EMS al fine di avviare l'esecuzione di procedure di diagnostica su un impianto. L'implementazione della feature richiede l'interazione con un servizio dell'EMS che si interfaccia all'unità di controllo dell'asset per richiedere l'esecuzione effettiva del comando; tale servizio dovrà inoltre comunicare l'esito positivo della richiesta, oppure indicare l'errore riscontrato.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.11.

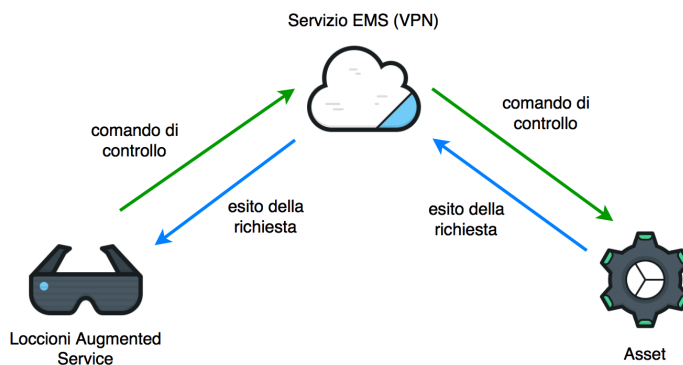


Figura 5.11: Architettura della funzionalità di invio di comandi di controllo.



### 5.3.4 Gestione dell'intervento

La gestione dell'intervento comprende le funzionalità di condivisione video, di consultazione di procedure guidate e di generazione del report di intervento.

#### Condivisione video

Questa funzionalità prevede che lo stream video acquisito dalla videocamera degli smart-glass sia trasmesso in rete e fruibile da un tecnico Loccioni addetto all'assistenza remota dell'operatore che si trova presso la sede del cliente. Idealmente lo stream dovrebbe includere l'audio ed essere bidirezionale, assumendo le caratteristiche di una videoconferenza che consenta ai due operatori di comunicare con sincronismo audiovisivo. Il tecnico che offre assistenza remota dalla sede del Gruppo dovrebbe dunque disporre di un applicativo per PC o dispositivi handheld che consenta la visualizzazione dello stream video dell'operatore presso il cliente, la trasmissione della propria voce ed eventualmente il video acquisito dalla webcam del PC o dalla videocamera del dispositivo handheld.

La progettazione e l'implementazione di questa feature richiede un'esplorazione preliminare delle tecnologie di riferimento attualmente disponibili per la trasmissione e la fruizione di feed video, al fine di adottare l'architettura più appropriata per il contesto applicativo specifico. Una possibile soluzione potrebbe prevedere l'installazione di un server di streaming in grado di accettare i feed video dei due operatori e di pubblicare ciascuno di essi su un URL.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.12.

#### Procedure di manutenzione guidate

La feature di consultazione di procedure guidate è fortemente orientata alla conoscenza e al know-how industriale; una delle questioni da affrontare in fase di progettazione è la scelta del supporto di memoria su cui mantenere il contenuto informativo delle procedure. Nonostante sia teoricamente possibile scegliere di archiviare le procedure di manutenzione sullo storage degli smart-glass, si è deciso che il loro

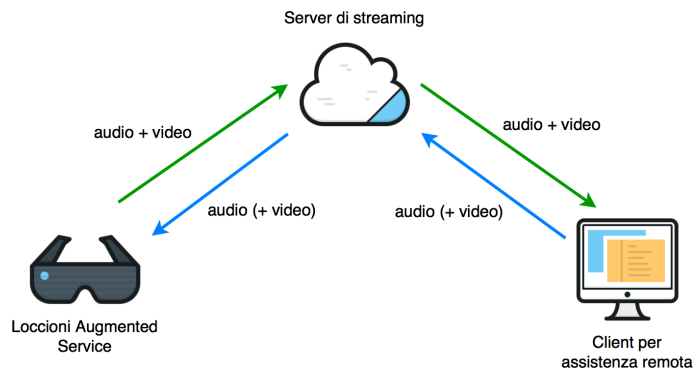


Figura 5.12: Architettura della funzionalità di condivisione video.

contenuto informativo dovrebbe essere pubblicato su un servizio web. Questa scelta non è stata dettata da ragioni legate alla potenzialmente limitata capacità di memoria del dispositivo, ma da una motivazione più sistemica: pubblicare la base di conoscenza delle procedure di manutenzione su un servizio web significherebbe renderla disponibile non solamente all'applicazione Loccioni Augmented Service, ma anche a tutto il resto dell'ecosistema software della BU Energy. Gli asset della stessa tipologia supportano le stesse procedure guidate, dunque il servizio web in questione dovrebbe essere progettato in modo che, dato in input il tipo di asset su cui è necessario effettuare la manutenzione, fornisca l'elenco delle procedure supportate da tale tipologia di impianto. Ad esempio, si può assumere che tutti gli impianti fotovoltaici installati dal Gruppo Loccioni presso le imprese clienti condividano lo stesso insieme di procedure di manutenzione, seppur non siano identici tra loro.

Durante un intervento, dopo che l'operatore ha scelto dall'elenco fornito dal servizio web la procedura che intende avviare, l'applicazione effettua un'ulteriore chiamata REST al servizio al fine di reperire i contenuti della procedura selezionata. A questo punto la procedura è pronta per essere visualizzata sul display degli smart-glass, uno step dopo l'altro. Gli step che guidano l'operatore mostrando contenuti in realtà aumentata presentano alcune problematiche analoghe a quelle già citate riguardo alla funzionalità di geolocalizzazione: anche in questo caso è necessario indagare le potenzialità dei framework di realtà aumentata attualmente disponibili per individuarne punti di forza e

limitazioni in fatto di tracciamento di contenuti aumentati all'interno del campo visivo dell'utente.

Come si è detto nel paragrafo 5.2.4, l'applicazione dovrebbe consentire la creazione di materiale per nuove procedure guidate al fine di alimentare i contenuti del relativo servizio web; ogniqualvolta l'operatore svolge un intervento di manutenzione per cui non esiste ancora una procedura guidata, l'applicazione può acquisire video e immagini da utilizzare come contenuti degli step della nuova procedura. Una volta che l'operatore ha completato l'intervento ed è tornato alla sede del Gruppo, può finalizzare l'editing del materiale acquisito e caricare sul servizio web la nuova procedura, che diverrà disponibile per gli interventi futuri.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.13.

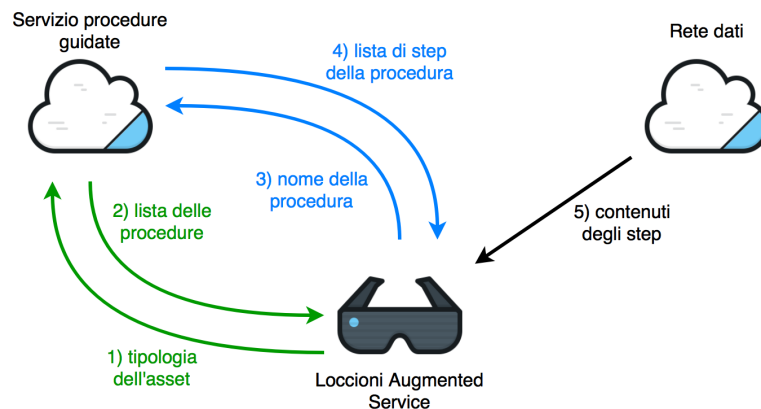


Figura 5.13: Architettura della funzionalità di consultazione di procedure guidate.

## Report di intervento

Ai fini della generazione del report di intervento, al termine di ogni passo di una procedura guidata l'operatore deve indicare se l'operazione eseguita allo step corrente è andata a buon fine oppure no. Si è deciso che tale indicazione dovrebbe avvenire vocalmente: così facendo ad ogni passo della procedura il tecnico può aggiungere una nota in linguaggio naturale e, nel caso in cui l'operazione da eseguire abbia esito negativo, può descrivere la causa di tale fallimento. Le note

vocali dovrebbero essere convertite in forma testuale ed incluse nel report di intervento, integrando le informazioni sull'esito dei singoli passi della procedura.

Il report ottenuto dovrebbe essere salvato sullo storage degli smart-glass e poter essere inviato tramite e-mail al cliente, al responsabile del team Service e ad altri operatori: gli indirizzi dei possibili destinatari sono contenuti tra le informazioni sul cliente caricate durante le fasi preliminari dell'intervento e dovrebbero essere disponibili nella rubrica integrata dell'applicazione.

Lo schema dell'architettura di alto livello di questa feature è riportata in Figura 5.14.

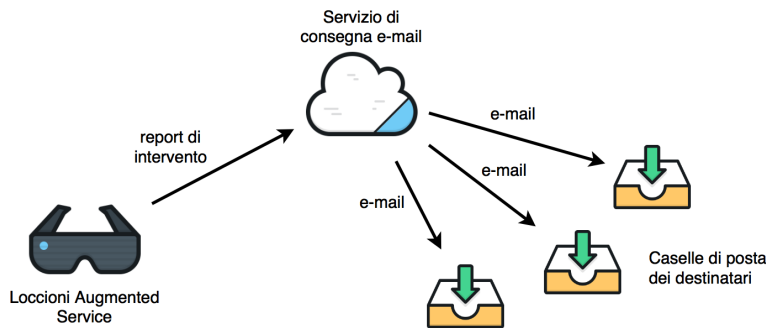


Figura 5.14: Architettura della funzionalità di generazione e invio del report di intervento.

## 5.4 Ingegnerizzazione

Al termine delle fasi di analisi dei requisiti e di progettazione dell'architettura è stato necessario definire gli strumenti tecnologici di riferimento per l'ingegnerizzazione del sistema Loccioni Augmented Service, ovvero la selezione della piattaforma hardware su cui sarà installata l'applicazione realizzata e degli strumenti software per la sua implementazione.

