

Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN
MECCANICA E SCIENZE AVANZATE DELL'INGEGNERIA

Ciclo 33

Settore Concorsuale: 09/A3 - PROGETTAZIONE INDUSTRIALE, COSTRUZIONI MECCANICHE
E METALLURGIA

Settore Scientifico Disciplinare: ING-IND/15 - DISEGNO E METODI DELL'INGEGNERIA
INDUSTRIALE

INTERFACCE DI REALTA' AUMENTATA PER APPLICAZIONI
EFFICIENTI E SICURE NELL'INDUSTRIA E NEI SERVIZI

Presentata da: Sergio Piastra

Coordinatore Dottorato

Marco Carricato

Supervisore

Francesca De Crescenzo

Co-supervisore

Gianni Caligiana

Esame finale anno 2021

Prefazione

Le prime righe di questa tesi hanno lo scopo di far comprendere al lettore da dove trae origine il presente elaborato e dove esso mi porterà in futuro. Esse descrivono di seguito come, mattone dopo mattone, la “struttura” del mio essere ingegnere sia stata costruita con scelte razionali, a volte anche inconsce, ma mai dettate dal caso e come ciò rappresenti una base stabile che mi dà fiducia nel continuare la costruzione in altezza.

A voler dire il vero, le prime righe di questa tesi nascono qualche ora fa, durante una corsa estiva per i campi assolati di Forlì, lungo le rive del fiume Montone, paesaggi naturali dove la storia si perde nella notte dei tempi. Metro dopo metro, i rumori della città scompaiono, sento solo il mio respiro, il caldo ed il canto delle cicale, il tutto ad un ritmo ripetitivo che scandisce ogni mio passo, con una cadenza ipnotica che offusca la mente e la libera da ogni pensiero. Il mio movimento ed il paesaggio attorno si fondono in un tutt'uno, percepisco lo scorrere della storia lungo le antiche rive che nascondono, tra giunchi e canneti, *“vetuste mura, chiese e fortezze, come dei cavalieri epici risvegliati, e, innumerevoli, emersi dalla terra ...”*. Il pensiero allora va ad un altro fiume, al russo Volga, a cui i precedenti versi del famoso cantautore ed attore teatrale russo Vladimir Vysotsky si riferiscono (*“Kak po Volge-matushke”*). Il parallelo non è casuale. Passo dopo passo il sentiero si fa sempre più stretto, inizio ad addentrarmi nella fitta vegetazione del lungofiume che nasconde il cammino da seguire, la mente inizia a districarsi tra giunchi e canneti di impenetrabili ricordi che occludono la luce su di un passato lontano, a cui mai avevo fatto caso, ma che ora stava affiorando con tutta la sua potenza del fiume più lungo e imponente d'Europa, il Volga appunto, sulle cui sponde nacque mio nonno. Ho solamente un vago ricordo di lui, ma, correndo, respiro dopo respiro, i miei occhi iniziano ad immedesimarsi nei suoi, la vista del fiume Montone inizia a dar luogo a quella del Volga, accumulati come sono da storia e natura, i miei percorsi si confondono con quelli di mio nonno, inizio a comprendere, come mai avevo fatto prima, di come io somigli a lui, non solo fisicamente, come altri mi riportano, ma anche intellettualmente: lui ingegnere navale, io ingegnere dei trasporti, carriera che ho scelto in modo totalmente autonomo, senza che nessuno mi intradasse volontariamente, ed a cui mai avevo pensato fermamente, preso come ero nel giocare a calcio ad alti livelli fino a poco prima di iniziare l'università. Ora è tutto chiaro. Ciò che riporta il libro *“Oltre i confini del pensiero - Fondamenti di etica delle relazioni umane”* di Guy M. Franca è tutto vero e si è manifestato in modo inequivocabile nel mio essere: i passaggi evolutivi dell'individuo sono influenzati in modo inconscio dall'ambiente circostante sin dalle prime fasi di vita, addirittura nel feto della madre. Correndo lungo le rive del Volga, mio nonno era affascinato dalle navi che *“tirano le chiatte dal Caspio senza mai voltarsi indietro”* (V. Vysotsky, *“Kak po Volge-matushke”*), così sognò di diventare ingegnere navale e ci riuscì, tramandando a me quella passione.

Il percorso di laurea in Ingegneria dei Trasporti da me intrapreso presso l'Università degli Studi di Roma “La Sapienza” ha portato allo studio non solo del settore navale, ma in generale di tutto il mondo complesso dei mezzi di spostamento passeggeri/merci e delle relative vie di comunicazione terrestri/aeree/marittime, analizzandone gli aspetti costruttivi e le modalità di esercizio. La progettazione di un sistema di trasporto ha come obiettivo ultimo quello della massimizzazione dell'efficienza tecnico-economica di un'opera e l'approccio alla sicurezza è il paradigma che permea l'intero iter accademico e rappresenta il leitmotiv che accompagna l'intera vita professionale dell'ingegnere dei trasporti. La questione della sicurezza è un aspetto fondamentale, perché, a volte, essa può condizionare la realizzazione dell'opera. L'ingegnere dei trasporti è abituato a considerare la sicurezza come un aspetto prescrittivo. In altre parole, esistono delle norme che riguardano dimensioni e parametri di esercizio da rispettare e, una volta soddisfatte, si è in linea con le condizioni di sicurezza, indipendentemente a quello che è il livello di rischio corrispondente a tali norme. Quest'ultime vengono fuori dall'esperienza. Per esempio, la ferrovia ha oltre cento anni di esperienza e durante tutto questo periodo si è visto che certe scelte iniziali hanno funzionato e quindi sono

diventate norme. Di conseguenza, i nuovi sistemi che verranno costruiti andranno a rispettare le norme che derivano dall'esperienza consolidata. Ciò però spesso contrasta con la possibilità di usufruire dell'innovazione tecnologica, che, per definizione, non è standardizzata e quindi non dispone di norme per garantirne la sicurezza. Per tale motivo, prima di essere applicata nella progettazione, l'innovazione tecnologica va certificata, in modo tale da renderla compatibile con le norme sulla sicurezza. La tematica dell'innovazione tecnologica ha una importanza fondamentale nel campo dell'ingegneria dei trasporti. D'altronde, chi ha fatto perizie nel campo giudiziario sa bene che, durante il processo, il pubblico ministero rivolgerà sempre, prima o poi, la seguente domanda: "ha l'innovazione tecnologica messo a disposizione delle soluzioni che, se fossero state applicate, avrebbero evitato l'incidente?" In questa situazione, la risposta di chi conosce l'innovazione tecnologica non può essere che positiva. Tuttavia, essa non è sufficiente a stabilire se è stato necessario applicarla oppure no, perché su tale decisione intervengono tanti elementi, uno fra tutti è il costo.

Nel campo dell'ingegneria dei trasporti il concetto di innovazione tecnologica è fortemente legato all'automazione ed all'ICT (information and communication technologies). Per quanto riguarda la prima, essa rappresenta il futuro del mondo dei trasporti, in quanto ovunque ormai si sente parlare di veicoli/velivoli a guida autonoma, metropolitane automatiche, stazioni ferroviarie e porti impresenziati. In merito all'ICT, la progettazione, il controllo e la gestione di un qualunque sistema atto a spostare passeggeri o merci sono inevitabilmente connessi con lo scambio di informazioni, l'analisi delle stesse e la scelta delle azioni da intraprendere (spesso in real-time), per massimizzarne la sicurezza e l'efficienza. Si pensi per esempio alla gestione di una flotta di veicoli/droni automatici adibiti al trasporto passeggeri/merci, tematica estremamente popolare al giorno d'oggi. Tale sistema richiede che vi sia uno scambio di informazioni costante fra ogni componente del sistema stesso (inclusi gli utenti, che richiedono un servizio tramite specifiche applicazioni su telefono) e le strategie di esercizio devono essere implementate traendo vantaggio dalle innovazioni fornite dalla tecnologia satellitare, informatica, elettronica e delle telecomunicazioni. Ciò permette l'esecuzione di un programma di esercizio in real-time, il miglioramento delle performance della flotta (incremento del numero di richieste servite, riduzione della distanza totale percorsa e dell'attesa degli utenti), ed ottimizzazione delle problematiche relative all'incertezza legata alla domanda, ai tempi di viaggio, alle cancellazioni del servizio ed al guasto dei veicoli/droni.

La mia convinzione della sussistenza del legame indissolubile tra ingegneria dei trasporti, ICT ed automazione si è poi consolidata durante il periodo di tirocinio di tesi magistrale, presso la Torre di Controllo ENAV S.p.A. dell'aeroporto di Roma-Fiumicino, il sovrano assoluto fra i sistemi italiani di gestione e controllo del traffico, visto il primato di cui gode all'interno della sua categoria per numero di passeggeri/movimenti orari (lo scalo romano batte di gran lunga tutti gli altri aeroporti italiani per statistiche di traffico) e vista la sua maggiore complessità rispetto alle altre modalità di trasporto terrestri e marittime (il traffico aereo si muove nelle tre dimensioni spaziali, il che ne aumenta il grado di complessità nella gestione e controllo). Tale contesto mi ha aperto gli occhi verso i mondi del software e dell'hardware, direzionandomi verso la comprensione della grande potenza assunta dall'ICT in tale ambito e la possibilità di renderlo completamente automatizzato, come già avviene con le stazioni ferroviarie ed i porti.

A tal proposito, compreso che il futuro dell'ingegnere dei trasporti sarà inevitabilmente sempre più compenetrato dall'ICT e dall'automazione, dopo la Laurea Magistrale in Ingegneria dei Trasporti ho deciso di intraprendere l'ulteriore percorso di studi di laurea in Ingegneria Informatica-Automatica.

Il completamento di tale iter, ha prodotto gli effetti desiderati: a pochi mesi dal conseguimento della laurea, grazie anche all'esperienza accumulata nel settore aereo (derivante da tre tesi complessive), sono stato assunto dall'Università di Bologna come assegnista di ricerca sul Progetto RETINA, avente come obiettivo l'investigazione dell'utilizzo, nelle torri di controllo aeroportuali, di strumenti innovativi di ICT, in particolare di dispositivi di Realtà Aumentata.

La motivazione alla base dello studio su cui si incentra il progetto RETINA risiede nell'evoluzione del modus

operandi nelle torri di controllo aeroportuali: lo sguardo del controllore è stato progressivamente spostato dall'ambiente reale che si trova fuori dai vetri della torre di controllo verso le interfacce pc e monitor all'interno della stessa (*head-down equipment*). Tali interfacce riducono la *situational awareness* del controllore perché lo forzano ad alternare continuamente lo sguardo tra l'interfaccia stessa e l'ambiente reale fuori dai vetri della torre di controllo. Al contrario, le tecnologie di Realtà Aumentata offrono l'opportunità di trasferire tutta l'informazione che è attualmente disponibile nella *head-down equipment* verso una visione cosiddetta "a testa alta" (*head-up view*), che porti il controllore a guardare costantemente fuori dai vetri di torre, mediante l'uso di informazione digitale correttamente allineata (registrata) nello spazio e nel tempo con il mondo fisico reale, indipendentemente dalle condizioni di visibilità e dalla distanza.

Il successo su scala mondiale conseguito dal Progetto RETINA ha portato alla decisione di approfondire ulteriormente il campo della Realtà Aumentata con l'attivazione del corso di dottorato in oggetto. Esso ha prodotto, come risultato finale, l'ideazione e l'implementazione di un "*Video-Based Adaptive and Collaborative Augmented Reality System*" che ha valore generale e può essere esteso ad ogni campo di applicazione del mondo dell'industria e dei servizi. Come casi di studio particolari, sono stati considerati i contesti delle torri di controllo aeroportuali e degli elicotteri. In questo modo, si è trattata, specificamente al campo aerospaziale, l'applicazione del suddetto sistema sia agli impianti di gestione/controllo, che ai mezzi di trasporto, le due macro-aree fondamentali che caratterizzano ogni settore dei trasporti (terrestre, marittimo, aereo). Visto il mio background, la ricerca relativa ai mezzi di trasporto ha aperto la sfida più affascinante per un ingegnere trasportista-automatico come me: l'applicazione della Realtà Aumentata ai veicoli/velivoli comporta l'implementazione dei più performanti sistemi ICT, che devono essere altamente rispondenti a scenari operativi in cui le alte velocità in gioco e la presenza di vibrazioni possono rendere estremamente complessa la registrazione della Realtà Aumentata con il mondo reale. Dal canto mio, spero in futuro di continuare ad approfondire questa linea di ricerca.

*Per la madre Volga, lungo il fiume nutrice,
vanno le merci, i velieri e le barche,
senza sforzarsi e senza stancarsi
porta le sue navi.*

*Navigando giù per il Volga, supero le rapide
e guardo alle dolci rive sulla destra,
lì il canneto si muove, si spezza nel mezzo,
a destra la riva si estende, a sinistra sale.*

*Il Volga ha sentito canzoni più drammatiche di "Dubinushka",
in esso l'acqua fu sferzata da proiettili di nemici
e fluiva per la madre Volga in nostro sangue,
si raffreddava con schiuma rossa lungo le rive.*

*A lungo nelle acque dolci si riversavano aspre lacrime.
Rive scoscese, rive piatte,
piangevano, calpestate da zoccoli taglienti,
ma ormai le ferite maligne sono state rimarginate dalle onde.*

*Cosa vi è successo, antiche città,
lì dove vi sono vetuste mura, chiese e fortezze,
come dei cavalieri epici risvegliati,
e, innumerevoli, emersi dalla terra.*

*Vogando le navi si sforzano,
tirano le chiatte dal Caspio, tirano, si affaticano,
tirano, senza mai voltarsi indietro, e per molte miglia
le rive piatte tirano quelle scoscese.*

Indice

1	Introduzione	7
2	Campi di applicazione della Realtà Aumentata	11
3	Stato dell'arte della ricerca sulla Realtà Aumentata applicata al campo aerospaziale.....	23
3.1	L'Augmented Reality applicata ai velivoli	23
3.2	L'Augmented Reality applicata al controllo del traffico aereo.....	25
3.3	Droni e spazio: nuove frontiere per l'Augmented Reality	33
4	Stato dell'arte delle tecnologie di Realtà Aumentata	35
4.1	Display	36
4.2	Tracking	39
5	Sviluppi di tracking.....	45
5.1	Il sensore di tracking Microsoft Kinect: funzionamento e proposte di ottimizzazione.....	45
5.2	Applicazioni di tracking.....	54
5.3	Unity3D.....	59
5.4	La navigazione interattiva di ambienti virtuali nel Progetto CASTLE	64
5.5	<i>Kinect Abstraction Interface</i> – tracking svincolato dall'hardware Kinect.....	70
6	Concettualizzazione, sviluppo ed integrazione di <i>frameworks</i> di Realtà Aumentata	73
6.1	Il dispositivo Microsoft HoloLens Head-Mounted Display	73
6.1.1	Interfaccia utente	74
6.1.2	Il processo di costruzione delle applicazioni	77
6.1.3	Microsoft HoloLens versione 2.....	81
6.2	Creazione delle prime interfacce pilota di Realtà Aumentata	83
6.3	La piattaforma di validazione del progetto RETINA: applicazione dell'AR alle torri di controllo aeroportuali.....	88
6.4	Concettualizzazione, sviluppo ed implementazione di un <i>sistema AR adattivo, collaborativo e video-based</i>	94
6.4.1	Sistema AR adattivo e collaborativo.....	95
6.4.2	Sistema AR video-based	100
7	Applicazione del Sistema AR video-based ai casi di studio: torri di controllo aeroportuali ed operatori <i>search and rescue</i> a bordo degli elicotteri	114
7.1	Caso di studio n° 1 - Torri di controllo aeroportuali.....	114
7.2	Caso di studio n° 2 - Operatori <i>search and rescue</i> a bordo degli elicotteri.....	118
8	Risultati.....	122
9	Conclusioni e sviluppi futuri	124
10	Bibliografia.....	127

Acronimi

ACK	Acknowledgment
AR	Augmented Reality
CAD	Computer-Aided Design
CASTLE	CAbin Systems design Toward passenger wellbEing
CAVE	Cave Automatic Virtual Environment
CPU	Central Processing Unit
DLP	Digital Light Processing
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Center for Air- and Space-flight)
EEG	Electroencephalography
ENAV	Ente Nazionale per l'Assistenza al Volo
FOV	Field Of View
FSC	Field-Sequential Color
GIS	Geographic Information System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HCD	Human Centred Design
HMD	Head-Mounted Displays
HMI	Human Machine Interface
HUD	Head-Up Display
ICT	Information and Communications Technology
ILS	Instrument Landing System
IMU	Inertial Measurement Unit
IMVS	Interactive Multi-view Video Streaming
IP	Internet Protocol
IR	InfraRed
LAN	Local Area Network

LCD	Liquid Crystal Display
LCoS	Liquid Crystal on Silicon
MR	Mixed Reality
NASA	National Aeronautics and Space Administration
OLED	Organic Light Emitting Diode
OOT	Out Of the Tower view generator
PP App	Pseudo-Pilot Application
QR	Quick Response
RETINA	Resilient Synthetic Vision for Advanced Control Tower Air Navigation Service Provision
SD	Spatial Displays
SDK	Software Development Kit
SLAM	Simultaneous Localization And Mapping
TCP	Transmission Control Protocol
TDA	Tipo di Dato Astratto
UDP	User Datagram Protocol
VR	Virtual Reality
VS	Visible Spectrum

1 Introduzione

Lo scopo di tale introduzione è quello di riassumere il concetto di Realtà Aumentata (Augmented Reality in lingua inglese, anche nota col nome di Mixed Reality, di seguito identificata con gli acronimi AR o MR rispettivamente) ed i principali obiettivi del corso di dottorato in oggetto. Inoltre, per conferire una chiara visione d'insieme della struttura di tale elaborato, nella parte finale di questo capitolo introduttivo verranno passate in rassegna le principali tematiche di trattazione riportate in questa tesi, con esplicito riferimento ai capitoli che seguono.

Grazie al suo grande potenziale nel mercato rivolto al consumo di massa, nonché in molti campi dell'industria e dei servizi, sin dalle prime applicazioni è emerso che l'AR rappresenta oggi uno dei fattori chiave coinvolti nel concetto di Industria 4.0. Tuttavia, nonostante si abbiano diverse soluzioni inerenti al mercato rivolto al consumo di massa, i prodotti AR destinati ai campi dell'industria e dei servizi ancora soffrono di mancanza di robustezza e di efficienza, il che ne limita l'uso effettivo.

Il concetto di AR si differenzia da quello molto popolare di Realtà Virtuale (Virtual Reality in lingua inglese, di seguito anche identificata con l'acronimo VR) dal momento che, mentre il primo arricchisce il mondo reale attorno all'utente con informazione digitale aggiuntiva, il secondo fornisce un ambiente completamente artificiale, immergendo l'utente in un mondo virtuale che sostituisce totalmente quello reale. In altre parole, l'AR può essere definita come la sovrapposizione, in *real-time*, di dati digitali (oggetti 2D/3D, stringhe di testo ed immagini) generati al computer sul mondo reale attorno all'utente [1] [2].

Il concetto di sovrapposizione e georeferenziazione degli oggetti virtuali sugli oggetti reali secondo la prospettiva visiva dell'utente è anche noto col nome di "registrazione" [3] [4] [5] [6]. Così, i dati digitali devono essere registrati nello spazio, nel tempo ed in funzione della luminosità con ciò che descrivono o rappresentano nel mondo reale. Specificamente, i dati digitali devono essere correttamente allineati con gli oggetti reali del mondo fisico che circonda l'utente (registrazione in spazio), senza considerevoli ritardi (registrazione in tempo) e con una luminosità dipendente dalle condizioni ambientali (registrazione in tonalità di luce). Se tale requisito non viene soddisfatto, l'esperienza è destinata a fallire. In Fig.1 si ha un esempio di applicazione dell'AR al settore dell'automotive: due linee verdi tracciate sul parabrezza sono correttamente registrate con i bordi della strada, fornendo supporto nella guida in condizioni di scarsa visibilità.

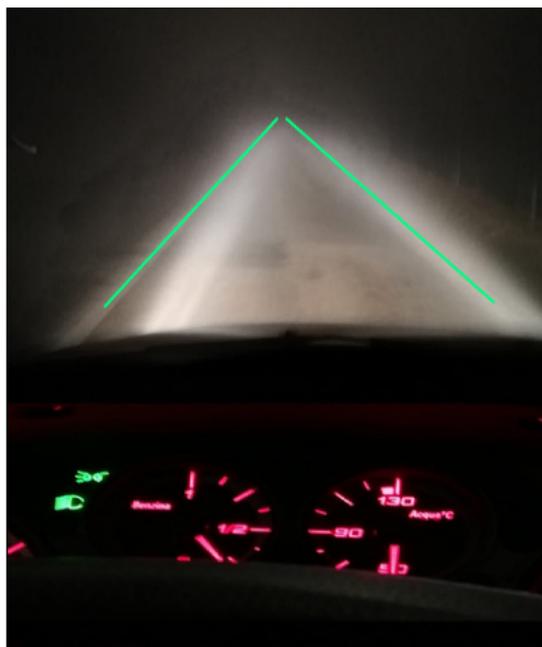


Fig. 1: il parabrezza delle automobili integrato con l'AR potrebbe aiutare il guidatore nel riconoscere i bordi della strada in condizioni di scarsa visibilità.

Poco sopra si è detto che la registrazione in tempo deve avvenire senza “considerevoli” ritardi o latenze. Ciò significa che non vi è una soglia standard di ritardo che discrimini le applicazioni efficienti da quelle destinate a fallire, ma l’aggettivo virgolettato sta ad indicare come la registrazione in tempo sia un fattore dipendente dall’applicazione che si va considerando. Per esempio, viste le velocità in gioco e le manovre consentite, un aereo caccia militare richiederà sicuramente dei ritardi di registrazione di entità molto minore rispetto a quelli consentiti ad un aereo civile. Ciò si traduce in una diversa efficienza degli algoritmi che governano i sistemi AR o in una differente maturità tecnologica e quindi in costi di implementazione che possono essere molto variabili a seconda dell’applicazione considerata.

Delineato il significato di AR, la ricerca in oggetto è stata caratterizzata dai seguenti due obiettivi:

1. Progettazione e sviluppo di efficienti interfacce AR (di seguito identificate anche col nome inglese di “AR *overlays*”) basate sull’integrazione di tecnologie innovative che producano delle soluzioni robuste alle limitazioni che presentano le attuali interfacce.
2. Test di tali interfacce in una selezione di campi di applicazione che possono beneficiare dall’introduzione dell’AR in termini di efficienza operativa e sicurezza.

Lo studio proposto si focalizza principalmente sul campo aerospaziale e, sebbene siano tre le tecnologie AR attualmente sul mercato (Fig.2), la trattazione si concentra su due di esse:

- *display* spaziali (*spatial displays* in lingua inglese, di seguito identificati con l’acronimo SD), schermi tipicamente ancorati nello spazio (ne sono esempio gli “*head-up displays*”

frequentemente usati nei *cockpit* degli aerei) il cui funzionamento non prevede l'uso di occhiali speciali o di altri dispositivi indossabili in testa da parte dell'utente;

- occhiali/*display* indossabili (più comunemente noti come *head-mounted display*, di seguito identificati con l'acronimo di HMD).

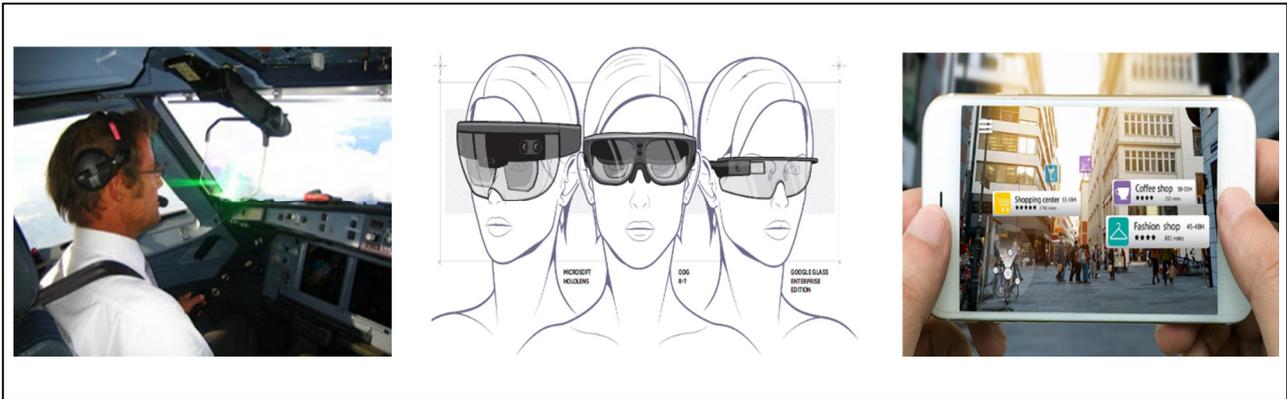


Fig. 2: le tre tecnologie AR attualmente sul mercato: *head-up display* (a sinistra), *head-mounted display* (al centro) e *hand-held display* (a destra). Soltanto le prime due verranno considerate nel presente elaborato.

Il terzo tipo di tecnologia, rappresentato dagli *hand-held display*, non viene considerato nella presente tesi, in quanto, nei due casi di studio afferenti al campo aerospaziale considerati in questo elaborato, vale a dire nelle torri di controllo e sugli elicotteri, il carico di lavoro fisico derivante dal tenere in mano i dispositivi di Realtà Aumentata rischierebbe di compromettere in modo inaccettabile contesti critici per la sicurezza quali risultano i suddetti ambiti.

Chiariti gli obiettivi della presente tesi, il contesto in cui essa si colloca ed i dispositivi AR presi a riferimento, prima di concludere questo capitolo introduttivo con la struttura generale di tale elaborato, è opportuno menzionare le tre componenti imprescindibili che caratterizzano ogni dispositivo AR. Così facendo, si potrà comprendere in modo più concreto l'iter generale (riportato più in basso) con cui si susseguono le tematiche ed i capitoli di tale studio. Detto ciò, i tre fattori fondamentali che costituiscono ogni dispositivo AR sono i seguenti:

- *display*: costituiscono i *monitor* su cui vengono rappresentate le interfacce AR e quindi interagiscono con la vista e con la percezione dell'utente, assolvendo un compito molto gravoso, vista la complessità del sistema occhi-cervello;
- *tracking dell'utente*: deve essere robusto e senza considerevoli ritardi; la corretta registrazione degli AR *overlays* con gli oggetti che descrivono o che rappresentano nel mondo reale dipende dalla prospettiva visiva dell'utente ed è un fattore critico nel funzionamento delle applicazioni AR;

- l'interazione gestuale: fortemente collegata al problema del tracking, consiste nel riconoscimento degli *input* gestuali da parte dell'utente.

Delineate le tre componenti che caratterizzano ogni dispositivo AR, la struttura generale dell'elaborato è composta come di seguito riportato.

Dopo il presente capitolo introduttivo ed i successivi due capitoli riguardanti rispettivamente i *campi di applicazione della Realtà Aumentata* (capitolo 2) e lo *stato dell'arte della ricerca sulla Realtà Aumentata applicata al campo aerospaziale* (capitolo 3), la trattazione si sposta ad analizzare lo *stato dell'arte delle tecnologie di Realtà Aumentata* (capitolo 4) in termini delle due componenti fondamentali che le costituiscono, ovvero *display* e *tracking*. Successivamente, dal momento che il tracking rappresenta il cuore di ogni sistema AR, il capitolo 5 si focalizza sulle applicazioni di tracking da me implementate e sulla loro integrazione all'interno del framework del progetto europeo CASTLE.

La ricerca condotta durante l'ultimo anno di studio e l'esperienza all'estero di sei mesi presso l'Università di Cranfield (UK) hanno permesso di conseguire importanti risultati nel campo dei sistemi di tracking, nonché nei seguenti ambiti:

- a) scambio di informazioni (collaborazione) tra dispositivi AR;
- b) *live video multi-streaming* verso dispositivi AR.

La tematica dei sistemi AR *multiuser* (a), nonché quella di proiettare in *real-time* video multipli su dispositivi AR (b), hanno aperto la strada verso l'ideazione e l'implementazione di un "*Video-Based Adaptive and Collaborative Augmented Reality System*" (capitolo 6), che può essere esteso a numerosi campi di applicazione del mondo dell'industria e dei servizi. Nel capitolo 7 la componente *Video-Based* è stata applicata specificamente al campo aerospaziale. Così, come casi di studio particolari, sono stati considerati i contesti delle torri di controllo aeroportuali e degli elicotteri, trattando, in questo modo, sia gli impianti di gestione/controllo che i mezzi di trasporto, le due macro-aree fondamentali in cui ricadono tutte le applicazioni AR che caratterizzano ogni settore dei trasporti (terrestre, marittimo, aereo).

Gli ultimi 3 capitoli (8, 9, 10) riportano rispettivamente una discussione finale, le conclusioni e la bibliografia.

2 Campi di applicazione della Realtà Aumentata

Prima di passare in rassegna i campi di applicazione più promettenti per l'AR, nonché quelli che si addicono di più alla ricerca espressa dal presente elaborato, è bene chiedersi quale sia il motivo che ha portato alla necessità di utilizzo dell'AR e quali siano gli effetti prodotti a seguito della sua introduzione. Una volta chiariti gli impatti conseguenti all'utilizzo della tecnologia AR, sarà più immediato comprendere i campi di applicazione chiave verso cui direzionare l'impiego dei dispositivi AR.

L'integrazione dell'informazione digitale aggiuntiva sovrapposta agli oggetti reali che si trovano intorno a noi è vista da molti studiosi come un passaggio epocale: l'invenzione della scrittura e della ruota segnano il superamento della dimensione dei rapporti "faccia a faccia" propri del clan o del villaggio, in seguito l'invenzione del computer segna il passaggio all'era dell'informazione su scala mondiale, infine l'AR ci catapulta dall'era dell'informazione a quella della conoscenza [7]. Infatti, come affermato da Michael E. Porter, direttore dell'Istituto di Strategia e Concorrenza della Facoltà di Economia di Harvard, l'esigenza di AR deriva dal fatto che «c'è un profondo divario tra l'enorme quantità di dati digitali che abbiamo a disposizione ed il mondo fisico in cui li applichiamo. Mentre la realtà è tridimensionale, i ricchi dati che oggi possono informare le nostre decisioni e le nostre azioni restano intrappolati su pagine e schermi bidimensionali. Questa discrasia tra mondo reale e mondo digitale limita la nostra capacità di sfruttare la massa di informazioni e di indicazioni generate da miliardi di prodotti intelligenti ed interconnessi in tutto il mondo» [8]. Così, l'AR si promette di colmare la discrasia fra il mondo digitale e quello fisico, riducendo la "distanza cognitiva" e minimizzando il "carico cognitivo". La prima rappresenta il gap fra la forma in cui l'informazione è presentata ed il contesto in cui essa è applicata. Il secondo dipende dallo sforzo mentale richiesto a processare un dato tipo di informazione, nonché dipende dalla distanza cognitiva. Un esempio concreto chiarirà il tutto (Fig.3).

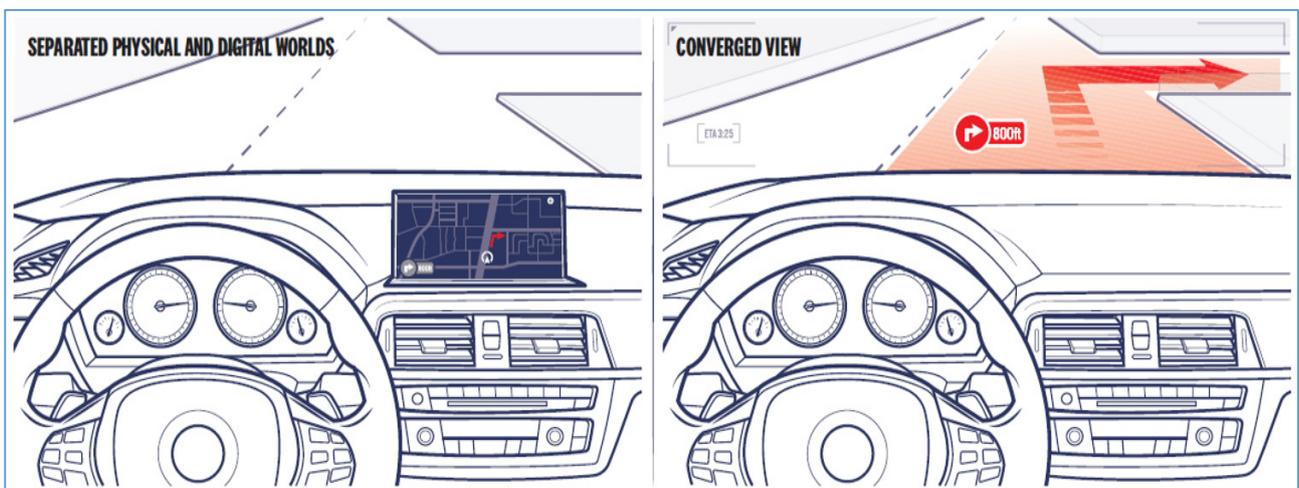


Fig. 3: in assenza dell'AR, la trasposizione mentale delle immagini GPS sulla strada reale risulta impegnativo e soggetto ad errori (sinistra); l'AR sovrappone i dati digitali direttamente sul mondo reale (destra) e riduce lo sforzo mentale necessario a connettere l'informazione digitale riguardante il mondo fisico al contesto in cui essa è applicata (fonte [8]).

Si consideri cosa accade durante la guida di un veicolo quando ci si riferisce al navigatore dello smartphone per seguire una direzione. Il guidatore consuma l'informazione dallo schermo, la trattiene nella sua memoria, la traduce nell'ambiente fisico reale ed agisce su di essa, il tutto operando contemporaneamente sul veicolo. A tal riguardo, si comprende facilmente come vi sia una significativa distanza cognitiva fra l'informazione digitale sullo schermo dello smartphone ed il contesto fisico nel quale tale informazione è applicata. La presenza di tale distanza crea un carico cognitivo. Al contrario, sovrapponendo l'informazione digitale direttamente sull'ambiente reale, l'AR permette di processare entrambi in modo simultaneo, eliminando la necessità di connettere mentalmente i dati digitali al mondo fisico. In questo modo, permettendo al guidatore di focalizzare totalmente la propria attenzione sulla strada, l'AR abbatte la distanza cognitiva e minimizza il carico cognitivo, in quanto:

- riduce lo sforzo mentale (“*mental workload*”) nel registrare l'informazione digitale col contesto reale,
- previene le distrazioni (e quindi incrementa la cosiddetta “*situation awareness*”),
- minimizza gli errori durante la guida (e quindi incrementa le “*performances*”).

Perciò, in generale, è bene tenere a mente che l'introduzione dell'AR produce sempre dei benefici in termini di:

- a) *workload*, *situational awareness* e *performances* degli utenti [9] [10];
- b) efficienza del sistema su cui gli utenti agiscono.

Da tutto ciò segue che l'AR permette agli utenti di migliorare la propria abilità nell'assorbire rapidamente ed accuratamente l'informazione, nel prendere decisioni e nell'eseguire i *task* richiesti in modo veloce ed efficiente.

Comprese le necessità che hanno portato alla nascita dell'AR e gli effetti prodotti a seguito della sua utilizzazione, di seguito vengono descritti quelli che costituiscono correntemente i campi di applicazione più popolari. Essi non esauriscono affatto l'intero spettro di tutti i possibili impieghi dell'AR, in quanto tale lista risulterebbe pressoché infinita, dal momento che ogni giorno compaiono nuove idee di applicazione. Lo scopo quindi è quello di fornire una panoramica generale e di espandere la propria riflessione ed inventiva sulla miriade di possibilità che l'AR offre.

I primi campi di applicazione che emergono in modo esplicito anche agli occhi dei più profani sono quelli legati al campo dell'industria, dove l'AR crea valore secondo due modalità principali:

1. diventando parte del prodotto stesso;
2. migliorando l'efficienza lungo tutta la filiera (*supply chain*).

In merito al punto 1, si è già osservato in precedenza come l'AR crei valore diventando parte del prodotto stesso, considerando l'esempio dell'ipotetico parabrezza innovativo per auto riportato nelle Fig.1 e Fig.3 precedentemente descritte. In realtà, il punto 1 di cui sopra rappresenta un cluster che può racchiudere una miriade di prodotti, afferenti di campi di applicazione più disparati. Tuttavia, i

settori dell'automotive, dei trasporti in generale e della difesa sono sicuramente di importanza preminente, visto che sono quelli che in assoluto investono maggiormente in AR [8] (nonché, in generale, in Ricerca e Sviluppo) e quindi il lettore sarà sicuramente indotto a far ricadere correttamente nel *cluster* del punto 1 l'intera generalità dei mezzi di trasporto terrestri, marittimi ed aerospaziali (inclusi i droni) di scopo civile e militare (Fig.4).

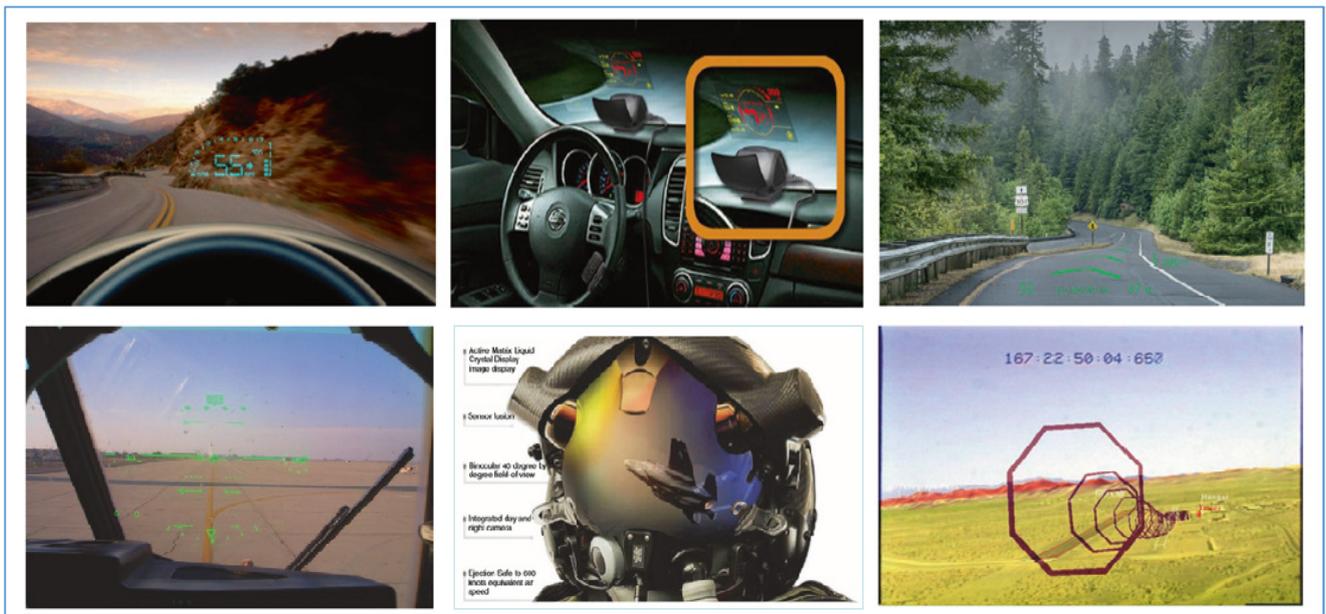


Fig.4: i settori dei trasporti e della difesa sono quelli che investono di più in AR: le immagini nella fila superiore si riferiscono all'applicazione dell'AR al campo dell'automotive, mentre la fila inferiore a quello aerospaziale civile/militare (fonte [7]).

In riferimento al secondo punto di cui sopra, è stato dimostrato che l'utilizzo dell'AR lungo tutte le attività della *supply chain* può migliorare l'efficienza di ognuna di esse, in quanto riduce drasticamente gli errori ed incrementa fortemente la produttività delle stesse [11]. Da ciò segue che ogni singolo settore componente la filiera produttiva diviene campo di applicazione per l'AR: dalla fase iniziale di design e sviluppo di prodotto, passando per la produzione, logistica/magazzino e marketing, arrivando alla fase finale del servizio post-vendita, ogni attività diviene terreno fertile per l'AR, come riportato in Fig.5.



Fig. 5: i campi di applicazione dell'AR spaziano lungo tutta la filiera industriale, in quanto l'AR migliora l'efficienza di ogni singola componente della supply chain (Fonti in senso orario: [ref 1](#), [ref 2](#), [ref 3](#), [ref 4](#), [ref 5](#)).

Durante il processo di design e sviluppo di prodotto, l'utilizzo dell'AR assume un significato fondamentale: ogni fase progettuale può essere profondamente analizzata attraverso un approccio sistematico che permetta all'utente di essere al centro di ogni *step* di avanzamento nel processo di realizzazione del prodotto. Le scelte progettuali vengono così indirizzate non più attraverso i software di modellazione che effettuano *rendering* sugli schermi bidimensionali dei pc, ma mediante la navigazione 3D in prima persona dei prodotti che si vanno creando, valutandone le dimensioni reali, i colori più adatti e quindi anche l'impatto emotivo che tutto ciò produce sull'utilizzatore finale, massimizzando in tale modo l'efficienza dell'intero iter del processo di progettazione, favorendo l'approccio HCD (*Human Centred Design*). Tale tematica verrà molto approfondita nel capitolo 5 dedicato al tracking, dove si tratteranno in modo esauriente gli obiettivi e le procedure sperimentali considerate nel progetto europeo CASTLE, che mira allo sviluppo di un nuovo processo di progettazione degli interni di cabina degli aerei, ponendo al centro dell'attenzione il comfort del passeggero sin dalle fasi iniziali del processo di progettazione.

Allo stesso modo di quanto esposto per la fase di design e sviluppo del prodotto, l'impiego dell'AR durante il processo di produzione assume una valenza cruciale. In merito a ciò, è significativo menzionare come la società Boeing abbia messo a confronto le *performances* degli operatori durante lo svolgimento di *task* complessi portati a termine mediante l'uso di istruzioni presentate su supporti cartacei e schermi 2D oppure facendo uso dell'AR: i benefici apportati da quest'ultima sono stati fortemente superiori [12].

L'AR fornisce anche un valido supporto nel processo di ottimizzazione delle operazioni logistiche, un settore sempre più strategico per le aziende concorrenti su scala mondiale. Nella pratica corrente, gli operatori all'interno dei magazzini eseguono solitamente più azioni durante la gestione di un ordine: devono individuare il prodotto corretto, scansionarlo e consegnarlo alla banchina di carico.

Tuttavia, le nuove soluzioni emergenti di *computer vision* possono identificare velocemente dove si trova un prodotto, oppure se si tratta del prodotto corretto, il tutto ad un ritmo molto più rapido di quanto potrebbe essere altrimenti svolto senza l'uso dell'AR. Così, tale tecnologia ha il potenziale di ridurre significativamente i costi delle operazioni di stoccaggio. Ne è esempio DHL, azienda logistica tedesca leader sulla scena mondiale, che sta testando i dispositivi di AR che consentono il riconoscimento degli oggetti in *real-time*, la lettura dei codici a barre e la navigazione all'interno del magazzino: indossando gli HMD, gli addetti al magazzino possono vedere le liste di prelievo digitali nel loro campo visivo e determinare il percorso migliore, riducendo i rischi ed i tempi di viaggio.

Da ultimo, ma non per questo meno importanti, sono le applicazioni dell'AR al marketing ed al servizio post-vendita, che costituiscono oggi delle attività indispensabili lungo la *supply chain*, alla stessa stregua delle altre già citate precedentemente. In relazione al marketing, nel momento in cui i clienti, prima di acquistare un prodotto, hanno la possibilità di vedere virtualmente il modo in cui esso si collocherà o funzionerà in un ambiente reale, ciò li porterà a sperimentare delle aspettative più accurate sul prodotto, una maggiore sicurezza nelle decisioni di acquisto e quindi una *customer satisfaction* di livello più elevato.

Inoltre, dal momento che quest'ultima rappresenta uno degli obiettivi strategici per vincere la concorrenza sul mercato, nel corso degli ultimi anni la filiera produttiva ha iniziato sempre di più a sviluppare ed inglobare al suo interno le attività del servizio post-vendita, ovvero quelle di assistenza al cliente, riparazione e manutenzione, dando vita a soluzioni innovative in cui tecnici specializzati effettuano assistenza da remoto o in loco avvalendosi di tecnologia AR per vedere la storia di funzionamento di un componente difettoso, ripararlo, mantenerlo ed effettuare diagnostica.

Fino ad ora sono stati analizzati i campi di applicazione dell'AR prendendo a riferimento le due macro-aree in cui essa diventa parte del prodotto stesso (1) ed i suoi impieghi lungo la filiera produttiva (2). La trattazione della *supply chain* ha permesso di estendere l'utilizzo dell'AR non solo al settore del secondario, ovvero dell'industria (attività di design e sviluppo di prodotto, nonché quella di produzione riportate sopra), ma anche al settore dei servizi, in quanto le attività della logistica, marketing e post-vendita sopracitate hanno un carattere trasversale e possono essere inglobate nel settore del terziario, specie se esternalizzate dall'azienda produttrice. Prima di approfondire ulteriormente il settore dei servizi, è bene concludere la trattazione di quello secondario con altre due sfere di applicazione chiave dell'AR, ovvero le costruzioni e l'industria dell'Energia, Oil&Gas (Fig.6).



Fig. 6: Altri due campi di applicazione molto promettenti del settore del secondario: le costruzioni (sinistra, fonte [ref 6](#)) ed Energia, Oil&Gas (destra, fonte [ref 7](#)).

In entrambi i contesti, l'AR, integrata con applicazioni CAD e tecnologia GIS (quest'ultima utilizzata per georeferenziare le interfacce AR), fornisce un valido supporto operativo, in quanto permette all'utente di sovrapporre all'ambiente reale la visualizzazione virtuale di strutture (giunti, travi, colonne, rinforzi di cemento armato, ecc.) impianti o condotte, sia in fase di costruzione, per controllare il corretto avanzamento dei lavori mediante il confronto tra lo stato di fatto e quello di progetto, che durante l'esercizio, per facilitare l'individuazione ed il raggiungimento di componenti nascoste all'interno delle pareti o sotto terra durante gli interventi di ristrutturazione, manutenzione, modifica o riparazione. Ne segue che l'utilizzo dell'AR in tali ambiti porta ad un notevole risparmio economico e di tempo. Si pensi ad esempio al caso di dover effettuare degli scavi in ambito urbano per raggiungere tubature sotto il manto stradale: l'AR integrato con la tecnologia GIS permette di visualizzare con estrema esattezza il punto di passaggio dell'opera nascosta sotto terra e quindi di operare con precisione chirurgica per limitare l'entità degli scavi ed i tempi di intervento, nonché le ripercussioni sul traffico e sull'inquinamento che ne conseguono. Questo tipo di applicazioni, caratterizzate dall'obiettivo di registrare l'AR con oggetti fisici reali non visibili perché nascosti od occlusi, prendono il nome di "visualizzazioni a raggi X" [13] [14] [15] [16]. La ricerca inerente a tale tematica riguarda anche la sfera di applicazione del campo medico, di cui si parlerà a breve.

Analogamente al settore secondario, quello dei servizi è anch'esso un terreno molto fertile per l'AR. Ne è esempio il campo medico, con diverse applicazioni documentate. L'ambito della chirurgia risulta sicuramente uno dei più promettenti: chi svolge un'operazione potrebbe sfruttare l'AR per visualizzare i parametri vitali del paziente ed altre innumerevoli informazioni sovrapposte allo stesso con codifiche di colore opportune ed intervenire in modo preciso e minimamente invasivo (Fig.7). Come nel caso descritto poco sopra relativamente agli impianti e condotte sotto terra, anche qui si potrebbe parlare di componenti simili, ma propri dell'anatomia dell'essere umano, come l'apparato circolatorio, scheletrico, sistema nervoso, il tutto non come prima nascosto sotto il manto stradale, ma sotto pelle. In forza di ciò, è stato scientificamente dimostrato che, in confronto alla visualizzazione 2D, quella stereoscopica (3D) comporta dei forti miglioramenti nelle *performances* dei chirurghi nelle teleoperazioni endoscopiche o negli interventi in laparoscopia [17] [18]. Inoltre, utilizzando l'innovativo concetto di *live video multi-streaming* su dispositivi HMD (capitolo 6), il potenziale di utilizzo della tecnologia AR nel campo medico cresce notevolmente: si pensi allo

svolgimento di un intervento complesso in modalità di videoconferenza dove chi opera viene supportato da remoto da un chirurgo luminare che si trova dall'altra parte del mondo, oppure al caso della riabilitazione, in cui il paziente sfrutta un dispositivo AR per comprendere la corretta esecuzione di un esercizio mostrato dal fisioterapista in videoconferenza.



Fig. 7: applicazione dell'AR al campo medico della chirurgia. Visualizzazione dei parametri vitali del paziente ed altre innumerevoli informazioni sovrapposte allo stesso con codifiche di colore opportune per intervenire in modo preciso e minimamente invasivo (fonte [ref 8](#))

Altro campo di applicazione di carattere dominante è quello della didattica, dell'addestramento e dello sport, dove l'AR spicca per la potenzialità di comprendere e simulare lo svolgimento di *task* reali.

Per quanto riguarda la didattica, la visualizzazione di modelli 3D in scala 1:1 e la possibilità di esplorarli ed osservarli da tutte le possibili angolazioni fornisce all'utente uno strumento di studio di ineguagliabile valore [19] [20] [21].

In merito all'addestramento, in [22] gli autori descrivono come l'uso dell'AR per il *training* ed il supporto nella manutenzione degli aeromobili sia di grande beneficio per la riduzione degli errori degli operatori, le cui responsabilità hanno impatto sulla sicurezza di centinaia di vite umane.

Infine, relativamente allo sport, numerose applicazioni sono già sul mercato e guidano l'utente a migliorare le proprie prestazioni in modo continuo durante la prova. Esempio fra tutti è il nuoto,

dove gli occhiali rappresentano un componente imprescindibile in ogni competizione. Così qualcuno ha pensato di renderli “aumentati” e quindi sono diventati a tutti gli effetti tecnologia AR [23]: mediante essi, il nuotatore in fase di allenamento può visualizzare in modo continuo le proprie prestazioni e confrontarle con quelle del giorno prima o con i risultati migliori (Fig.8), oppure letteralmente gareggiare contro un umanoide digitale.



Fig. 8: applicazione dell'AR nello sport del nuoto. Gli occhiali del nuotatore diventano dispositivi AR che permettono di visualizzare le prestazioni in modo continuo (fonte [ref 9](#))

L'elevato potenziale di impiego dell'AR nei campi di applicazione fin qui descritti si riflette anche sulle attività connesse al campo della “*security*”. Le persone che accedono a luoghi di aggregazione di massa o strutture/infrastrutture molto affollate (centri commerciali, stazioni metropolitane/ferroviarie ed aeroporti) possono essere scansionate da videocamere che, mediante algoritmi di riconoscimento facciale, confrontano le acquisizioni con i profili presenti nei database delle forze dell'ordine e, in caso di presenza di soggetti pericolosi, mettono in allerta gli addetti alla sicurezza operanti con dispositivi HMD (Fig.9).

Tali considerazioni possono essere anche estese al concetto di “*healthcare-security*”, reso oggi molto popolare a causa della pandemia da COVID-19 che stiamo vivendo. In questo ambito, l'unica differenza col caso di cui sopra sarebbe costituita dall'uso di termocamere, che, invece di confrontare le immagini acquisite con i profili a rischio, risulterebbero adibite al calcolo della temperatura corporea. Una soluzione già disponibile negli Emirati Arabi e nell'aeroporto internazionale di Roma-Fiumicino è quella di dotare gli operatori addetti alla “*healthcare-security*” di speciali dispositivi HMD abilitati alla misurazione della temperatura corporea. Tuttavia, la convenienza di tale soluzione va debitamente valutata sulla base dei costi che comporta la presenza fisica degli operatori a tutti gli ingressi dell'ambiente da controllare, il che potrebbe risultare molto dispendioso e di scarsa convenienza rispetto all'impresenziamento delle entrate ed automatizzazione del processo con termocamere fisse che allertano gli operatori solo in caso di necessità di intervento.

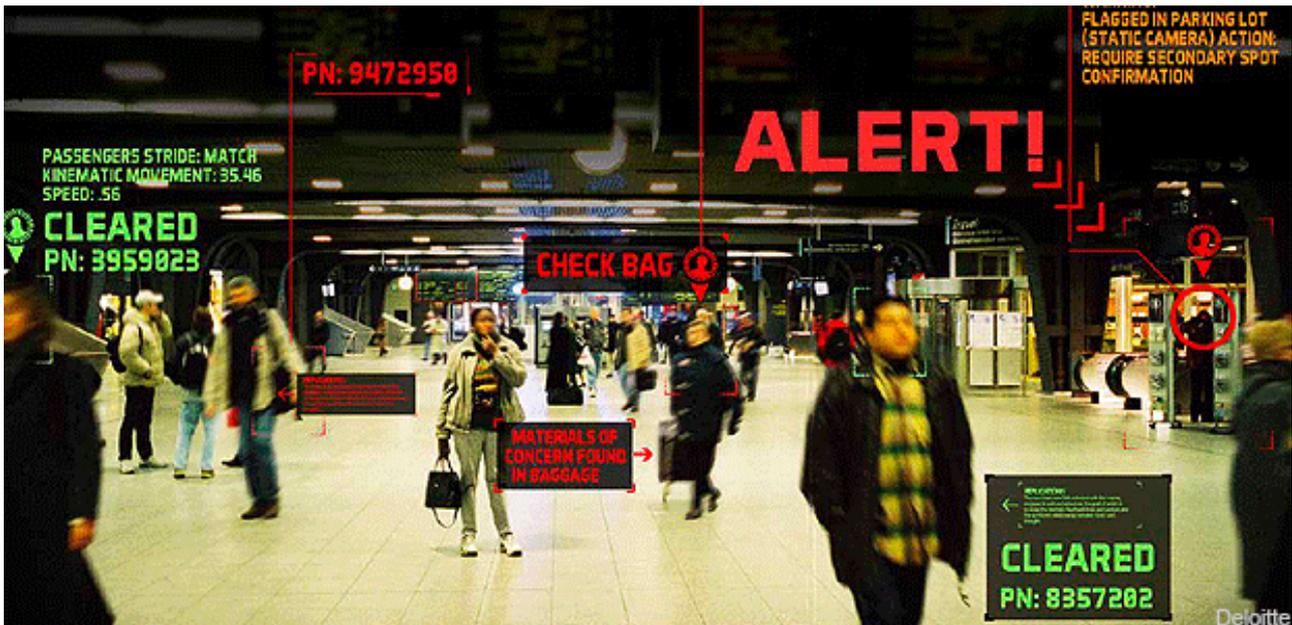


Fig. 9: AR integrata con sistemi di riconoscimento facciale nel campo della security. L'operatore viene messo in allerta nel caso in cui venga identificato un soggetto pericoloso a seguito del confronto fra l'immagine acquisita ed i profili presenti nei database delle forze dell'ordine (fonte [ref_10](#))

Spostandoci nella sfera dei servizi di carattere pubblico, le attività svolte dagli operatori di primo soccorso, come vigili del fuoco, personale sanitario, forze dell'ordine, esercito, protezione civile ed operatori "search and rescue", si prestano in modo esemplare ad essere coadiuvate dall'utilizzo dell'AR, specie se essa viene integrata con sistemi di georeferenziazione o mappe di ambienti indoor in cui i suddetti operatori sono chiamati ad intervenire. Così, proiettando le informazioni di navigazione direttamente sul visore AR, la *situational awareness* risulta massimizzata, non potendo essere altrimenti in tali contesti caratterizzati da vite in pericolo.

Vista l'importanza della tematica, diverse soluzioni sono state proposte per svolgere tali "life-critical tasks". Ne è esempio quella ideata dall'École Polytechnique Fédérale di Losanna (Svizzera) che ha sviluppato un HMD per vigili del fuoco che permette loro di valutare le temperature degli ambienti attraverso delle termocamere (Fig.10).

Il settore degli operatori di primo soccorso si coniuga fortemente con il risultato finale di questa tesi, ovvero con il "Video-Based Adaptive and Collaborative Augmented Reality System", ideato, sviluppato ed implementato nel corso dei tre anni di dottorato. Infatti, l'utilizzo di un sistema AR "video-based" e "collaborative" presenta nei suddetti ambiti un altissimo potenziale, in quanto garantirebbe uno scambio di informazioni e di video (*video-based*) tra operatori e supervisori ("collaborative"), questi ultimi aventi il ruolo di ottimizzare le strategie di intervento in *real-time* da una centrale operativa. Il caso di studio sugli elicotteri considerato nella parte finale di questo elaborato riguarderà proprio gli operatori *search and rescue* a bordo degli stessi.



Fig. 10: sistemi AR per vigili del fuoco. Le informazioni digitali rappresentate derivanti dall'integrazione di opportuna sensoristica permettono lo svolgimento dei task massimizzando la situational awareness e la sicurezza dell'operatore (fonte [ref 11](#)).

Sebbene il settore dei servizi comprenda moltissimi altri campi di applicazione molto promettenti per l'AR, come per esempio quelli dell'arte e dell'intrattenimento, ecc., si vuole concludere questa trattazione ritornando al settore dei trasporti. Tutte le applicazioni AR afferenti ad esso si possono dividere in 2 macro-aree funzionali fondamentali: navigazione e controllo/gestione. L'elevato potenziale per l'impiego della tecnologia AR è già stato analizzato precedentemente in riferimento ai mezzi di trasporto sia del campo civile che di quello militare. Così, trattata la macro-area della navigazione, il focus si concentra ora sulla macro-area degli impianti di controllo/gestione, indispensabile in ogni sistema di trasporto (terrestre, marittimo, aereo), in quanto, ai fini della sicurezza, se c'è navigazione, parimenti deve esistere un organo atto a monitorarla e regolarla secondo norme prestabilite.

Tale seconda macro-area è di solito caratterizzata dalla presenza di operatori che sperimentano una visione panoramica sull'ambiente circostante che deve essere controllato e gestito. Questo è il caso dei centri di controllo all'interno degli aeroporti, porti e terminal container. L'utilizzo dell'AR in tali ambiti apporta significativi benefici, come la riduzione del carico di lavoro (*workload*) degli operatori, così come il miglioramento delle *performance* e della *situation awareness* degli stessi [24], producendo in tale modo un impatto positivo non solo nei confronti del personale addetto, ma in generale sulla sicurezza e sull'efficienza dell'intero sistema controllato e gestito [3] [25]. Ciò è stato anche confermato dai risultati di RETINA, un progetto di ricerca europeo coordinato dall'Università di Bologna a cui lo scrivente ha contribuito mediante un contratto di assegno di ricerca, con lo scopo di investigare l'applicazione dell'AR nelle torri di controllo aeroportuali (Fig.11). Essendo accomunati dal paradigma della visione panoramica dall'alto, le soluzioni introdotte da RETINA, di cui si parlerà più avanti nel corso di questo elaborato, possono essere

concettualmente estese anche a tutti gli altri centri di controllo del settore dei trasporti, primi tra tutti quelli portuali e dei terminal container (Fig.12).

Così, vista la valenza e la possibilità di estensione di tale tematica, parallelamente a quello sugli elicotteri, l'altro caso di studio del "*Video-Based Adaptive and Collaborative Augmented Reality System*" considerato in questo elaborato ha riguardato il contesto delle torri di controllo aeroportuali. In questo modo, lo *step* principale introdotto da tale elaborato sta nell'aver ideato, sviluppato ed implementato nei tre anni di dottorato un sistema AR flessibile, il cui carattere trasversale fa sì che esso possa essere efficientemente applicato, apportando innovazione, ad entrambe le macro-aree funzionali della navigazione e del controllo/gestione in cui possono essere fatte ricadere tutte le applicazioni AR afferenti non solo al campo aerospaziale (trattato specificatamente in questa tesi), ma in generale a quello dei trasporti di carattere civile e militare.



Fig. 11: Progetto RETINA - Applicazione dell'AR alle torri di controllo aeroportuali.



Fig. 12: L'investigazione, da parte del Progetto RETINA, dell'applicazione dell'AR alle torri di controllo aeroportuali ha aperto la strada verso l'estensione dell'impiego dell'AR in tutti gli altri centri di controllo del settore dei trasporti, primi tra tutti quelli portuali e dei terminal container, tutti accomunati dal paradigma della visione panoramica dall'alto dell'ambiente da controllare e gestire.

Per concludere, questo capitolo ha voluto conferire al lettore la consapevolezza del forte potenziale espresso dalle tecnologie AR, mediante la trattazione delle necessità che hanno portato alla loro nascita, degli effetti prodotti a seguito della loro utilizzazione e dei campi di applicazione più popolari in cui attualmente il mercato sta investendo di più, ma che non esauriscono affatto l'intero mondo di tutti i possibili impieghi dell'AR, che sono pressoché infiniti. Così, insieme al capitolo introduttivo, la speranza della trattazione fin qui considerata è quella di aver fornito una panoramica generale di cosa sia l'AR e delle motivazioni alla base del suo impiego.

Con i capitoli che seguono la trattazione abbandonerà il carattere di generalità fin qui espresso e si addenterà sempre più nello specifico, passando per lo stato dell'arte della ricerca sull'AR applicata al campo aerospaziale e terminando con la concettualizzazione, sviluppo ed implementazione del *"Video-Based Adaptive and Collaborative Augmented Reality System"*.

3 Stato dell'arte della ricerca sulla Realtà Aumentata applicata al campo aerospaziale

3.1 L'Augmented Reality applicata ai velivoli

Il concetto di AR è nato con l'aviazione e, più precisamente, a bordo dei velivoli. Le prime applicazioni risalgono al 1937, quando l'AR fu utilizzata in Germania per fornire supporto agli aerei da combattimento tedeschi [7]. Dapprima, la tecnologia impiegata è stata quella degli *head-up display* integrati nei *cockpit* dei velivoli, inizialmente militari e poi, a partire dagli anni '70, anche commerciali. In seguito, l'utilizzo è stato esteso anche all'aviazione generale. Tipicamente basati su una grafica bidimensionale, gli *head-up display* sono degli schermi trasparenti “*see-through*” che permettono ai piloti di sovrapporre, alla visione del mondo circostante, un'informazione digitale aggiuntiva, principalmente rappresentata da indicazioni di *routing*, aiuti alla navigazione, identificazione degli ostacoli, supporto alle procedure di atterraggio ed ai sistemi di puntamento (aerei caccia). Un considerevole interesse da parte della ricerca è oggi focalizzato sulla possibilità di superare il carattere bidimensionale di tali dispositivi e sviluppare soluzioni tridimensionali, atte a produrre vantaggi in termini di sicurezza ed efficienza rispetto alle classiche tecnologie 2D [26].

Verso la fine degli anni '60, gli *head-up display* sono stati messi in concorrenza con i primi prototipi della soluzione alternativa rappresentata dagli HMD. I vantaggi nell'utilizzo di quest'ultimi sono stati approfonditamente analizzati in [27] [28] [29] [30]. I primi due studi effettuano una descrizione dello stato dell'arte degli HMD per piloti ed esaminano lo sviluppo di tale tecnologia a partire da mezzo secolo fa, quando si iniziò a comprendere il suo maggiore potenziale rispetto alle soluzioni *head-up display*, principalmente sotto gli aspetti di un'area di applicazione più vasta dell'informazione digitale aggiuntiva (con le soluzioni HMD i dati AR seguono la direzione dello sguardo del pilota e quindi l'area di applicazione degli stessi è molto più ampia rispetto a quella caratteristica degli *head-up display*, dove i dati AR sono pertinenti ad un'area più limitata e relativa solamente alla direzione frontale dell'aereo), un incremento della *situational awareness* ed una riduzione dell'accomodazione visiva. Gli ultimi due studi riportati riguardano il campo di applicazione degli elicotteri e quello degli aerei commerciali rispettivamente. Nel primo dei due, sono stati condotti degli esperimenti sull'impiego degli HMD per valutare gli aspetti relativi alla sicurezza nella navigazione in condizioni di scarsa visibilità (minore agli 800 m): sedici piloti professionisti sono stati coinvolti nello studio ed uno dei principali risultati ottenuti dall'utilizzo degli HMD è stato quello della riduzione del carico di lavoro (*workload*) dei piloti nell'evitare le collisioni con gli ostacoli. Nell'altro studio, ovvero in quello relativo agli aerei commerciali, la ricerca si è focalizzata sull'impiego del dispositivo Microsoft HoloLens come strumento di ausilio alla navigazione e sostituzione del supporto fornito dal copilota durante le procedure di emergenza, con lo scopo di incrementare la capacità di gestire *task* complessi in tempi stretti. Ventiquattro piloti professionisti sono stati coinvolti negli esperimenti svolti mediante un simulatore di volo relativo ad un Airbus A320, atto a confrontare la situazione di stato di fatto, senza impiego degli HMD, con quella di progetto, caratterizzata dall'utilizzo degli stessi. Come risultato, le istruzioni e l'informazione digitale aggiuntiva fornite dal dispositivo Microsoft HoloLens HMD riducono i tempi di reazione, il carico di lavoro (*workload*) e migliorano la *situational awareness* dei piloti.

Ad oggi l'applicazione dell'AR ai velivoli sembra sempre più protendere verso le soluzioni HMD rispetto a quelle degli *head-up display*. La ricerca militare, che, dalla notte dei tempi, ha sempre spianato la strada verso il futuro delle applicazioni civili, è fortemente orientata verso i dispositivi AR indossabili. Un primo esempio è rappresentato dall'aereo caccia militare statunitense F-35. Esso è stato il primo velivolo da combattimento a volare senza l'integrazione nel *cockpit* dell'*head-up display*, la cui funzionalità è stata rimpiazzata da speciali HMD, nati nel 2014 dalla collaborazione delle società Rockwell Collins (statunitense) ed Elbit Systems (israeliana); integrati all'interno dei caschi dei piloti, sono caratterizzati da un costo che si aggira attorno ai 400.000 dollari e rappresentano oggi uno dei dispositivi più all'avanguardia in termini di tecnologia AR sul mercato [31] (Fig.13).

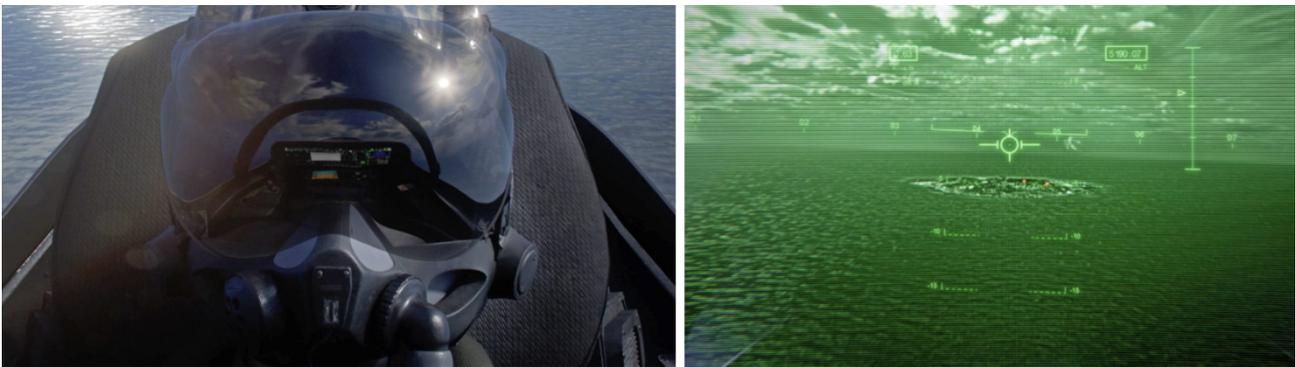


Fig. 13: casco per piloti dell'aereo caccia militare statunitense F35 (a sinistra); visione del pilota da dentro il casco integrato con l'AR (a destra).

Lo scenario dei dispositivi indossabili a bordo degli aerei militari vede un altro esempio di eccellenza, il che mostra un trend non di certo destinato a fallire, anzi, in netto sviluppo: il progetto britannico Tempest, reso pubblico all'air show di Farnborough nel 2018, è frutto di una partnership tra le società BAE Systems, Leonardo, MDBA, Rolls-Royce e MoD e mira alla creazione del primo caccia *stealth* multiruolo di "sesta generazione", in cui uno dei concetti più innovativi riguarda l'idea di un "*cockpit* indossabile" mediante HMD; in altre parole, l'utilizzo dell'AR e dell'interazione gestuale con strumenti virtuali è pensato come elemento sostitutivo dei *cockpit* odierni e rivoluzionario l'intera cabina di pilotaggio (Fig.14).



Fig. 14: il progetto britannico Tempest mira alla creazione del primo caccia stealth multiruolo di “sesta generazione”, in cui uno dei concetti più innovativi riguarda l’idea di un “cockpit indossabile” mediante HMD, che andrebbe a sostituire la strumentazione presente negli odierni cockpit delle cabine di pilotaggio degli aerei militari [32].

Sulla scia di quella militare, l’introduzione degli HMD nel campo dell’aviazione civile non si fa attendere. Ne è esempio la tecnologia TopMax, un innovativo dispositivo indossabile creato dalla società Thales ed atto a fornire ai piloti tutte le informazioni indispensabili di cui essi hanno bisogno durante le fasi di volo e di manovra all’interno degli aeroporti [33].

Mentre nel XX secolo la ricerca sull’AR condotta nell’ambito del settore aerospaziale era esclusivamente destinata allo sviluppo di applicazioni atte ad operare a bordo dei velivoli, con l’inizio del nuovo millennio essa ha iniziato ad esplorare un nuovo campo di applicazione molto promettente del settore aerospaziale, ovvero quello relativo ai sistemi di controllo/gestione a terra.

3.2 L’Augmented Reality applicata al controllo del traffico aereo

Anche se in realtà la prima concezione dell’utilizzo dell’AR per gli scopi del controllo del traffico aereo risale al 1996 [34], è nei primi anni del 2000 che tale direzione di impiego si è definitivamente consolidata con la ricerca portata avanti dalla NASA. In un primo lavoro sono state studiate le specifiche funzionali atte a caratterizzare i *display* AR destinati ad essere usati nelle torri di controllo [35]. Al fine di ciò, tre differenti *field of view* (FOV)

diagonali binoculari, rispettivamente di 14°, 28° e 47°, sono stati esaminati per determinare l'effetto sulle capacità degli utenti ad identificare gli aerei al suolo o in atterraggio. I risultati attestano che, rispetto agli altri due, il FOV di 47° apporta un netto miglioramento nell'efficacia di identificazione dei velivoli e che valori molto superiori ai 50° non producono benefici molto marcati. L'anello debole di tale ricerca è stato quello di non coinvolgere controllori reali all'interno degli esperimenti. A seguito di tali risultati, in un secondo lavoro la NASA ha sviluppato dei prototipi di dispositivi HMD e li ha testati coinvolgendo il personale operante nelle torri di controllo aeroportuali [36]. L'obiettivo è stato quello di arricchire la visione dell'ambiente reale fuori dai vetri di torre con informazione digitale aggiuntiva sugli aerei in arrivo o in partenza, in modo tale da gestire il traffico a vista, indipendentemente dalle condizioni atmosferiche. Dai primi esperimenti effettuati è risultato che, sebbene i controllori, in modo unanime, abbiano valutato molto positivamente il potenziale rappresentato da tale tecnologia (soprattutto nella risoluzione di molte problematiche relative al controllo dei movimenti degli aeromobili a terra, al coordinamento con altre strutture e veicoli, all'acquisizione dell'informazione e al monitoraggio delle "runway incursions"), d'altro canto hanno anche constatato una insufficiente maturità tecnologica, manifestata in forma di ritardi di registrazione dei dati virtuali con il mondo reale, eccessivo discomfort fisico dovuto alla pesantezza del dispositivo, *field of view* (FOV) molto limitato e bassa luminosità dei dati virtuali.

L'imaturità tecnologica dei dispositivi HMD ha portato gli scienziati coinvolti nel progetto IMAN (parte del framework "iPort" promosso dal Ministero Federale Tedesco degli Affari Economici e dell'Energia) ad analizzare, sviluppare e testare soluzioni di interfacce AR per controllori, mediante la proiezione delle stesse sui vetri virtuali di un simulatore di torre di controllo o su di un pannello semitrasparente antistante l'ambiente virtuale [37]. Tale ricerca, che ha avuto come scopo quello di progettare un'opportuna HMI atta a ridurre il carico di lavoro mentale degli operatori, ha seguito un approccio HCD (*Human Centred Design*) mediante il coinvolgimento di molti controllori che hanno partecipato attivamente ad ogni fase della progettazione. Attraverso l'analisi delle possibili posizioni occupate dagli utenti all'interno della sala del simulatore di torre ed in base alla configurazione dei pannelli semitrasparenti utilizzati, nonché alla disposizione dei vetri di torre virtuali, è stata definita l'area nella quale ogni AR overlay deve ricadere, in modo tale che la vista o l'attività svolta da ciascun controllore non sia disturbata dall'operato dei suoi colleghi impegnati in altri *task*. Il fattore caratteristico di tale ricerca è stato quello di applicare sistemi tecnologici diversi rispetto agli HMD per la fornitura dell'AR ai controllori di torre, ovvero i cosiddetti *display* spaziali SD, i cui aspetti caratteristici ai fini della loro progettazione ed applicazione nelle torri di controllo sono stati dapprima studiati in [38] e poi in [6].

Nel corso degli anni, il progresso della tecnologia HMD ha permesso di risolvere o fortemente ridurre le problematiche constatate dalle ricerche condotte dalla NASA. Gli attuali dispositivi [39] sono caratterizzati da pesi ed ingombri di gran lunga inferiori, da potenti processori *embedded*, da autonomie molto più estese, tempi di ricarica ridotti al minimo e dalla possibilità di adattare la luminosità degli oggetti virtuali in base alle caratteristiche dell'ambiente considerato. Tali sviluppi tecnologici hanno portato i dispositivi HMD ad essere visti sotto una nuova luce, pienamente rispondente ai requisiti di impiego di tali strumenti nei contesti operativi delle torri di controllo aeroportuali [40].

Questa fiducia nei confronti dell'AR è stata presto ripagata dalla tecnologia stessa mediante il successo conseguito da RETINA (Resilient Synthetic Vision for Advanced Control Tower Air Navigation Service) [41], un progetto europeo di ricerca esplorativa finanziato dalla Commissione Europea nell'ambito del programma SESAR (Single European Sky ATM Research) ed a cui lo scrivente ha preso parte mediante un contratto di assegno di ricerca. Il punto di forza di tale ricerca

è stato quello di condurre uno studio che ha portato la partnership (coordinata dall'Università di Bologna) coinvolta nei due anni di progetto ad investigare l'impiego dell'AR nelle torri di controllo aeroportuali mediante un metodo scientifico, ovvero attraverso un processo mirante a giustificare ogni *step* di avanzamento della ricerca, sin dalle fasi iniziali di scelta della tecnologia AR orientata all'utenza [42], passando per la valutazione dell'impatto che l'impiego di tale tecnologia produce sui *task* svolti dai controllori [43], passando per numerosi brainstorming con gli *stakeholders* (controllori ed esperti di traffico aereo) per decidere quali informazioni mostrare e la simbologia da utilizzare nelle interfacce AR, fino ad arrivare all'implementazione ed alla validazione delle stesse in un ambiente di torre simulato in laboratorio. Persino la motivazione alla base dello studio su cui si è incentrato il progetto RETINA trova solida giustificazione: non deriva, tout court, dalla semplice volontà di portare l'AR nelle torri di controllo aeroportuali, ma trova fondamento nell'evoluzione del *modus operandi* nelle stesse, dove lo sguardo del controllore è stato progressivamente spostato dall'ambiente reale che si trova fuori dai vetri della torre di controllo verso le interfacce pc e *monitor* all'interno della stessa (*head-down equipment*). Le interfacce classiche bidimensionali, oltre a risultare meno efficaci degli strumenti AR in termini di precisione della stima della distanza di oggetti lontani [44], riducono la *situational awareness* del controllore perché lo forzano ad alternare continuamente lo sguardo tra il *display head-down* e l'ambiente reale fuori dai vetri della torre di controllo. Al contrario, le tecnologie AR offrono l'opportunità di trasferire tutta l'informazione che è attualmente disponibile nella *head-down equipment* verso una visione cosiddetta "a testa alta" (*head-up view*), che porta il controllore a guardare costantemente fuori dai vetri di torre, mediante l'uso di informazione digitale correttamente allineata (registrata) nello spazio e nel tempo con il mondo fisico reale, indipendentemente dalle condizioni di visibilità e dalla distanza (Fig.15).