### 5.4.1 Strumenti tecnologici di riferimento

L'applicazione Loccioni Augmented Service in linea di principio può essere installata su qualsiasi smart-glass che soddisfi i prerequisiti

hardware presentati nel paragrafo 5.3 e che sia dotato di una piattaforma di sviluppo che supporti –nativamente o tramite l'utilizzo di un software framework di terze parti– l'implementazione delle funzionalità di realtà aumentata descritte nel paragrafo 5.2.

### **Piattaforma hardware e ambiente di sviluppo**

Attualmente la business unit Energy ha a disposizione dei Google Glass Explorer Edition: dopo aver verificato che il dispositivo soddisfi i prerequisiti appena descritti, si è deciso di adottarlo come piattaforma hardware dell'applicazione Loccioni Augmented Service. Le funzionalità del software sviluppato richiedono la connessione ad una rete dati (Wi-Fi o rete cellulare) ed alla rete telefonica: non essendo i Google Glass dotati di scheda SIM è necessario che l'operatore abbia con sé uno smartphone ed esegua il pairing Bluetooth tra i due dispositivi.

Il livello di interattività richiesto all'applicazione Loccioni Augmented Service e la necessità di accedere ai sensori hardware dello smart-glass ha suggerito l'adozione di un approccio di progettazione che promuove il Glass Development Kit come piattaforma di sviluppo di riferimento. L'IDE utilizzato per la creazione e il coding dell'applicazione Loccioni Augmented Service è stato Android Studio<sup>4</sup>, un ambiente di sviluppo basato su IntelliJ IDEA. Android Studio è un IDE in grado di fornire supporto in tutte le fasi del processo di sviluppo, comprendendo un insieme di sofisticati strumenti per la creazione, l'editing, il debugging, il profiling e l'esecuzione di applicazioni Android.

### **Framework di realtà aumentata**

Come si è detto nei paragrafi 5.3.2 e 5.3.4, la realizzazione delle funzionalità di realtà aumentata dell'applicazione ha richiesto l'analisi e la comparazione dei framework e delle tecnologie di realtà aumentata allo stato dell'arte; al termine di questa fase di scouting si è deciso di adottare Metaio [22] come framework di riferimento per l'applicazione Loccioni Augmented Service. Metaio è un software framework per lo sviluppo di applicazioni in realtà aumentata su piattaforme Android,

---

<sup>4</sup><https://developer.android.com/sdk/index.html>

iOS, Windows, OS X e Unity<sup>5</sup>. Il framework include un insieme di API per il tracciamento della scena inquadrata dalla videocamera del dispositivo e per la visualizzazione dei contenuti virtuali da sovrapporre ad essa. I tipi di contenuto virtuale che Metaio è in grado di gestire includono immagini 2D, clip video e modelli 3D statici ed animati. I metodi di tracciamento supportati dal framework sono molteplici; nel seguito si fornisce una descrizione sintetica delle macro-categorie in cui sono organizzati; l'elenco completo delle modalità di tracciamento è riportato in Figura 5.15.

- **Detection:** riconoscimento di QR code e codici a barre
- **Fiducial:** riconoscimento di marker predefiniti, marker LLA e immagini di forma rettangolare, aventi bordo scuro e disposte su uno sfondo chiaro
- **Sensor-based:** tracciamento basato su sensore GPS, accelerometro, giroscopio e bussola. Utile per realizzare contenuti virtuali geolocalizzati
- **Planar:** riconoscimento di pattern grafici di forma arbitraria, ovvero immagini con bordo di forma qualsiasi
- **Dummy:** metodo di tracciamento per oggetti con posa fissa rispetto alla visuale dell'utente. Utile per visualizzare informazioni di overlay solidali con l'inquadratura della videocamera
- **Markerless:** tracciamento della scena reale utilizzando tecnologie basate su SLAM (Simultaneous Localization And Mapping), modelli CAD e altri algoritmi 2D o 3D. Non richiede l'impiego di marker o pattern da riconoscere
- **Instant:** i dati utilizzati per il tracciamento sono estratti automaticamente da un'immagine acquisita dalla videocamera del dispositivo.

---

<sup>5</sup><https://unity3d.com/>

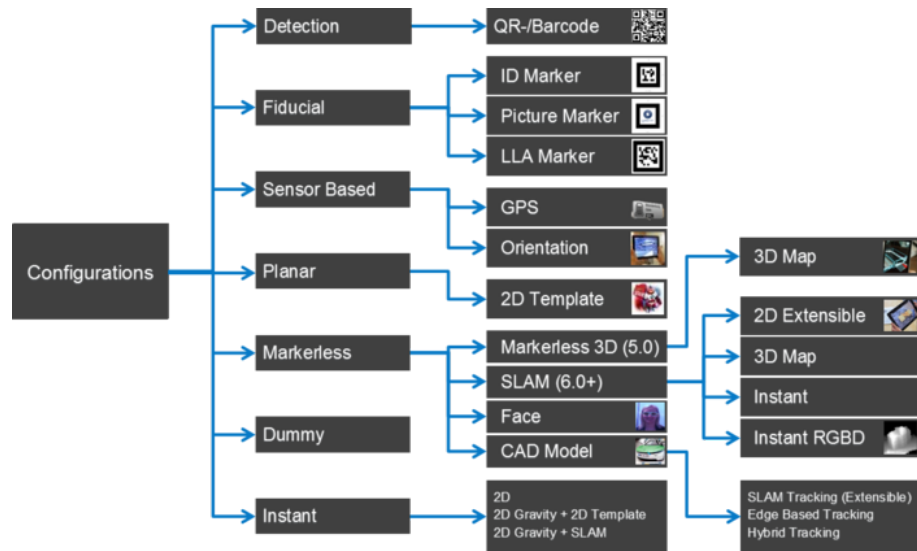


Figura 5.15: I metodi di tracciamento supportati da Metaio.

## 5.4.2 Struttura dell'applicazione

L'applicazione *Loccioni Augmented Service* è composta da due progetti Android distinti: *LoccioniService* e *LoccioniGlassLibrary*. Il progetto *LoccioniGlassLibrary* è identificato dal package `com.loccioni.glasslibrary` e comprende un insieme di API di utilità generale e l'implementazione delle funzionalità che potrebbero essere riutilizzate in futuro da altri *glassware* sviluppati dalla BU Energy, ovvero le feature di geolocalizzazione e di selezione del cliente. La funzionalità di geolocalizzazione degli asset, ad esempio, potrebbe essere riutilizzata durante i tour aziendali che il Gruppo *Loccioni* occasionalmente organizza per i dirigenti di imprese partner; durante la visita gli ospiti potrebbero indossare i *Google Glass* e vedere attorno a loro la disposizione e le caratteristiche degli asset installati presso la sede del Gruppo. Tornando alla struttura dell'applicazione, il progetto *LoccioniService*, identificato dal package `com.loccioni.service`, include invece l'implementazione delle restanti funzionalità, inerenti esclusivamente alle attività del team *Service*.

Ciascuno dei due progetti è organizzato in moduli Android. *LoccioniService* è composto dai moduli `app` e `libstreaming`: il primo rappresen-

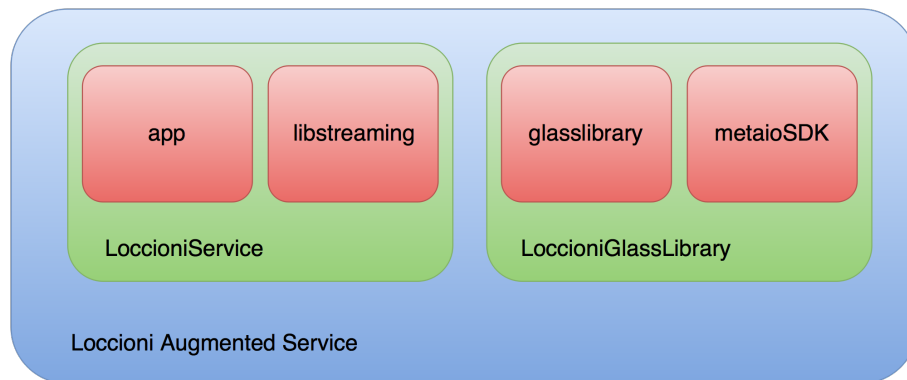


Figura 5.16: Struttura dei progetti e dei moduli dell'applicazione Loccioni Augmented Service.

ta il core dell'applicazione e include la maggior parte delle funzionalità; il secondo comprende le classi di supporto per l'implementazione dello streaming video. Il progetto `LoccioniGlassLibrary` comprende invece i moduli `glasslibrary` e `metaioSDK`: il primo contiene le API di utilità e include le feature di selezione del cliente e di geolocalizzazione, mentre `metaioSDK` contiene l'implementazione per Android del framework `Metaio` e offre un insieme di API per realizzare applicazioni di realtà aumentata. La struttura dei progetti e dei moduli che compongono `Loccioni Augmented Service` è illustrata in Figura 5.16.

Il glassware è realizzato da un'immersion composta da molteplici `activity` Android, ciascuna delle quali implementa una schermata della GUI. L'applicazione `Loccioni Augmented Service` può essere avviata pronunciando "Ok Glass, Service" dalla home card della timeline; al suo avvio viene presentata una splash card che mostra, oltre al logo dell'applicazione, il nome dell'impresa cliente attualmente selezionata (si veda Figura 5.17).

Il menu contestuale della splash card consente di accedere alle macrofunzionalità del glassware: in particolare la voce di menu `Assets` mostra la lista degli impianti del cliente e permette di svolgere tutte le attività inerenti ad un singolo asset. Le feature principali dell'applicazione sono accessibili tramite i menu contestuali della splash card e della card degli asset; alcune voci di menu sono replicate su entrambi i menu al fine di facilitarne l'accesso all'operatore (si veda Figura 5.18).