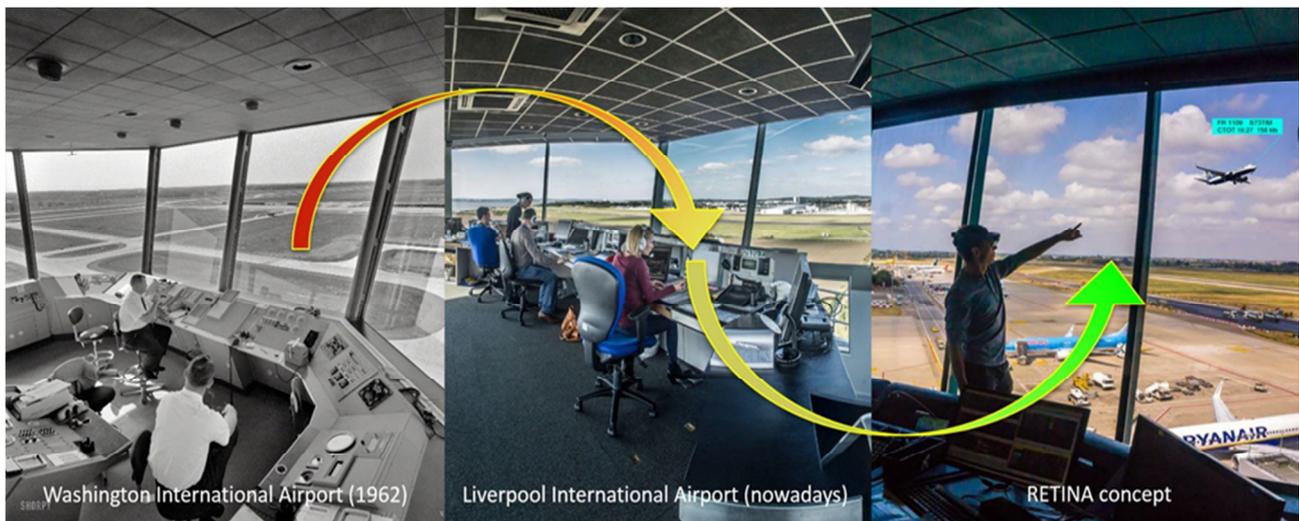


Fig. 15: processo di evoluzione delle torri di controllo: l'introduzione dell'automazione ha trasferito lo sguardo del controllore sulle interfacce "head-down" dentro le torri di controllo, mentre RETINA lo riporta nuovamente sul mondo reale fuori dai vetri di torre.

Il processo di validazione di RETINA ha riguardato due tipi di tecnologia AR: HMD e SD, quest'ultimi immaginati come grandi schermi trasparenti coincidenti con i vetri di torre. Una lunga

campagna di *human-in-the-loop- simulations* in laboratorio ha coinvolto i controllori del traffico aereo, che sono stati inseriti all'interno di un ambiente virtuale immersivo rappresentato dal Sistema di Proiezione CAVE, simulante con altissimo grado di realismo l'ambiente della torre di controllo dell'aeroporto di Bologna. Per riprodurre fedelmente l'operato reale, ai controllori è stata fornita una replica dell'interfaccia *head-down* presente nelle torri di controllo e sono stati messi in comunicazione radio con una postazione "*pseudopilot*", avente il ruolo di svolgere le veci dei piloti e quindi di movimentare gli aerei a seguito delle istruzioni impartite dal controllore, aggiornando così l'ambiente virtuale in cui il controllore era immerso. Al fine di confrontare fra loro i risultati ottenuti, sono stati eseguiti tre tipi fondamentali di test, ognuno caratterizzato a sua volta da diverse condizioni di traffico e di visibilità (Fig.16): un primo test ha considerato la situazione di stato di fatto, cioè l'attuale modo di lavorare dei controllori (senza AR); un secondo test ha considerato l'uso dell'AR per mezzo della tecnologia HMD (ed in particolare della prima versione del dispositivo Microsoft HoloLens) ed un terzo test ha considerato l'impiego di SD. Quest'ultima soluzione è stata simulata via software all'interno dell'ambiente virtuale immersivo rappresentato dal Sistema di Proiezione CAVE, in quanto tale tecnologia, che potrebbe essere pensata coincidere con *head-up display* dai formati molto ampi (delle stesse dimensioni dei vetri di torre), non è ancora in commercio al giorno d'oggi.



Fig. 16: Durante il processo di validazione di RETINA sono state considerate tre differenti modalità operative: la situazione di stato di fatto, cioè l'attuale modo di lavorare dei controllori senza AR (a sinistra), l'impiego dell'AR per mezzo dei display spaziali coincidenti con i vetri di torre (al centro) e l'uso dell'AR per mezzo della tecnologia HMD (a destra).

I risultati delle validazioni hanno mostrato che, sostituendo le interfacce *head-down* con tecnologie AR, RETINA riduce il carico di lavoro dei controllori ed aumenta la *situational awareness* e le prestazioni degli stessi [45]. Inoltre, l'AR presenta un impatto positivo non solo sull'operatività del controllore, ma anche sull'efficienza aeroportuale e, di conseguenza, sui costi e sulle tematiche di impatto ambientale. Infatti, attualmente, al diminuire della visibilità, vengono applicate alcune procedure specifiche al fine di mantenere dei livelli di sicurezza opportuni nella gestione del

traffico. Tali procedure sono basate su limitazioni nel distanziamento degli aerei e nella percorribilità di alcune aree aeroportuali, che pregiudicano in modo estremamente negativo l'efficienza aeroportuale (cioè il numero di movimenti orari). Al contrario, l'AR garantisce ai controllori una visione sintetica del traffico e del layout aeroportuale anche in condizioni di scarsa visibilità, mantenendo la *situational awareness* ed il carico di lavoro dei controllori agli stessi livelli propri delle condizioni di buona visibilità (Fig.17).

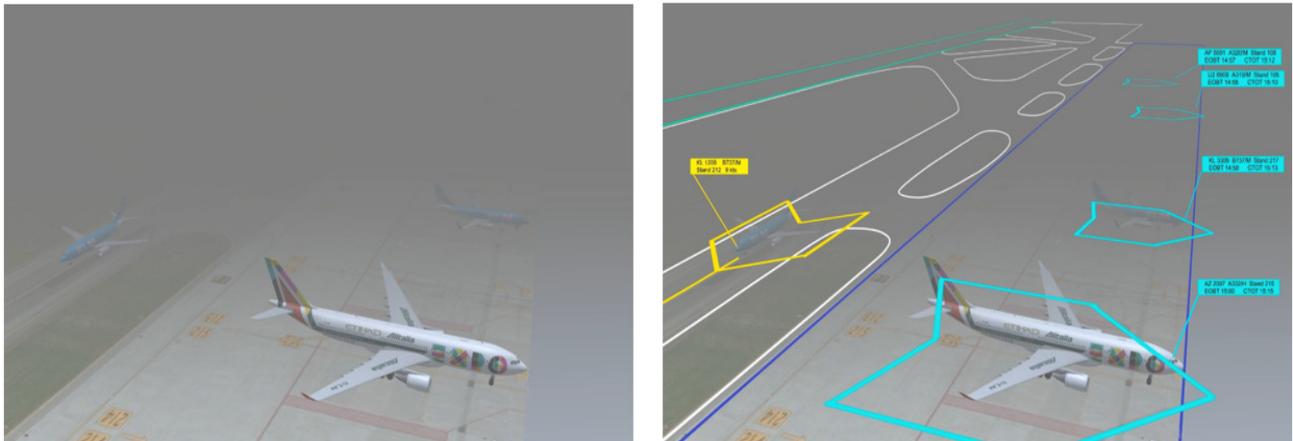


Fig. 17: Vista fuori dai vetri di torre in condizioni di scarsa visibilità: confronto tra assenza di Realtà Aumentata (sinistra) e uso della stessa (destra). L'AR applicata in condizioni di scarsa visibilità permette di mantenere la *situational awareness* ed il carico di lavoro dei controllori agli stessi livelli propri delle condizioni di buona visibilità.

Grazie a ciò, la maggioranza delle limitazioni presenti in scarsa visibilità (dovute all'impossibilità da parte del controllore di vedere l'ambiente reale fuori dai vetri di torre) possono essere rimosse (come le limitazioni sull'uso di *intermediate holding points*, di vincoli sul numero di partenze gestite contemporaneamente, di vincoli sul numero di operazioni di *pushback* da *stands* contigui), con conseguente effetto positivo sull'efficienza e quindi anche sui costi e sull'impatto ambientale. I risultati ottenuti da RETINA sono stati confermati anche in altre ricerche [25] [46]. L'ultimo lavoro citato rappresenta uno studio portato avanti dal DLR, centro nazionale di ricerca aerospaziale tedesco, nell'ambito del progetto "Adaptive Controller Support Components" (AdCoSCo). In tale contesto, le sperimentazioni eseguite mediante *human-in-the-loop-simulations* hanno visto il coinvolgimento di sei controllori atti a valutare l'impiego dell'AR in un ambiente di torre simulato. Dai test si è evinto che l'AR apporta un effettivo miglioramento nelle *human performances* in termini di riduzioni del carico di lavoro (*workload*), incremento della *situational awareness* e di una migliore accessibilità dell'informazione, presentata esattamente nel momento in cui si ha necessità di essa.

I risultati molto positivi conseguiti hanno portato RETINA ad uscire dall'ambiente di laboratorio, ovvero dal porto della ricerca esplorativa, per farlo approdare in quello della ricerca industriale, andando a portare la soluzione HMD nell'ambiente reale della torre di controllo. Per quanto riguarda gli SD, sebbene abbiano un potenziale superiore agli HMD grazie ad una migliore ergonomia (nulla da indossare) ed un maggiore campo visivo (coincidente teoricamente con quello umano), la soluzione degli schermi trasparenti coincidenti con i vetri di torre è ancora lungi

dall'essere implementata, poiché a livello costruttivo non esistono in commercio *head-up display* così ampi ed inoltre, non essendo ad uso individuale, lo scenario “multiuser” è ancora tematica di ricerca. Si pensi infatti al caso in cui due controllori in posizioni diverse osservino il medesimo aereo: i dati digitali ad esso relativi comparirebbero in due punti diversi sui vetri di torre, il che creerebbe confusione in quanto ciascun controllore vede entrambe le informazioni, anche quelle destinate al suo collega. Ne segue che, a differenza degli HMD, il filone della ricerca industriale intrapreso da RETINA non ha incluso la tematica degli SD.

Altro capitolo molto interessante è rappresentato dalla ricerca inerente all'applicazione dell'AR alle torri di controllo remote.

A partire dai primi anni del nuovo millennio, con lo sviluppo di tecnologie di proiezione dei video ad alta risoluzione, algoritmi di compressione delle immagini, comunicazioni a banda larga e sistemi elettronici ad alto potere computazionale, il paradigma del controllo di un aeroporto da remoto ha preso sempre più piede. Specificatamente, tale concetto è stato introdotto dal DLR nel 2002 e descrive un centro di controllo che gestisce il traffico aeroportuale basando la sorveglianza su immagini video e quindi non più sulla visione fuori dai vetri di torre, come avviene nelle classiche strutture con vista panoramica sull'ambiente aeroportuale circostante. In tale modo, le torri di controllo remote possono essere localizzate fuori dal sedime dell'aeroporto da gestire, volendo anche dall'altra parte del mondo. Così, predisponendo nell'aeroporto da controllare un sistema di telecamere che coprano in modo efficiente tutte le aree di movimentazione degli aeromobili, il centro di controllo può gestire il traffico aereo da remoto, basandosi sulle immagini video in *real-time* derivanti dalle telecamere in loco (Fig.18).



Fig. 18: prima torre di controllo remota al mondo implementata in Svezia nel 2015: l'aeroporto di Örnsköldsvik/Gideå (OER/ESNO) è operato dal centro di controllo remoto situato a Sundsvall/Midlanda (SDL/ESNN).

Le specifiche fondamentali affinché tale sistema operi in sicurezza riguardano la qualità delle immagini video e la latenza delle stesse, entrambe soggette a vincoli molto stringenti affinché i rispettivi valori si mantengano sempre entro delle soglie di sicurezza [47]. L'applicazione dell'AR nelle torri di controllo remote (Fig. 19) vede molta ricerca a riguardo.



Fig. 19: applicazione dell'AR ai pannelli video di una torre di controllo remota (fonte: [ref. 12](#))

Uno dei primi studi è stato condotto dal DLR nell'ambito del progetto RapTOr (Remote Airport Tower Operation Research) [48]. In esso è stato implementato un centro di controllo remoto caratterizzato da visione panoramica ed interfacce AR, atto a gestire l'aeroporto sperimentale di Braunschweig, situato nel centro di ricerca del DLR. Successivamente, i risultati conseguiti da RapTOr sono stati ulteriormente sviluppati nel progetto RAiCe (Remote Airport Traffic Control Center), avente lo scopo di sorvegliare e controllare da remoto più aeroporti di piccola taglia. In RAiCe il centro di controllo remoto implementato in RapTOr è stato integrato con nuove funzionalità [49], quali la visualizzazione dei dati di traffico provenienti dall'A-SMGCS (Advanced Surface Movement Guidance and Control System), la capacità di processamento delle immagini in *real-time* per la rilevazione degli aerei in atterraggio con un tracking continuo nel tempo (e non ad istanti discreti coincidenti con gli intervalli di campionamento radar) e la possibilità di effettuare il

replay del panorama completo di 180°. L'esperienza pressoché decennale del DLR nella tematica delle torri di controllo remote è riportata in un libro pubblicato nel 2016, dove viene effettuata un'ampia trattazione sullo stato dell'arte della relativa ricerca, nonché sugli aspetti tecnici di progettazione, implementazione e validazione dei centri "videopanoramici", che, da un unico luogo, possono controllare e gestire più aeroporti, migliorando l'efficienza, nonché riducendo i costi di esercizio rispetto alle tradizionali strutture di controllo poste all'interno dei sedimi aeroportuali [50]. L'abbattimento dei costi e l'incremento dell'efficienza di gestione sono aspetti che riguardano particolarmente gli aeroporti di taglia medio-piccola, con un basso numero di movimenti o stagionali. In tali contesti non c'è convenienza a mantenere in loco il servizio di controllo e gestione del traffico aereo e la tendenza è quella di dismettere i relativi servizi dal sedime aeroportuale e praticarli da remoto, seguendo la stessa tendenza, ormai da anni consolidata, delle stazioni ferroviarie impresenziate. In linea con ciò, gli studi del DLR, leader nel settore della ricerca sulle torri remote, si sono soprattutto concentrate sui piccoli aeroporti regionali [51] [52], di importanza strategica per i territori che li ospitano in quanto tali infrastrutture sono prese di mira dal traffico di massa delle compagnie low-cost, producendo quindi indotto di servizi commerciali ed ausiliari che favoriscono un introito economico considerevole per le amministrazioni locali.

L'integrazione della tecnologia AR nei contesti delle torri remote è un campo di ricerca atto a produrre benefici non solo in termini di efficienza e costi, ma anche in termini di sicurezza e training [53]. In tale studio riferito, gli autori hanno implementato un prototipo sperimentale di torre remota integrata con interfacce AR, atto all'analisi di situazioni di rischio, nonché al training dei controllori. La flessibilità del sistema permette anche di studiare, progettare e validare nuove procedure di controllo in condizioni meteo differenti.

Delineate le principali linee di ricerca sull'AR applicata alle torri remote in base agli scopi di utilizzo (controllare più aeroporti da uno stesso centro, training) ed ai benefici stimati (efficienza, costi, sicurezza), è bene infine riportare un ultimo filone di ricerca i cui studi si differenziano per tipologia di videocamere impiegate nell'aeroporto che deve essere gestito da remoto [54] [55]. Nel primo studio il centro di controllo vede l'impiego delle interfacce AR sovrapposte ad immagini video derivanti dalla fusione di camere a spettro visivo (VS) e camere a spettro infrarosso (IR). In Fig.20 sono riportati esempi di fusione in ambiente diurno ed ambiente notturno. Nella seconda ricerca le immagini video derivano invece da telecamere IP, ovvero dispositivi che generano un segnale video in forma digitalizzata e in formato pronto per la trasmissione diretta su rete dati, senza necessità di conversione analogico-digitale, e in grado di essere controllate direttamente tramite la rete dati stessa.

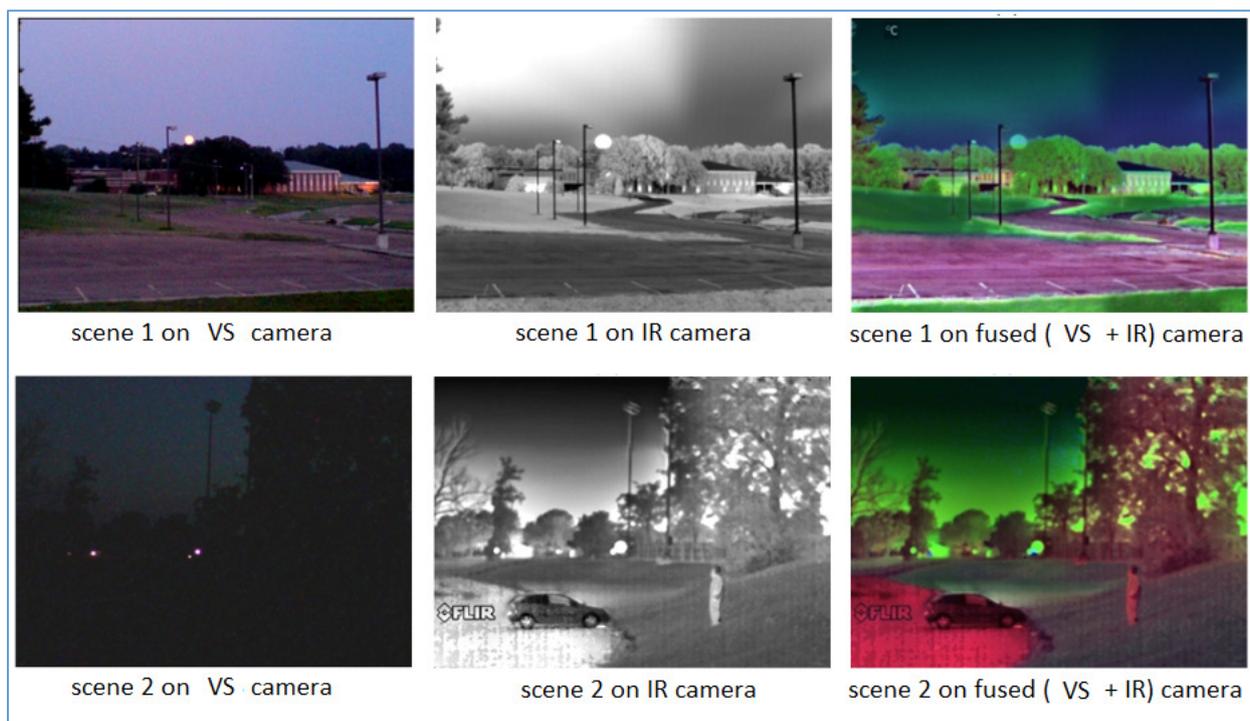


Fig. 20: Esempi di fusione tra camere a spettro visivo (VS) e camere a spettro infrarosso (IR) in ambiente diurno (riga sopra) ed ambiente notturno (riga sotto) (fonte: [ref 13](#))

3.3 Droni e spazio: nuove frontiere per l'Augmented Reality

L'industria dei droni rappresenta oggi un campo di applicazione molto promettente per l'AR. In questo caso non ha senso parlare di *head-up display* o HMD, in quanto non è l'uomo al centro dell'attenzione e quindi non si ha a che fare con schermi "a testa alta" o "montati in testa": ora il termine giusto è quello di "video *see-through AR*", in quanto la visione diretta sul mondo reale è in questo caso rimpiazzata da un'immagine video dello stesso sulla quale viene sovrapposta l'informazione digitale aggiuntiva. In altre parole, il drone riprende la scena dall'alto mediante una o più telecamere, il/i relativo/i video viene/vengono "aumentati" con interfacce AR ed infine il tutto viene osservato da remoto in tempo-reale o meno. Molta ricerca a riguardo tratta soprattutto due macro-aree principali, ovvero quella degli operatori di primo soccorso, come vigili del fuoco, personale sanitario, forze dell'ordine, esercito, protezione civile [56] e *search and rescue* [57], campo di applicazione molto fertile per l'AR come descritto nel precedente capitolo, oppure la macro-area dell'agricoltura di precisione [58]. Così, parallelamente al settore del secondario e del terziario fin qui descritti nel presente e precedente capitolo, la macro-area dell'agricoltura di precisione permette di estendere l'impiego dell'AR anche al settore del primario, completando in questo modo la trattazione dell'AR in tutti e tre i settori dell'economia, ovvero agricoltura, industria e servizi.

Non ancora appagata dall'essere ambita da ogni settore produttivo, l'AR supera i confini del nostro mondo e si proietta nello spazio [59]. Il primo approccio della Realtà Aumentata nel mondo

dell'esplorazione spaziale è avvenuto a bordo della Stazione Spaziale Internazionale nel 2015, attraverso il progetto Sidekick, una collaborazione tra NASA e Microsoft. Insieme a quello che era un normale carico di rifornimento, vennero consegnati agli astronauti anche un paio di Microsoft HoloLens HMD, utilizzati per scopi manutentivi in collaborazione da remoto: mediante tali visori collegati in live video *streaming*, gli astronauti mostravano il *task* in cui erano impegnati ai tecnici della stazione di terra e quest'ultimi inviavano informazioni dettagliate su come svolgere il *task*, andando direttamente a disegnare le istruzioni che gli astronauti vedevano in forma di informazione digitale aggiuntiva sovrapposta all'ambiente in cui operavano (Fig.21).

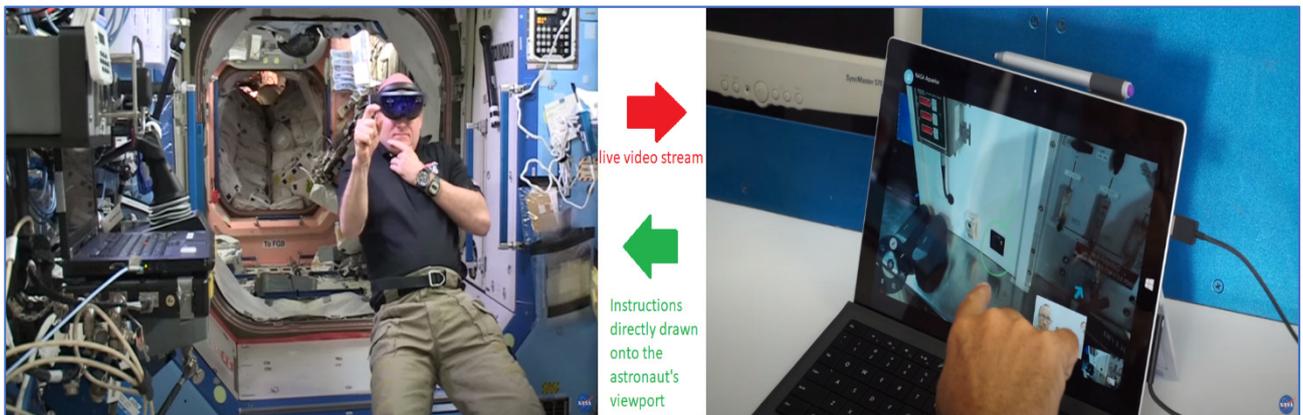


Fig. 21: collaborazione da remoto per scopi manutentivi tra astronauti indossanti Microsoft HoloLens HMD (a sinistra) e tecnici della stazione di terra (a destra) nell'ambito del progetto Sidekick, collaborazione tra NASA e Microsoft (fonte: [ref 14](#)).

4 Stato dell'arte delle tecnologie di Realtà Aumentata

A seguito dei precedenti capitoli riguardanti *i campi di applicazione della Realtà Aumentata* e *lo stato dell'arte della ricerca sulla Realtà Aumentata applicata al campo aerospaziale*, l'iter di trattazione prosegue con l'analisi dello stato dell'arte delle tecnologie AR in termini delle due componenti fondamentali che le costituiscono, ovvero *display* e *tracking*.

Il modulo di tracking rappresenta il cuore dei sistemi AR (Fig. 22): esso acquisisce le immagini mediante videocamere standard o di profondità (di quest'ultime si parlerà più avanti nella sezione 4.2 dedicata al tracking) e calcola posizione ed orientamento dell'utente, e dunque del suo punto di vista, in tempo-reale. Successivamente, sulla base di tale computazione, il modulo di *rendering* combina l'immagine originale acquisita dal sensore di tracking con i dati/modelli virtuali e riproduce l'immagine "aumentata" sul *display*. Questo descritto è in sintesi il processo di funzionamento di ogni sistema AR. A conferma di ciò, si riporta in Fig. 23 il ciclo di vita di un frame (ovvero l'istante di tempo tra due "refreshes" di *display* successivi) del Microsoft Hololens See-Through HMD e quindi il suo funzionamento: rispetto alla figura originale presa dal sito Microsoft, sono stati introdotti tre riquadri (di colore giallo, ocra e ciano) aventi lo scopo di suddividere l'intero funzionamento nei tre moduli funzionali introdotti in Fig. 22, ovvero tracking, CPU/GPU e *display*.

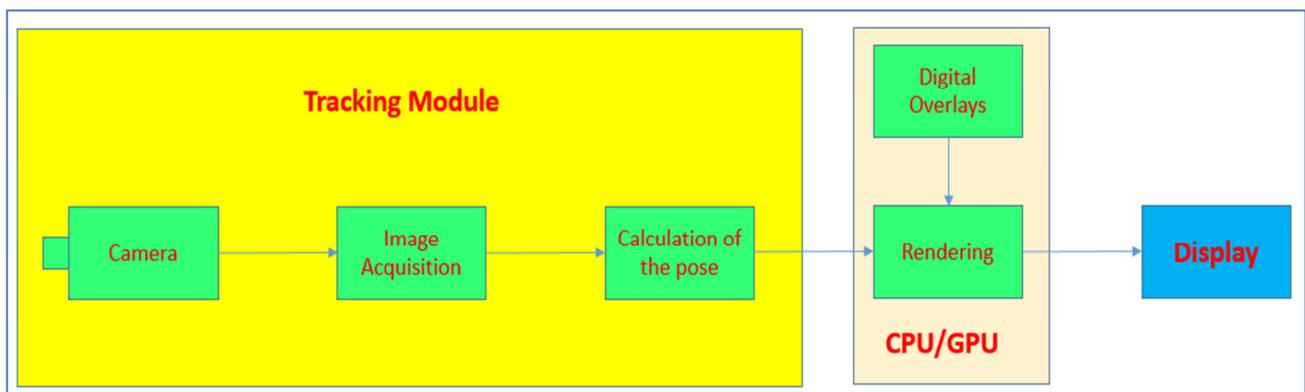


Fig. 22: sintesi del processo di funzionamento di ogni sistema AR

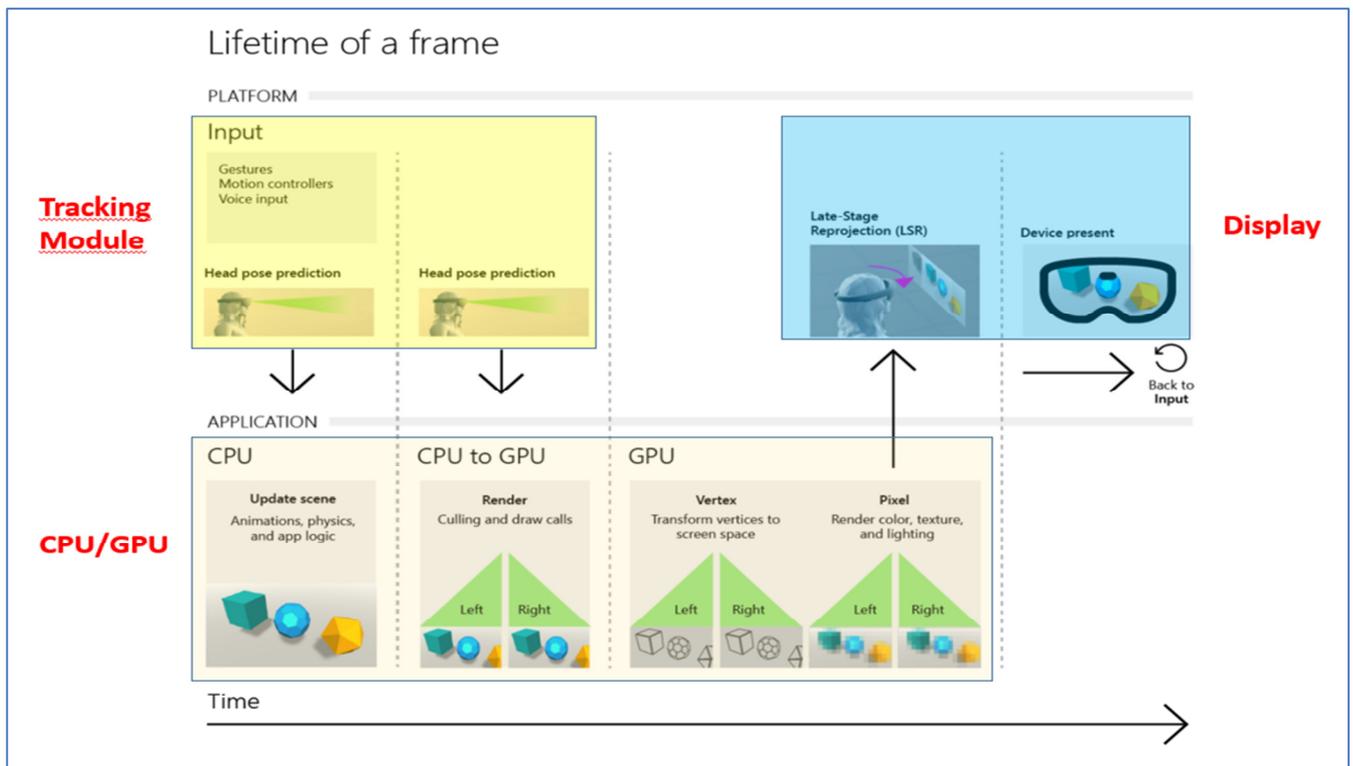


Fig. 23: funzionamento del Microsoft HoloLens See-Through HMD, (fonte: [ref 15](#)).

Nei sistemi AR più avanzati, dal sensore al *display* intercorrono al massimo 50 ms: l'acquisizione impiega generalmente 17 ms ed il *rendering* fino a 23 ms, lasciando solamente 10 ms per il calcolo della configurazione dell'utente (posizione e orientamento) [7].

Seguono le sezioni relative all'analisi dei *display* e dei moduli di tracking. Il terzo modulo funzionale fondamentale relativo al *rendering* non verrà trattato in questa sede in quanto l'analisi delle CPU/GPU esula dal focus principale riguardante i sistemi AR.

4.1 Display

Il panorama tecnologico degli AR *display* è costituito da molteplici soluzioni, quali schermi LCD, OLED, DLP, LCoS (Liquid Crystal on Silicon), OLED-on-Silicon Microdisplays e MicroLEDs. I *display* LCD e OLED sono impiegati per formati più ampi, mentre i microdisplay (LCoS, DLP, OLED) sono generalmente utilizzati per i dispositivi indossabili HMD. Con il passare del tempo, lo sviluppo di processori sempre più potenti ha prodotto un netto miglioramento nel processo di generazione delle immagini, sia in termini di nitidezza, che di colore. D'altro canto però, tali sviluppi comportano maggiori ritardi in termini di latenza *end-to-end* sulla *pipeline* tra il processo di generazione delle immagini e quello del trasferimento dell'*output* al *display*, in quanto il miglioramento qualitativo delle immagini si paga inevitabilmente a prezzo di un incremento del tempo computazionale per generarle, nonché di quello necessario per trasferirle lungo la *pipeline*. Specificamente, le tecnologie AR inviano dati al *display* in modo sequenziale, secondo le tecniche

raster-line (Fig. 24) o *matrix-element* derivanti dal metodo *raster scan* [60], in cui il *display* deve ricevere un'intera immagine prima di poter dare il via alla presentazione dei pixel della stessa, introducendo così il *display* stesso un ritardo che è pari al lasso di tempo che intercorre fra la ricezione di interi *frames* di immagini o più lungo, nel caso il metodo *raster scan* sia integrato con la tecnica del *field sequential color* (FSC) [61], attraverso cui la percezione di un colore da parte dell'utente è frutto di una successione di immagini monocolori (Fig.25).

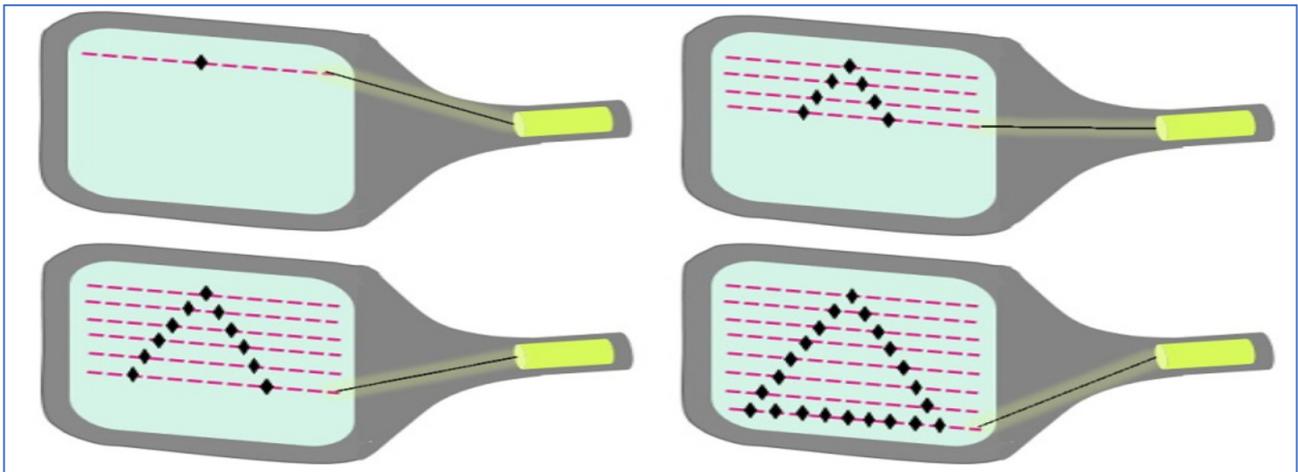


Fig. 24: tecnica *raster-line* per la visualizzazione di un'immagine a schermo. Il fascio di elettroni costruisce l'immagine punto per punto procedendo per linee (fonte: [ref. 16](#)).



Fig. 25: tecnica del *field sequential color* (FSC): la percezione di un colore da parte dell'utente è frutto di una successione di immagini monocolori (fonte: [ref. 17](#)).

Il metodo *raster scan* non è adatto per quelle applicazioni AR che richiedono dei bassi ritardi di registrazione dei dati digitali con il mondo reale visto dall'utente, tantomeno in caso di integrazione con la tecnica FSC. Pertanto, basandosi su tale connubio, i *display* DLP risentono negativamente dall'essere caratterizzati da elevate latenze nella presentazione dell'*output* su schermo. I *display* FSC LCoS (tecnologia su cui sono basati i dispositivi HMD Microsoft HoloLens) rispondono meglio in termini di latenza rispetto ai DLP, ma anch'essi devono attendere la ricezione di un intero *frame* prima di poter mostrare il primo colore su schermo [7]. La tecnologia LCoS rappresenta un ibrido

tra i *display* DLP ed LCD: la proiezione viene riflessa su un chip proprio come avviene nei DLP ma i colori vengono creati in maniera analoga alla tecnologia LCD, ovvero per mezzo di pannelli a cristalli liquidi.

Sulla scia delle tecnologie LCD ed OLED, i *display* microLED, in cui ogni pixel è composto da matrici di LED microscopici, sono considerati la tecnologia AR del futuro. I *display* basati sul paradigma della proiezione, come LCoS e DLP, necessitano di una sorgente di luce aggiuntiva per produrre immagini su schermo, mentre la soluzione rappresentata dai MicroLED è auto-emissiva, fornendo significativi miglioramenti in termini di peso, risoluzione, luminosità, contrasto, consumo energetico e latenza. Un confronto fra le caratteristiche funzionali principali e costruttive dei *display* LCD, OLED e microLED è riportato in Fig.26, dove il dato più significativo in termini di differenze riguarda la latenza, ovvero il *response time*: microLED batte OLED di tre ordini di grandezza, nonché LCD (e quindi anche LCoS e DLP) di sei ordini di grandezza.

Display technique	LCD	OLED	Micro LED
Technological type	Back light/LED	Self-Illumination	Self-Illumination
Contrast ratio	5,000:1	∞	∞
Lifetime	Medium	Medium	Long
Response time	ms	μ s	ns
Operating temperature	-40 to 100°C	-30 to 85°C	-100 to 120°C
Cost	Low	Medium	High
Energy consumption	High	Medium	Low
Viewing Angle	Low	Medium	High
PPI (Wearables)	Max 250 ppi	Max 300 ppi	Above 1500 ppi
PPI(VR)	Max 500 ppi	Max 600 ppi	Above 1500 ppi

Fig. 26: confronto fra le caratteristiche funzionali principali dei *display* LCD, OLED e microLED (fonte: [ref 18](#)).