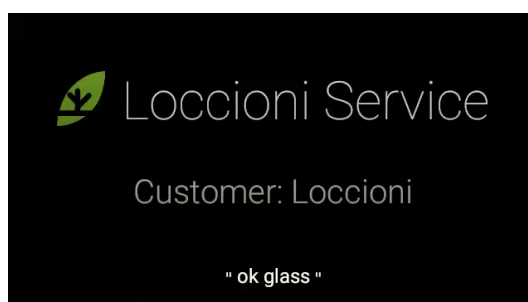


Figura 5.17: Splash card dell'applicazione Loccioni Augmented Service.

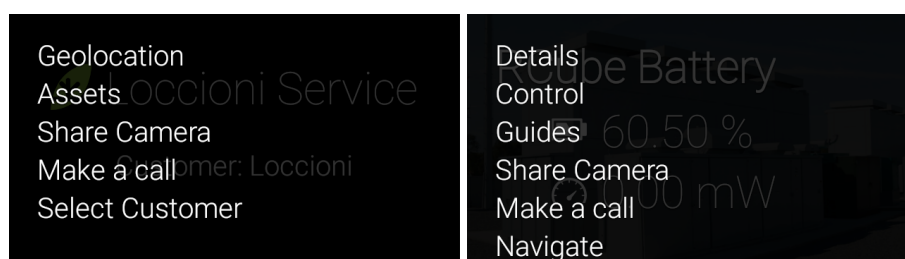


Figura 5.18: Menu della splash card (a sinistra) e della card degli asset (a destra).

### 5.4.3 Gestione del cliente

Nel seguito si descrivono i principali aspetti implementativi delle funzionalità di selezione del cliente e di composizione di una chiamata telefonica hands-free.

#### Selezione del cliente

La prima operazione che l'applicazione intraprende al suo avvio è il caricamento dei dati dell'ultimo cliente selezionato in precedenza. Nel caso in cui sia la prima volta che si avvia il glassware, viene caricato automaticamente il cliente di default *Loccioni*, utilizzato per la manutenzione degli asset installati presso la sede del Gruppo.

L'activity Android che implementa la splash card è definita dalla classe `SplashActivity` contenuta nel package `splash` del modulo `app`, il cui compito principale è reperire i dati inerenti all'ultimo cliente selezionato, in modo che siano disponibili a tutte le funzionalità dell'applicazione. Il componente software che si occupa del recupero effettivo

di tali informazioni è rappresentato dalla classe `CustomerDataResolver` contenuta nel package `customerdata` del modulo `glasslibrary`, la quale adempie al proprio compito caricando dapprima i dati dell'impresa cliente e in seguito i dati di funzionamento correnti degli asset di sua proprietà.

La selezione di un nuovo cliente avviene accedendo al menu della splash card e selezionando la voce *Select Customer*: a questo punto è possibile selezionare l'impresa cliente scegliendo il suo nome da una lista di card oppure inquadrando con la videocamera degli smart-glass un QR code che rappresenta il nome del cliente. La feature di selezione del cliente tramite la scansione di un QR code è stata realizzata apportando alcune modifiche minori ai sorgenti del progetto *BarcodeEye*<sup>6</sup>.

Idealmente il recupero delle informazioni sui clienti dovrebbe avvenire contattando il servizio di anagrafica di cui si è parlato nel paragrafo 5.3.1 ma, per questioni organizzative e di tempo, tale servizio non è stato realizzato in questo lavoro tesi. Attualmente il caricamento dei dati inerenti ai clienti avviene invece interpretando un opportuno file JSON che simula la risposta del web service mancante. Il file in questione è nominato `customers.json` ed è mantenuto sullo storage degli smart-glass all'interno del path `Loccioni\GlassLibrary\Customers`. L'implementazione della funzionalità di selezione del cliente è stata strutturata in modo che, qualora in futuro il servizio di anagrafica venga implementato, le modifiche da apportare al codice applicativo saranno minimali. Un esempio della struttura dei dati anagrafici di un cliente è riportata in Listing 5.1. Al fine di reperire i dati aggiornati degli asset del cliente, è necessario utilizzare le web API REST dell'EMS, effettuando periodicamente una chiamata all'URL definito dall'attributo `assetsUrl` del file JSON del cliente. La struttura e la gestione dei dati correnti degli impianti sarà descritta in maggior dettaglio nel paragrafo 5.4.5, quando si illustrerà la funzionalità di visualizzazione di informazioni in tempo reale.

```
1 {
2   "name": "Continental",
3   "email": "info@continental.com",
4   "phoneNumber": "123 456 7890",
```

---

<sup>6</sup><https://github.com/BarcodeEye/BarcodeEye>

```
5  "assetsUrl": "http://172.17.0.124:8092/api/v2/
    smartdevices/",
6  "operators": [
7    {
8      "name": "Lorenzo Spadari",
9      "email": "l.spadari@loccioni.com",
10     "phoneNumber": "123 456 7890"
11   },
12   {
13     "name": "Marco Piccinini",
14     "email": "m.piccinini@loccioni.com",
15     "phoneNumber": "123 456 7890"
16   }
17 ]
18 }
```

Listing 5.1: Esempio della struttura dei dati anagrafici di un'impresa cliente

### Chiamata telefonica hands-free

L'operatore incaricato dell'intervento di manutenzione può contattare l'impresa cliente accedendo alla rubrica integrata dell'applicazione. Per utilizzare questa funzionalità l'operatore accede al menu della splash card dell'applicazione e seleziona la voce *Make a call*. L'activity che implementa la feature è realizzata dalla classe `MakeACallActivity` contenuta nel package `call` del modulo `app`; il suo compito è presentare l'elenco dei recapiti telefonici caricati durante la selezione del cliente e presentarli su una lista di card. Dopo che l'operatore seleziona la card relativa al contatto da chiamare, lo smartphone accoppiato con i Google Glass instaura la telefonata; l'operatore a questo punto può utilizzare il microfono e l'altoparlante degli smart-glass per conversare senza dover impegnare le mani. L'attivazione della telefonata è stata implementata utilizzando i meccanismi standard della piattaforma Android, ovvero creando un *Intent* che rappresenti la volontà di effettuare una chiamata telefonica ad un numero specifico; si veda lo stralcio di codice riportato in Listing 5.2.

```
1  Intent dialIntent = new Intent();
2  dialIntent.putExtra("com.google.glass.extra.
    PHONE_NUMBER", phoneNumber);
3  dialIntent.setAction("com.google.glass.action.
    CALL_DIAL");
```

```
4  sendBroadcast(dialIntent);
```

Listing 5.2: Stralcio di implementazione della chiamata telefonica

#### 5.4.4 Localizzazione dell'asset

Nel seguito si presentano i principali aspetti implementativi delle funzionalità di navigazione verso l'asset che necessita di manutenzione e di geolocalizzazione degli asset dell'impresa cliente.

##### Navigazione

La feature di navigazione verso l'asset di non è stata implementata ex-novo: per la sua realizzazione ci si è avvalsi del glassware *Google Maps*, l'applicazione stock di navigazione dei Google Glass. Per attivare questa funzionalità, come prima cosa l'operatore seleziona la voce *Assets* dal menu della splash card dell'applicazione; così facendo l'operatore vede sul display la GUI dell'activity *AssetsOverviewActivity* contenuta nel package *asset.overview* del modulo *app*, che presenta la lista degli impianti del cliente e le informazioni di overview su ciascuno di essi (ulteriori dettagli su questa feature sono contenuti nel paragrafo 5.4.5). A questo punto l'operatore sceglie l'impianto per cui desidera ricevere indicazioni stradali ed infine seleziona la voce *Navigate* dal menu della card. In maniera analoga a quanto accade con la funzionalità di chiamata telefonica hands-free, l'interazione con la piattaforma Android e la conseguente esecuzione del glassware *Google Maps* avviene creando un opportuno Intent che contenga le coordinate GPS della destinazione da raggiungere. Si veda lo stralcio di codice riportato in Listing 5.3.

```
1  GpsCoordinates gpsCoords = asset.getGpsCoords();
2  double lat = gpsCoords.getLatitude();
3  double lon = gpsCoords.getLongitude();
4  Uri destinationUri = Uri.parse("google.navigation:q
    =" + lat + "," + lon);
5  Intent intent = new Intent(Intent.ACTION_VIEW,
    destinationUri);
6  startActivity(intent);
```

Listing 5.3: Stralcio di implementazione dell'avvio della navigazione



Figura 5.19: Screenshot della navigazione verso un asset con Google Maps.

### Geolocalizzazione

La feature di geolocalizzazione degli asset dell'impresa cliente è stata realizzata con l'ausilio del framework di realtà aumentata Metaio. Per visualizzare le icone virtuali che indicano la posizione degli asset rispetto al campo visivo dell'operatore si è impiegata la modalità di tracciamento *sensor-based* supportata dal framework. L'utilizzo di questo metodo di tracciamento consente l'elaborazione dei dati ricevuti dal sensore GPS, dal giroscopio e dalla bussola confrontandoli con le coordinate GPS degli impianti del cliente.

Essendo il display dei Google Glass non immersivo, le funzionalità di realtà aumentata sono state realizzate in modalità video see-through: questo significa che, proprio come accade con un dispositivo handheld, la scena reale non viene osservata direttamente dall'utente, bensì tramite l'inquadratura della videocamera che viene visualizzata sul display del dispositivo. Le icone virtuali generate dunque sono visibili solamente nell'area del display dei Google Glass, sovrapposte ad una miniatura della scena reale che l'utente vede attorno a sé.

L'informazione fornita dalle icone virtuali comprende, oltre alla localizzazione degli asset del cliente, il nome dell'impianto e la potenza istantanea che sta producendo o consumando. Inoltre, ciascuna icona mostrata adotta un codice di colori che esprime lo stato di funzionamento dell'impianto a cui è associata:

- **icona di colore verde:** l'asset sta funzionando correttamente
- **icona di colore arancione:** l'asset si trova in stato di warning

- **icona di colore rosso:** l'asset si trova in stato di allarme.

La funzionalità di geolocalizzazione è implementata dall'activity `GeolocationActivity` contenuta nel package `geolocation` del modulo `glas-library`, e può essere avviata selezionando la voce *Geolocation* dal menu della splash card dell'applicazione. La classe `GeolocationActivity` estende `ARViewActivity`, l'activity di base per interagire con il framework di Metaio. L'implementazione della callback `onCreate()` di `GeolocationActivity` contiene l'istruzione con cui si seleziona la modalità di tracciamento, riportata in Listing 5.4: la stringa "GPS" indica al framework di utilizzare la modalità di tracciamento sensor-based che utilizza i sensori di localizzazione.

```
1 public class GeolocationActivity extends
    ARViewActivity implements ICustomerDataObserver
    {
2     ...
3
4     @Override
5     public void onCreate(Bundle savedInstanceState) {
6         ...
7         boolean result = metaioSDK.
            setTrackingConfiguration("GPS", false);
8         ...
9     }
10    ...
11 }
```

Listing 5.4: Stralcio di implementazione del settaggio della modalità di tracciamento

La callback `loadContents()` ereditata da `ARViewActivity` viene invocata all'avvio dell'activity e si occupa della creazione degli oggetti virtuali e del caricamento dei file ad essi associati. I contenuti virtuali che Metaio è in grado di gestire comprendono immagini, clip video e modelli 3D statici ed animati, e sono rappresentati nel codice applicativo da oggetti Java che implementano l'interfaccia `IGeometry`. All'interno del metodo `loadContents()` dunque si popola una lista di oggetti di tipo `IGeometry` (uno per ogni asset installato presso l'impresa cliente) e si caricano i file immagine delle icone virtuali da visualizzare sul display. Inoltre, il metodo si occupa dell'inizializzazione di un radar che aiuta l'operatore ad individuare gli asset che si trovano al

di fuori del proprio campo visivo. Le istruzioni di disegno degli oggetti virtuali da sovrapporre all'inquadratura della videocamera sono contenute nell'implementazione della callback `onDrawFrame()`, invocata ogniqualvolta il framework richiede il refresh del display. La sua implementazione dunque realizza il calcolo del posizionamento sul display di ciascuna icona, in funzione della posizione dell'operatore e della direzione in cui sta guardando.

Rivolgendo lo sguardo verso l'icona di un impianto e toccando il touchpad degli smart-glass, l'operatore ottiene le informazioni in tempo reale di overview e di dettaglio dell'asset, accedendo alla funzionalità illustrata nel paragrafo 5.4.5.

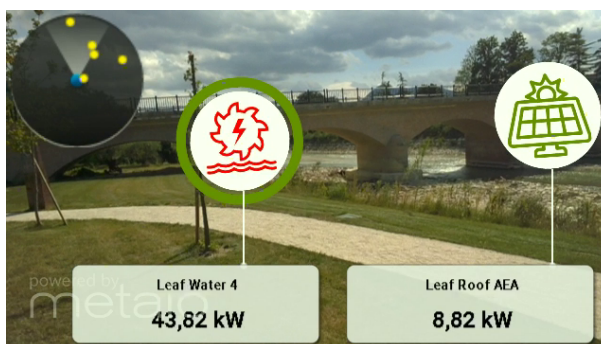


Figura 5.20: Screenshot della geolocalizzazione degli asset.

### 5.4.5 Gestione dell'asset

Per questioni organizzative e di tempo la funzionalità di invio di comandi di controllo e di diagnostica non è stata realizzata in questo lavoro di tesi. L'implementazione di questa feature richiederà l'interazione con il software di basso livello dell'EMS al fine di interfacciarsi con l'unità di controllo dell'asset.

Nel seguito si descrivono i principali aspetti implementativi della visualizzazione di informazioni in tempo reale sugli asset.

#### Informazioni in tempo reale

L'accesso alla feature di visualizzazione in tempo reale di informazioni di overview e di dettaglio sugli impianti avviene selezionando la voce

*Assets* nel menu della splash card dell'applicazione. Le informazioni di overview comprendono le proprietà più rilevanti di un impianto e sono presentate sulla card realizzata dalla classe `AssetsOverviewActivity` contenuta nel package `asset.overview` del modulo `app`. La card di overview presenta lo stato di funzionamento dell'impianto adottando il codice di colori utilizzato dalle icone virtuali impiegate nella funzionalità di geolocalizzazione: qualora l'asset stia funzionando normalmente il suo nome apparirà di colore verde, qualora invece sia in stato di warning o di allarme apparirà rispettivamente di colore arancione o rosso. Accedendo al menu della card di overview di un asset e selezionando la voce *Details* lo smart-glass visualizza le informazioni di dettaglio dell'impianto, mostrando tutte le proprietà di funzionamento disponibili. La card di dettaglio è implementata dalla classe `AssetDetailActivity` contenuta nel package `asset.detail` del modulo `app`.

In Listing 5.5 si riporta uno stralcio della struttura dei dati correnti degli asset, reperibili all'URL specificato dall'attributo `assetsUrl` disponibile tra le informazioni del cliente.

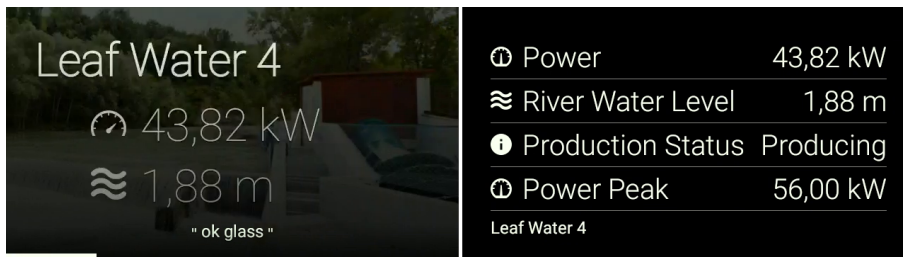


Figura 5.21: Screenshot delle card che mostrano informazioni di overview e di dettaglio.

```

1  [
2    {
3      ...
4      "DeviceStatus": "Warning",
5      "DisplayName": "RCube Battery",
6      "IsResetable": true,
7      "Name": "ESS",
8      "OverviewProperties": [
9        {
10         ...
11         "Name": "State Of Charge",

```



```
12         ...
13         "Value": {
14             "Timestamp": "/Date(1425454076+0100)/",
15             "Type": "Measurement",
16             "Uom": "%",
17             "Value": 80,
18             "ValueStr": "80,00 %"
19         },
20         "Values": null
21     },
22     ...
23 ],
24 "Type": "BaseEnergyStorageSystem"
25 },
26 {
27     ...
28     "DeviceStatus": "Error",
29     "DisplayName": "Leaf Water 1",
30     "IsResetable": false,
31     "Name": "Leaf Water 1",
32     "OverviewProperties": [
33         {
34             ...
35             "Name": "Power",
36             ...
37             "Value": {
38                 "Timestamp": "/Date(1425454587+0100)/",
39                 "Type": "Measurement",
40                 "Uom": "W",
41                 "Value": 43822.4609375,
42                 "ValueStr": "43,82 kW"
43             },
44             "Values": null
45         },
46         {
47             ...
48             "Name": "River Water Level",
49             ...
50             "Value": {
51                 "Timestamp": "/Date(1425454547+0100)/",
52                 "Type": "Measurement",
53                 "Uom": "cm",
54                 "Value": 188,
55                 "ValueStr": "1,88 m"
56             },
```

```
57         "Values": null
58     },
59     ...
60 ],
61     "Type": "HydroElectric"
62 },
63 ...
64 ]
```

Listing 5.5: Stralcio della struttura dei dati correnti degli asset

### 5.4.6 Gestione dell'intervento

Nel seguito si presentano i principali aspetti implementativi delle feature di condivisione video, di consultazione di procedure guidate e di generazione del report di intervento.

#### Condivisione video

I Google Glass della BU Energy sono equipaggiati con la versione KitKat del sistema operativo Android. A partire da questa versione la funzionalità di videochiamata non è più supportata poiché, secondo gli ingegneri Google, la feature non era in grado di soddisfare gli standard di qualità che l'azienda impone ai propri prodotti [50]. L'assenza di un supporto nativo alla funzionalità di videochiamata ha reso necessaria una fase di scouting sulle tecnologie e i framework attualmente disponibili per la realizzazione di applicazioni di streaming e videoconferenza. Al termine di questa esplorazione si è deciso di adottare *Wowza Streaming Engine*<sup>7</sup> come tecnologia di riferimento per la funzionalità di condivisione video dell'applicazione. Wowza Streaming Engine comprende un server di contenuti multimediali altamente configurabile, in grado di abilitare la fruizione di stream multimediali live e on-demand su molteplici tipologie di dispositivo. A differenza di altri servizi di streaming testati durante la fase esplorativa, Wowza Streaming Engine è in grado di mantenere una latenza di trasmissione esigua (minore di 2 secondi), una caratteristica essenziale in contesti applicativi in cui si instaura una comunicazione audiovisiva in tempo reale.

<sup>7</sup><http://www.wowza.com/products/streaming-engine>

In caso di necessità di assistenza remota l'operatore utilizza la funzionalità di chiamata telefonica hands-free per contattare uno dei tecnici assegnati all'impresa cliente; i loro recapiti sono stati recuperati in fase di caricamento dei dati del cliente. Durante la conversazione l'operatore presso il cliente abilita la feature di condivisione video selezionando la voce di menu *Share camera* accessibile sia dal menu della splash card dell'applicazione, sia dal menu della card che presenta le informazioni di overview sugli asset. In questo modo il flusso video è gestito da Wowza Streaming Engine, mentre i due flussi audio sono gestiti dalla rete telefonica; questa combinazione consente di ottenere una trasmissione video con latenza trascurabile e una comunicazione audio di latenza sostanzialmente nulla. La trasmissione video è unidirezionale: l'implementazione attuale non consente all'operatore che fornisce assistenza di trasmettere il video acquisito, ad esempio, dalla webcam del proprio PC. Questa feature risulta attualmente inessenziale, ma potrebbe essere considerata in futuro come possibile miglioramento della funzionalità di collaborazione remota.