Accanto alla soluzione dei microLED, un'ulteriore linea di ricerca molto promettente è quella condotta dal Massachusetts Institute of Technology (MIT) dove lo studio è rivolto alla sperimentazione di *display* trasparenti costituiti da nano-particelle che selettivamente diffondono luce ad una lunghezza d'onda di progetto. Tale impiego comporterebbe notevoli benefici in termini di semplicità e bassi costi di realizzazione, ampi angoli di visione e scalabilità verso grandi formati [62].

4.2 Tracking

Come detto nel capitolo introduttivo, il ritardo di registrazione delle interfacce AR con il mondo reale è un fattore che va limitato a seconda dell'applicazione che si va considerando e, più in particolare, è inversamente dipendente dall'entità dei movimenti traslatori e rotatori che il contesto specifico di applicazione dell'AR impone all'utente: ad esempio, considerando i *task* svolti da un pilota di aerei militari e da un operatore logistico in un magazzino, risulta evidente che il primo caso richiede ritardi di registrazione molto più bassi rispetto all'altro, in quanto i due contesti comportano velocità di rotazione della testa con picchi di 2000 (in alcune situazioni speciali [7]) e di 50 gradi al secondo rispettivamente. Ciò si traduce in una diversa efficienza degli algoritmi di tracking che governano i sistemi AR e/o in una differente qualità sensoristica utilizzata.

Gli HMD generalmente integrano al loro interno i moduli di tracking ed il calcolo della posizione ed orientamento dell'utente viene effettuato sulla base della configurazione nello spazio 3D del dispositivo indossato, che, nei casi più avanzati, viene valutata attraverso sensori inerziali (accelerometri e giroscopi) e/o dati provenienti da sensori di localizzazione (magnetometri e GPS), e/o dati provenienti da connessioni Wi-Fi con dispositivi di telefonia mobile e relative apps. Le tecniche base e più datate per effettuare il tracking del dispositivo non usano la sensoristica descritta poco sopra, ma dei “*fiducial markers*” predefiniti, che corrispondono ad oggetti posti nello spazio 3D con la funzione di rappresentare dei punti di riferimento per telecamere integrate negli HMD, atte a riconoscere gli spostamenti e le rotazioni rispetto a tali punti di riferimento, che possono essere costituiti da semplici codici QR (Fig. 27).



Fig. 27: tracking per mezzo dei fiducial markers rappresentati da codici QR: telecamere integrate nell'HMD sono atte a riconoscere gli spostamenti e le rotazioni rispetto a tali punti di riferimento collocati nello spazio attorno all'utente (fonte: [ref 19](#)).

In sintesi, in caso di uso di tecniche basate su *fiducial markers*, riferendosi senza perdita di generalità al caso di *markers* rappresentati da codici QR, il processo di tracking del dispositivo e quindi dell'utente può essere schematizzato dalla seguente successione di operazioni, con ciclo iterativo dalla seconda in poi (Fig. 28):

1. Disposizione dei *fiducial markers* nello spazio 3D;
2. Individuazione dei *markers* all'interno di ogni *frame* catturato dalla telecamera mediante tecniche di *image processing*:
 - a. analisi di un *frame* per identificare i contorni squadrati esterni dei codici QR che possono essere candidati a rappresentare dei *markers* (Fig.28 - 2a); in caso di riconoscimento, passaggio al punto b, altrimenti analisi del frame successivo;
 - b. trasformazione delle prospettive per disporre i probabili candidati nella loro forma canonica (Fig.28 - 2b);
 - c. suddivisione dei candidati in celle per verificare le posizioni dei colori bianchi e neri, conversione in matrici di bit (colore nero rappresentato da bit 0 e colore bianco da bit 1) e confronto della stesse con matrici di bit precaricate che forniscono il modello di paragone per determinare se i candidati appartengono ad uno specifico "dizionario" (Fig.28 - 2c). In caso di riconoscimento, passaggio al punto 3, altrimenti ritorno al punto a.
3. Stima della configurazione dei *markers* (Fig.28 - 3);
4. Trasformazione di coordinate e stima della configurazione della telecamera e quindi del dispositivo, ovvero dell'utente.

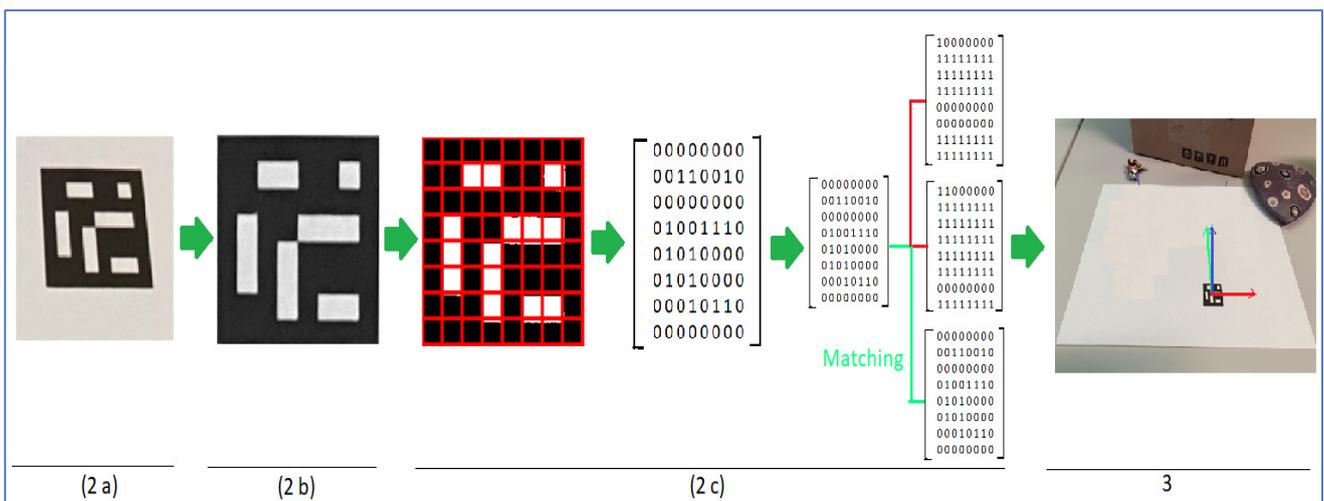


Fig. 28: determinazione della configurazione del dispositivo nel caso di tracking con *fiducial markers*: 2a) analisi di ogni *frame*; 2b) trasformazione delle prospettive; 2c) suddivisione dei candidati in celle, conversione in matrici di bit e confronto della stesse con matrici di bit precaricate; 3) stima della configurazione dei *markers*. Fonte: [ref 20](#)

Un approccio differente e più innovativo è rappresentato dalle tecniche di localizzazione SLAM (Simultaneous Localization and Mapping), che non fanno uso dei *fiducial markers*. Esse sono state sviluppate per far sì che il tracking non dipenda dalla disposizione di punti di riferimento attorno all'utente e quindi da una iniziale preparazione "ad hoc" dell'ambiente circostante, ma per permettere al dispositivo di creare una mappa del mondo fisico che lo circonda e di localizzare sé stesso all'interno di tale mappa in tempo-reale. Tali tecniche sono nate nel campo della robotica con lo scopo di far navigare i robot in ambienti a loro sconosciuti, senza dover utilizzare *fiducial markers* o l'ausilio di sensori GPS differenziali ad alta precisione, che del resto potrebbero essere solamente impiegati in contesti all'aria aperta, dove è possibile usufruire della localizzazione satellitare. Il principio di funzionamento delle tecniche SLAM si basa essenzialmente sull'integrazione di due componenti, un'unità di misura inerziale, meglio nota col termine IMU (Inertial Measurement Unit) ed una telecamera di profondità. L'IMU è costituita da sensori inerziali quali accelerometri e giroscopi che permettono di valutare rispettivamente le traslazioni e le rotazioni relative ai tre assi coordinati del dispositivo in cui l'IMU è integrata. Le unità di misura inerziali più avanzate includono anche la presenza di magnetometri che permettono di determinare l'orientamento assoluto rispetto al Nord terrestre. Mediante l'aggiunta di tali ulteriori sensori, si parla di IMU a nove gradi di libertà, cosicché i tre gradi di libertà aggiuntivi rispetto ai sei forniti dagli accelerometri e giroscopi sono sfruttati per fornire dati supplementari da integrare con quelli provenienti dai giroscopi in modo tale da incrementare l'accuratezza delle misure (Fig. 29).

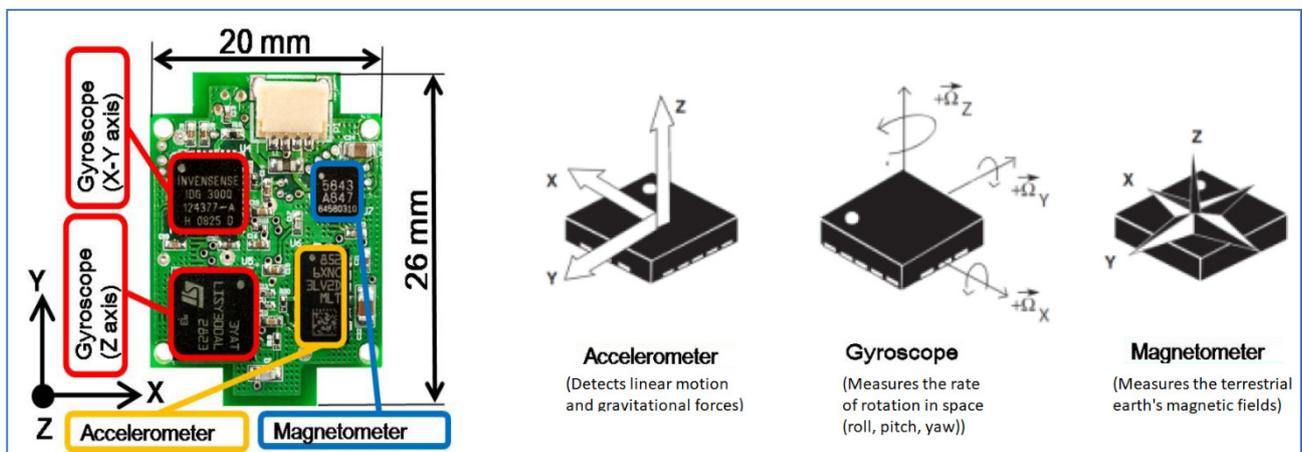


Fig. 29: Esempio di unità di misura inerziale (IMU) a nove gradi di libertà, costituita da accelerometri, giroscopi e magnetometri.
Fonte: [ref 21](#)

Per quanto riguarda il secondo componente fondamentale su cui poggiano le tecniche SLAM, ovvero la telecamera di profondità, anche detta "time-of-flight camera", il suo principio di funzionamento si basa sull'emissione di fasci codificati di luce infrarossa che colpiscono gli oggetti del mondo fisico circostante e vengono riflessi dagli stessi, tornando indietro verso la telecamera di profondità. Così, mediante il calcolo del tempo impiegato da ogni segnale luminoso emesso nel

ritornare (Fig. 30), la telecamera di profondità è in grado di misurare la distanza degli oggetti illuminati in modo da creare una immagine 3D dell'ambiente scansionato [63] [64] (Fig. 31).

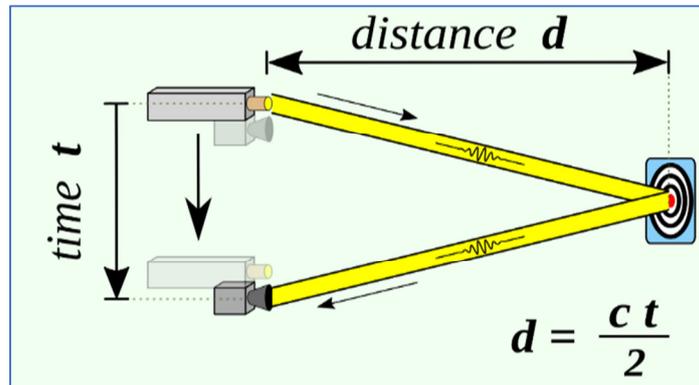


Fig. 30: mediante il calcolo del tempo impiegato da ogni segnale luminoso emesso nel ritornare, la telecamera di profondità è in grado di misurare la distanza degli oggetti illuminati. Fonte: [ref_22](#)

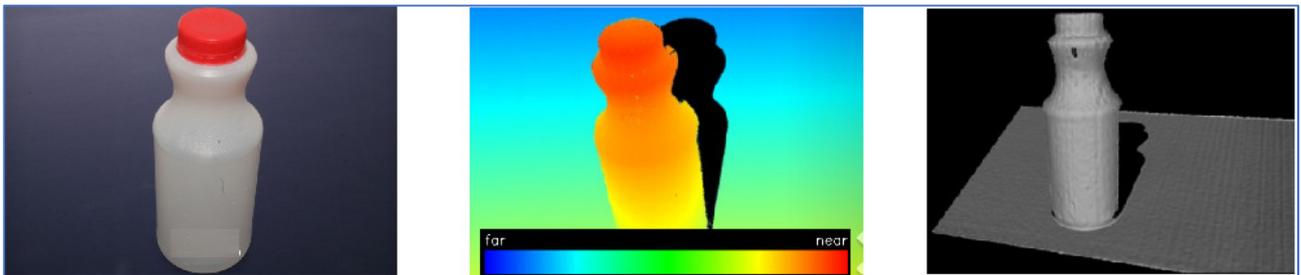


Fig. 31: telecamera di profondità in azione: flacone collocato nello spazio 3D su cui la sorgente invia fasci di luce codificati (sinistra); acquisizione dei fasci di luce codificati riflessi dal flacone e creazione della mappa di profondità in cui ad ogni pixel è associata una distanza (centro); ricostruzione 3D dell'ambiente scansionato (destra). Fonte: [ref_23](#)

La mappatura e ricostruzione 3D del mondo fisico (*spatial mapping*) effettuate dalla *time-of-flight camera* si raffinano sempre più con l'aumentare del numero di segnali di ritorno analizzati, in quanto le computazioni delle distanze vengono memorizzate e perfezionate dai nuovi dati che sopraggiungono mediante algoritmi genetici basati sul paradigma della “sopravvivenza della specie più forte” (*survival of the fittest*), atti ad eliminare quei dati storici che non sono più utili perché sostituiti da acquisizioni più attendibili, come effettuato con l'algoritmo ORB-SLAM [65]. Analizzato il comportamento di ciascuno dei due componenti fondamentali sull'integrazione dei quali si basano le tecniche SLAM, si comprende il principio di funzionamento generale delle stesse: la telecamera di profondità viene utilizzata per costruire la mappa dell'ambiente circostante, mentre

l'IMU per identificare i movimenti dell'utente e localizzarlo all'interno di tale mappa. Un completo ed esauriente esame della ricerca condotta in passato sulle tecniche SLAM, nonché un'analisi dei relativi sviluppi futuri è condotto in [66]. Numerosi studi sono rivolti all'analisi dell'integrazione SLAM-GPS per ambienti all'aperto. Tale connubio ha lo scopo di incrementare il grado di robustezza degli algoritmi di mappatura 3D [67] [68] [69] e molti lavori sono incentrati sul campo di applicazione dei droni [70] [71].

Per quanto riguarda gli SD, al contrario degli HMD, generalmente non dispongono di sensori di tracking integrati. Gli *head-up display* nei *cockpit* degli aerei, rappresentanti più famosi dei *display* spaziali, basano il proprio funzionamento su tre moduli separati: il sensore di tracking (1) cattura i movimenti e le rotazioni della testa del pilota, comunica tali informazioni al computer di bordo (2), che a sua volta calcola il posizionamento corretto dei dati digitali secondo la prospettiva visuale del pilota ed aggiorna il *display* AR (3) con nuove rappresentazioni. Il paradigma di funzionamento è quindi lo stesso di quello degli HMD rappresentato in Fig. 22 di cui sopra: il modulo di tracking fornisce dati in *input* alla CPU/GPU che renderizza opportunamente gli AR *overlays* sul *display*. Le uniche differenze rispetto agli HMD potrebbero risiedere nel modulo di tracking. L'uso del condizionale è d'obbligo in quanto, in caso di tecniche basate su *fiducial markers* disposti sull'utente, tutto continua a funzionare esattamente come descritto per i dispositivi indossabili: i movimenti dell'utente davanti al sensore di tracking vengono rilevati sulla base della stima della configurazione dei markers (in questo contesto dinamici, mentre con gli HMD statici) visti dalla telecamera (ora statica, mentre prima dinamica in quanto integrata nel dispositivo indossabile). Tutto cambia in relazione al tracking che non fa uso di *fiducial markers*, in quanto in questo caso non sono applicabili le tecniche SLAM caratterizzate dall'integrazione IMU/telecamera di profondità che sono state considerate nel caso degli HMD: a differenza di quest'ultimi, gli SD non seguono fisicamente l'utente nei suoi movimenti perché sono fissi e quindi non ha senso considerare l'IMU. Ciò che invece continua ad avere valenza è la telecamera di profondità. In sintesi, il paradigma di funzionamento generale è il seguente. L'utente si muove nel campo di copertura visiva della telecamera di profondità, quest'ultima acquisisce i pixels ed associa ad essi una distanza dipendente dalla lontananza di ciò che in essi è rappresentato mediante il concetto di *time-of-flight* visto in Fig.30. Successivamente avviene il riconoscimento dell'utente mediante:

1. separazione della scena di primo piano (*foreground*) da quella di sfondo (*background*) e successiva separazione del corpo umano dalla scena di primo piano [72] [73] [74] [75] [76];
2. suddivisione del corpo umano in regioni per l'individuazione dei punti di interesse soggetti a tracking [77] (Fig. 32)

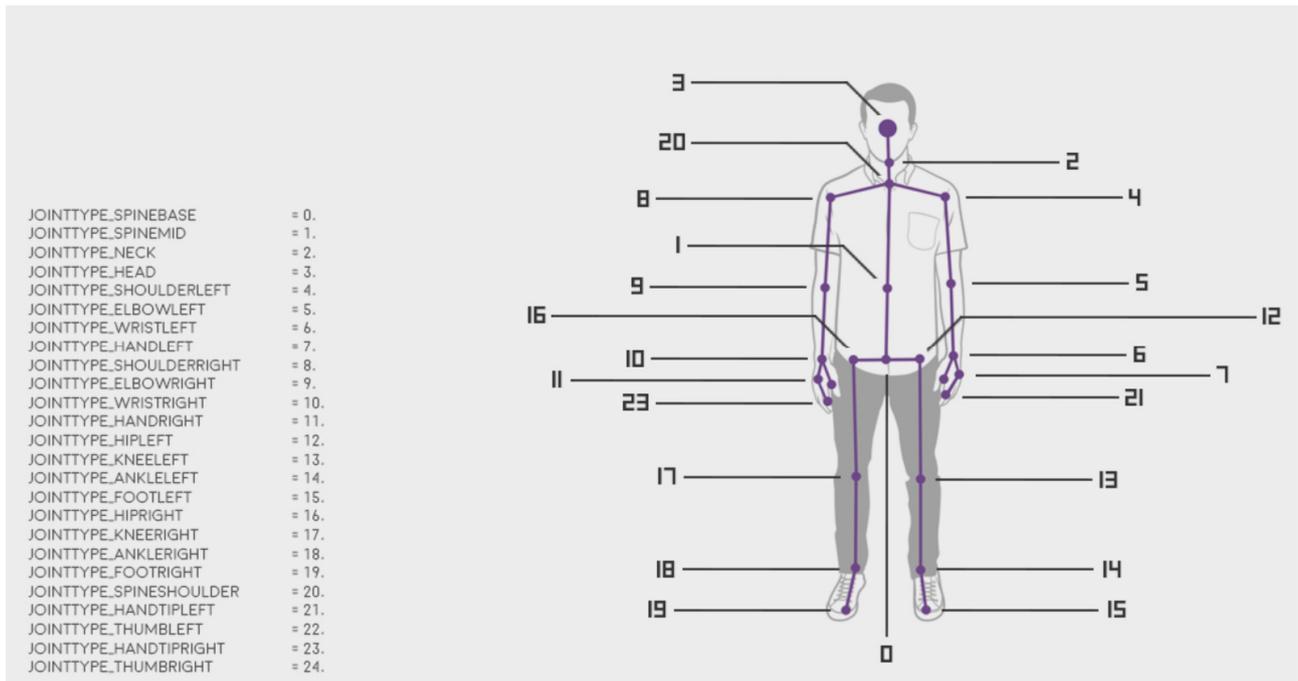


Fig. 32: individuazione dei punti di interesse soggetti a tracking da parte della camera di profondità relativa al sensore di tracking Microsoft Kinect. Fonte: [ref. 24](#)

Per approfondire gli aspetti del paradigma di funzionamento generale sopra riportato, si rimanda la trattazione a [78], dove viene riportata un’esaustiva analisi delle tecniche di tracking del movimento e del corpo umano studiate in passato ed attualmente oggetto di ricerca, nonché una completa rassegna delle camere di profondità attualmente sul mercato, specificandone i vantaggi e gli svantaggi.

5 Sviluppi di tracking

Come è stato detto nelle prime righe del capitolo precedente, il modulo di tracking rappresenta il cuore dei sistemi AR (Fig. 22). Per tale motivo, la trattazione si focalizza innanzitutto sulle applicazioni di tracking dell'utente, basamento imprescindibile su cui poggiano:

- la navigazione di ambienti virtuali di cabina di aereo nell'ambito del progetto europeo CASTLE (Cabin Systems design Toward passenger wellBeing);
- la registrazione di numerose applicazioni AR sviluppate in ambiente di laboratorio sia nell'ambito del progetto RETINA che durante il corso di dottorato (capitolo 6).

Le applicazioni di tracking sono state inizialmente realizzate mediante il sensore Microsoft Kinect versione 2 (v2). In seguito, l'interfaccia software del Kinect è stata opportunamente modificata per acquisire dati da qualunque hardware operante con un numero minore o uguale di punti di interesse soggetti a tracking e collocati in linea di massima nelle stesse posizioni di quelli del Kinect (Fig.32). Prima di passare alla descrizione delle applicazioni, è opportuno delineare il principio di funzionamento di tale dispositivo, che, per le caratteristiche di qualità-prezzo che lo contraddistinguono, viene largamente utilizzato per scopi di intrattenimento, professionali, nonché di ricerca, costituendo a tutti gli effetti il più famoso sensore di tracking su scala mondiale.

5.1 Il sensore di tracking Microsoft Kinect: funzionamento e proposte di ottimizzazione

Il sensore di tracking Microsoft Kinect v2 è equipaggiato con una telecamera RGB, un'altra di profondità, una sorgente di luce infrarossa e microfoni per il riconoscimento vocale (Fig.33). Le specifiche tecniche fondamentali sono riportate in Fig.34.



Fig. 33: sensore di tracking Microsoft Kinect v2. Fonte: [ref 25](#)

Kinect v2		
Infrared/depth camera	Resolution	512 × 424 px
	Field of view (h×v)	70.6° × 60.0°
	Angular resolution	0.14°/px
	Operating range	0.5–4.5 m
Color camera	Resolution	1920 × 1080 px
	Field of view (h×v)	84.1° × 53.8°
Frame rate		30 Hz
Minimum latency		20 ms
Dimensions (w×d×h) [mm]		249 × 66 × 67
Mass (without cables)		970 g
Connection type		USB 3.0

Fig.34: specifiche tecniche del sensore di tracking Microsoft Kinect v2.

Il principio di funzionamento non è basato sull'impiego di *fiducial markers*, bensì sul concetto del *time-of-flight* visto in Fig.30. Così, la telecamera di profondità acquisisce i *pixels* ed associa ad essi una distanza dipendente dalla lontananza di ciò che in essi è rappresentato (Fig.31). Successivamente avviene l'identificazione dei *pixels* "attivi" (Fig.35 a)), ovvero associati ad oggetti mobili (separazione del *foreground* dal *background*) ed infine la suddivisione del corpo umano in regioni (Fig.35 b)) per l'individuazione dei punti di interesse soggetti a tracking (Fig.35 c)).

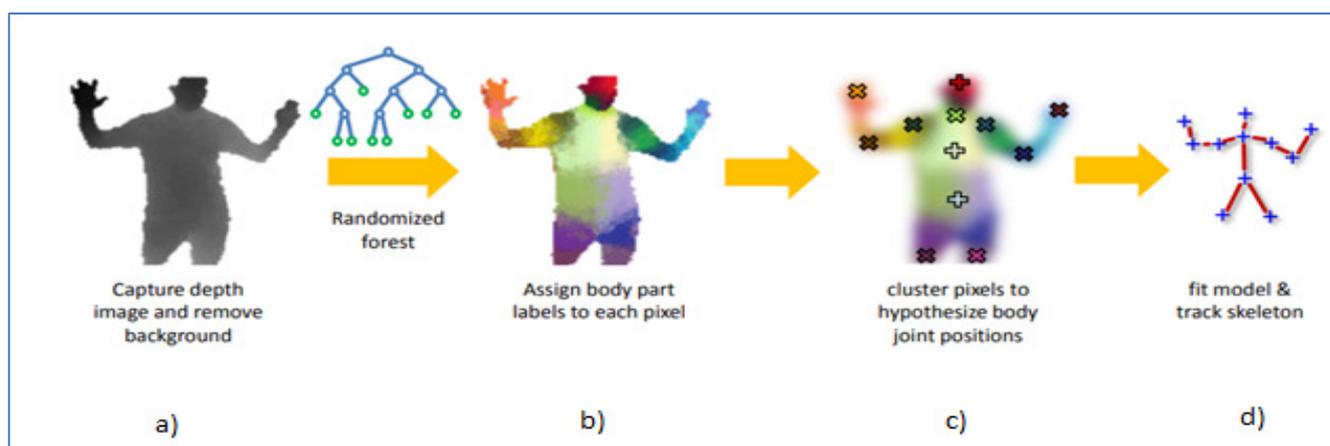


Fig. 35: pipeline del sistema di tracking Kinect. Fonte: [ref 26](#)

Il passaggio fondamentale nella pipeline del Kinect è l'impiego dell'algoritmo della Foresta Decisionale Casuale (Randomized Forest o Randomized Decision Forest) per assegnare un'area di appartenenza del corpo umano ad ogni *pixel* attivo (passaggio a-b) nella figura soprastante. Tale algoritmo, effettuando riconoscimento di immagini o di oggetti all'interno delle stesse, è largamente utilizzato nei campi della *computer vision* e *machine learning*, trovando ampio impiego in una moltitudine di applicazioni nel mondo dell'industria e dei servizi. In quest'ultimo spicca per importanza il campo medico, che impiega l'algoritmo della Foresta Decisionale Casuale per effettuare la diagnosi automatica da scansioni radiologiche. Il termine "Foresta" indica un insieme di alberi, ognuno dei quali in campo informatico è un Tipo di Dato Astratto (TDA) non lineare utilizzato per memorizzare informazioni in modo gerarchico e caratterizzato dalla proprietà fondamentale che ogni nodo diverso dalla radice possiede un unico padre. L'aggettivo "Decisionale" specializza il TDA albero, in quanto, nel percorrerlo, ad ogni nodo interno è associata una decisione avente solamente possibilità affermativa o negativa (sì/no): tale albero è caratterizzato come "binario", poiché ogni nodo non può possedere un numero arbitrario di figli, bensì al più due. L'albero decisionale può essere per esempio utilizzato da parte di un sistema automatico per riconoscere se una immagine di *input* rappresenta una scena all'aria aperta od al chiuso, come riportato in Fig.36, dove ad ogni nodo interno dell'albero la funzione di test mira a valutare il colore della parte alta o bassa dell'immagine.

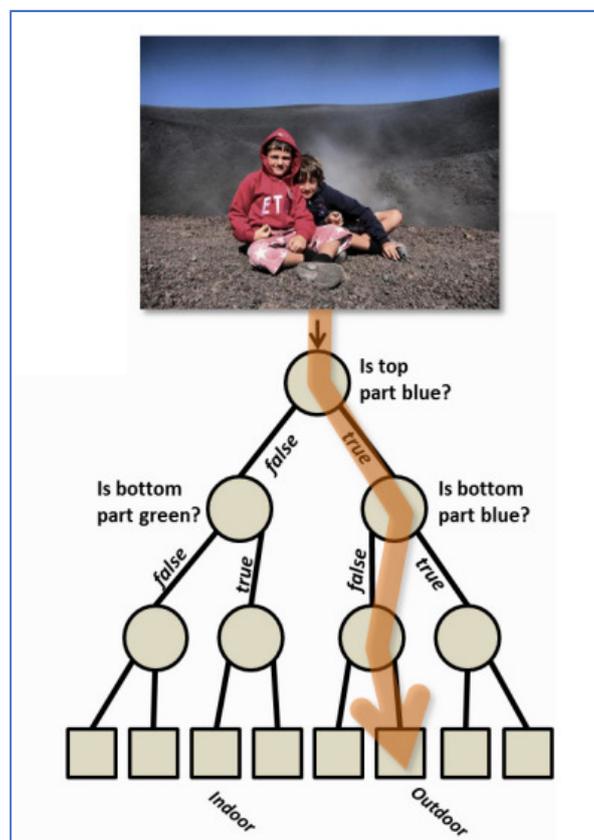


Fig. 36: esempio di albero decisionale. Fonte: [ref_27](#)

In modo del tutto analogo all'esempio appena introdotto, l'albero decisionale viene utilizzato nella problematica dell'individuazione dell'area del corpo umano a cui appartiene il pixel dell'immagine acquisita dal sensore Kinect. Infatti, la problematica del discernimento tra *aria aperta* o *chiuso* è opportunamente sostituita da quella della distinzione tra le varie aree del corpo umano, stavolta non avendo più in *input* un'immagine, bensì l'unità che la compone, ovvero il *pixel*. In questo caso però tale *input* non viene valutato facendo riferimento ad un solo albero, bensì a più di essi, ovvero ad una "Foresta Decisionale". La complicazione deriva dal fatto che l'utente può assumere diverse pose e quindi ciascun *pixel* attivo viene valutato mediante un'analisi più approfondita, correlando tra loro diverse risposte ottenute al raggiungimento di una foglia in ciascun albero. Infine, tale Foresta Decisionale è anche "Casuale" in quanto, ad ogni nodo interno di ciascun albero, la funzione di test è atta a valutare cosa succede in un intorno di raggio *random* dal *pixel* di *input*. Specificamente viene calcolato se un *pixel* target (diverso da quello di *input*) all'interno di tale intorno appartiene o meno al corpo dell'utente e, in caso affermativo, se ne calcola anche la profondità. In questo modo, man mano che si percorre il ramo dell'albero, confrontando le informazioni del *pixel* target e di quello di *input*, ad ogni funzione di test si riescono ad ottenere degli indizi sempre più precisi in riguardo all'area di appartenenza del *pixel* di *input*, che verrà espressa attraverso una distribuzione probabilistica al raggiungimento della foglia dell'albero. Tale distribuzione $p_t(c|v)$ sarà relativa a ciascuna area di appartenenza "c" del *pixel* di *input* "v" e deriverà dalla composizione delle probabilità risultanti che ciascuna area di appartenenza avrà accumulato durante il percorso decisionale lungo il ramo dell'albero che è stato percorso (Fig.37). La distribuzione di probabilità finale del *pixel* di *input* "v" dell'immagine "I" sarà mediata attraverso il numero totale T di alberi della foresta:

$$P(c|I, v) = \frac{1}{T} \sum_{t=1}^T P_t(c|I, v).$$

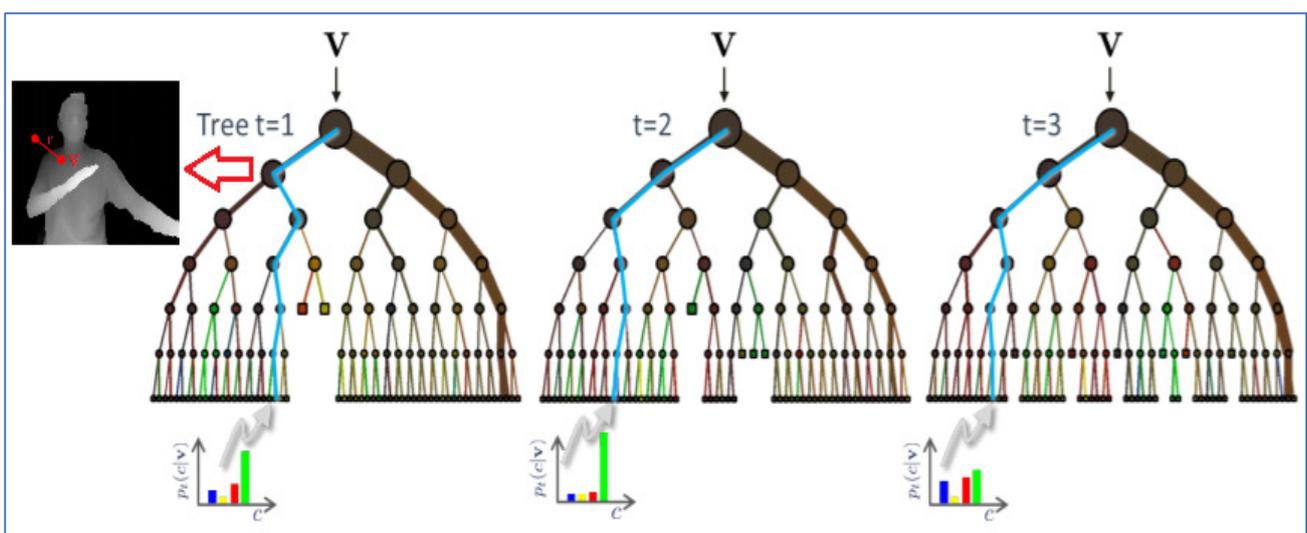


Fig. 37 Algoritmo della Foresta Decisionale Casuale. Fonte: [ref_28](#)

Il numero di alberi, la loro struttura, nonché le funzioni di test ed i valori di probabilità di ogni nodo interno sono state determinate a priori in fase di progettazione del Kinect e derivano da processi di addestramento del sensore basati su movimenti non di persone fisiche, bensì di modelli sintetici realizzati al computer. In questo modo è stato possibile generare automaticamente un database di riferimento con un ingente quantitativo di informazioni su un'ampia gamma di posture e dimensioni che può assumere il corpo umano (Fig.38). Tale database viene impiegato per il confronto e l'identificazione dei movimenti reali delle persone fisiche su cui effettuare il tracking durante la vita operativa del sensore.



Fig. 38: Addestramento del sensore Kinect (a destra) mediante modelli sintetici del corpo umano (a sinistra) atti a generare un'ampia gamma di posture e dimensioni di riferimento per il confronto con quelle reali delle persone fisiche su cui effettuare il tracking durante la vita operativa del sensore. Fonte: [ref 29](#)

Sebbene la letteratura sul sensore Kinect esaminata al riguardo non ne faccia menzione, al fine di implementare fisicamente la percorrenza di ciascun albero della Foresta Decisionale Casuale (Fig.37), esso può essere rappresentato mediante la struttura dati informatica della “lista collegata”. Essa può essere costruita tramite l'utilizzo di componenti che, nella programmazione ad oggetti, definiamo classi. Quest'ultime, nel caso in esame, saranno di due tipi, rispettivamente *Albero* e *Nodo*. Così, la classe *Albero* rappresenterà il gruppo che conterrà tutti gli “oggetti *Albero*” della Foresta Decisionale Casuale, mentre la classe *Nodo* identificherà il cluster che racchiuderà tutti gli “oggetti *Nodo*”. In tale contesto, ogni oggetto *Albero* sarà rappresentato da un attributo che contiene il puntatore all'oggetto *Nodo* radice, mentre ogni oggetto *Nodo* sarà appresentato dai seguenti tre attributi: funzione di test, puntatore al figlio sinistro, puntatore al figlio destro. Così, nel caso del seguente *Albero* binario in Fig.39,

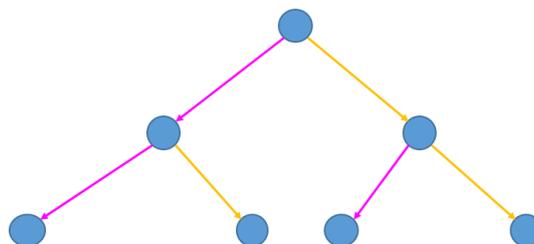


Fig. 39 *Albero binario*

la lista collegata ad esso associata è rappresentata in Fig.40.