La selezione della voce di menu *Share camera* comporta l'avvio dell'activity `ShareCameraActivity` contenuta nel package `videostream` del modulo `app`, responsabile della connessione al server Wowza e della trasmissione del flusso acquisito dalla videocamera tramite il protocollo RTSP (Real Time Streaming Protocol). L'activity contatta il server di streaming recuperando il suo indirizzo IP e i parametri di autenticazione dal file di configurazione `config.properties` mantenuto nella directory `Loccioni` dello storage degli smart-glass. Il server Wowza deve essere installato preventivamente su una macchina dotata di un indirizzo IP raggiungibile da entrambi gli operatori che partecipano alla collaborazione remota; il suo compito è accettare il flusso video trasmesso da `ShareCameraActivity` e pubblicarlo su un URL del protocollo RTSP. L'operatore che fornisce assistenza può vedere in tempo reale la visuale del tecnico presso il cliente accedendo a tale URL con un qualunque software di riproduzione multimediale, come ad esempio VLC Media Player<sup>8</sup>.

---

<sup>8</sup><http://www.videolan.org/vlc/>



Figura 5.22: Screenshot della card di streaming video.

### Procedure di manutenzione guidate

La consultazione di procedure guidate è accessibile selezionando la voce *Assets* dal menu della splash card, scegliendo l'asset di interesse e selezionando la voce *Guides* dal menu della card di overview dell'impianto. L'elenco delle procedure guidate disponibili per la tipologia dell'asset selezionato dovrebbe essere recuperato contattando il servizio web di cui si è parlato in fase di definizione dell'architettura di questa funzionalità (paragrafo 5.3.4). In questo lavoro di tesi tale servizio non è stato implementato, dunque l'elenco delle procedure guidate disponibili per ciascuna tipologia di impianto è caricata da un file JSON che simula la risposta del web service mancante; tali file sono contenuti nella directory `Loccioni\Service\Guides\Asset types` dello storage degli smart-glass. L'insieme delle procedure disponibili per la tipologia dell'asset selezionata è presentata su una lista di card implementata dalla classe `GuideTitlesActivity` contenuta nel package `guide` del modulo `app`. Dopo che l'operatore ha selezionato la procedura guidata che intende avviare, l'applicazione carica la sequenza di step che la compongono; in assenza del servizio web di cui sopra, attualmente questo processo avviene interpretando un file JSON avente il nome della procedura, contenuto nella directory `Loccioni\Service\Guides` dello storage dei Google Glass. Un esempio della struttura di una procedura guidata è riportata in Listing 5.6.

```
1 {  
2   "id": "proci",  
3   "title": "Sostituzione batteria",
```

```
4
5  "steps": [
6    {
7      "id": "step1",
8      "title": "Premere il pulsante di reset",
9      "type": "AR",
10     "source1": "http://pcddeterminant.altervista.
11             org/yellow_button.png",
12     "source2": "http://pcddeterminant.altervista.
13             org/push.png"
14   },
15   {
16     "id": "step2",
17     "title": "Scollegare il cavo B",
18     "type": "ImageAndAudio",
19     "source1": "http://pcddeterminant.altervista.
20             org/maint_image.jpg",
21     "source2": "http://pcddeterminant.altervista.
22             org/voce.mp3"
23   },
24   {
25     "id": "step3",
26     "title": "Sostituire la batteria",
27     "type": "Video",
28     "source1": "http://pcddeterminant.altervista.
29             org/maint_video2.mp4",
30     "source2": ""
31   },
32   {
33     "id": "step4",
34     "title": "Ricollegare il cavo B",
35     "type": "Text",
36     "source1": "Ricollegare il cavo B",
37     "source2": ""
38   }
39 ]
40 }
```

Listing 5.6: Esempio della struttura di una procedura guidata

Gli step della procedura selezionata sono presentati sul display degli smart-glass secondo l'implementazione delle classi `GuideStepActivity` e `GuideStepAdapter` contenute nel package `guide.step` del modulo `app`. Ogni step di una procedura può utilizzare differenti contenuti mul-

timediali per istruire l'operatore sulla prossima azione da compiere sull'asset:

- **testo:** fornisce una descrizione testuale dell'azione da intraprendere nello step corrente
- **immagine:** mostra un'immagine che, ad esempio, evidenzia il componente dell'asset con cui è necessario interagire
- **audio:** riproduce una clip audio che descriva le azioni da compiere in linguaggio naturale
- **immagine con audio:** mostra un'immagine sul display e riproduce una clip audio
- **video:** riproduce un breve video tutorial
- **indicazioni in realtà aumentata:** sul display degli smart-glass compaiono delle indicazioni virtuali sovrapposte all'impianto che guidano l'operatore nello step corrente.

Di particolare interesse è l'utilizzo di tecniche di realtà aumentata per supportare l'operatore durante l'intervento di manutenzione: questa funzionalità è stata realizzata con l'ausilio di Metaio. Così come è avvenuto per la feature di geolocalizzazione degli asset, date le caratteristiche del display dei Google Glass anche in questo caso l'esperienza di realtà aumentata avviene in modalità video see-through. Durante l'esecuzione di uno step in realtà aumentata l'obiettivo del framework è riconoscere la porzione dell'asset inquadrata dagli smart-glass e sovrapporvi un'indicazione virtuale. Il pattern da riconoscere è dunque una foto dell'impianto inquadrato dal punto di vista dell'operatore: la modalità di tracciamento più adatta in questa circostanza è *2D template*, appartenente alla categoria *planar* di Metaio (si veda Figura 5.15 a pagina 101). A differenza della modalità sensor-based impiegata per la geolocalizzazione degli asset, la modalità 2D template non prevede una configurazione predefinita per il tracciamento, dunque è necessario preparare un file di testo che contenga i parametri da utilizzare per il processo di tracciamento, tra cui i nomi dei file relativi all'immagine del pattern da riconoscere e all'indicazione virtuale da sovrapporvi. Durante la consultazione di una procedura guidata,

quando l'utente giunge ad uno step in realtà aumentata l'applicazione avvia `ARGuideStepActivity`, classe che estende l'activity di base di Metaio `ARViewActivity`. L'implementazione dell'activity prevede il recupero dei file relativi all'immagine pattern e all'indicazione virtuale, così come la preparazione del file di testo atto a contenere i parametri di configurazione del tracciamento.



Figura 5.23: Screenshot di uno step in realtà aumentata.

Durante l'esecuzione di uno step qualunque l'operatore può accedere al menu di `GuideStepActivity` usando il touchpad degli smart-glass oppure pronunciando "Ok Glass" seguito da uno dei seguenti comandi:

- **Completed:** indica che lo step corrente ha avuto esito positivo e avanza allo step successivo della procedura
- **Previous:** torna allo step precedente della procedura
- **Replay:** riproduce nuovamente le istruzioni dello step corrente
- **Voice note:** assegna una nota vocale allo step corrente
- **Failed:** indica che lo step corrente ha avuto esito negativo, assegna una nota vocale che descriva la causa del fallimento e avanza allo step successivo della procedura
- **Abort:** termina la procedura

L'applicazione *Loccioni Augmented Service* agevola la creazione di nuove procedure guidate consentendo l'acquisizione di video durante

l'esecuzione di un intervento di manutenzione. L'operatore può accedere a questa funzione scorrendo tutta la lista delle procedure guidate esistenti per l'asset selezionato e scegliendo la card *Create a new video guide*. L'implementazione della classe `NewVideoGuideActivity` contenuta nel package `guide` del modulo `app` si occupa della registrazione di una clip video utilizzando la videocamera degli smart-glass; il file della clip acquisita è rinominato opportunamente e salvato sullo storage degli smart-glass all'interno della directory `DCIM`. Il video ottenuto può in seguito essere scaricato su PC, editato ed utilizzato in uno o più step della nuova procedura.

### Report di intervento

Al termine di una procedura guidata l'applicazione genera un report dell'intervento di manutenzione che comprende l'esito di ciascuno step della procedura e la trascrizione delle note vocali. L'activity `ReportActivity` contenuta nel package `report` del modulo `app` mostra il report sul display degli smart-glass e lo salva su un file di testo archiviato nella directory `Loccioni\Service\Maintenance reports` dello storage degli smart-glass. È inoltre possibile inviare il report tramite e-mail ad uno o più destinatari scelti dalla rubrica integrata dell'applicazione. La composizione e la consegna dell'e-mail contenente il report è stata realizzata con l'ausilio di *SendGrid*<sup>9</sup>, un servizio cloud per la gestione e il recapito automatico di e-mail e newsletter. L'invio dell'e-mail del report tramite `SendGrid` è implementata dalla classe `SendEmailTask` contenuta nel package `report.email` del modulo `app`. In Figura 5.24 è riportato un esempio di report di intervento ricevuto tramite e-mail.

---

<sup>9</sup><https://sendgrid.com/>



**LOCCIONI**

**Report: Ciclo di pulizia impianto idroelettrico**

Data: 2015-06-26

Cliente: Loccioni

Step	Descrizione	Esito	Note
1	Premere il pulsante di reset	Ok	-
2	Premere il pulsante START CICLO PULIZIA	Ok	-
3	Verificare che la paratoia di scarico si apra e che la sgrigliatrice si liberi dai detriti	Fallito	The sluice can't open

Report generato da Loccioni Service Glassware

<http://www.loccioni.com/>

Figura 5.24: Un esempio di e-mail contenente il report di intervento.



# Capitolo 6

## Collaudo del sistema

In questo capitolo si descrivono le prove sul campo effettuate per il collaudo delle funzionalità del sistema software sviluppato, evidenziando i risultati ottenuti e le problematiche che non sono state risolte nel contesto di questo lavoro di tesi. Infine, si presentano alcune valutazioni finali sul sistema Loccioni Augmented Service.

### 6.1 Collaudo delle funzionalità

Le funzionalità implementate dal glassware sono state collaudate su alcuni impianti installati presso la sede del Gruppo Loccioni, simulando il processo di manutenzione che normalmente avviene presso le sedi delle imprese clienti. Gli asset presi in considerazione durante la fase di collaudo sono i seguenti:

- *RCube Battery*: impianto di storage per l'accumulo di energia prodotta da fonti rinnovabili
- *Leaf Water 1*: impianto per la produzione di energia idroelettrica installata su un canale artificiale del fiume Esino
- *Leaf Roof AEA*: impianto fotovoltaico installato sul tetto dell'edificio *AEA*
- *Leaf Roof Summa*: impianto fotovoltaico installato sul tetto dell'edificio *Summa*

- *Leaf Roof Lab*: impianto fotovoltaico installato sul tetto dell'edificio *Leaf Lab*
- *Spray Momentum*: asset automotive per il controllo del processo di iniezione di carburante all'interno di motori Diesel per automobili.

La BU Energy del Gruppo Loccioni gestisce gli impianti fotovoltaici, elettrici e di storage di cui sopra, mentre lo *Spray Momentum* è gestito dalla BU Mobility. Presso la sede del Gruppo sono installati numerosi altri asset che non sono stati considerati ai fini del collaudo dell'applicazione Loccioni Augmented Service. Al fine di illustrare più efficacemente i test eseguiti durante la fase di collaudo del sistema, in Figura 6.1 si riporta la dislocazione degli asset appena presentati.

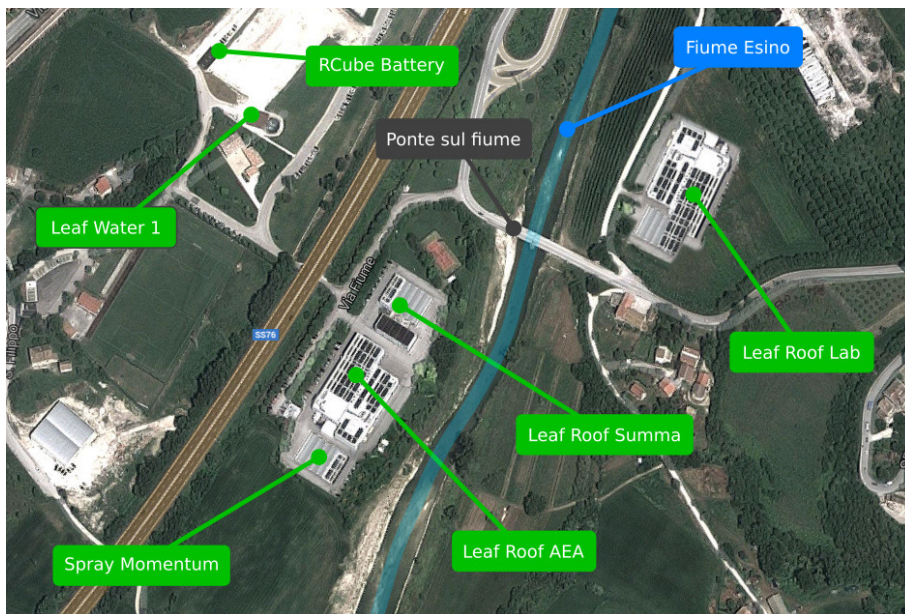


Figura 6.1: Vista satellitare della sede Loccioni. Gli asset considerati durante le fasi di collaudo sono indicati in colore verde.

L'impresa cliente denominata *Loccioni* a cui si è accennato nel paragrafo 5.4.3 è stata utilizzata per il collaudo delle funzionalità dell'applicazione, e in futuro potrà essere selezionata sul glassware qualora sia necessario mantenere gli asset della sede del Gruppo. Le

informazioni associate a tale cliente comprendono gli impianti appena presentati e le informazioni di contatto di alcuni operatori tecnici di riferimento.

La fase di collaudo del sistema software ha compreso le seguenti prove sul campo:

- Ottenimento di indicazioni stradali verso l'asset Leaf Water 1
- Geolocalizzazione degli impianti sul ponte del fiume Esino
- Collaborazione remota tramite la trasmissione di uno stream video su rete locale
- Creazione e consultazione di procedure di manutenzione guidate per gli impianti Leaf Water 1 e Spray Momentum

### 6.1.1 Navigazione verso Leaf Water 1

La feature di navigazione è stata collaudata richiedendo al glassware le indicazioni stradali per raggiungere l'impianto Leaf Water 1 partendo dall'edificio Leaf Lab. Quando l'applicazione invoca Google Maps il percorso stradale viene calcolato correttamente, ma talvolta lo smart-glass non riesce ad agganciare il segnale GPS. Più precisamente, all'avvio di Google Maps l'applicazione localizza correttamente la posizione dell'utente, ma quando quest'ultimo inizia a dirigersi verso la sua destinazione sembra che il glassware non aggiorni la sua posizione, continuando a mostrare l'utente nel punto di partenza. L'area della sede Loccioni risulta coperta da segnale GPS, dunque è probabile che tale problematica sia causata da un bug del software di comunicazione con lo smartphone accoppiato, che in alcune circostanze impedisce allo smart-glass di reperire correttamente i dati dal sensore GPS del dispositivo handheld.

### 6.1.2 Geolocalizzazione sul ponte del fiume

La feature di geolocalizzazione degli asset è stata collaudata in corrispondenza del ponte sul fiume Esino: tale posizione permette di vedere gli impianti Leaf Roof Lab, Leaf Roof Summa e Leaf Roof AEA, consentendo dunque di valutare l'accuratezza della posizione delle icone

virtuali visualizzate sul display dello smart-glass. Durante le prove effettuate si è sperimentato qualche problema di allineamento delle icone rispetto alla posizione effettiva degli asset: la causa di questa problematica è da imputare probabilmente alla precisione del sensore giroscopico e/o della bussola dello smart-glass. Infatti, eseguendo il porting della feature di geolocalizzazione su smartphone Android, il posizionamento delle icone virtuali è risultato più preciso ed affidabile. La ricalibrazione manuale dei sensori dello smart-glass sembra non essere sufficiente a risolvere questo comportamento inatteso.

### 6.1.3 Streaming video su rete locale

Il collaudo della funzionalità di condivisione video è avvenuto su rete locale, assegnando allo smart-glass e al server Wowza Streaming Engine due indirizzi IP appartenenti alla stessa rete privata. I risultati ottenuti con questa configurazione sono positivi: la latenza del feed video ricevuto è minore di 2 secondi. In uno scenario reale lo stream dovrà attraversare la rete pubblica per poter essere fruito dall'operatore remoto, dunque le prestazioni di questa feature dipenderanno dalla banda disponibile e dal traffico di rete. Ad ogni modo, solamente la componente video della comunicazione risulterebbe affetta da questi fattori, poiché l'audio transiterebbe comunque sulla rete cellulare.

### 6.1.4 Procedure guidate su Leaf Water 1 e Spray Momentum

La consultazione di procedure di manutenzione guidate è stata collaudata sull'impianto idroelettrico Leaf Water 1 e sul macchinario Spray Momentum. Per quanto riguarda la Leaf Water 1, si è creata una procedura per eseguire il ciclo di pulizia dell'asset. Durante il normale funzionamento dell'impianto, il flusso d'acqua in ingresso, proveniente da un canale artificiale del fiume Esino, passa attraverso un filtro che impedisce ad eventuali corpi estranei –come rami di alberi e altri detriti trasportati dal fiume– di entrare nell'impianto e raggiungere le turbine. Tale filtro è denominato *sgrigliatrice* e deve essere periodicamente ripulito affinché rilasci i detriti accumulati nel flusso d'acqua



Figura 6.2: L'impianto idroelettrico Leaf Water 1.

di scarico dell'impianto. La procedura di test creata è composta dai seguenti passi:

- come prima cosa, un'indicazione in realtà aumentata suggerisce all'operatore di premere il pulsante di reset dell'impianto
- in seguito, un breve video indica la posizione del pulsante da premere per avviare il ciclo di pulizia
- infine, un breve video raccomanda all'operatore di verificare che la paratoia di scarico dell'impianto si apra e che la sgrigliatrice si liberi dai detriti.

Per quanto riguarda invece l'impianto Spray Momentum, si è creata una procedura guidata che consente di riportare gli assi di rotazione del macchinario in posizione iniziale. L'asset è infatti composto di tre parti meccaniche principali –denominati *assi*– che possono ruotare indipendentemente le une dalle altre; l'esecuzione di determinate attività sul macchinario richiede che i suoi assi si trovino inizialmente in una posizione nota, e la procedura creata serve proprio a questo scopo. Gli step che compongono la procedura riproducono una serie di clip video che indicano all'operatore le operazioni da compiere sull'interfaccia software offerta dallo Spray Momentum.