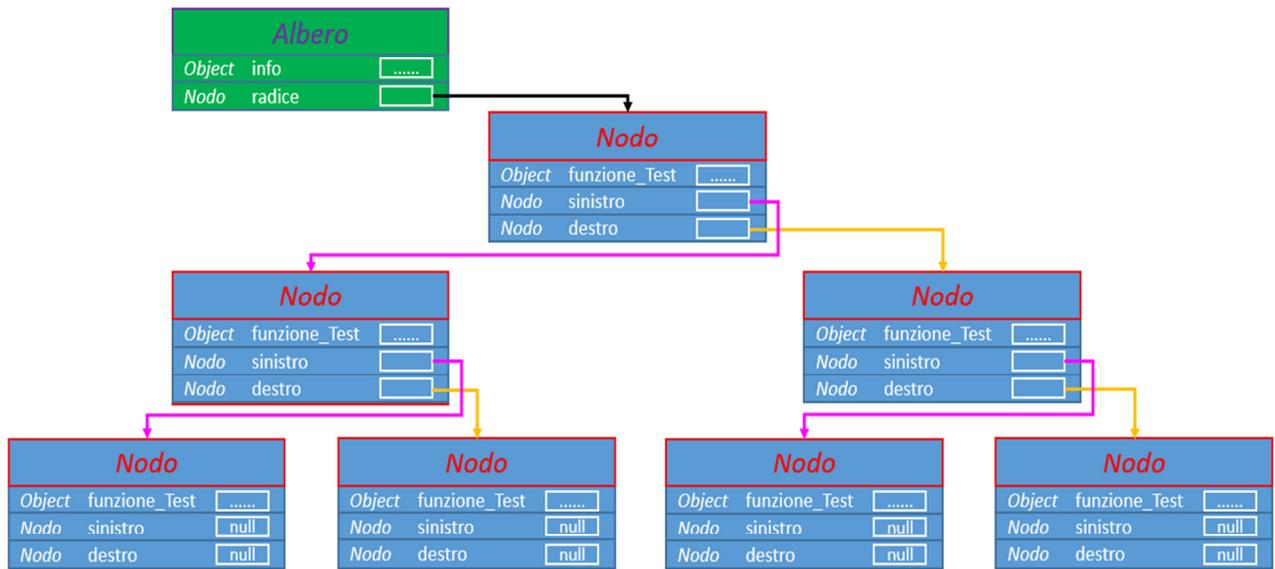


Fig. 40: rappresentazione del TDA Albero binario di Fig.39 mediante la struttura dati della lista collegata

Una seconda struttura dati atta a poter rappresentare il TDA *Albero* è costituita dall'*array*, ovvero dalla lista indicizzata. In essa, la radice è memorizzata nella posizione di indice 1 e, se un nodo è nella posizione di indice i , allora il suo figlio sinistro è nella posizione di indice $2i$, mentre il suo figlio destro è nella posizione di indice $2i + 1$. Così, a differenza di prima, in tale rappresentazione per passare da padre a figlio non si ha necessità di alcun puntatore. In questo modo si naviga molto efficientemente raddoppiando gli indici e senza utilizzare i puntatori, il che produce impatti positivi in termini di allocazione di memoria. Così, nel caso del seguente *Albero* binario in Fig.41,

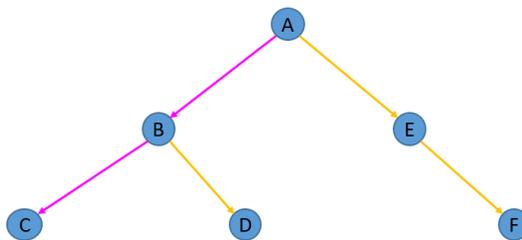


Fig. 41: Albero binario

l'*array* ad esso associato è rappresentato in Fig.42. Si noti che in questo caso ai nodi sono state aggiunte delle lettere per individuare in modo semplificato il posizionamento di ogni oggetto *Nodo*

all'interno dell'*array*. In realtà gli oggetti *Nodo* sono da intendersi collocati nelle rispettive celle dell'*array*, senza alcun puntatore ad essi afferente.

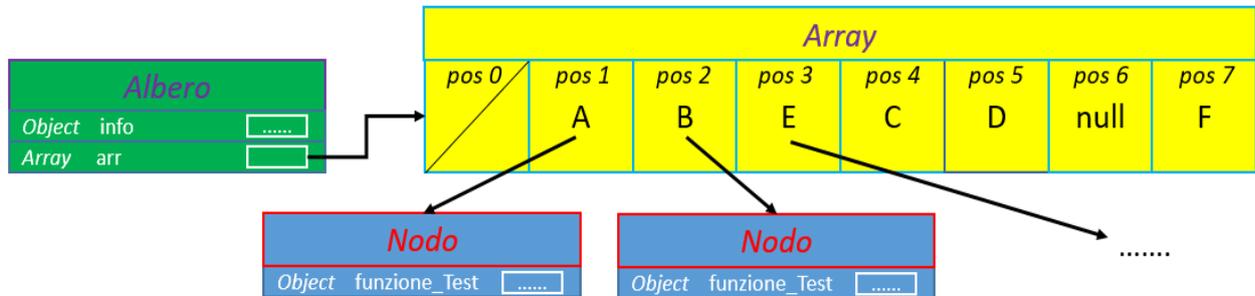


Fig.42: rappresentazione del TDA Albero binario di Fig.41 mediante la struttura dati della lista indicizzata array.

Tale seconda rappresentazione indicizzata è particolarmente efficiente per alberi binari completi, ovvero caratterizzati dalle seguenti due proprietà:

- 1) ogni nodo interno presenta sempre due figli;
- 2) le foglie sono tutte allo stesso livello dell'albero.

In base a tale definizione, l'albero di Fig.39 è un albero binario completo, mentre quello di Fig.41 non lo è. Facendo riferimento al primo, l'*array* corrispondente avrebbe riempito anche la posizione 6 lasciata vuota nel secondo caso. Da ciò segue che, nel caso di alberi completi, l'*array* si mantiene compatto e del tutto riempito, senza lo spreco inutile di allocazioni di memoria come quella lasciata libera nella posizione 6 in Fig.42.

Si noti che in entrambe le strutture dati di lista collegata e di *array* introdotte, partendo dalla radice dell'albero è possibile utilizzare i collegamenti (puntatori ed indici rispettivamente) ai nodi figli per poter raggiungere ogni nodo dell'albero. In base a questa assunzione, ogni oggetto *Nodo* rappresenta implicitamente il sottoalbero di cui è radice e quindi l'algoritmo di percorrenza del ramo di ogni *Albero* della Foresta Decisionale Casuale rappresentata in Fig.37 può essere descritto in modo conciso mediante il seguente algoritmo ricorsivo, dove con quest'ultimo termine si indica il fatto che, all'interno della propria struttura, l'algoritmo richiama sé stesso per percorrere il sottoalbero sinistro o destro:

Algoritmo Distribuzione_Probabilità_Zone_Appartenenza (A, r, p)

Input: albero A, radice r dell'albero A, pixel corrente p

Output: distribuzione di probabilità di appartenenza di p alle varie zone del corpo umano

```
if (r.sinistro != null and r.destro != null) // se visito un nodo interno
    if (funzione di test in r su p produce valore positivo)
        // attraversamento ricorsivo sottoalbero sinistro:
        return Distribuzione_Probabilita_Zone_Appartenenza (A, r.sinistro, p)
    else
        // attraversamento ricorsivo sottoalbero destro:
        return Distribuzione_Probabilita_Zone_Appartenenza (A, r.destro, p)
else // se visito una foglia
    // Restituisco la distribuzione di probabilità ad essa associata:
    return r.getProbabilità()
```

Il costo computazionale di tale algoritmo, dato in funzione della dimensione dell'*input* rappresentata dal numero n di nodi dell'albero binario, oscilla tra $\log n$ ed n , rispettivamente costo di caso migliore e di caso peggiore. Il primo si ha nella situazione di albero completo. Infatti, dalle due proprietà di completezza riportate sopra, segue che, in un albero binario completo, in ogni livello i ci sono esattamente 2^i nodi. Indicando con k il numero di livelli, si hanno 2^k foglie ed il numero totale di nodi n è:

$$n = \sum_{i=0}^k 2^i = 2^{k+1} - 1$$

Da ciò segue che in un albero completo si hanno 2^k foglie e $2^k - 1$ nodi interni. In tale modo, partendo dal livello zero corrispondente alla radice, un albero binario completo di n nodi ha un numero di livelli k ed un'altezza h pari a:

$$k = \log_2(n + 1)$$

$$h = k - 1 = \log_2(n + 1) - 1$$

Semplificando e trascurando i moltiplicatori costanti del cambiamento di base logaritmico, in un albero binario di n nodi l'altezza h è:

$$\log n \leq h \leq n$$

Così, l'altezza minima si ha in caso di albero completo, quella massima se l'albero è degenero (tutti i nodi con un solo figlio) e tutti i valori intermedi a tale *range* sono ammissibili (Fig.43).

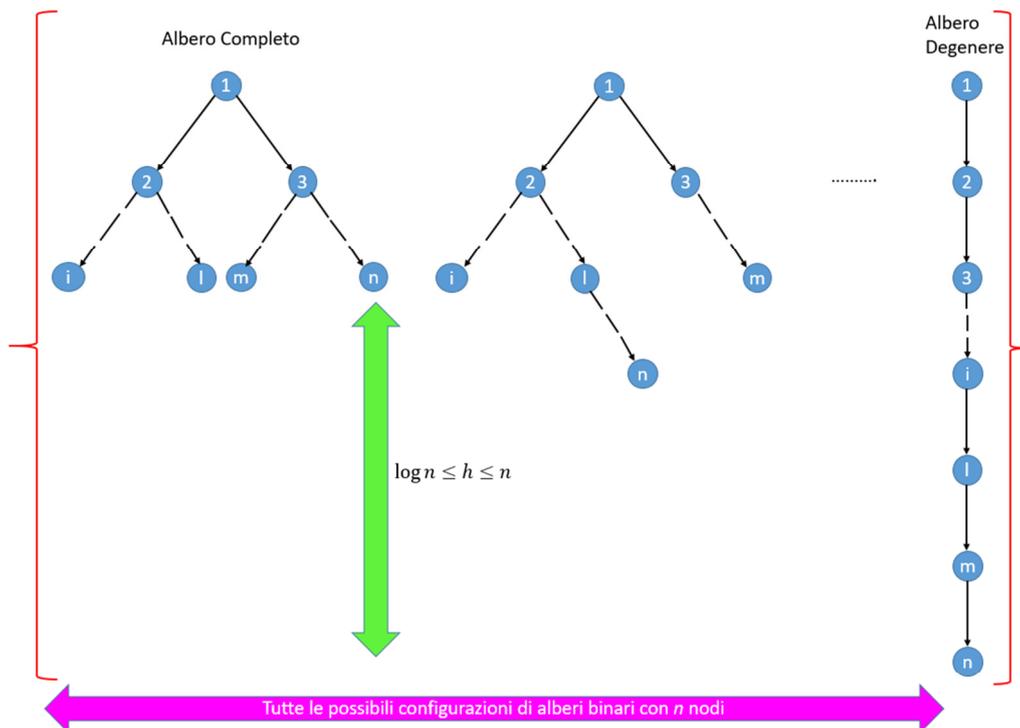


Fig.43: oscillazione dell'altezza tra il valore minimo $\log n$ (nel caso di albero binario completo) ed il valore massimo n (nel caso di albero binario degenerare) al variare di tutte le possibili configurazioni di alberi binari con n nodi.

Dal momento che l'altezza di un albero coincide con l'altezza del suo ramo più lungo, il costo computazionale dell'algoritmo di percorrenza del ramo di ogni albero della Foresta Decisionale Casuale si valuta proprio nella condizione più sfavorevole di massima distanza radice-foglia ed oscilla quindi tra $\log n$ ed n . Dal salto esponenziale che intercorre tra i due valori estremi, si comprende come sia fortemente vantaggioso affinché ogni albero della Foresta Decisionale Casuale sia costruito in modo tale da risultare completo.

Per concludere, il presente paragrafo ha fornito una descrizione molto approfondita del funzionamento del sensore Kinect, basandosi sulla letteratura presente al riguardo. In più, laddove quest'ultima è risultata carente, si è cercato di proporre metodi e strumenti aventi lo scopo di trattare la problematica del tracking non solo sotto gli aspetti implementativi, ma anche in relazione a quelli di ottimizzazione e di efficienza computazionale algoritmica. L'integrazione di tecniche di *machine learning* (per l'addestramento automatico del sensore), *computer vision* ed *image processing* (per il riconoscimento dei pattern "imparati" durante l'addestramento) descritta sopra permette di costruire un sistema di riconoscimento basato sull'identificazione di parti del corpo umano mediante un'analisi locale di ogni *pixel*. Ciò consente di superare i classici schemi tradizionali del tracking, dove l'utente viene coinvolto in una fase di calibrazione iniziale, in cui si richiede all'individuo di assumere una posizione di inizializzazione (la classica configurazione a T con le braccia estese lateralmente), che, successivamente, permette al sensore di seguire il movimento del corpo sulla base di un semplice *matching*: se in un primo frame acquisito viene identificata una gamba, nel frame successivo essa non può essere soggetta ad assumere una configurazione molto diversa rispetto a quella del frame precedente. In tali schemi classici, se

l'utente viene "perso" a causa dell'interposizione di un oggetto ocludente l'individuo alla vista del sensore, l'utente è costretto ad assumere nuovamente la posizione di calibrazione per far partire nuovamente il corretto funzionamento del sensore. Al contrario, con il nuovo approccio tali problematiche vengono superate, in quanto il tracking non dipende dalla storia delle posizioni assunte dall'individuo nei *frames* precedenti, bensì da un'analisi locale di ogni *pixel* e dal confronto di configurazioni del corpo assunte durante il processo di addestramento del sensore. Si potrebbe quindi affermare che tale metodologia non ha memoria e l'efficienza precedentemente trattata, nonché la robustezza appena descritta fanno sì che essa costituisca una direzione molto promettente sia nel campo delle applicazioni industriali, che in quello della ricerca futura.

5.2 Applicazioni di tracking

Opportune considerazioni sull'unione degli *stream* del Kinect provenienti dalla telecamera RGB e da quella di profondità hanno portato alla simulazione via software di un *display* spaziale AR di ampie dimensioni. Come detto nel capitolo introduttivo, gli *head-up display* frequentemente usati nei *cockpit* degli aerei sono un esempio di *display* spaziale, che, come ogni altro sistema AR, basa il proprio funzionamento sui tre moduli separati mostrati in Fig.22: sensore di tracking, CPU/GPU e *display* AR. Tuttavia, tale tecnologia non è presente sul mercato in grandi formati e, per ovviare a questa problematica, si è inizialmente pianificato di creare un *display* spaziale di ampie dimensioni mediante l'integrazione tra tracking fornito dal Kinect, un potente modulo CPU/GPU ed un *display* AR in grande formato. Anche questa idea però è stata presto abbandonata per via dell'assenza sul mercato di schermi AR *see-through* di ampie dimensioni. In tale modo, non potendo integrare soluzioni hardware, il *display* AR è stato opportunamente simulato via software nel Laboratorio di Realtà Virtuale del Dipartimento di Ingegneria Industriale del campus di Forlì attraverso il sistema CAVE (Cave Automatic Virtual Environment), che rappresenta un ambiente immersivo di Realtà Virtuale costituito da ampi schermi retroproiettati ed una pedana. La simulazione via software della tecnologia del *display* spaziale di ampie dimensioni mediante la piattaforma integrata Kinect + CAVE è stata implementata collocando il sensore di tracking sulla pedana sotto lo schermo centrale (Fig.44).



Fig.44: la piattaforma integrata CAVE + Kinect.

In questa configurazione, mediante codice di programmazione C#, nonché librerie di sviluppo software per Kinect, due *stream* di dati sono stati opportunamente convogliati dal sensore al CAVE, rispettivamente quello relativo alla telecamera RGB e l'altro concernente la telecamera di profondità. Come descritto nel precedente paragrafo e mostrato in Fig.35, dal secondo *stream* deriva l'estrapolazione del tracking dell'utente e la relativa rappresentazione è stata sovrapposta sugli schermi del CAVE allo *stream* derivante dalla telecamera RGB (Fig. 45).

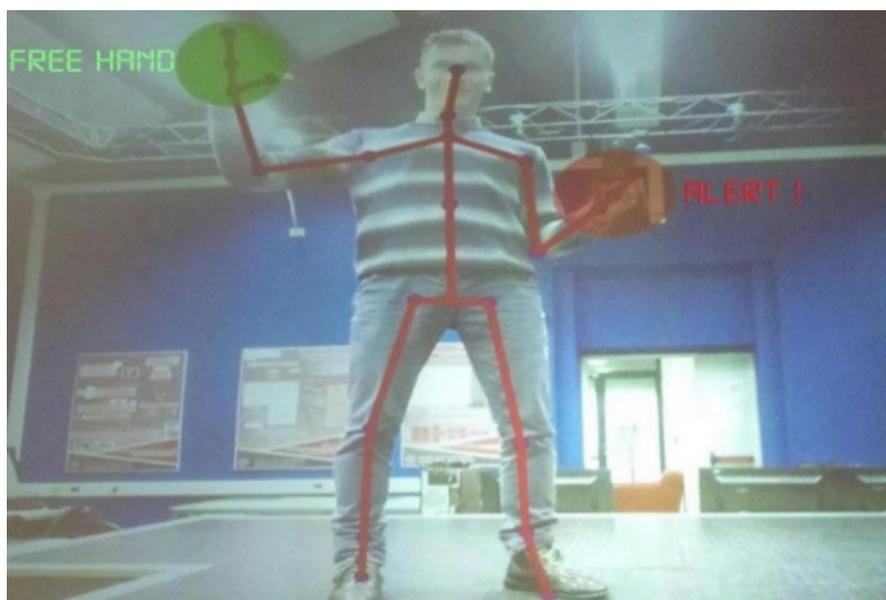


Fig.45: stream della telecamera RGB e della telecamera di profondità del Kinect integrati nel CAVE.

La situazione descritta e rappresentata nella precedente figura può essere immaginata nel seguente modo (Fig.46): lo *stream* della telecamera RGB viene proiettato sugli schermi del CAVE, mentre lo *stream* del tracking (giunti dello scheletro indicati in rosso in Fig. 45) estrapolato dalla telecamera di profondità è convogliato verso un ipotetico *display* AR di ampio formato, collocato immediatamente davanti agli schermi del CAVE. Tali considerazioni permettono di simulare via software un *display* spaziale AR di ampie dimensioni, in quanto lo *stream* del tracking può essere pensato come un AR overlay registrato in ogni momento con lo *stream* della telecamera RGB.

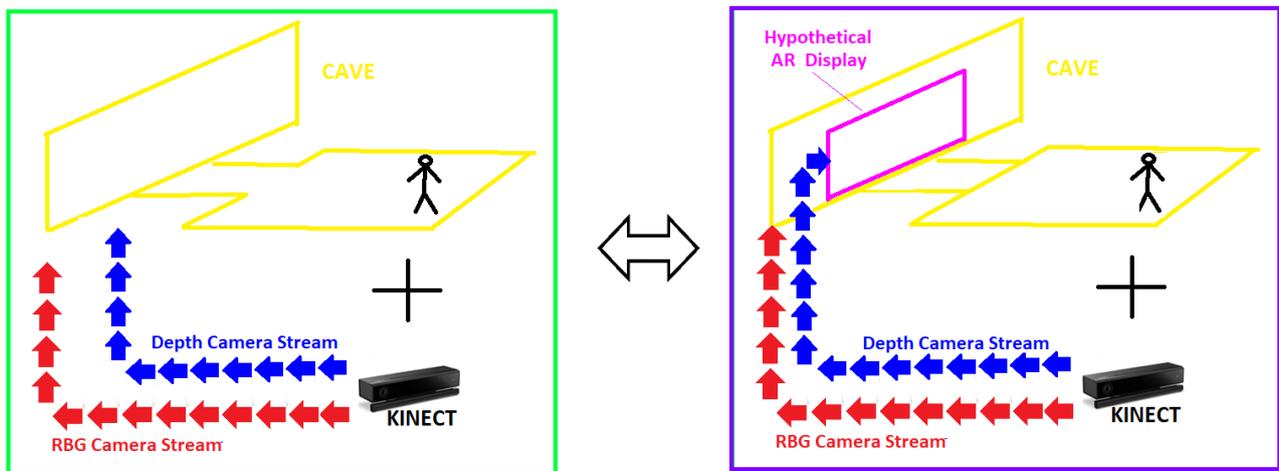


Fig.46: la situazione descritta in Fig.45 e qui rappresentata a sinistra è equivalente alla concettualizzazione di un ipotetico display AR di ampio formato simulato attraverso la piattaforma integrata CAVE + Kinect taforma integrata CAVE + Kinect e mediante separazione opportuna degli stream della telecamera RGB e della telecamera di profondità del Kinect.

Nell'applicazione descritta in Fig.45 viene anche mostrato come il Kinect dia la possibilità di valutare lo stato delle mani: l'utente tiene un oggetto con la mano sinistra, cosicché essa risulterà valutata come chiusa e quindi di stato differente rispetto alla mano destra libera. Ciò fornisce l'opportunità di intravedere numerose applicazioni nel campo della security, soprattutto se il tracking del corpo umano (1), abbinato con la rilevazione dello stato delle mani (2), viene ulteriormente integrato con la terza ed ultima *feature* messa a disposizione del Kinect v2, ovvero il tracking facciale (Fig.47). Mediante quest'ultimo, non solo è possibile rilevare il volto di un utente, ma è anche possibile accedere in *real-time* a più di mille punti di interesse per effettuare riconoscimento facciale. Il tracking del volto avviene sulla base dello stesso principio di funzionamento del tracking del corpo visto nel paragrafo precedente: identificazione di parti del corpo umano mediante un'analisi locale di ogni *pixel*.



Fig.47: tracking facciale (a sinistra) ed integrazione con lo stream derivante dalla telecamera RGB del Kinect (a destra)
 Fonte fig. destra: [ref_30](#)

L'integrazione tra tracking facciale e corporeo permette di incrementare l'efficacia del sensore Kinect nel distinguere configurazioni assunte dall'utente altrimenti non discernibili. Considerando infatti solamente il tracking del corpo, al di sopra delle spalle il Kinect rileva solamente i punti di interesse che corrispondono al collo ed alla testa (Fig.32), cosicché, passando tra due punti infiniti piani, non è possibile valutare come è orientato il piano del volto. Ciò rischia di far fallire applicazioni che richiedono di riuscire ad identificare situazioni in cui il corpo dell'utente è orientato in una certa direzione, mentre quella in cui guarda l'utente è completamente differente. Purtroppo, l'integrazione tra gli *stream* del Kinect relativi al tracking facciale ed a quello del corpo richiede elevate risorse computazionali ed un'efficiente gestione simultanea dei due *stream* risulta di problematica implementazione, a meno che non si ricorra a sistemi multiprocessore (*multicore*) e/o programmazione *multithreading*. Attraverso i primi è possibile conferire la gestione dei due *stream* ad unità di calcolo differenti in modo tale da eseguirli in parallelo, mentre con la programmazione *multithreading* i due *stream* possono essere gestiti da un unico programma che suddivide l'esecuzione del codice in *threads*, ovvero in *task* operanti parallelamente (Fig.48).

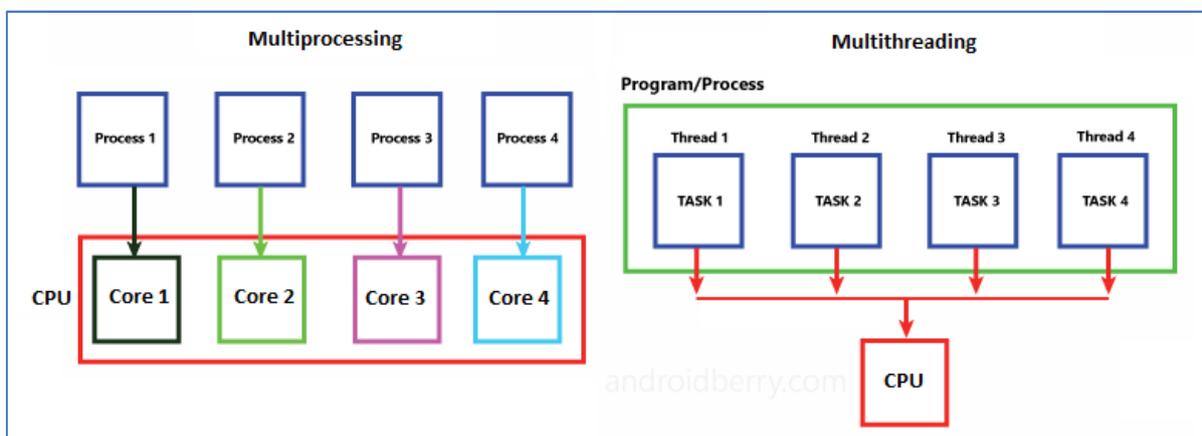


Fig.48: differenza tra sistemi multiprocessore (*multicore*) e programmazione *multithreading*. Fonte: [ref_31](#)

La strada che si è scelto di percorrere per integrare i due *stream* è stata quella della programmazione *multithreading*, resa possibile mediante il software Unity3D, dove i due *stream* di tracking del corpo e del volto (di Fig.45 e Fig.47) vengono processati efficientemente in modo simultaneo, come riportato in Fig.49: la posa assunta durante lo *screenshot* acquisito durante il funzionamento del software vuole evidenziare il diverso orientamento del volto rispetto a quello del corpo.

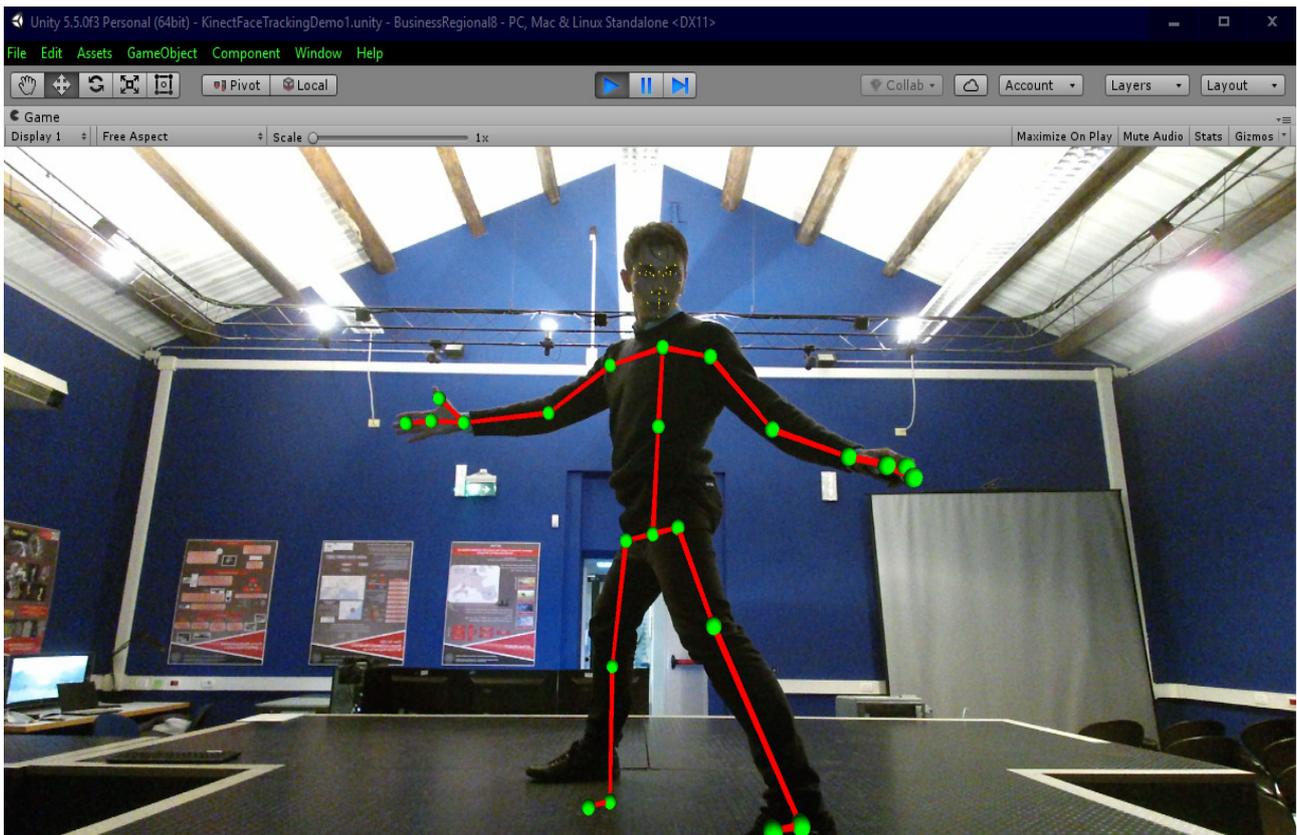


Fig.49: software Unity in opera per integrare i due *stream* di tracking del corpo e del volto.

Introdotta il software Unity3D, la trattazione segue con un paragrafo atto ad evidenziare le caratteristiche fondamentali del programma. Esso è stato lo strumento fondamentale da me utilizzato per l'implementazione di

- 1) *visualizzazione stereoscopica,*
- 2) *navigazione interattiva,*
- 3) *comportamento dinamico nel tempo*

di ambienti virtuali e di Realtà Aumentata caratterizzanti le applicazioni che hanno permeato l'intero corso di dottorato, nonché due progetti europei di spicco per successo conseguito e partnership coinvolta, ovvero CASTLE e RETINA, di cui si parlerà più avanti.

5.3 Unity3D

Unity3D è un software per lo sviluppo di videogiochi, ovvero un *game engine* e, come tale, è caratterizzato dai seguenti tre moduli fondamentali:

- 1) motore grafico (o motore di *rendering*): generazione di immagini da un modello matematico;
- 2) motore fisico: simulazione del comportamento degli oggetti secondo le leggi della fisica newtoniana e quindi attraverso il concetto di forza, massa, accelerazione, velocità, attrito, etc..;
- 3) *tool* di programmazione in linguaggio C# o Javascript mediante ambiente di sviluppo *MonoDevelop* o *Visual Studio*: conferimento di un comportamento variabile nel tempo agli oggetti.

Inoltre, Unity3D aggiunge una funzionalità molto utile alle tre classiche che caratterizzano ogni *game engine* riportate sopra: mette a disposizione dell'utente una *game live preview* che consente di visualizzare ed apportare modifiche all'applicazione che si va costruendo durante il *run* (test) dell'applicazione stessa.

Tuttavia, Unity non è un software di modellazione 3D, come lo sono invece per esempio 3D Studio Max o Maya. Abbinato con tali strumenti professionali e quindi importando modelli creati con essi, Unity sostanzialmente “anima” la scena mediante il motore fisico ed il tool di programmazione (ultimi due punti dell'elenco riportato sopra). Impiegato massivamente dall'industria dei videogiochi e del cinema per la creazione di ambienti di realtà virtuale o aumentata, Unity, insieme ad Unreal, secondo statistiche e ricerche [79] è oggi il migliore *game engine* in circolazione, nonché sicuramente quello più usato al mondo, come riportato in Fig.50.

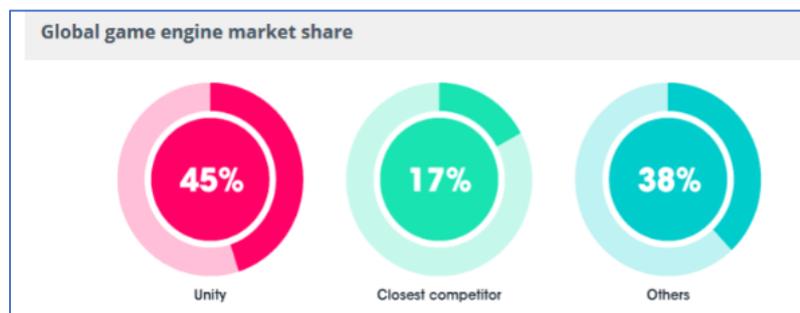


Fig.50: spartizione del mercato globale dei game engines (anno 2016). Fonte: [ref. 32](#)

Tanto successo è dovuto soprattutto alla sua portabilità: le applicazioni create con Unity possono essere esportate su numerose piattaforme, fra cui tutti i sistemi operativi desktop, *mobile*, *console* e web. Ciò ne fa una scelta formidabile:

- 1) per i teams di sviluppatori che desiderano portare la loro applicazione su più piattaforme;
- 2) nel caso inizialmente non si sappia il genere di applicazione da creare e la tecnologia su cui investire.

L'interfaccia di Unity è organizzata in pannelli, di cui i principali sono quello *Project*, *Scene*, *Game*, *Console*, *Hierarchy* ed *Inspector*, evidenziati in Fig.51.

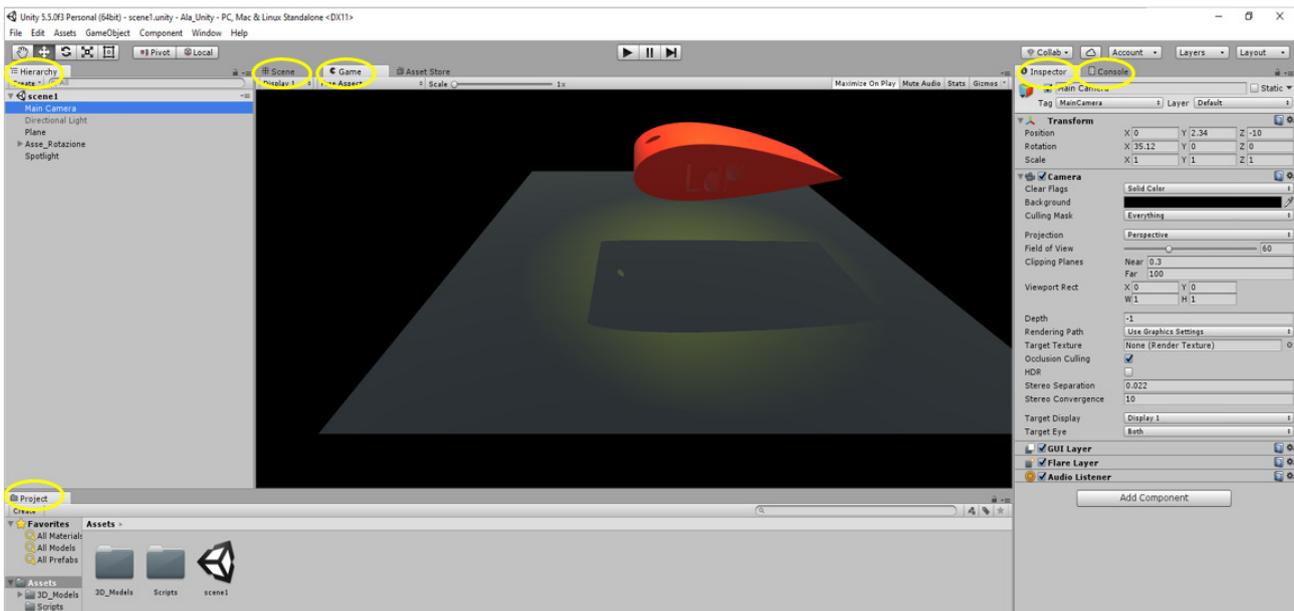


Fig.51: l'interfaccia di Unity con i sei pannelli principali evidenziati in giallo: *Project*, *Scene*, *Game*, *Console*, *Hierarchy*, *Inspector*.

Il pannello *Project* visualizza tutti i file (i cosiddetti “*Assets*”) presenti nell'applicazione. All'inizio è vuoto e progressivamente si popola con i file che vengono importati o creati all'interno dell'applicazione. È buona pratica ordinare gli *Assets* per categoria, come evidenziato in Fig.51, dove tutti i files sono stati ordinati tra le *directories* “*3D_Models*” e “*Scripts*”. La prima contiene tutti i modelli 3D importati e costruiti per esempio con i software di modellazione prima menzionati. La seconda contiene tutti i codici di programmazione creati dentro all'applicazione Unity stessa (mediante l'ambiente di sviluppo *MonoDevelop* o *Visual Studio* già citati precedentemente) ed atti a conferire una dinamica agli elementi presenti dentro la scena, ovvero il pannello *Scene*. Questo è il cuore di Unity e costituisce la rappresentazione dell'ambiente dell'applicazione ai fini della modellazione. L'ambiente *Scene* viene navigato attraverso le classiche funzioni di spostamento, rotazione e zoom ed è probabilmente così chiamato in quanto corrisponde alla scena in campo cinematografico dove opera il regista del film: analogamente, il regista dell'applicazione Unity, ovvero lo sviluppatore di turno incaricato di costruire la stessa, organizza la scena, l'esamina da tutte le possibili angolazioni e decide infine dove sistemare la telecamera con cui sarà coerente la visione del fruitore dell'applicazione durante il funzionamento della stessa. Tale visione è prodotta nel pannello *Game*, che quindi fornisce la rappresentazione dell'ambiente

dell'applicazione dal punto di vista della videocamera attualmente attiva. In altre parole, tornando al parallelo col campo della cinematografia discusso prima, il pannello *Game* fornisce il film che vede lo spettatore al cinema. In fase di costruzione dell'applicazione, lo si utilizza al fine di testare la stessa e per tale motivo è anche noto col nome di *game live preview* di cui si è parlato all'inizio del paragrafo. Unity mette a disposizione dello sviluppatore una *game live preview* molto innovativa, che consente di visualizzare ed apportare modifiche all'applicazione che si va costruendo durante il test (anche detto *run* nel gergo della programmazione) della stessa. Così, premendo il tasto "play" del pannello *Game*, Unity inizia ad eseguire l'applicazione e, corrispondentemente, tutti gli *Scripts*, nonché il motore fisico, verranno inizializzati ed eseguiti, riportando le informazioni di *debug* in un altro pannello, ovvero la *Console*. Così, essa rappresenta l'altro strumento a cui fa affidamento lo sviluppatore per controllare che tutto proceda secondo la logica prestabilita, in quanto fornisce avvisi ed informazioni sulla tipologia di errori a cui si va inevitabilmente incontro durante il processo di programmazione.

Delle volte non è così semplice risolvere gli errori a tempo di esecuzione dell'applicazione (ovvero durante il test della stessa), perché essi potrebbero essere segnalati dai driver di verifica di Unity molto più avanti rispetto al reale punto del codice dove è avvenuto lo sbaglio da parte dello sviluppatore. Un esempio classico è rappresentato dagli errori logici, che scaturiscono dall'implementazione di funzionalità differenti da quelle desiderate e che non vengono identificati a tempo di compilazione, ovvero prima di eseguire il programma. Si consideri per esempio il seguente metodo riportato in Fig.52.