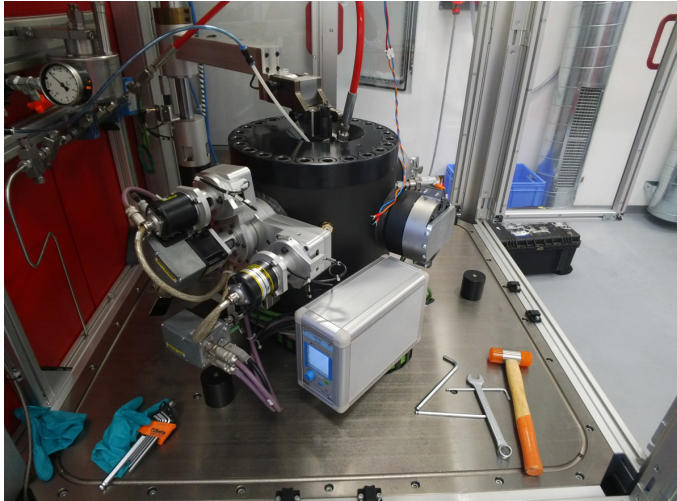


Figura 6.3: L'impianto automotive Spray Momentum.

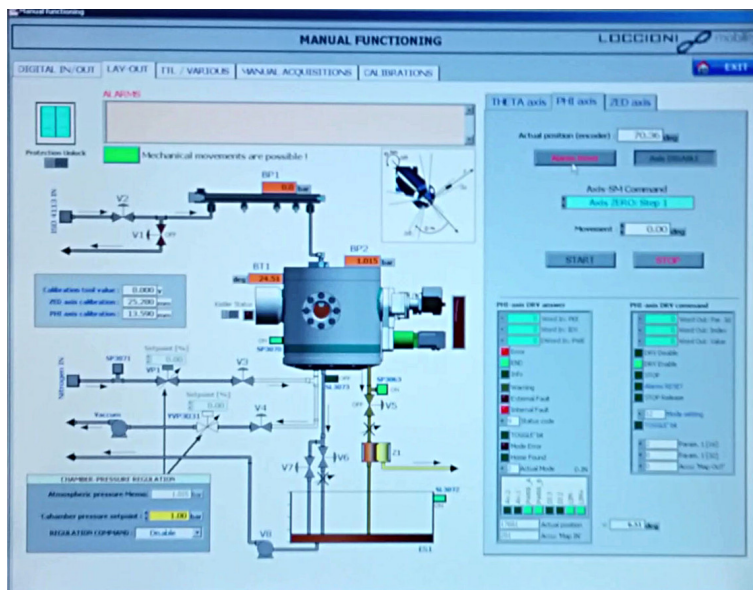


Figura 6.4: L'interfaccia software dello Spray Momentum.



Il collaudo di procedure che istruiscono l'operatore per mezzo di testo, immagini, clip audio e video ha prodotto risultati positivi; l'esecuzione di step che forniscono indicazioni in realtà aumentata ha invece evidenziato alcune problematiche legate al riconoscimento della parte di impianto su cui sta lavorando l'operatore. Le condizioni luminose dell'ambiente infatti influiscono notevolmente sull'esito del riconoscimento, così come la distanza dal target inquadrato: per riconoscere con successo la parte dell'impianto di interesse è talvolta necessario portare la videocamera degli smart-glass ad una distanza molto ravvicinata da essa (circa 10 cm). Una possibile soluzione a queste problematiche potrebbe essere l'utilizzo della modalità di tracciamento di Metaio basata su *ID marker* (si veda Figura 5.15 a pagina 101). Dopo alcuni test è infatti emerso che l'algoritmo di riconoscimento ID marker è più robusto rispetto a quello della modalità *2D template* attualmente usato: risulta meno affetto dalle condizioni ambientali ed è in grado di riconoscere l'impianto con successo da una distanza maggiore. Tuttavia, mentre la modalità *2D template* utilizza come target da riconoscere una foto della parte di asset di interesse, la modalità *ID marker* impiega un fiducial che sarebbe necessario stampare e applicare opportunamente su ogni parte dell'impianto che si desidera riconoscere (si veda Figura 6.5): una soluzione di questo tipo risulterebbe dunque poco scalabile e difficilmente praticabile, considerando inoltre l'elevato numero di asset mantenuti dal team Service.



Figura 6.5: Esempi di immagini target utilizzate nelle modalità *2D template* (a sinistra) e *ID marker* (a destra) di Metaio.

### 6.1.5 Collaudo delle altre funzionalità

Le funzionalità del glassware non citate finora –la selezione del cliente, le telefonate hands-free, la visualizzazione di informazioni sugli asset in tempo reale e la trasmissione tramite e-mail del report di intervento– non hanno richiesto prove sul campo ad hoc, e sono state collaudate durante le fasi di sviluppo e debug dell'applicazione Loccioni Augmented Service.

In particolare, la feature della telefonata hands-free sembra funzionare correttamente; una limitazione tecnologica riscontrata riguarda il trasduttore a conduzione ossea: sebbene risulti talvolta inefficace in ambienti rumorosi, è comunque possibile utilizzare l'auricolare in dotazione con lo smart-glass per ottenere un volume più alto durante la conversazione.

La funzionalità di visualizzazione di informazioni sugli asset in tempo reale non ha presentato particolari problematiche: i dati visualizzati sullo smart-glass sono consistenti con quelli pubblicati dalle web API REST dell'EMS. Questa feature funziona correttamente con impianti idroelettrici, fotovoltaici e di storage. Altri tipi di asset –come ad esempio lo Spray Momentum della BU Mobility– non espongono proprietà e informazioni in tempo reale, dunque non consentono di accedere a tale funzionalità.

Infine, il collaudo delle funzionalità di salvataggio del report di intervento sullo storage dello smart-glass e della sua trasmissione tramite e-mail ha avuto esito positivo e non ha evidenziato alcuna criticità.

## 6.2 Valutazione del sistema

Considerando gli obiettivi raggiunti e le problematiche ad oggi irrisolte, si ritiene che i risultati conseguiti dalla realizzazione dell'applicazione Loccioni Augmented Service siano soddisfacenti. Come era lecito immaginarsi, le principali criticità su cui ci si è imbattuti durante le fasi di sviluppo e collaudo del glassware riguardano l'integrazione di tecnologie di realtà aumentata sullo smart-glass. In particolare, le problematiche che affliggono le procedure di manutenzione in realtà aumentata evidenziano il fatto che, attualmente, uno degli ostacoli alla diffusione di applicazioni AR su sistemi basati su smart-glass è

rappresentato dalla difficoltà di progettare e realizzare algoritmi di tracciamento efficienti, efficaci e che al contempo richiedano una potenza di calcolo compatibile con quella disponibile su questa tipologia di dispositivi.

La soluzione hands-free realizzata ovviamente non dispone di un livello di maturità tale da poter essere impiegata come strumento standard del workflow di manutenzione del team Service; qualunque dispositivo, attrezzatura o software deve infatti soddisfare determinati requisiti di qualità ed affidabilità per poter essere impiegato in un contesto lavorativo reale. Tuttavia, alcune funzionalità del sistema sviluppato sono attualmente usabili e potrebbero quindi essere impiegate durante lo svolgimento di alcune attività del processo di manutenzione.

La progettazione e la realizzazione del sistema Loccioni Augmented Service ha inoltre consentito di valutare le potenzialità offerte dalla sinergia tra sistemi basati su smart-glass e tecnologie di realtà aumentata, così come di analizzare alcune criticità e complicazioni che possono emergere durante lo sviluppo di questi sistemi.



# Capitolo 7

## Conclusioni

La diffusione nel mercato di massa e in ambito enterprise di sistemi MAR basati su smart-glass è attualmente ostacolata da alcune problematiche di varia natura: ad oggi questi sistemi non sono ancora socialmente accettati da una parte dell'utenza e non hanno ancora raggiunto un livello di maturità tecnologica tale da poter apportare benefici in molteplici ambiti applicativi. Ad ogni modo, il processo di costruzione di dispositivi mobili e lo sviluppo di tecnologie di realtà aumentata negli ultimi anni stanno sperimentando una repentina evoluzione. Inoltre, alcune previsioni (riportate nel paragrafo 3.2.1) indicano che entro la fine del decennio si potrebbe assistere ad un considerevole incremento del fatturato del mercato dei dispositivi di realtà aumentata e del relativo numero di unità vendute.

Il successo e la diffusione di questi sistemi dipenderà da una moltitudine di fattori, tra cui l'utilità e la qualità delle applicazioni software disponibili, l'esperienza percepita dagli utenti e l'efficacia delle interfacce grafiche e delle modalità di interazione supportate. Risulta dunque di fondamentale importanza approfondire queste tematiche, elaborando nuove GUI e UI progettate appositamente per questi dispositivi e proponendo metodologie di sviluppo software che siano in grado di catturare e rappresentare opportunamente le caratteristiche peculiari di questi sistemi.

Come si è detto, l'obiettivo di questo lavoro di tesi è stato lo studio e l'elaborazione di una soluzione hands-free basata su smart-glass per supportare le attività di customer care del team Service del Gruppo

Loccioni. Il percorso seguito durante la preparazione di questo lavoro ha previsto i seguenti passi:

- Inizialmente si è provveduto a raccogliere e studiare materiale inerente ad Augmented Reality, Mobile Augmented Reality e sistemi hands-free basati su smart-glass.
- In seguito si è ricercato materiale su casi di studio in cui i sistemi MAR siano stati impiegati con successo in molteplici ambiti applicativi, con un focus particolare sul settore industriale.
- Dopo aver approfondito il contesto in cui il lavoro di tesi si sarebbe inserito, si è analizzata la problematica da risolvere interagendo con alcuni membri del team Service al fine di elaborare i requisiti e l'architettura dell'applicazione Loccioni Augmented Service.
- Successivamente si è presa confidenza con il modello di programmazione dei Google Glass, studiando la documentazione inerente al Glass Development Kit, ai pattern di sviluppo di riferimento e alle best practice da seguire.
- Infine si è proceduto con l'ingegnerizzazione del glassware Loccioni Augmented Service, l'implementazione delle sue funzionalità e il collaudo del sistema realizzato.

La soluzione hands-free progettata è attualmente in fase prototipale e in futuro potrebbe essere estesa con nuove funzionalità e migliorie. Si potrebbero ad esempio sviluppare le feature e i componenti software che non sono state implementate in questo lavoro di tesi: *(i)* la funzionalità di invio di comandi di controllo e di diagnostica agli asset, *(ii)* un servizio web che fornisca le coordinate GPS degli asset e *(iii)* un servizio web che gestisca l'elenco e i contenuti delle procedure guidate inerenti a ciascuna tipologia di impianto.

Inoltre si potrebbero risolvere le questioni emerse durante le fasi di collaudo del sistema: la problematica di maggior rilevanza riguarda il processo di riconoscimento e tracciamento degli asset su cui l'operatore esegue le procedure di manutenzione guidate in realtà aumentata. Per risolvere questa ed altre limitazioni si potrebbe ad esempio adottare un framework di realtà aumentata differente da Metaio, oppure

si potrebbe effettuare un porting dell'applicazione su uno smart-glass differente dai Google Glass; in questi casi una parte del codice applicativo andrebbe sicuramente riscritta, ma l'architettura del sistema proposta nel paragrafo 5.3 rimarrebbe comunque valida. Attualmente, lo smart-glass Epson Moverio BT-200 (presentato nel paragrafo 3.5.2) potrebbe rappresentare il candidato che meglio concilia funzionalità, prestazioni e prezzo.

Ulteriori sviluppi futuri potrebbero inoltre indagare la possibilità di rivedere l'intera architettura del sistema hands-free, delegando alcune funzioni ad uno smartphone che l'operatore porta con sé. L'intento sarebbe quello di realizzare un sistema distribuito in cui il core dell'applicazione e i task computazionalmente onerosi sono in esecuzione sullo smartphone, mentre la GUI dell'applicazione viene presentata sul display dello smart-glass. In questo modo sarebbe possibile avvalersi della potenza computazionale dello smartphone –tipicamente maggiore di quella di uno smart-glass– beneficiando al contempo dell'interfaccia hands-free, della UI e delle funzionalità di realtà aumentata offerte dallo smart-glass.





# Bibliografia

- [1] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.
- [2] Ronald T Azuma et al. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.
- [3] Ronald Azuma, Yohan Baillet, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47, 2001.
- [4] Tobias Höllerer and Steve Feiner. Mobile augmented reality. *Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK*, 21, 2004.
- [5] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.
- [6] Thomas A Furness. The super cockpit and its human factors challenges. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 30, pages 48–52. SAGE Publications, 1986.
- [7] Steve Mann. Wearable computing: A first step toward personal imaging. *Computer*, 30(2), 1997.
- [8] Thomas P Caudell and David W Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *System Sciences, 1992. Proceedings of*

- the Twenty-Fifth Hawaii International Conference on*, volume 2, pages 659–669. IEEE, 1992.
- [9] Steven Feiner, Blair MacIntyre, Tobias Höllerer, and Anthony Webster. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. *Personal Technologies*, 1(4):208–217, 1997.
- [10] The surprising reason oil companies love google glass. <http://www.fastcolabs.com/3031171/the-surprising-reason-oil-companies-love-google-glass>.
- [11] Wearable systems for industrial automation. <http://www.beckhoff.com/english.asp?highlights/google-glass/default.htm>.
- [12] Easyjet is testing the use of augmented reality to minimise passenger delays. <http://www.hapakenya.com/easyjet-is-testing-the-use-of-augmented-reality-to-minimise-passenger-delays/>.
- [13] Zhanpeng Huang, Pan Hui, Christoph Peylo, and Dimitris Chantzopoulos. Mobile augmented reality survey: A bottom-up approach. *arXiv preprint arXiv:1309.4413*, 2013.
- [14] DWF Van Krevelen and R Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 9(2):1, 2010.
- [15] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202. IEEE Computer Society, 2008.
- [16] Artoolkit. <http://www.hitl.washington.edu/artoolkit/>.
- [17] Studierstube. <http://studierstube.icg.tugraz.at/main.php>.
- [18] Dwarf. <http://far.in.tum.de/Chair/ProjectDwarf>.

- [19] D'fusion. <http://www.t-immersion.com/products/dfusion-suite>.
- [20] Wikitude. <https://www.wikitude.com/>.
- [21] Vuforia. <https://www.qualcomm.com/products/vuforia>.
- [22] Metaio. <https://www.metaio.com/>.
- [23] Layar. <https://www.layar.com/>.
- [24] Blippar. <https://blippar.com/>.
- [25] Aurasma. <http://www.aurasma.com/>.
- [26] Junaio. <http://www.junaio.com/>.
- [27] Sap launches two enterprise applications supporting vuzix' m100 smart glasses. <http://www.prnewswire.com/news-releases/sap-launches-two-enterprise-applications-supporting-vuzix-m100-smart-glasses-279541752.html>.
- [28] Smart glasses market report 2015. <http://www.augmentedreality.org/#!/smartglassesreport/c88h>.
- [29] Smart glasses market report 2015: towards 1 billion shipments. <http://www.slideshare.net/comogard/smart-glasses-2015-for-distribution>.
- [30] Augmented/virtual reality to hit \$150 billion disrupting mobile by 2020. <http://www.digi-capital.com/news/2015/04/augmentedvirtual-reality-to-hit-150-billion-disrupting-mobile-by-2020/#.VaYwLvnt1Bc>.
- [31] Nassir Navab, A Bani-Kashemi, and Matthias Mitschke. Merging visible and invisible: Two camera-augmented mobile c-arm (camc) applications. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 134–141. IEEE, 1999.
- [32] Google glass returns to help kids with autism. <http://www.imedicalapps.com/2015/07/google-glass-autism/>.

- [33] A new google glass app uses augmented reality, and dance, to help parkinson's sufferers. <http://www.fastcocrete.com/3048038/a-new-google-glass-app-uses-augmented-reality-and-dance-to-help-parkinsons-sufferers>.
- [34] Firenze: primo prelievo di fegato coi google glass. <http://www.webnews.it/2014/11/24/firenze-prelievo-fegato-google-glass/>.
- [35] Chris Argenta, Anne Murphy, Jeremy Hinton, James Cook, Todd Sherrill, and Steve Snarski. Graphical user interface concepts for tactical augmented reality. In *SPIE Defense, Security, and Sensing*, pages 76880I–76880I. International Society for Optics and Photonics, 2010.
- [36] David C Roberts, Stephen Snarski, Todd Sherrill, Alberico Menozzi, Brian Clipp, and Patrick Russler. Soldier-worn augmented reality system for tactical icon visualization. In *SPIE Defense, Security, and Sensing*, pages 838305–838305. International Society for Optics and Photonics, 2012.
- [37] Simon Julier Yohan, Simon Julier, Yohan Baillot, Marco Lanzagorta, Dennis Brown, and Lawrence Rosenblum. Bars: Battlefield augmented reality system. In *In NATO Symposium on Information Processing Techniques for Military Systems*. Citeseer, 2000.
- [38] Mini augmented vision: a revolutionary display concept offering enhanced comfort and safety. <http://www.miniusanews.com/newsrelease.do?id=770&mid=1>.
- [39] Patrick Dähne and John N Karigiannis. Archeoguide: System architecture of a mobile outdoor augmented reality system. In *null*, page 263. IEEE, 2002.
- [40] Gudrun Klinker, Oliver Creighton, Allen H Dutoit, Rafael Kobylinski, Christoph Vilsmeier, and B Brugge. Augmented maintenance of powerplants: A prototyping case study of a mobile ar system. In *Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on*, pages 124–133. IEEE, 2001.

- [41] Stuart Goose, S Guven, Xiang Zhang, Sandra Sudarsky, and Nassir Navab. Paris: fusing vision-based location tracking with standards-based 3d visualization and speech interaction on a pda. In *Proc. IEEE DMS 2004 (International Conference on Distributed Multimedia Systems)*, pages 75–80. Citeseer, 2004.
- [42] Ubimax enters into strategic cooperation with vuzix - focus on industrial wearable computing applications for smart glasses. <http://www.ubimax.de/index.php/en/news/press-releases/item/139-ubimax-enters-into-strategic-cooperation-with-vuzix-focus-on-industrial-wearable-computing-applications-for-smart-glasses&Itemid=329>.
- [43] Vuzix and evolar introduce smartpick for the enterprise and industrial markets. <http://www.prnewswire.com/news-releases/vuzix-and-evolar-introduce-smartpick-for-the-enterprise-and-industrial-markets-255473861.html>.
- [44] Mark Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.
- [45] Alessandro Ricci, Luca Tummolini, Michele Piunti, Olivier Boisier, and Cristiano Castelfranchi. Mirror worlds as agent societies situated in mixed reality environments. In *13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014): The 17th International Workshop on Coordination, Organisations, Institutions and Norms*, pages AAMAS2014–W22, 2014.
- [46] Cristiano Castelfranchi, Michele Piunti, Alessandro Ricci, and Luca Tummolini. Ami systems as agent-based mirror worlds: Bridging humans and agents through stigmergy., 2012.
- [47] Lost lake cafe, seattle restaurant, kicks out patron for wearing google glass. [http://www.huffingtonpost.com/2013/11/27/lost-lake-cafe-google-glass\\_n\\_4350039.html](http://www.huffingtonpost.com/2013/11/27/lost-lake-cafe-google-glass_n_4350039.html).
- [48] Google quietly distributes new version of glass aimed at workplaces - wsj. [http://www.wsj.com/article\\_email/google-](http://www.wsj.com/article_email/google-)

quietly-distributes-new-version-of-glass-aimed-at-workplaces-1438283319-1MyQjAxMTI1MDMzMDczNjA3Wj.

- [49] Thad Starner. Project glass: An extension of the self. *Pervasive Computing, IEEE*, 12(2):14–16, 2013.
- [50] Google’s kit kat update for glass is coming later this week, but video calling is out for now. <http://thenextweb.com/google/2014/04/15/google-kitkat-update-glass-coming-later-week-video-calling-now/>.