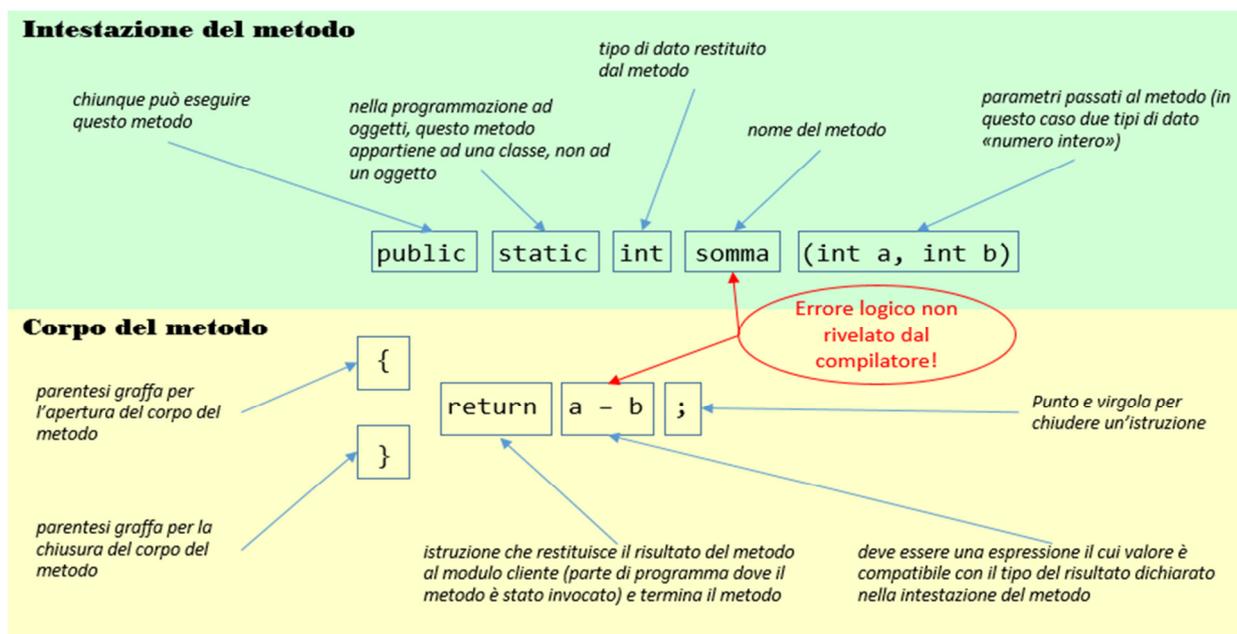


Fig.52: Esempio di errore logico non rivelato dal compilatore.

Tale metodo restituisce un numero sbagliato rispetto alla specifica di sommare due interi (errore logico), ma per il compilatore non ci sono problemi perché il nome del metodo può essere scelto a

piacere dallo sviluppatore. Tuttavia, durante l'esecuzione del programma, si supponga che tale valore restituito vada assegnato ad una variabile in seguito modificata da una successione di moltissime altre operazioni ed infine risultante a divisore col valore di zero: i driver di verifica di Unity generano a tempo di esecuzione il messaggio di errore “*Division by zero*” riportante la posizione nel codice dove esso si è manifestato, sebbene l'errore effettivo si sia verificato molto prima del punto segnalato.

Per quanto riguarda il pannello *Hierarchy*, tutto ciò che viene inserito in esso ha immediato riscontro nella scena, ovvero viene inserito nel pannello *Scene*. Entrambi, al momento della creazione di una nuova applicazione, contengono solamente la telecamera principale e la luce che illumina la scena, identificate rispettivamente con i nomi di *Main Camera* e *Directional Light* in Fig.51. Man mano che l'applicazione viene costruita, l'*Hierarchy* (e quindi la *Scene*) viene popolato con elementi nativi o importati. Con i primi si identificano quelli messi a disposizione dal programma (come la stessa *Main Camera* e *Directional Light*), mentre i secondi rappresentano quelli importati dall'esterno negli Assets, ovvero nel pannello *Project*. Quest'ultimo può essere paragonato ad un magazzino dove arrivano le merci che poi verranno utilizzate nella scena. Tuttavia, se nulla viene trasferito nell'*Hierarchy* dal pannello *Project*, l'applicazione sarà vuota, a meno che non sia caratterizzata da soli elementi nativi, quest'ultimi non presenti tra gli Assets. Gli elementi nativi possono essere quindi immaginati come dei beni di proprietà della troupe cinematografica, che, come tali, sono presenti nella scena, ma non nel magazzino delle merci provenienti dai fornitori (pannello *Project*). L'*Hierarchy* fonda il proprio nome sulla relazione padre-figlio che caratterizza i suoi elementi, che danno spesso vita ad un albero gerarchico molto complesso con molteplici livelli di annidamento. Il motivo di ciò risiede principalmente nel concepire una migliore organizzazione e gestione della scena mediante il concetto di ereditarietà:

- alla disattivazione di un elemento padre, anche tutti i figli diventano inattivi: in tale modo, in qualunque momento si potrebbero disabilitare con un solo click tutti gli elementi facenti parte di un certo gruppo, per essere poi riabilitati altrettanto facilmente in un secondo momento, senza dover agire singolarmente su ciascuno di essi, il cui numero potrebbe essere molto elevato;
- alla rimozione di un elemento padre, anche tutti i figli vengono rimossi;
- quando un elemento padre viene trasformato (cioè scalato, spostato o ruotato), i figli vengono anch'essi trasformati rispetto agli assi su cui è stato trasformato il genitore; così, se per esempio il padre viene scalato di un valore doppio rispetto al proprio asse x , anche i suoi figli vengono scalati al doppio della loro dimensione originale rispetto alla direzione x del genitore; la potenza di ciò consiste nel fatto che, in questo modo, se si ha necessità di scalare un modello complesso con una certa struttura gerarchica, basta agire sulla radice della gerarchia e non su ciascuno dei suoi componenti.

La nozione di padre-figlio, oltre a permettere una migliore organizzazione e gestione della scena, ha anche un'altra valenza molto importante, che è legata al concetto di cambiamento di base. Dal momento che in Unity non è possibile cambiare il sistema di riferimento di un elemento o di una gerarchia, per trasformarli rispetto ad un sistema di riferimento diverso rispetto alla propria base, si introduce nella scena l'elemento nativo coincidente con una terna di riferimento e si instaura il rapporto padre-figlio tra essa e l'elemento o la gerarchia che si vuole trasformare. Così, si immagini l'esempio di una scena in cui sia presente un modello di umanoide caratterizzato da una struttura gerarchica con radice avente terna di riferimento posta nella testa del modello. Nel caso si volesse

ruotare l'umanoide attorno ad una base posta all'altezza dei piedi, in questo punto della scena occorre dapprima introdurre un nuovo elemento nativo coincidente con una terna di riferimento ed infine stabilire la relazione di parentaggio con la radice dell'umanoide, cosicché da sfruttare la proprietà di ereditarietà riportata nel terzo punto dell'elenco visto poco sopra.

Ogni elemento dell'*Hierarchy* è sempre caratterizzato da tre fattori:

- 1) nome;
- 2) posizione nella scena;
- 3) una serie di proprietà (ciò non riguarda l'elemento nativo coincidente con la terna di riferimento, caratterizzato solamente dai precedenti due punti dell'elenco).

Tali caratteristiche sono riportate nel pannello *Inspector*, che, essendo proprio di ogni elemento, ne rappresenta la sua carta d'identità (nome, posizione = residenza, serie di proprietà = anno, luogo di nascita e così via). Il secondo punto dell'elenco è in realtà contraddistinto non solo dalla posizione, bensì anche dalla rotazione e scala, che, nel complesso, costituiscono la voce *Transform* (trasformazioni). Essa riporta la configurazione della terna locale dell'elemento rispetto alla configurazione della terna globale posta nel punto 0,0,0 dello spazio nella scena. Ciò non è vero quando l'elemento gode di una relazione di parentaggio: i valori delle trasformazioni del figlio sono in relazione a quelli del padre. Sulla base di tale concetto, se un padre viene trasformato, i valori di trasformazione del figlio non cambieranno perché il figlio non è stato trasformato rispetto al padre, ovvero la terna di riferimento del figlio vede quella del padre sempre nello stesso modo.

Per concludere, in questo paragrafo sono state riportate le caratteristiche fondamentali dell'interfaccia di Unity ed essa è stata descritta mediante una trattazione che ha voluto sottolineare il carattere di stretto legame con cui ogni pannello è intrinsecamente collegato agli altri senza soluzione di continuità. In generale, evitando di portare avanti un ragionamento per “compartimenti stagni”, la comprensione delle funzionalità messe a disposizione di un software, il successivo domandarsi del perché la scelta sia ricaduta su di esse invece che su altre, nonché l'ausilio derivante dall'immaginare esempi di facile intuizione (pannello *Project* equiparato al magazzino merci, *Scene* correlato col *set* cinematografico, *Game* pensato come il film visto dallo spettatore ed *Inspector* come la carta d'identità), permettono di identificare le strette connessioni e le logiche che intercorrono tra i vari moduli dell'interfaccia messa a disposizione dello sviluppatore, velocizzando in massima misura l'apprendimento di un programma complesso come lo è Unity. La trattazione segue con la descrizione di come l'integrazione in Unity del tracking del corpo e del viso forniti dal Kinect sia stata efficientemente applicata al progetto europeo CASTLE per implementare la visualizzazione stereoscopica e la navigazione interattiva di ambienti virtuali di cabina di aereo.

5.4 La navigazione interattiva di ambienti virtuali nel Progetto CASTLE

CASTLE, acronimo che sta per “Cabin System design Toward passenger wellbEing”, è un progetto europeo che è incluso nel programma Clean Sky 2 del framework Horizon 2020 e mira allo sviluppo di un innovativo processo di design degli interni di cabina degli aerei, ponendo al centro dell’attenzione il comfort del passeggero sin dalle fasi iniziali della progettazione. Il coordinatore del progetto è la società Geven e la partnership include studi di progettazione come Acumen e Pininfarina, tre università italiane (Bologna, Napoli e Torino) ed altre importanti società, come Leonardo e Siemens.

La motivazione alla base dello studio sta nel fatto che gli aerei richiedono continui miglioramenti nel comfort del passeggero per competere con la concorrenza sul mercato globale. Così, il progetto si pone come obiettivo la creazione di soluzioni innovative riguardanti interni di cabina che massimizzano il comfort ed il benessere del passeggero nel caso degli aerei turboelica adibiti a trasporto regionale e business jet (Fig.53).



Fig.53: Le due tipologie di aeromobili studiate dal progetto CASTLE: Aerei Regional e Business Jet

Per soddisfare tale obiettivo, viene utilizzato l’approccio progettuale *human centred design* (HCD), ovvero incentrato sui desideri e le esigenze degli utenti finali. Esso fornisce strumenti e metodi per collezionare dati sull’impatto prodotto sugli utenti dal design dei differenti componenti di cabina, sin dalle prime fasi di progettazione degli stessi. La ricerca è stata applicata a diversi elementi che sono stati selezionati come quelli che più di tutti influenzano il comfort in termini di *human factors* e ergonomia, come per esempio i sedili, i tavolini, il rivestimento di cabina, le cappelliere dei bagagli, la cucina, i bagni, l’illuminazione, ecc. .

Il processo di design di ogni componente (*item*), ovvero l’approccio HCD del CASTLE, è riportato in Fig.54 e coincide con l’integrazione tra la *piattaforma di valutazione del comfort degli items* (al centro della figura) ed il *loop di valutazione delle soluzioni di design* (a sinistra).

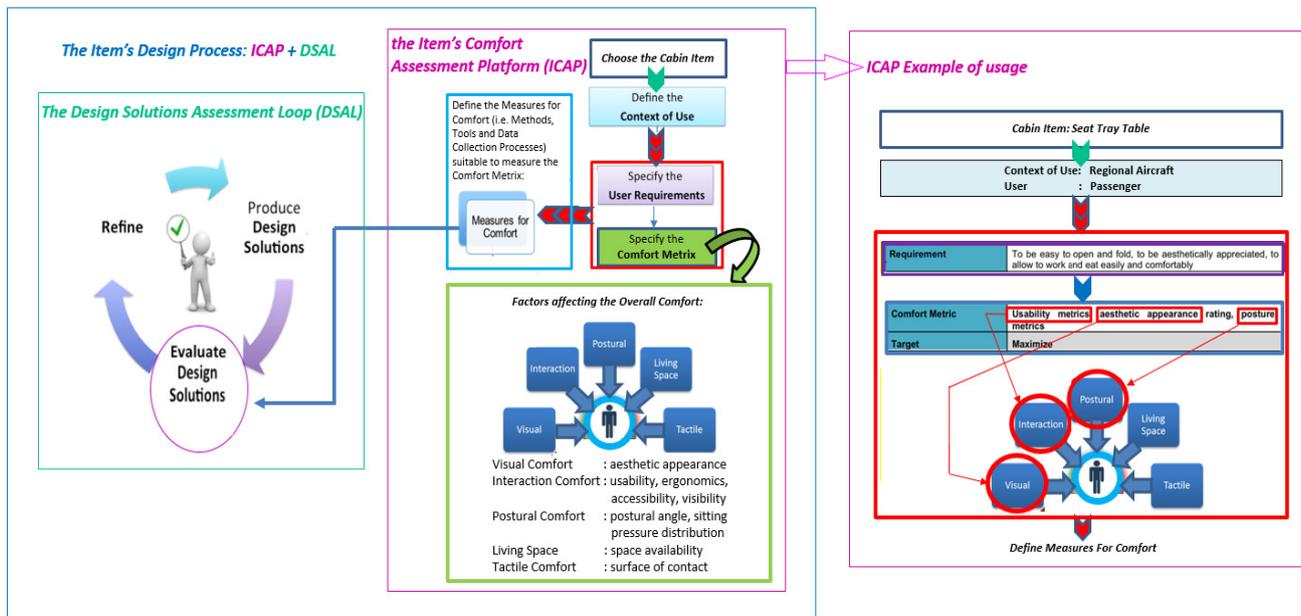


Fig.54: l'approccio human centred design (HCD) del progetto CASTLE – schema generale

Nella *piattaforma di valutazione del comfort degli items*, secondo un processo a cascata in cui gli *step* successivi sono funzione di quelli precedenti, per ogni componente viene definito un *contesto d'uso*, un *set di requisiti* da parte dell'utenza e delle *metriche di comfort*, quest'ultime selezionate considerando tutti i fattori che influenzano il *comfort totale*, costituito dal *comfort visivo*, di *interazione*, *posturale*, *spaziale* e *tattile*. Così, per esempio, nel caso dei tavolini (si veda l'esempio d'impiego della *piattaforma di valutazione del comfort degli items* nella parte destra della figura), il *set di requisiti* contempla caratteristiche come la facilità di apertura e stabilità, l'estetica gradevole, il comfort di utilizzo nelle attività lavorative e di consumazione dei cibi, cosicché le *metriche di comfort* riguardano la facilità d'uso, l'estetica ed il *comfort posturale*. Sulla base di quest'ultime vengono definite le *misure di comfort*. Dal momento che costituiscono l'*output* della *piattaforma di valutazione del comfort degli items* e l'*input* del *loop di valutazione delle soluzioni di design*, le *misure di comfort* rappresentano il collante che connette i due moduli, permettendo così di *valutare le soluzioni di design* secondo un processo ciclico. Così, nel *loop*, quest'ultime vengono dapprima prodotte, poi valutate mediante le *misure di comfort* e quindi successivamente modificate o rifinite, entrando nuovamente nel *loop* secondo un processo iterativo.

Le *misure di comfort* rappresentano un gruppo di metodi, strumenti e processi di collezione dati adatto a misurare le *metriche di comfort* (Fig.55).

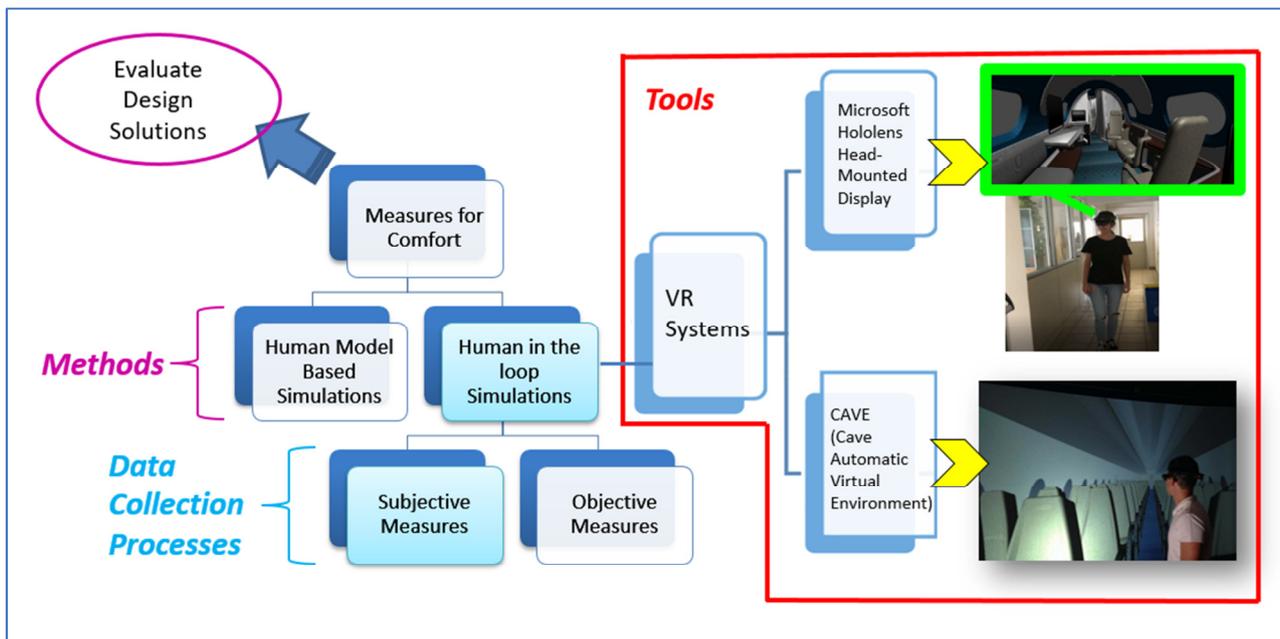


Fig.55: misure di comfort

Il ruolo svolto dall'Università di Bologna nel progetto CASTLE è stato quello della valutazione delle soluzioni di design riguardanti aerei *regional* e *business jet* [80]. Essa si è basata su un processo di collezione dati forniti dagli utenti. Questi sono stati coinvolti in esperimenti svolti mediante il metodo delle cosiddette simulazioni *human-in-the-loop* effettuate per mezzo di due strumenti differenti di Realtà Virtuale: il sistema CAVE + Kinect ed il visore Microsoft Hololens HMD (Fig.55) presenti nel Laboratorio di Realtà Virtuale del Dipartimento di Ingegneria Industriale del campus di Forlì. Per quanto riguarda il dispositivo Microsoft Hololens, seppur strumento proprio della Realtà Aumentata, è stato efficientemente utilizzato nel CASTLE in condizioni di luminosità del Laboratorio molto basse, in modo tale da riuscire a mascherare il mondo reale visto attraverso il *display see-through* con quello artificiale costruito dallo stesso. In questo modo, ciascun utente è stato sottoposto ad una esperienza di cabina mediante un ambiente virtuale immersivo e durante l'esperimento sono state raccolte sia informazioni qualitative soggettive mediante questionari, sia dati quantitativi oggettivi come coinvolgimento, stress, ansia ed emozioni che l'utente ha percepito nell'interazione col l'ambiente virtuale (raccolti mediante sensori EEG).

Il *task* da me svolto è stato quello di implementare gli ambienti di Realtà Virtuale sulle piattaforme rappresentate dal sistema CAVE + Kinect e dal visore Microsoft Hololens HMD, fornendo in tal modo all'utente la possibilità della visualizzazione stereoscopica e della navigazione interattiva secondo la propria prospettiva visuale degli ambienti virtuali di cabina. Specificamente, una volta acquisiti dai partners del progetto CASTLE i modelli 3D delle soluzioni progettuali da valutare, essi sono stati opportunamente lavorati all'interno ed all'esterno di Unity (mediante i programmi Rhinoceros e Blender) per minimizzare:

1. la perdita di informazioni dovuta alla conversione dei files tra i diversi formati di compatibilità dei software di modellazione dei partners e di Unity [81];

2. la pesantezza dei modelli 3D acquisiti.

Implementati gli ambienti virtuali, essi sono stati resi navigabili e, per mantenere efficienti prestazioni nello svolgimento di tali attività, sono stati infine opportunamente valutati gli aspetti di *rendering* delle applicazioni. Relativamente al visore, la navigazione secondo la prospettiva visuale dell'utente è stata implementata secondo il *pattern* proprio dei classici dispositivi di Realtà Virtuale: all'interno dell'ambiente artificiale si sceglie un punto di partenza iniziale, cosicché l'utente può spostarsi a partire da esso camminando nel mondo virtuale in scala 1:1 che ha attorno (si veda l'immagine relativa al caso degli HMD in Fig.55). Diversa è stata l'implementazione per la piattaforma CAVE + Kinect: il sensore di tracking cattura la posizione e l'orientamento del corpo e del volto ed invia tali dati al software Unity, che calcola ed aggiorna costantemente la matrice di proiezione degli schermi in modo solidale al punto di vista dell'utente [6] che si muove sulla pedana del CAVE. Al fine di rendere l'applicazione non eccessivamente sensibile, si è impostata una efficiente azione di filtro atta ad impedire che dei minimi spostamenti e rotazioni della testa condizionino l'esperienza di navigazione. Inoltre, dal momento che i quattro metri di pedana del CAVE non sono sufficienti a navigare ambienti vasti come per esempio tutta la lunghezza della fusoliera dell'aereo, il movimento longitudinale dell'utente è stato scalato mediante un fattore moltiplicativo. Quest'ultimo non è stato invece contemplato per gli spostamenti trasversali e verticali, che quindi corrispondono a quelli naturali.

Per dare la possibilità di interagire con gli ambienti, in essi è stato inserito un avatar, ovvero un manichino virtuale avente lo scopo di ripercorrere i movimenti effettuati dall'utente. L'altezza dell'avatar è stata adattata a quella di ogni utente: in funzione di tale valore immesso al partire dell'applicazione di navigazione, l'avatar viene opportunamente scalato, essendo nota la sua altezza e scala di origine. La funzionalità dell'interazione non è stata predisposta nel caso del visore Microsoft HoloLens, in quanto la prima versione di cui si dispone nel Laboratorio non prevede il tracking delle mani e quindi l'avatar non è stato considerato. Al contrario, nella piattaforma CAVE + Kinect la posizione dei giunti e del volto del manichino virtuale è stata collegata a quella dell'utente, quest'ultima catturata dal sensore Kinect come descritto nel paragrafo 5.2. Di conseguenza, l'avatar replica i movimenti dell'utente ed è in grado di entrare in contatto (collisione) con gli oggetti dell'ambiente virtuale, al che un riscontro audio-visivo mediante suoni ed effetto scintilla è fornito all'utente per rivelare e simulare l'interazione avvenuta. Per rendere maggiore la consapevolezza di quest'ultima, nel CAVE l'utente dispone anche di una vista esocentrica che permette di identificare la parte del corpo con la quale l'avatar è entrato in collisione, non sempre velocemente individuabile mediante la prospettiva visuale dell'utente allo scaturire del suono di interazione (Fig.56).



Fig.56: navigazione secondo la prospettiva visuale dell'utente, interazione mediante avatar e vista esocentrica (in basso a destra) durante le simulazioni human-in-the-loop volte a valutare un ambiente di cabina mediante la piattaforma CAVE + Kinect

Un possibile sviluppo futuro all'interno del progetto CASTLE potrebbe riguardare la compilazione *online* dei questionari di valutazione degli ambienti. Finora tale processo è stato eseguito mediante questionari cartacei dopo aver terminato la navigazione. Il vantaggio di effettuarlo durante le simulazioni *human-in-the-loop* ridurrebbe i tempi e migliorerebbe l'attendibilità delle risposte degli utenti, in quanto esse verrebbero fornite guardando direttamente al modello che si sta navigando, senza dover ricordare a posteriori le esperienze vissute durante gli esperimenti. Il concetto di questionario *online* è stato già efficientemente testato (Fig.57) e risulta pronto per essere implementato all'interno del *framework* del progetto CASTLE.



Fig.57: esempio di valutazione della cucina di bordo mediante un questionario online

Tale nuova opportunità di svolgere gli esperimenti mediante una *user interface* ha portato alla concezione di una *navigazione interattiva* che è anche *assistita*. Essa non solo apporterebbe i suddetti benefici al processo di valutazione degli ambienti virtuali, ma permetterebbe anche di intraprendere la strada verso la formulazione di nuove forme d'interazione uomo-computer, ovvero verso una *natural user interface* atta ad eliminare gli ostacoli alla comunicazione tra uomo e macchina ed impiegabile in numerosissimi contesti di applicazione [82] [83] [84] [85] [86] [87].

Per riassumere, la navigazione di ambienti virtuali secondo la prospettiva visuale dell'utente implementata mediante il sensore di tracking Kinect è stata successivamente integrata con una funzione interattiva ed una *user interface*. Ne risulta una *navigazione interattiva assistita* realizzata mediante un processo in cascata che ha visto il succedersi delle seguenti operazioni:

1. importazione dei modelli 3D e costruzione degli ambienti virtuali mediante il *game engine* Unity;
2. integrazione del sensore Kinect col Sistema di proiezione CAVE;
3. implementazione del tracking del corpo e del viso in linguaggio di programmazione C# mediante librerie di sviluppo software per Kinect;
4. integrazione dei due *stream* di tracking del corpo e del viso all'interno di Unity;
5. calcolo ed aggiornamento costante della matrice di proiezione degli schermi in modo solidale al punto di vista dell'utente che si muove sulla pedana del CAVE;
6. implementazione dell'azione di filtro dei movimenti;
7. introduzione di un avatar nell'ambiente virtuale per replica dei movimenti dell'utente;

8. implementazione dell'interazione tra l'avatar e gli oggetti dell'ambiente virtuale mediante feedback audio-visivo.
9. Integrazione di una *user interface* per assistere l'utente durante la navigazione.

Sebbene queste operazioni descritte siano state applicate nell'ambito del progetto CASTLE, nel loro insieme esse costituiscono un pattern procedurale fondamentale per l'implementazione di un concetto di *navigazione interattiva assistita* che è indipendente dal contesto applicativo considerato. In tal senso, ne risulta che lo schema operativo sopra riportato non è destinato ad essere confinato all'approccio progettuale HCD legato unicamente agli interni di cabina di aerei, bensì ad essere generalizzato ed applicato a numerosi ambiti e filoni di ricerca, come quello molto promettente delle *natural user interfaces*. Tale carattere flessibile evidenziato sarà reso ancora più marcato nel prossimo paragrafo conclusivo del presente capitolo sul tracking, dove la *navigazione interattiva assistita* si svincolerà anche dal sensore Kinect fin qui considerato, per risultare adattabile a qualunque hardware di tracking.

5.5 *Kinect Abstraction Interface* – tracking svincolato dall'hardware Kinect

Il presente capitolo sugli sviluppi di tracking implementati durante il corso di dottorato di ricerca è iniziato con la descrizione del sensore Kinect, rivelatosi il fulcro su cui si è incentrato il concetto di *navigazione interattiva assistita*, sviluppata e realizzata come fin qui riportato. In questo paragrafo conclusivo il cerchio si chiude ritornando al sensore Kinect, non focalizzandosi sul suo hardware, di cui, come si vedrà a breve, si potrà fare anche a meno, bensì concentrandosi sulla sua interfaccia software di acquisizione dati, sviluppata durante i primi due anni di studio (come descritto nei precedenti paragrafi) e resa flessibile nell'anno finale. In particolare, durante il periodo di studio all'estero presso il Through-life Engineering Services Institute dell'Università di Cranfield (UK), tutte le applicazioni implementate nel Laboratorio di Realtà Virtuale del Dipartimento di Ingegneria Industriale del campus di Forlì non potevano essere utilizzate perché il sistema di tracking operativo nel laboratorio d'oltremarina era diverso dal Kinect e rappresentato dall'ART Tracking System. Esso è basato su *markers* (Fig.58) e su un gruppo di telecamere equipaggiate con emettitori IR e quindi funzionanti mediante il principio del *time-of-flight* (Fig.30), sfruttando le proprietà retroriflettenti dei *markers*.



Fig.58: sistema di tracking ART facente uso di markers. Fonte: [ref 33](#)

La descrizione della piattaforma del sistema ART è riportata in Fig.59. L'uso di più telecamere è implementato per rendere maggiormente robusto l'output di tracking risultante dalle misurazioni di ciascun dispositivo. Il protocollo di trasporto utilizzato è quello UDP e le misure di tracking vengono inviate facendo riferimento ad una frequenza di sincronizzazione tra le diverse videocamere. I pacchetti dati viaggiano lungo connessioni *Ethernet* verso le applicazioni eseguite da uno o più pc raggruppati sulla rete LAN. La documentazione relativa a tale sistema fornisce codice sorgente in C++ su come ricevere e fare il *parsing* dei dati di *output* del sistema di tracking (contenuti nei pacchetti UDP).

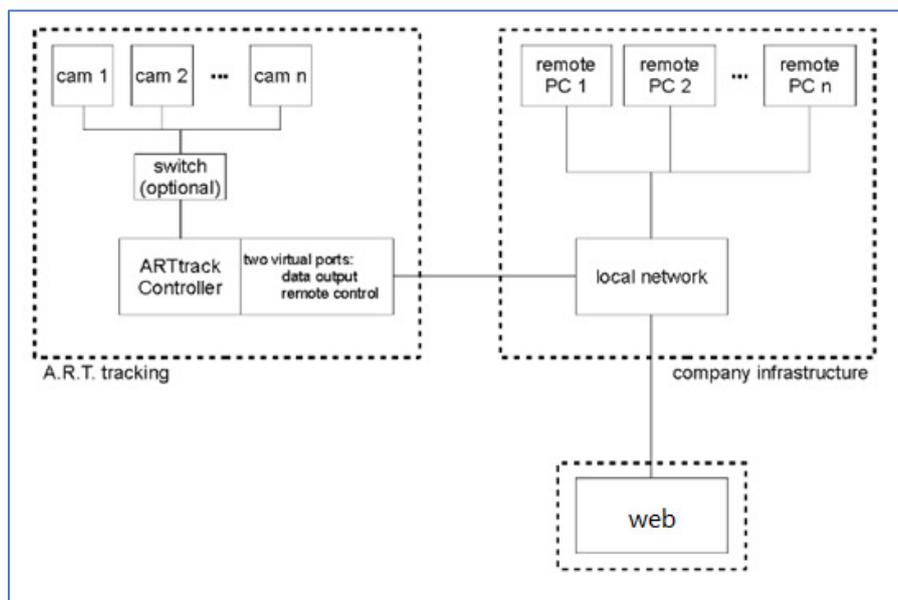


Fig.59: la piattaforma del sistema di tracking ART. Fonte: documentazione ART

Al fine di risolvere la problematica relativa alla dipendenza delle applicazioni di navigazione ad un particolare hardware di tracking, l'interfaccia software sviluppata per il Kinect:

1. è stata dapprima modificata (Fig.60 – 1° step) per acquisire dati di tracking specifici del sistema ART;
2. ed è stata successivamente generalizzata in una cosiddetta “Kinect Abstraction Interface” (Fig.60 – 2° step) per acquisire dati da qualunque hardware operante con un numero minore o uguale di punti di interesse soggetti a tracking e collocati in linea di massima nelle stesse posizioni di quelli del Kinect (Fig.32); per esempio, se lo scopo è effettuare solamente il tracking della testa, si può utilizzare l'IMU del dispositivo Microsoft Hololens (Fig.60 – 3° step).

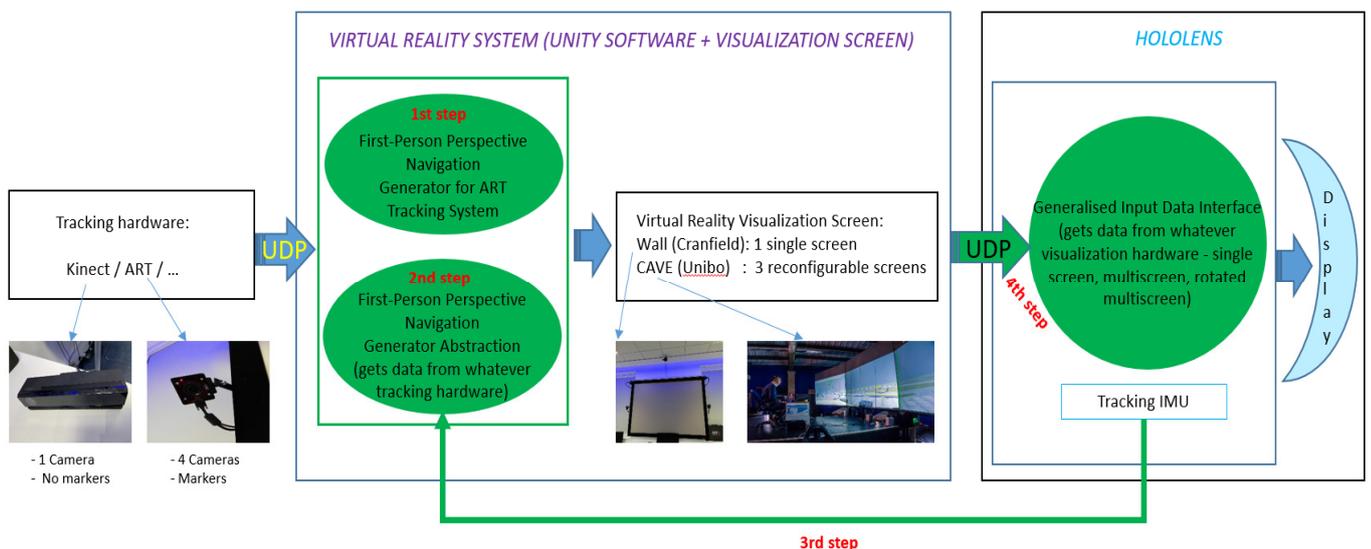


Fig.60: modifiche all'interfaccia software del Kinect per l'acquisizione dei dati dal sistema di tracking ART (1° step) e successivamente da qualunque hardware di tracking (2° step), come per esempio dall'IMU dell'Hololens (3° step).

Il 4° step del suddetto schema rappresentato in Fig.60 introduce il seguente capitolo sulla concettualizzazione, sviluppo ed integrazione di un framework di Realtà Aumentata: l'implementazione della connessione UDP in input al dispositivo Microsoft Hololens fornisce ad esso dei dati di riferimento indispensabili per registrare gli AR overlays con gli ambienti

- *virtuali*: riprodotti dai sistemi di proiezione su schermi nelle sperimentazioni di laboratorio;
- *reali*: caratterizzanti il mondo fisico attorno all'utente nelle applicazioni operative del mondo dell'industria e dei servizi.

6 Concettualizzazione, sviluppo ed integrazione di *frameworks* di Realtà Aumentata

In tale capitolo verranno descritte tutte le principali implementazioni di interfacce AR realizzate mediante il dispositivo Microsoft HoloLens HMD. La trattazione parte da un'analisi del dispositivo e quindi l'attenzione verrà inizialmente focalizzata sulla sua interfaccia utente, sul processo di costruzione delle applicazioni, nonché sulle caratteristiche della nuova seconda versione da pochi mesi sul mercato europeo. Successivamente la trattazione sarà incentrata sulle prime interfacce pilota AR realizzate all'interno del Laboratorio di Realtà Virtuale del Campus di Forlì. Tali implementazioni hanno costituito i componenti fondamentali su cui sono state costruite la piattaforma di validazione del progetto europeo RETINA, nonché il *sistema AR adattivo, collaborativo e video-based*, risultato finale del corso di dottorato di ricerca.

6.1 Il dispositivo Microsoft HoloLens Head-Mounted Display

Sebbene da qualche mese la seconda versione del dispositivo sia stata resa disponibile ad un numero limitato di clienti europei, la trattazione sarà qui incentrata sulla prima versione, di cui si è fatto uso durante l'intero corso di dottorato. Tuttavia, avendo potuto valutare direttamente il nuovo visore durante il periodo di studio all'estero di sei mesi presso il Through-life Engineering Services Centre dell'Università di Cranfield, a fine paragrafo verranno riportati i principali sviluppi caratterizzanti la seconda versione.

Il dispositivo Microsoft HoloLens fornisce interfacce AR mediante lo schema di funzionamento evidenziato nelle figure 22 e 23 del capitolo 4 sullo stato dell'arte delle tecnologie dell'AR. Il visore è caratterizzato da un *display* FSC LCoS e da un modulo di tracking operante mediante la tecnica SLAM che si basa essenzialmente sull'integrazione di telecamere di profondità ed IMU. Il funzionamento in dettaglio è stato descritto ai paragrafi 4.1 e 4.2, dove viene trattato il panorama tecnologico dell'AR inerentemente ai *display* ed ai moduli di tracking rispettivamente. In questo paragrafo sul dispositivo Microsoft HoloLens l'attenzione è focalizzata su:

1. interfaccia utente (*user interface*);
2. modus operandi nella costruzione delle sue applicazioni;
3. sviluppi caratterizzanti la seconda versione del dispositivo rilasciata recentemente sul mercato europeo.

6.1.1 Interfaccia utente

Una volta indossato il dispositivo Microsoft HoloLens, l'utente interagisce attraverso un'interfaccia utente (Fig.61) composta da pannelli olografici (AR *overlays*) che si sovrappongono alla visione sul mondo reale e permettono di:

- calibrare il dispositivo per adattare l'esperienza olografica all'utente operativo (pannello *Calibration*);
- istruire i nuovi utenti su come interagire con il dispositivo (*Learn Gestures*);
- richiedere assistenza a Cortana, software di riconoscimento vocale sviluppato da Microsoft che fornisce supporto all'utente (*Cortana*);
- configurare il sistema (*Settings*);
- interfacciarsi con il web (*Microsoft Edge*);
- visualizzare/condividere immagini e riprese catturate con il visore (*Photos*);
- eseguire applicazioni native (*Built-in apps*);
- eseguire applicazioni caricate dall'utente sul dispositivo (*Apps*);
- importare applicazioni dal web (*Windows Store*).



Fig.61: l'interfaccia utente Microsoft HoloLens

L'interfaccia utente è navigabile mediante tre modalità di interazione: orientamento della testa, *gestures* (ovvero particolari gesti effettuati muovendo le mani) e riconoscimento vocale. Esse non solo permettono di selezionare voci della *user interface*, ma influenzano anche in modo diretto il processo di progettazione di quelle applicazioni in cui sarà richiesto di interagire con l'utente mediante *input* forniti dallo stesso.

Orientando il proprio sguardo, l'utente dispone di un puntatore in grado di preselezionare (a) le voci della *user interface* di Fig.61, (b) gli ologrammi delle applicazioni e (c) le *meshes* dello *spatial mapping* (ricostruzione 3D dell'ambiente circostante, come descritto nel paragrafo 4.2 sul tracking). Ciò è rappresentato in Fig.62, dove è anche mostrata la presenza di un cursore per dare all'utente la possibilità di comprendere quale oggetto del mondo virtuale (*user interface* o ologrammi delle applicazioni) o reale (*meshes* dello *spatial mapping*) è effettivamente centrato dal "puntatore".

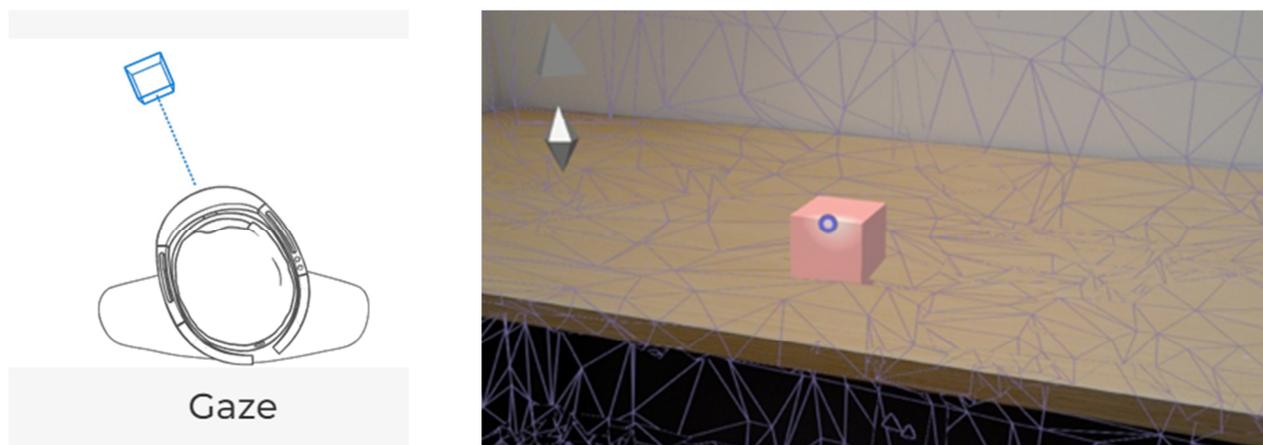


Fig.62: orientando diversamente la testa, l'utente è in grado di mirare ad ologrammi oppure a *meshes* dello *spatial mapping* (modello triangolare dell'ambiente 3D circostante) e quindi rispettivamente ad oggetti del mondo virtuale o reale. Un cursore (anello blu sull'ologramma rappresentato da un cubo rosa nell'immagine di destra) è introdotto come feedback per l'utente al fine di comprendere dove è effettivamente centrata la direzione dell'orientamento della propria testa. Fonte: [ref 34](#)

Una volta mirato un oggetto mediante il *gaze*, lo *step* successivo diventa quello di dare la possibilità di agire sull'oggetto stesso (selezionarlo in caso di *user interface*, oppure associare ad esso una qualsiasi azione in caso degli ologrammi delle applicazioni o *meshes* dello *spatial mapping*). Ciò viene predisposto attraverso le *gestures* o la voce.

La prima versione del dispositivo Hololens non effettua riconoscimento delle mani, bensì di tre gesti con esse effettuati. Questi vengono rilevati all'interno di un FOV trapezoidale posto davanti all'utente e sono identificati rispettivamente con i nomi "*bloom*", "*ready*" e "*tap*" (Fig.63). Quando le mani non sono all'interno dell'area visiva del dispositivo, oppure non effettuano i tre gesti riportati, esse verranno ignorate dall'Hololens.

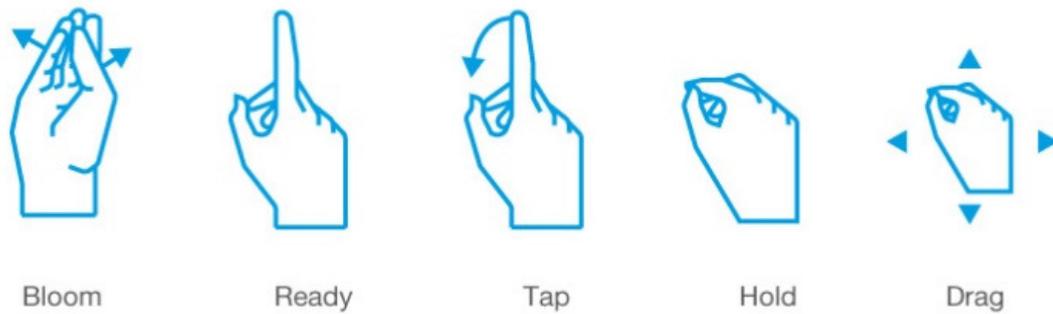


Fig.63: gestures identificate dal dispositivo Microsoft HoloLens versione 1. Fonte: [ref_35](#)

Per quanto riguarda la navigazione dell'interfaccia utente:

- con il gesto *bloom* si ritorna al menu iniziale mostrato in Fig.61,
- la successione rapida *ready-tap-ready* permette di realizzare il pattern “*press and release*” che corrisponde al comando di “selezione” (equivalente al *click* del mouse),
- la sequenza *ready-tap* crea il gesto *hold* con cui è possibile spostare, ridimensionare e scorrere le finestre mediante movimento della mano.

Eccezion fatta per la *gesture bloom*, che presenta il significato descritto sopra, è possibile programmare combinazioni di sequenze composte da *ready* e *tap* (magari anche temporizzate) per creare nuove *gestures* da utilizzare come *input* nelle applicazioni costruite.

In relazione alla modalità di interazione vocale, essa è supportata dallo stesso software caratterizzante tutte le applicazioni costruite mediante la Piattaforma Universale Windows (descritta in 6.1.2). Il riconoscimento vocale può essere utilizzato in sostituzione o integrazione delle *gestures*. Nel primo caso, l'interazione vocale permette di evitare il processo di digitazione su tastiere olografiche mediante puntamento con la testa e *gestures*, così da accelerare le procedure di inserimento del testo e ridurre l'*effort* dell'utente. Nel secondo caso, l'interazione vocale risulta particolarmente utile quando nelle applicazioni HoloLens vengono progettate interfacce utente molto complesse, in cui, insieme alla modalità di procedere di finestra in finestra mediante selezioni attraverso *gestures*, si fornisce all'utente anche la possibilità di saltare diversi blocchi di menu annidati mediante un solo comando vocale. Ciò velocizza il processo di navigazione e va quindi a favore di una progettazione di una *natural user interface* (si veda il paragrafo 5.4 relativo al Progetto CASTLE) altamente efficiente.

6.1.2 Il processo di costruzione delle applicazioni

Sebbene le applicazioni per Hololens possano essere create anche con il *game engine* Unreal, Unity rappresenta quello maggiormente documentato e, per tale motivo, si è fatto riferimento ad esso nella costruzione delle applicazioni. Questa può essere immaginata simile al processo di realizzazione di una pagina web: si organizza la struttura in modo gerarchico (il pannello *Hierarchy* in Unity, l'*html* per la pagina web, confrontati in Fig.64), successivamente si associa un comportamento agli elementi della struttura (in Unity si associano gli *scripts* in linguaggio *c#* o *javascript* agli oggetti del pannello *Hierarchy*, mentre nel caso della pagina web si associano gli *scripts* in linguaggio *javascript* ai *tag* dell'*html*), prevedendo, in caso, anche l'interazione tra comportamenti (ovvero l'interazione tra *scripts*, come per esempio nel caso in cui l'esecuzione di un codice di programmazione dipende dall'*output* di un altro).

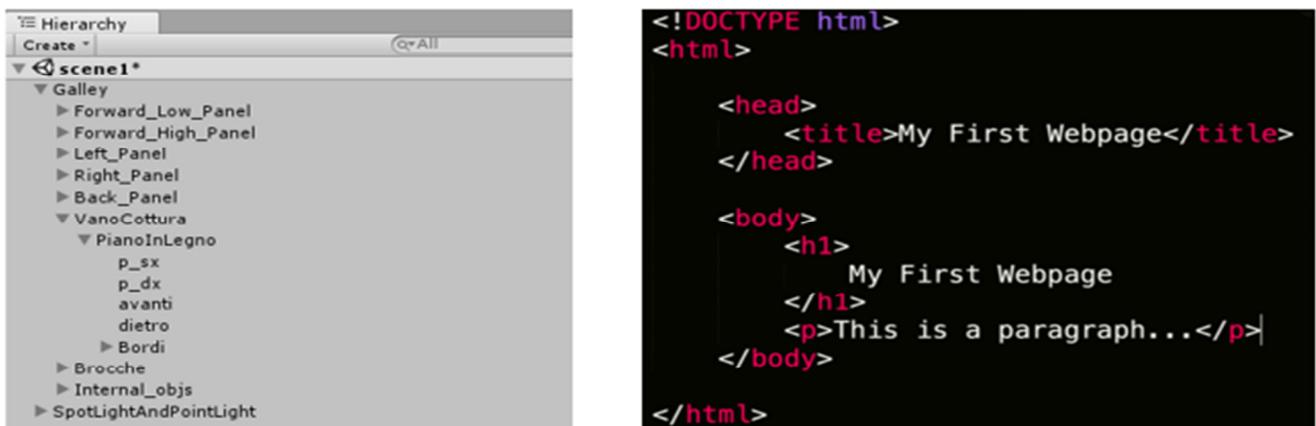


Fig.64: strutture gerarchiche del pannello *Hierarchy* di Unity (sinistra) e del codice *html* di una pagina web (destra)

Una volta realizzate le applicazioni, si effettua il *deploy* delle stesse, ovvero la distribuzione sul software dell'emulatore Hololens o sul *device* stesso. Per fare ciò, il primo passo da compiere è quello di far conoscere a Unity che l'*App* da esportare non dovrà essere caratterizzata da una visione 2D, bensì da una esperienza immersiva *olografica*. Questo viene garantito abilitando gli SDK (Software Development Kit) preposti ad ottemperare a tale specifica funzione, come in Fig.65, dove nel caso in esame Unity individua solamente il Mixed Reality SDK della piattaforma Windows 10.



Fig.65: abilitazione in Unity del Mixed Reality SDK della piattaforma Windows 10

Lo *step* successivo è quello di “impacchettare” l’applicazione per renderla compatibile con la Piattaforma Universale Windows (più comunemente conosciuta col termine inglese di *Universal Windows Platform*). L’aggettivo “Universale” non fa riferimento all’hardware di destinazione dell’app, bensì a risorse ed API (Application Programming Interface) condivise a cui hanno accesso gli sviluppatori durante il processo di realizzazione della stessa. In altre parole, lo scopo di tale piattaforma creata da Microsoft ed introdotta in Windows 10 è quello di fornire supporto agli sviluppatori, mettendo a disposizione degli stessi un *set* di risorse e strumenti condivisi da impiegare nei processi di realizzazione, configurazione, gestione, aggiornamento e sicurezza di app “universali”, tali da poter essere eseguite su dispositivi differenti, senza la necessità di dover riscrivere o modificare la struttura delle stesse per renderle compatibili con ognuno di essi (Fig.66).



Fig.66: la Piattaforma Universale Windows atta alla creazione di app eseguibili su dispositivi differenti. Fonte: [ref_36](#)

Una volta creato il pacchetto di compatibilità con la Universal Windows Platform, la procedura conclusiva è quella di caricarlo nell'ambiente di sviluppo integrato (Integrated Development Environment o IDE) Visual Studio (sviluppato da Microsoft), il quale infine verrà incaricato di *compilarlo* e “*deployarlo*” sull'emulatore Hololens o sul dispositivo stesso, nello stesso modo in cui agisce un ufficio postale che controlla e spedisce un pacco a destinazione.

Il *loop* di creazione delle applicazioni *Unity-Visual Studio-Emulatore-Hololens* è riportato in Fig.67. Al fine di velocizzare il processo, è buona prassi testare l'applicazione sull'emulatore, per poi infine collaudarla in ultima istanza sul *device*. Vi sono anche altre modalità di passaggio da Unity al dispositivo, escludendo Visual Studio dal processo. Tuttavia, la metodologia descritta rappresenta lo standard maggiormente condiviso.

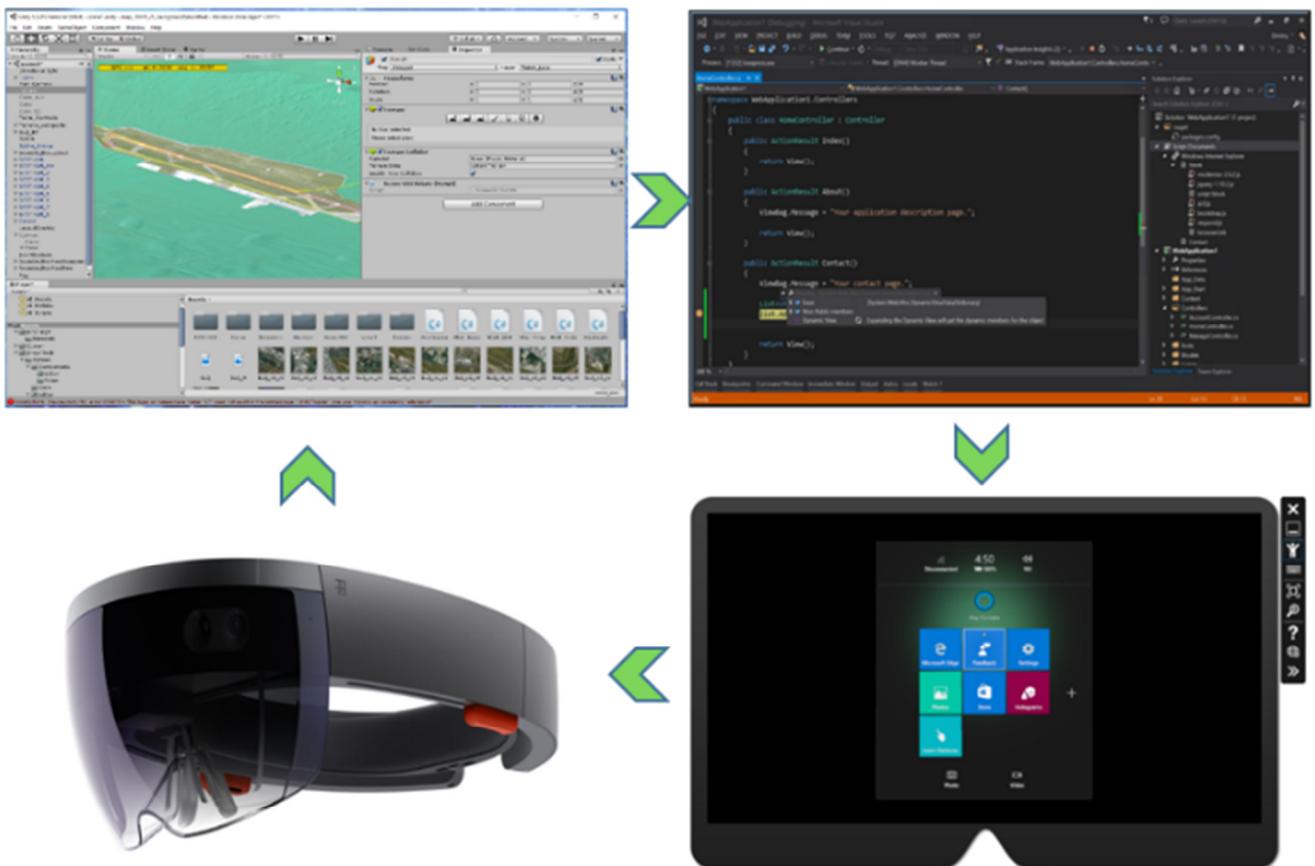


Fig.67: loop di creazione delle applicazioni da Unity (in alto a sinistra) a Hololens (in basso a sinistra): *Unity-Visual Studio-Emulatore-Hololens*

Una volta allineate le funzionalità implementate nell'applicazione con quelle derivanti dalle specifiche di progetto mediante il *loop* di creazione delle applicazioni riportato in Fig.67, il test finale di un'applicazione consiste nel valutare gli impatti da essa prodotti sulle *performances* di

sistema del dispositivo HoloLens. Per fare ciò, si fa uso del *Windows Device Portal for HoloLens*, ovvero di un *web server* ospitato dal dispositivo. Al *Device Portal* ci si connette mediante Wi-Fi o USB attraverso un portale di accesso al web (browser). Tale *web server* permette di configurare e gestire l'HoloLens da remoto ed include molti strumenti per effettuare il debug e l'ottimizzazione delle applicazioni. Di particolare interesse sono quelli chiamati *Processes* e *System Performance*. Il primo mostra in dettaglio tutte le applicazioni ed i processi di sistema al momento in esecuzione sul dispositivo. Il secondo strumento visualizza le performances di sistema. Nello specifico, riporta i grafici *real-time* delle informazioni diagnostiche di sistema (Fig.68), tra cui le seguenti metriche principali:

- *Frame rate* (frequenza fotogrammi): fotogrammi al secondo;
- *GPU*: percentuale del totale disponibile;
- *CPU*: percentuale del totale disponibile;
- *I/O*: letture e scritture;
- *Rete*: dati ricevuti e inviati;
- *Memoria*: memoria totale, in uso, riservata disponibile.

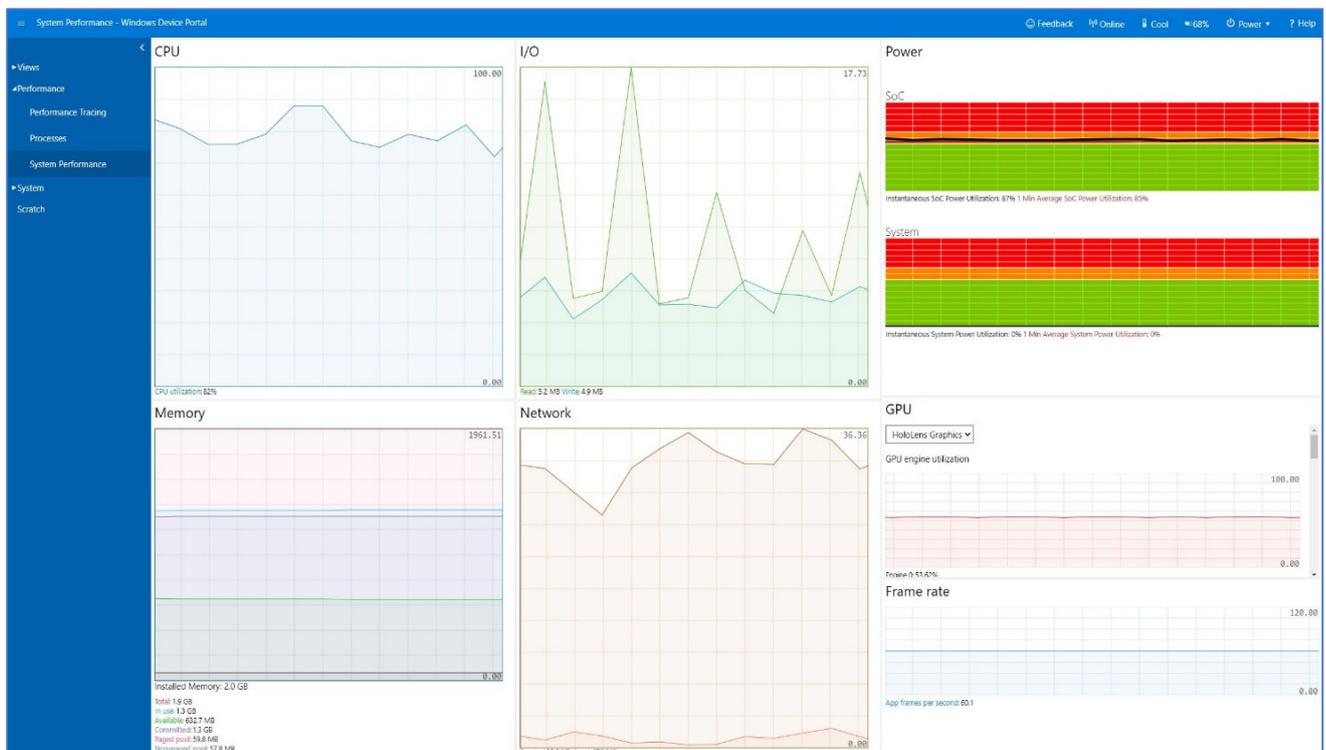


Fig.68: performances di sistema del dispositivo Microsoft HoloLens visualizzati mediante il Device Portal.

L'uso di tali strumenti è di importanza fondamentale ai fini di una corretta valutazione dell'applicazione creata. Spesso mediante essi ci si accorge di malfunzionamenti o veri e propri *crash* (uscita improvvisa dall'applicazione) non poco dopo l'inizio del funzionamento dell'applicazione, bensì solamente dopo molti minuti di esecuzione della stessa. Questo è il caso per esempio della problematica nota come *memory leak* ("perdita o fuoriuscita di memoria") costituita da un consumo di memoria dovuto alla mancata eliminazione di dati non più utilizzati da parte dei processi, ma che si accumulano durante l'esecuzione e generano il *crash* dell'applicazione. Analogamente, per risolvere problematiche legate ad una bassa frequenza di fotogrammi (applicazioni lente o a scatti), occorre spesso incrociare le informazioni relative al *frame rate* insieme a quelle riguardanti la *GPU* e *CPU*, nonché il numero e tipo di processi in esecuzione parallela, per poi così valutare se intervenire sugli aspetti di *rendering*, oppure su quelli relativi ad un uso non corretto o ridondante del codice di programmazione, o, in ultima analisi, su una riduzione o diversa organizzazione degli oggetti virtuali visualizzati nell'applicazione e disposti nel pannello *Hierarchy* del software Unity (per maggiori informazioni si veda paragrafo 5.3). Infine, nel caso di applicazioni che contemplano l'uso di flussi di *input/output*, l'analisi degli *stream* (letture/scritture da/su files, dati ricevuti/inviati dalla/alla rete) permette di ottenere informazioni molto importanti sulla quantità di informazione da trattare e sulle modalità di gestione della stessa, al fine di evitare le suddette problematiche di *memory leak* e di frequenza di fotogrammi. Infatti, una inefficiente gestione dei flussi di *input* ed *output* rischierebbe di causare un esaurimento delle allocazioni di memoria disponibili per eseguire l'applicazione, oppure l'incapacità di processare l'informazione a causa di un eccessivo carico computazionale a cui vengono sottoposte la *CPU* o la *GPU*.

6.1.3 Microsoft Hololens versione 2

Da qualche mese la seconda versione è stata resa disponibile ad un numero limitato di mercati nazionali. Tra questi vi è l'Inghilterra, dove ho avuto modo di valutare il nuovo dispositivo durante il periodo di studio all'estero di sei mesi presso il Through-life Engineering Services Centre dell'Università di Cranfield. Le specifiche tecniche dei due visori sono messe a confronto in Fig.69.

Specs	HoloLens 1	HoloLens 2
Display resolution	1280×720 (per eye)	2048 × 1080 px (per eye)
Holographic density	>2.5K radiants (light points per radian)	>2.5K radiants (light points per radian)
Field of view (FOV)	34°	52°
Weight	579 grams	566 grams
Eye-tracking	no	yes
Processor	Intel 32-bit (1GHz)	Qualcomm Snapdragon 850
Camera	2.4 MP, HD video	8MP stills, 1080p video
Flip-up visor	no	yes
Voice command	yes	yes
Biometric Security (Iris Scanning)	no	yes
Hand tracking	no	both hands full tracking

Fig.69: confronto tra le specifiche tecniche delle due versioni HoloLens.

Sebbene secondo le specifiche tecniche gli aspetti grafici della nuova versione risultino essere molto migliorati rispetto alla precedente, il dispositivo da me testato non rispecchiava affatto tali dichiarazioni ed anzi, ho potuto constatare uno sgradevole effetto arcobaleno molto marcato sul *display*, nonché una grafica di visualizzazione peggiore. Ciò però potrebbe derivare dal fatto che il dispositivo da me sperimentato rappresentava un prodotto ancora in fase di collaudo, essendo stata l'Università di Cranfield tra i primi acquirenti (se non addirittura l'unico in quel momento) sul mercato europeo a disporre della seconda versione del dispositivo. Ciononostante, esso presenta notevoli miglioramenti dal punto di vista del processore installato (molto più potente), del FOV (molto più ampio), nonché nuove funzionalità costituite dal tracking delle mani (Fig.70) e degli occhi.

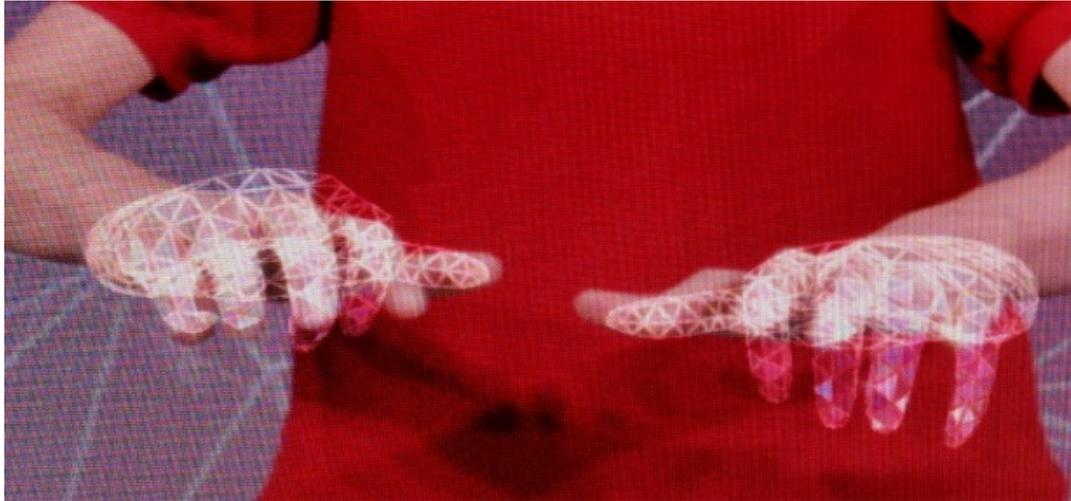


Fig.70: tracking delle mani nella seconda versione HoloLens. Fonte: [ref_37](#)

Dopo aver descritto le modalità di approccio con il dispositivo Microsoft HoloLens, ovvero le caratteristiche della sua *user interface*, le procedure di navigazione della stessa (coincidenti con quelle di interazione col dispositivo), nonché il modus operandi nella costruzione delle sue applicazioni, il prossimo paragrafo riguarderà la trattazione delle prime interfacce AR pilota che sono state con esso create.

6.2 Creazione delle prime interfacce pilota di Realtà Aumentata

Le iniziali applicazioni di registrazione delle interfacce AR con l'ambiente reale sono state implementate in ambiente *indoor* ed *outdoor*.

Per quanto riguarda il primo, è stata realizzata una navigazione guidata di un ambiente reale sfruttando le funzionalità dello spatial mapping e dell'IMU messe a disposizione dell'HoloLens. È stata sviluppata un'applicazione di guida di un percorso basata sull'indicazione di una direzione da percorrere per andare da una posizione A ad una posizione B. L'ambiente utilizzato per la sperimentazione è la sede di Forlì del Dipartimento di Ingegneria Industriale. L'applicazione implementa il tracciamento di un percorso di colore verde che porta dal Laboratorio di Realtà Virtuale situato al piano terra ad un ufficio posto al primo piano della sede (Fig.71).

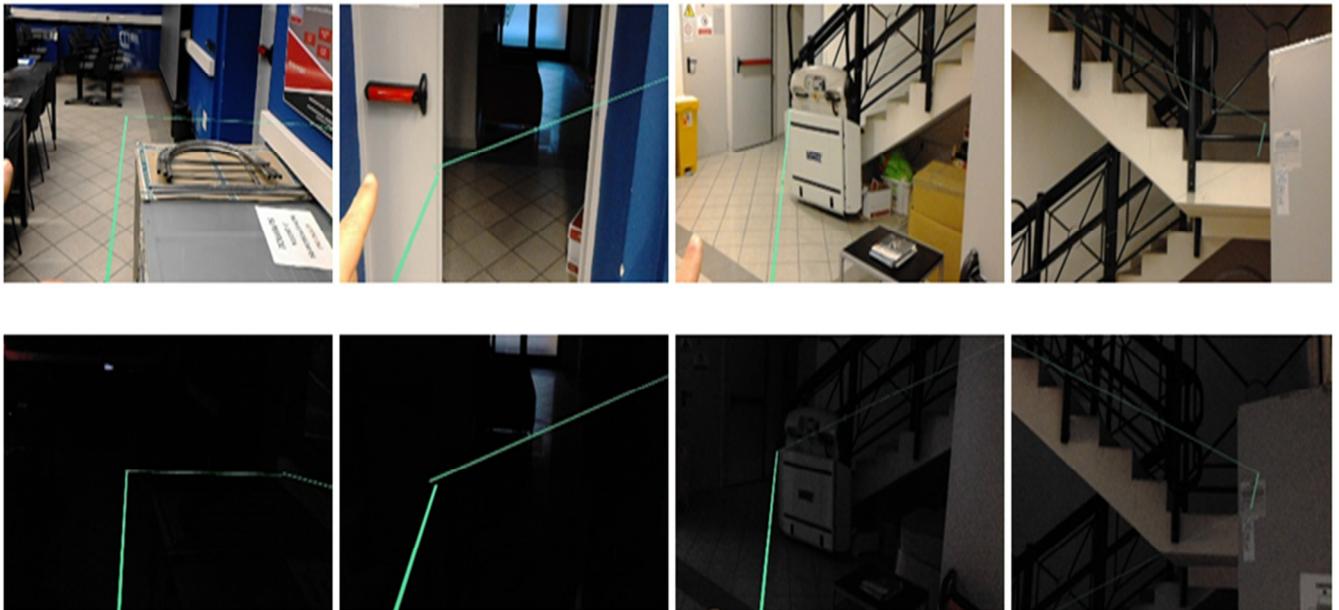


Fig. 71: navigazione guidata di un ambiente reale mediante tracciamento di un percorso di colore verde da parte del dispositivo Hololens – sequenza di quattro immagini (lab, uscita dal lab, atrio, scale) catturate dal visore in condizioni di buona e scarsa visibilità (rispettivamente riga sopra e riga sotto).

Lo *step* base di funzionamento dell'applicazione consiste nel collocare l'ologramma del percorso verde costruito con Unity in modo coerente con il reale cammino che deve essere tracciato dal dispositivo Hololens. Una volta scelto il posizionamento opportuno, l'ologramma viene ancorato mediante *gestures* allo *spatial mapping* effettuato da Hololens e contestualmente si effettua il salvataggio dei dati del posizionamento dell'oggetto virtuale *online* sul *web server* del *Device Portal* descritto in 6.1.2. Successivamente, durante gli usi futuri dell'applicazione, non occorrerà ripercorrere lo *step* base di posizionamento dell'ologramma del percorso verde, in quanto al partire della stessa, l'Hololens riconosce l'ambiente già mappato in precedenza e dal *Device Portal* basterà scaricare i dati di posizionamento salvati una volta per tutte sul *web server*, per ritrovare l'ologramma perfettamente registrato con l'ambiente reale. Ciò avviene in quanto l'Hololens effettua lo *spatial mapping* (1), riconosce l'ambiente (2) ed applica i dati di posizionamento dell'ologramma precedentemente salvati all'ambiente mappato (3). Tale procedimento del *matching* degli ologrammi con lo *spatial mapping* è analogo a quello del *matching* delle *textures* con le *meshes* di riferimento effettuato da Unity all'atto dell'importazione di un modello 3D in formato *obj*: insieme a questo deve essere contestualmente importato anche il file formato *mtl* (prodotto in fase di realizzazione dell'*obj*) che presenta i dati di posizionamento secondo cui avviene la registrazione delle *textures* con le *meshes* di riferimento.

Oltre ad accelerare tutti i processi operativi che richiedono di effettuare una registrazione iniziale delle interfacce AR con il mondo reale, il salvataggio dei dati di ancoraggio permette anche di conseguire risvolti applicativi importanti dal punto di vista della creazione di mappe relative ad ambienti al chiuso: esse possono fornire supporto agli operatori di primo soccorso per la navigazione di ambienti a loro sconosciuti ed in condizioni di danneggiamento strutturale a causa di calamità naturali, di incendio, di scarsa visibilità, ecc. . Così, il visore AR permette di massimizzare

la *situational awareness* di tali operatori impegnati in situazioni critiche per la sicurezza, evitando l'uso delle mani per la consultazione di mappe e garantendo una visione a testa alta del percorso da intraprendere.

Parimenti, il salvataggio dei dati di ancoraggio sul *web server* del *Device Portal* ha anche permesso di accelerare l'esecuzione degli scenari di validazione del progetto europeo RETINA: nelle simulazioni *human-in-the-loop* che hanno coinvolto i controllori del traffico aereo, il dispositivo Hololens era stato messo nella condizione di recuperare immediatamente i dati di registrazione delle interfacce AR con l'ambiente reale nel caso fossero subentrati malfunzionamenti del visore.

Per quanto riguarda gli ambienti *outdoor*, lo *spatial mapping* effettuato da Hololens può non essere di aiuto, specie in caso di ampi spazi aperti, dove l'uso della tecnica del *time-of-flight* mediante *stream IR* non riesce a garantire segnali di ritorno al visore. Così, dal momento che la tecnica SLAM è soggetta a fallire, nel caso di ambienti *outdoor* l'uso della localizzazione satellitare (funzionalità assente nell'Hololens) rappresenta la migliore soluzione alla problematica del calcolo ed aggiornamento della posizione. Per rispondere a tale esigenza, si è realizzata una applicazione in cui è stata implementata una comunicazione dati dal sistema operativo Android di un telefono smartphone al visore Hololens al fine di georeferenziare quest'ultimo. Attraverso tale flusso dati, il dispositivo Hololens è stato abilitato a conoscere: (1) le proprie coordinate GPS, (2) il proprio orientamento rispetto al Nord (*heading*), nonché (3) la posizione di altri oggetti attorno, come quella dell'aereo in Fig.72.



Fig. 72: immagine catturata dal dispositivo Hololens funzionante in ambiente outdoor: posizione GPS, heading e posizione aereo sono comunicati dal sistema Android del telefono a Hololens, che sovrappone i dati digitali AR (posizione, heading e mirino) alla visione sul mondo reale.

In tale applicazione, le stringhe di testo relative a posizione ed *heading*, nonché il mirino sono stati concepiti come AR *overlays* sempre rigidamente vincolati ad uno stesso punto del FOV Hololens, mentre l'overlay "Fire!" è attivato solo se il mirino al centro del FOV si registra con l'aereo (e quindi solo se l'orientamento della testa si allinea con la direzione dell'aereo).

Nelle applicazioni per sistemi di puntamento, come quelle a cui si riferisce la figura soprastante, per incrementare la *situational awareness* potrebbe risultare molto importante avere degli AR *overlays* sempre rigidamente vincolati ad uno stesso punto del FOV (si parla anche di "Terminator Vision HUD"). Tuttavia, in altri contesti, le specifiche potrebbero essere differenti. Si pensi ad una applicazione AR nel settore del turismo, dove un visore potrebbe fornire informazioni digitali relative alla storia di un monumento che si sta osservando. In questi casi l'overlay AR potrebbe essere non più vincolato rigidamente ad un preciso punto del FOV dell'Hololens, bensì trascinato in modo *smooth* coerentemente a dove si guarda. Così, in modo simile a quanto fatto per effettuare il tracking dell'utente nel progetto CASTLE (paragrafo 5.4), sono state realizzate applicazioni in cui al movimento dell'ologramma è stata applicata un'azione di filtro passa-basso che permette di evitare:

- 1) Uno spostamento a scatti dell'overlay per movimenti lenti della testa;
- 2) il fastidioso tremolio dell'overlay dovuto ai movimenti minimi a cui la stessa è inevitabilmente costantemente soggetta.

L'azione di filtro è stata dimensionata mediante un *trade-off* tra movimento *smooth* e latenza dello stesso, imponendo il vincolo che l'ologramma debba sempre trovarsi nel FOV dell'Hololens. Così, la velocità di trascinamento dell'ologramma non può essere troppo bassa, perché altrimenti esso uscirebbe dal FOV a seguito di movimenti rapidi della testa, ma, al tempo stesso, non può essere troppo alta, perché altrimenti l'ologramma si sposterebbe a scatti per movimenti lenti della testa. In ogni caso, il suddetto *trade-off* va valutato a seconda delle applicazioni e, più in particolare, in base alle caratteristiche di ampiezza e velocità di rotazione della testa a cui l'utente sarà sottoposto durante la specifica applicazione in esame.

In altre applicazioni risulta infine conveniente vincolare l'ologramma ad un punto dello spazio attorno all'utente, perché per esempio l'ologramma può godere della particolarità di essere visualizzato in modo sporadico, per cui non ha senso tenerlo costantemente nel FOV dell'utente. Quindi sono state realizzate applicazioni che, insieme all'azione di filtro degli spostamenti, prevedono anche il posizionamento fisso dell'overlay mediante *gesture ready-tap-ready* descritta in 6.1.1. In altre parole, in esse è possibile decidere tra l'opzione di avere l'overlay sempre nel FOV, oppure se ancorarlo in una parte fissa del mondo che circonda l'utente (in modo tale da avere il FOV completamente libero per altre funzionalità), per poi, in caso, riprendere nuovamente il trascinamento con una nuova *gesture ready-tap-ready*.

Così, riassumendo, le prime interfacce AR pilota sono state implementate sia in ambiente *indoor* che *outdoor*. In ciascuno di essi si sono stati considerati i tre seguenti scenari applicativi in ordine di rigidità del vincolo di ancoraggio dell'overlay AR ad un punto del FOV:

- vincolo rigido: overlay AR ancorato in un punto fisso del FOV;

- vincolo elastico: overlay trascinato in modo smooth da un punto fisso del FOV;
- vincolo assente: l'overlay non segue il FOV, ma è posizionato staticamente in un punto attorno all'utente.

Gli scenari descritti non sono stati introdotti a caso: essi rappresentano dei pattern di riferimento che sono stati implementati nel *framework* AR finale risultante da tale elaborato. Finora si è parlato prevalentemente di ancoraggio più o meno elastico degli ologrammi ad un punto del FOV dell'Hololens o di posizionamento statico degli stessi (registrazione statica, come nel caso dell'applicazione *indoor* del percorso verde lab-ufficio). Il passo successivo compiuto è stato quello di preparare la strada alla registrazione degli AR *overlays* con oggetti reali dinamici. Per fare ciò non si è operato nella direzione del riconoscimento immagini, bensì in quella di un invio di dati in *input* ad Hololens mediante connessioni Wi-Fi, dai quali il dispositivo estrapola informazioni sul posizionamento degli oggetti reali con cui registrare gli AR *overlays*. Tali *stream* di dati in *input* al visore hanno avuto le seguenti provenienze: *workstations*, *web* (siti e *databases*) e *smartphones*. Sulla prima origine si basano gli *stream* di comunicazione dati verso Hololens nell'architettura della piattaforma di validazione del progetto RETINA, descritta nel seguente paragrafo.

6.3 La piattaforma di validazione del progetto RETINA: applicazione dell'AR alle torri di controllo aeroportuali

L'architettura della piattaforma di validazione del progetto europeo RETINA, descritto in 3.2 ed atto ad investigare l'impiego dell'AR nelle torri di controllo aeroportuali, è riportata in Fig.73.

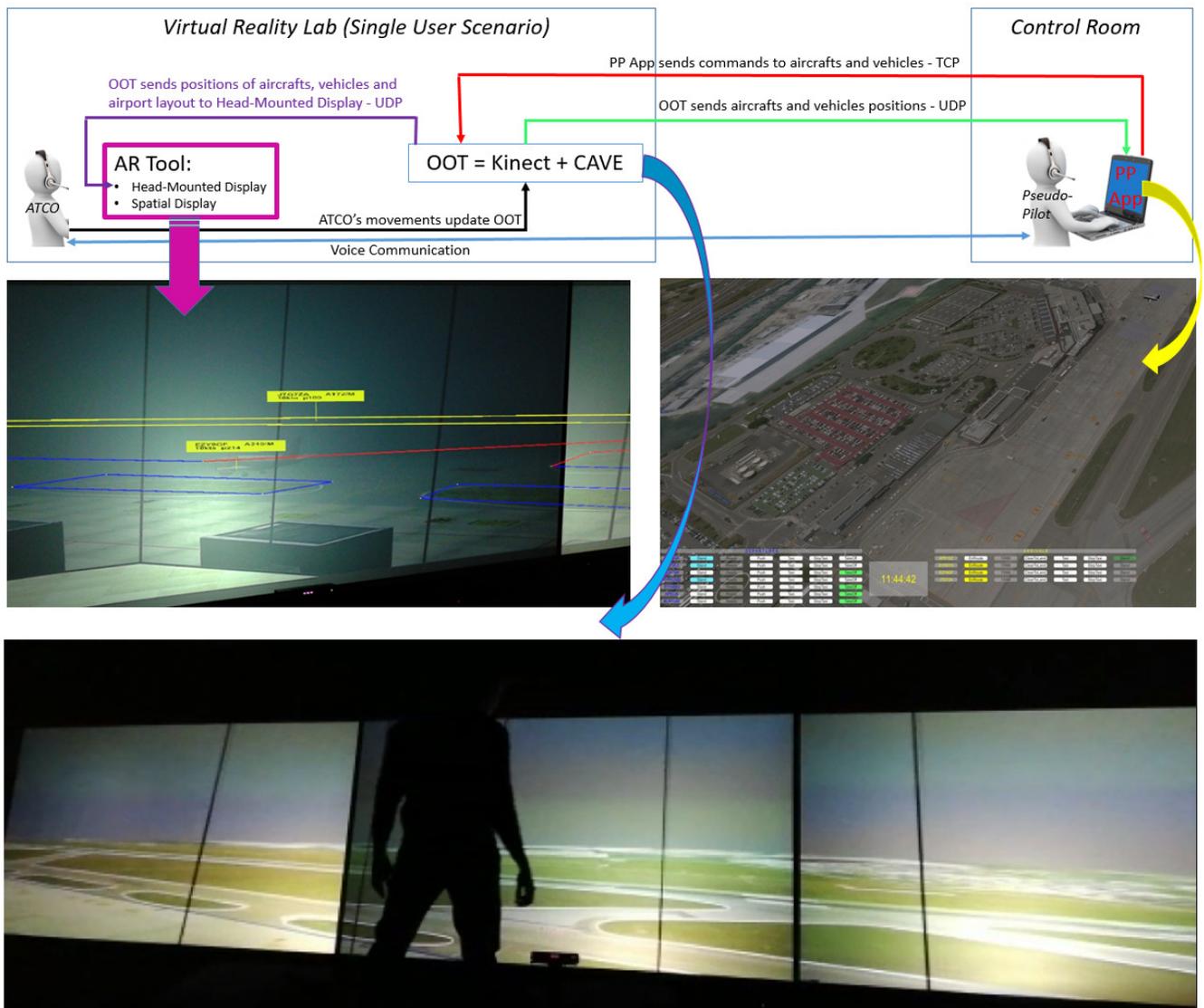


Fig. 73: architettura della piattaforma di validazione del progetto RETINA

L'architettura della piattaforma di validazione consiste in tre moduli principali: l'*Out of the Tower View Generator* (OOT), l'*Augmented Reality Tool* (AR Tool) e la *Pseudo-Pilot Application* (PP App).

Il cuore dell'intero sistema è rappresentato dall'*Out of the Tower View Generator*, ovvero dal sistema integrato CAVE + Kinect. I controllori del traffico aereo coinvolti nelle simulazioni *human-in-the-loop* sono stati in grado di navigare secondo la propria prospettiva visuale, calcolata ed aggiornata grazie al tracciamento continuo del punto di vista rilevato mediante il Kinect, l'ambiente virtuale della torre di controllo dell'aeroporto di Bologna riprodotto sul CAVE (in modo simile alla navigazione degli ambienti di cabina di aereo del progetto CASTLE riportata in 5.4). Le simulazioni sono state svolte presso il Laboratorio di Realtà Virtuale del Dipartimento di Ingegneria Industriale del Campus di Forlì (Fig.16), coinvolgendo singolarmente un controllore alla volta, a cui è stato richiesto di gestire sia la pista di volo che la movimentazione sulle vie di circolazione ed all'interno dei parcheggi. L'ambiente virtuale è stato implementato mediante una ricostruzione fotorealistica dell'Aeroporto di Bologna. Il modello 3D risultante è stato infine anche georeferenziato. Il traffico aereo simulato viene movimentato su curve *splines* formate da concatenazioni di curve di *Bezier*. Quest'ultime sono costruite per mezzo di sequenze di punti. La curva inizia nel primo e finisce nell'ultimo, ma non passa attraverso i punti intermedi. Essi servono a modellare la curva. Questa viene parametrizzata secondo un parametro $0 \leq t \leq 1$ e viene tracciata attraverso un processo di interpolazione [88]. Le curve di Bezier sono comunemente impiegate in applicazioni di *path planning* [89] [90].

Le modalità di assetto e le velocità degli aerei in arrivo ed in partenza che si muovono sulle *splines* del modello 3D sono state implementate prendendo a riferimento i comportamenti dei velivoli osservati nella realtà (Fig.74).



Fig. 74: particolare di un aereo Alitalia in decollo nel modello 3D.

Successivi sviluppi nel campo delle *applicazioni georeferenziate in tempo reale* hanno portato il Laboratorio di Realtà Virtuale a considerare l'ipotesi di riferirsi non più a scenari di traffico simulato, bensì in tempo reale. In particolar modo, qualora la piattaforma fin qui descritta venisse integrata con l'applicazione implementata mostrata in Fig.75, le simulazioni *human-in-the-loop*, mediante le quali i controllori sono stati immersi in un ambiente virtuale avente un alto grado di fedeltà rispetto al contesto operativo reale, potrebbero essere caratterizzate dall'aggettivo *real-time*.

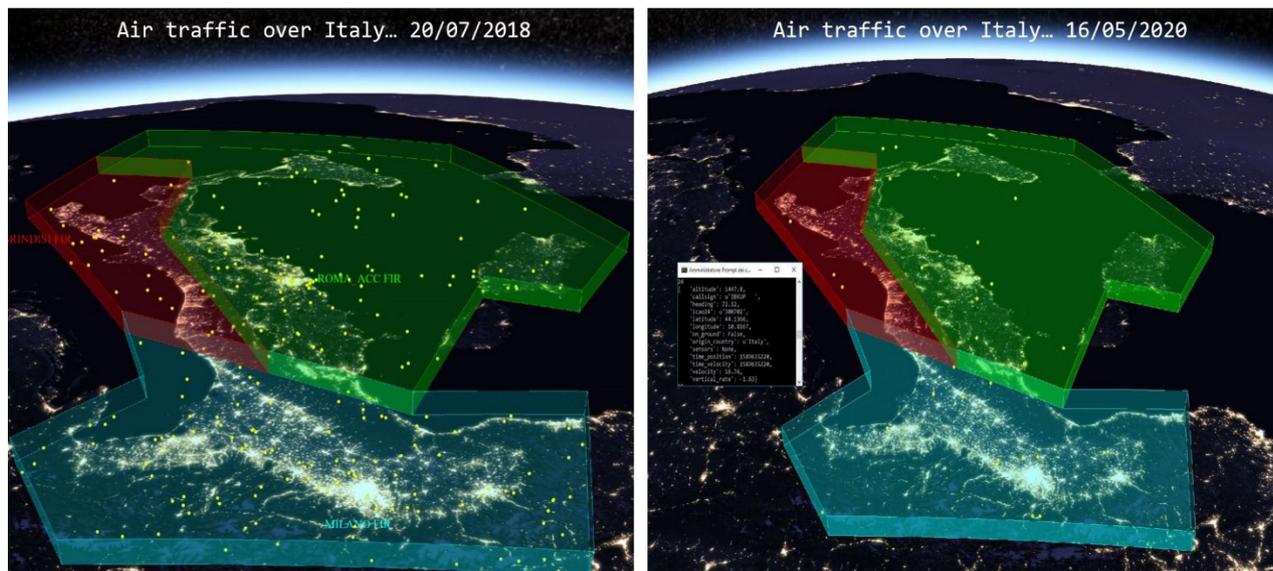


Fig.75: traffico aereo in tempo reale nello spazio aereo italiano diviso nei tre settori di controllo di Roma, Milano e Brindisi.

Il modello 3D è collocato al centro dell'architettura della piattaforma di validazione di Fig.73 non a caso: esso comunica attraverso protocolli di scambio dati con gli altri due moduli: PP App ed AR Tool.

La PP App è stata creata con il software Unity e permette allo pseudo-pilot di monitorare ed aggiornare lo stato dei voli nel modello OOT secondo le istruzioni (*clearances*) impartite dal controllore mediante una comunicazione vocale stabilita via software VoIP Skype. Essa quindi comunica con l'OOT per mantenere i dati coerenti fra le due applicazioni. Così, in base alle *clearances* decise dal controllore, mediante la PP App lo pseudo-pilot effettua le seguenti azioni:

- instrada gli aeromobili/veicoli su uno specifico percorso,
- arresta e fa ripartire gli aeromobili sulle vie di circolazione,
- autorizza i velivoli al decollo o all'atterraggio.

L'AR Tool, ovvero lo strumento AR impiegato nella validazione (visore Microsoft Hololens o SD coincidente con i vetri della torre di controllo), deriva i dati rilevanti dal modulo OOT e fornisce al controllore gli AR *overlays* correttamente registrati in tempo e spazio con gli oggetti statici e dinamici dell'ambiente virtuale rappresentato per mezzo dell'OOT. In relazione al visore, il funzionamento è il seguente (Fig.76):

1. il sensore Kinect effettua il tracking del controllore,
2. il CAVE aggiorna costantemente la matrice di proiezione degli schermi in modo solidale al punto di vista del controllore,
3. l'Hololens indossato dal controllore:
 - a. riceve dal CAVE i dati riguardanti il posizionamento del layout aeroportuale e degli aeromobili per mezzo di una connessione UDP,
 - b. converte le posizioni rispetto al proprio sistema di riferimento
 - c. costruisce la griglia del layout aeroportuale ed aggiorna il posizionamento delle *labels* degli aerei, sovrapponendo queste ed altre informazioni all'ambiente virtuale fornito dall'OOT, come riportato in Fig.76, dove gli *overlays* di colore ciano (*labels* degli aerei), blu (vie di circolazione), rosso (vie di circolazione proibite) e giallo (pista di volo con procedura di atterraggio in atto) costituiscono l'informazione digitale aggiuntiva sovrapposta allo scenario aeroportuale in condizioni di bassa visibilità.

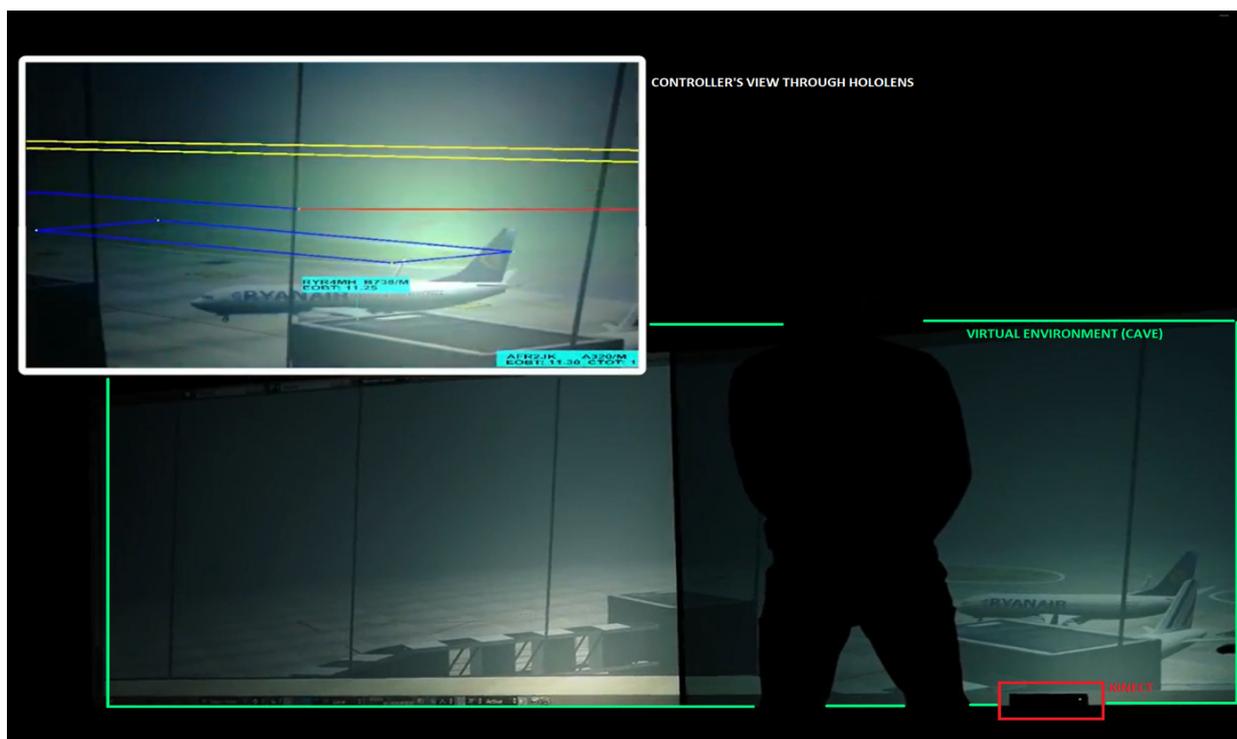


Fig. 76: AR Tool rappresentato dal visore Microsoft Hololens – pattern di comunicazione Kinect – CAVE – Hololens.

In relazione al *display* spaziale, come descritto in 3.2, la soluzione è stata simulata via software all'interno dell'ambiente virtuale immersivo OOT, in quanto tale tecnologia, che potrebbe essere pensata coincidere con *head-up display* dai formati molto ampi (delle stesse dimensioni dei vetri di torre), non è ancora in commercio al giorno d'oggi. In questo modo, il *display* spaziale è stato immaginato operativo in nove su trentadue vetri di torre, evidenziati per mezzo di un *frame* di colore bianco in Fig.77. Come risultato, l'informazione digitale aggiuntiva è resa visibile se l'oggetto a cui essa si riferisce è posizionato all'interno del *frame*, invisibile altrimenti.



Fig. 77: AR Tool costituito dal display spaziale simulato nel CAVE nel frame bianco (coincidente con nove vetri di torre su 32 totali).

Descritti i tre moduli principali componenti l'architettura della piattaforma di validazione, l'attenzione si focalizza sul sistema di comunicazione mediante il quale essi sono messi in comunicazione tra loro. Per permettere lo scambio di informazioni, sono stati considerati i protocolli di trasporto UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*), usati in combinazione con il protocollo di livello di rete IP.

Per quanto riguarda il primo, esso è caratterizzato da servizi di trasmissione ridotti all'osso ed è per questo molto leggero. In questo modo garantisce una trasmissione dati molto veloce, ma, al

contempo, inaffidabile: quando un processo invia un messaggio, non c'è alcuna garanzia che i dati giungano a quello di destinazione.

Al contrario, TCP offre molti servizi aggiuntivi alle applicazioni. Innanzitutto, garantisce un trasferimento dati affidabile. Mediante l'uso di

- controllo di flusso,
- numeri di sequenza,
- accettazioni (*acknowledgments*) e temporizzazioni (*timers*),

TCP garantisce la trasmissione dell'informazione dal processo mittente a quello destinatario. Per tale motivo, TCP è necessariamente un protocollo più complesso e quindi più lento di UDP, ma spedisce i dati senza perdita di *bytes*.

Dal momento che le applicazioni *real-time* richiedono trasferimenti dati ad alta velocità possono tollerare delle perdite di informazione, il modello di servizio TCP non è particolarmente adatto a tale tipo di applicazioni e contesti [91]. Così, le comunicazioni uscenti dal modulo OOT sono state implementate per mezzo del protocollo UDP: se durante la trasmissione viene perduto un *byte* con l'informazione relativa alla posizione di un aereo, il pacchetto dati mancante verrà subito sostituito da un altro caratterizzato da una nuova posizione.

D'altro canto, quando lo pseudo-pilot aggiorna il modulo OOT mediante la PP App, l'operatore deve assicurarsi che i comandi impartiti agli aeromobili od ai veicoli vengano trasmesse senza doverle ripetere più volte. Da ciò segue che la comunicazione PP App – OOT è stata implementata mediante il modello di servizio TCP.

Descritto il metodo di funzionamento del dispositivo Microsoft Hololens nelle validazioni del progetto RETINA, nonché la trasmissione dati UDP in esso entrante per la ricezione delle informazioni riguardanti il posizionamento del layout aeroportuale e degli aeromobili, risulta ora chiarificato il 4° step dello schema rappresentato in Fig.60 al paragrafo 5.5: l'implementazione della connessione UDP in *input* al dispositivo Microsoft Hololens fornisce ad esso dei dati di riferimento indispensabili per registrare gli AR *overlays* con gli ambienti virtuali riprodotti dai sistemi di proiezione su schermi nelle sperimentazioni di laboratorio. L'interfaccia di *input* del dispositivo Microsoft Hololens è stata opportunamente implementata per ricevere dati da qualunque hardware di visualizzazione a schermo singolo, multischermo e multischermo ruotato. Ciò è stato ottenuto mediante la trasmissione ad Hololens della seguente quaterna di informazioni relative ad ogni posizionamento inviato (Fig.78):

- numero totale di schermi;
- dimensione degli schermi;
- in quale schermo si trova il posizionamento;

- coordinate del posizionamento nel sistema di riferimento proprio di ogni schermo.

Attraverso tali informazioni il dispositivo Hololens effettua una conversione di coordinate nel proprio sistema di riferimento e registra propriamente gli AR *overlays* con i relativi oggetti statici o dinamici rappresentati nell'ambiente virtuale.

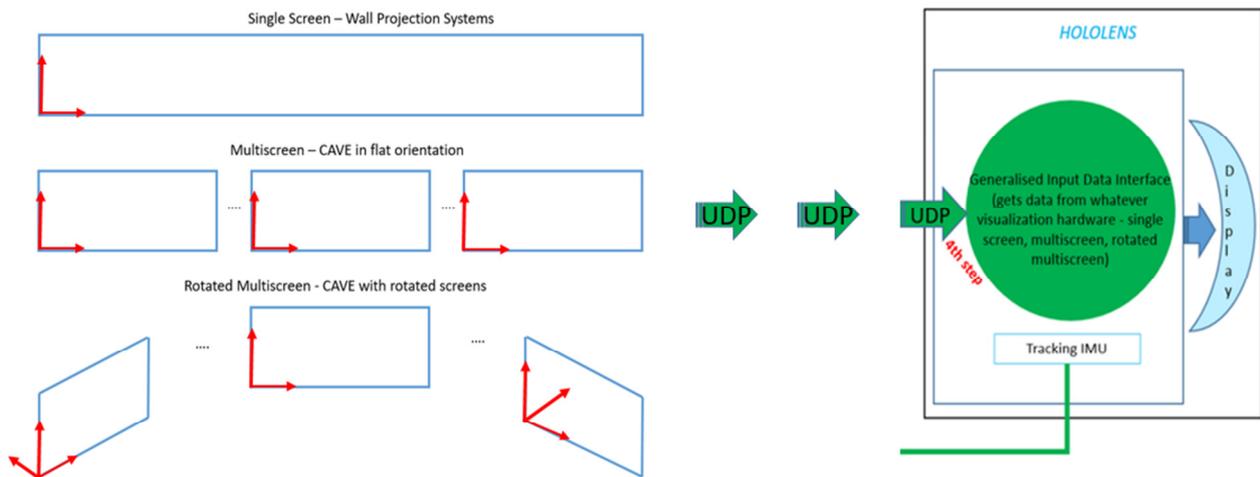


Fig. 78: interfaccia generalizzata di input creata per Hololens - ricezione dati da schermi singoli, multipli piani e multipli ruotati. La parte destra della figura è un estratto della Fig.60 al paragrafo 5.5.

6.4 Concettualizzazione, sviluppo ed implementazione di un *sistema AR adattivo, collaborativo e video-based*

Gli studi sull'AR condotti nell'ambito del progetto RETINA hanno aperto la strada verso la concettualizzazione di un framework AR fondato sul paradigma della visione panoramica dall'alto e la cui applicazione non trova solamente campo fertile nel contesto delle torri di controllo aeroportuali, bensì può essere estesa ad entrambe le macro-aree in cui ricadono tutte le applicazioni AR relative al settore dei trasporti (terrestri, marittimi, aerei), sia di carattere civile che militare:

- *AR per centri di controllo*: AR come supporto al controllo ed alla gestione di aeroporti, porti e terminal container;
- *AR per veicoli*: AR come supporto alla navigazione (mezzi di trasporto terrestri, *automotive*, navi, elicotteri, aerei).

Tali applicazioni ricadono nei cosiddetti contesti *safety-critical*, dove il fallimento delle strategie può causare la perdita di vite umane. Per massimizzare le prestazioni e minimizzare i rischi, due sono le principali leve su cui intervenire:

- *teamwork*: spartizione dei *task* fra operatori;
- *human factors*: miglioramento delle *human performances* in termini di riduzioni del carico di lavoro (*workload*), incremento della *situational awareness* e di una migliore accessibilità dell'informazione utile per eseguire il *task*, presentata esattamente nel momento in cui si ha necessità di essa.

La componente *teamwork* ha dato origine alla concettualizzazione di un *sistema AR multiutente*, ovvero *adattivo e collaborativo*, mentre quella *human factors* ha portato la ricerca ad esplorare la direzione dell'integrazione di video in tempo reale all'interno dei sistemi AR. Il risultante *sistema AR adattivo collaborativo e video-based* è descritto per componenti nei due paragrafi che seguono.

6.4.1 Sistema AR adattivo e collaborativo

Lo scenario multiutente non è stato considerato per i sistemi AR rappresentati dagli SD, dal momento che il loro svantaggio principale risiede nell'impossibilità di poter gestire la presenza simultanea di più utenti. Infatti, al fine di poter collimare i dati digitali con la scena reale, ogni utente ha necessità di vedere gli AR *overlays* che sono coerenti con la propria prospettiva visuale, filtrando quelli che sono destinati ad altri utenti che osservano la scena da posizioni differenti. In tal senso, è necessario adottare un *display AR* in grado di fornire in modo selettivo a ciascun utente dei differenti canali di visualizzazione, senza dover integrare alcuno strumento indossabile in tale sistema (per definizione il funzionamento degli SD non prevede l'utilizzo di dispositivi indossabili come occhiali o visori AR). Tale tipo di tecnologia, ancora oggetto di ricerca, permetterebbe anche la personalizzazione dell'informazione destinata allo specifico utente, sulla base di prospettiva visuale, *task* svolto e preferenze personali [42].

Al contrario, la soluzione degli HMD non è caratterizzata dalla stessa problematica che contraddistingue gli SD, dal momento che la tecnologia indossabile rappresenta una fonte di informazione destinata a ciascun singolo utente e quindi personalizzabile. Ciò comporta la fattibilità dello scenario multiutente per i dispositivi HMD ed essa ha focalizzato la ricerca su di un *sistema AR adattivo e collaborativo*, dove:

- gli utenti condividono lo stesso ambiente, ma sperimentano differenti visualizzazioni AR: uno stesso visore può essere programmato per fornire automaticamente differenti *set* di

overlays a seconda del *task* svolto, delle fasi operative, nonché delle condizioni ambientali (*sistema AR adattivo*);

- lo scambio di informazioni avviene fra tutte le componenti e fra tutti i livelli – fra gli operatori, nonché fra gli stessi ed i loro supervisori, quest’ultimi aventi il compito di ottimizzare le strategie di *teamwork* in tempo reale (*sistema AR collaborativo*). Sebbene il termine “collaborativo” venga comunemente impiegato con l’accezione di un sistema AR/VR atto ad integrare più utenti in uno spazio condiviso indipendentemente dalla loro collocazione geografica [92] [93], nel caso di questa tesi il significato di “collaborativo” verrà utilizzato come sinonimo di “cooperativo” ed andrà a caratterizzare un sistema AR dove gli utenti condividono fisicamente lo stesso ambiente ed interagiscono simultaneamente tra loro.

Il concetto è chiarificato dall’esempio applicativo riportato nella seguente Fig.79. L’immagine ritrae due controllori di volo, rispettivamente *tower* (avente responsabilità sulle piste di decollo e di atterraggio) e *ground* (incaricato di gestire la movimentazione sulle vie di circolazione e nelle zone dei parcheggi), che condividono lo stesso ambiente virtuale di una torre di controllo (simulato per mezzo del sistema CAVE) ed indossano il dispositivo Hololens. Il controllore *tower* autorizza l’aereo appena atterrato a liberare la pista mediante la bretella di uscita “B” e conferisce la gestione del volo al controllore *ground*, la cui responsabilità sarà quella di instradare l’aereo lungo uno specifico percorso che lo condurrà al parcheggio di destinazione.

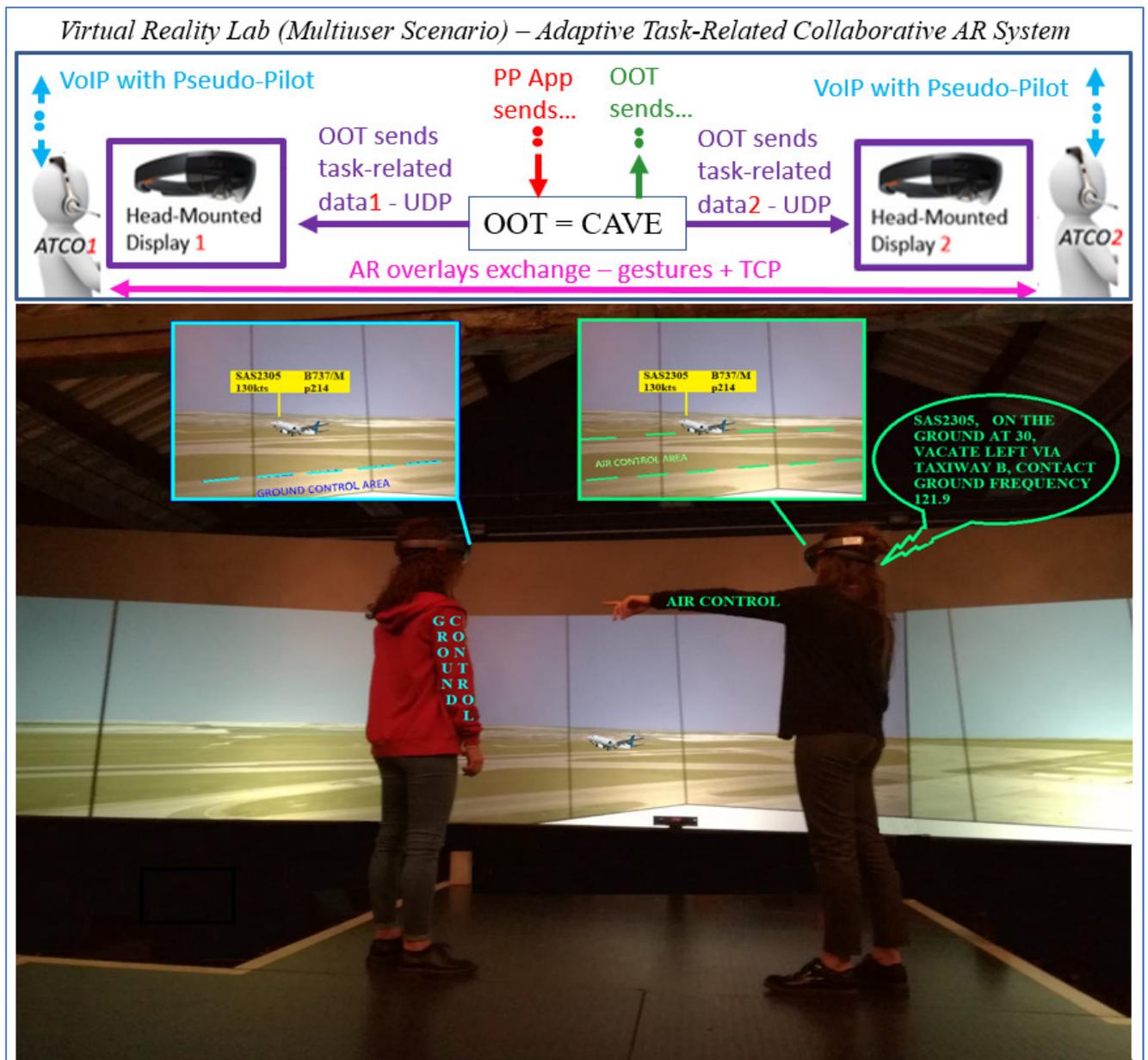


Fig.79: un esempio di sistema AR adattivo e collaborativo – applicazione alle torri di controllo aeroportuali.

Entrambi i controllori sono coinvolti in un *sistema AR multiutente* che è:

- *adattivo*: gli operatori condividono lo stesso ambiente (il CAVE), ma sperimentano differenti visualizzazioni AR in cui le relative aree di responsabilità sono rappresentate rispettivamente da linee tratteggiate di colore verde e ciano;
- *collaborativo*: lo scambio di informazioni tra i due dispositivi Holens permette al controllore *tower* di disimpegnare la propria responsabilità sul volo e passare la gestione dello stesso all'operatore *ground* per mezzo di una *gesture*, cosicché tutte le informazioni associate al volo (*label* gialla) compariranno nel gruppo di AR *overlays* visualizzati dal controllore *ground*, mentre scompariranno automaticamente da quello dell'altro operatore.

La piattaforma multiutente riportata nello schema di Fig.79 deriva da quella a singolo utente di Fig.73.

A differenza di quest'ultima, il modulo OOT non prevede l'integrazione del Kinect, dal momento che ogni utente non è messo nella condizione di poter navigare l'ambiente virtuale della torre di controllo secondo la propria prospettiva visuale, in quanto essa è diversa rispetto a quella di un altro utente. In questo caso quindi il CAVE non aggiorna la matrice di proiezione degli schermi in modo solidale al punto di vista dell'utente come avveniva con la piattaforma a singolo utente, ma proietterà costantemente una stessa visione verso l'ambiente aeroportuale che si trova fuori dai vetri del centro di controllo, condivisa da tutti gli utenti. Inoltre, a differenza della piattaforma a singolo utente, il modulo OOT invia l'informazione agli HMD in modo selettivo, ovvero trasmettendo ad ogni operatore i dati rilevanti richiesti da ciascuno di essi per lo svolgimento del proprio *task*. Analogamente alla modalità di invio dei comandi all'OOT da parte dello pseudo-pilot, quando l'informazione viene trasmessa fra visori indossati dagli operatori, occorre garantire a quest'ultimi che le loro azioni vengano inviate senza necessità di doverle ripetere più volte. Ciò significa che, come nel caso della comunicazione PP App – OOT, la trasmissione dati tra visori fa affidamento ad un protocollo di scambio dati TCP. La connessione PP App – OOT è caratterizzata da un transito dati unidirezionale, quella fra visori è *full-duplex*, ovvero gli utenti possono inviarsi tra loro informazioni in modo contemporaneo, senza necessità di prenotare il canale di comunicazione (come avviene nel caso della telefonia, in cui gli utenti possono parlare contemporaneamente). La problematica riscontrata nell'implementazione della connessione TCP tra visori Hologens è stata quella che la Universal Windows Platform (per maggiori informazioni si veda 6.1.2) non riconosce le librerie "System.Net.Sockets" utilizzate nella creazione della connessione TCP per la PP App e quindi esse non vengono compilate. La Universal Windows Platform richiede infatti librerie diverse per implementare le connessioni TCP e specificamente quelle chiamate "Windows.Networking.Sockets". Si potrebbe allora pensare di utilizzare quest'ultime per la creazione di applicazioni funzionanti su Hologens, ma ciò non è possibile in quanto Unity a sua volta non riconosce tali librerie e quindi riporta errore a tempo di compilazione, non permettendo di creare il pacchetto di compatibilità per Hologens con la Universal Windows Platform. La soluzione è stata quella di racchiudere il codice specifico di una piattaforma all'interno del corpo delle seguenti condizioni `#if !UNITY_EDITOR` e `#if UNITY_EDITOR`. La prima ha la funzione di dire al compilatore di ignorare completamente il blocco di codice relativo alla condizione, qualora si stia compilando una applicazione Unity; la seconda invece sta a significare di non tener conto delle istruzioni che seguono, se si sta compilando per una qualsiasi altra piattaforma. Quest'ultima quindi sarà la condizione accettata nel caso della Universal Windows Platform. Da ciò segue che, nel caso in cui si considerassero anche altre piattaforme, si potrebbe manifestare la necessità di aggiungere altre specifiche condizioni, che non sono risultate utili nel caso in oggetto. Oltre a risolvere la problematica di dover conciliare diversi sistemi di compilazione, le condizioni introdotte hanno permesso anche di conferire flessibilità al codice di programmazione sviluppato: un unico programma volto a gestire connessioni TCP può essere in tal modo utilizzato da vari *client* (piattaforme), che, attraverso differenti adattatori (librerie di sviluppo utilizzate), riescono ad impiegarlo nelle loro applicazioni.

In definitiva ogni dispositivo Hologens è caratterizzato da:

- una connessione UDP entrante con dati provenienti dal modulo OOT;
- una connessione TCP entrante/uscente con dati provenienti/inviati da/verso l'altro dispositivo Hololens.

Entrambe le connessioni fanno uso della libreria di sviluppo “Windows.Networking.Sockets” prima discussa. Questa è basata sul concetto di programmazione asincrona *multithreading*. Mentre in quella sincrona ogni *thread* assume un *task* e lo completa (come visto in Fig.48), nel modello asincrono un *thread* può interrompere l'esecuzione di un *task* per iniziare l'esecuzione di un altro, come riportato in Fig.80 nel caso di una programmazione asincrona a singolo *thread*.



Fig.80: programmazione asincrona a singolo thread. Fonte: [ref 38](#)

Nel modello asincrono *multithread* gli stessi *task* possono essere gestiti da più *threads*. Così, per esempio, in Fig.81 si ha che il *Task 4* è stato preso in carico dal *Thread 1*, per poi essere completato dal *Thread 2*. Analogamente, il *Task 6* viene lavorato dal *Thread 2*, *Thread 3* e *Thread 4*.

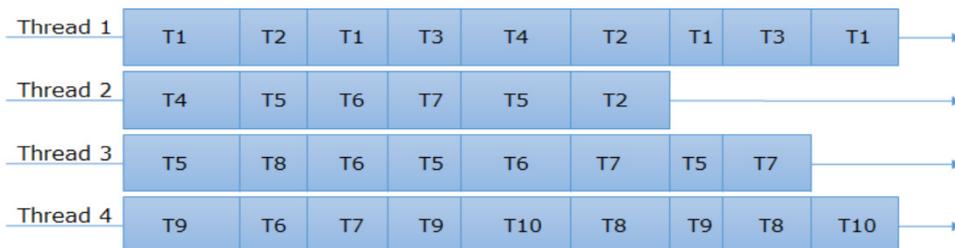


Fig.81: programmazione asincrona multithread. Fonte: [ref 39](#)

La programmazione asincrona *multithread* presenta il vantaggio di poter ottimizzare la gestione dei *threads*: appena uno di essi si libera dal completamento dei *task*, ne può assumere altri in fase di lavorazione presso altri *threads*, alleggerendo in tal modo il loro lavoro e quindi riducendo i tempi totali di computazione. Un esempio significativo che chiarifica il concetto è rappresentato dalla cucina di un ristorante: le pietanze possono essere preparate una dopo l'altra (modello sincrono), oppure cucinate contemporaneamente (asincrono) da un solo cuoco (singolo *thread*), oppure da più cuochi (*multithreading*).

L'impatto positivo che la programmazione asincrona *multithreading* apporta nelle operazioni di *networking*, ovvero nelle implementazioni delle connessioni TCP/UDP, è manifestato dal fatto che le operazioni sui pacchetti dati in entrata (ricezione, estrazione dell'informazione utile e *parsing* ed elaborazione della stessa), nonché su quelli in uscita (elaborazione dell'informazione, formazione dei pacchetti ed invio degli stessi) sono effettuate in modo parallelo (*multithreading*) e non bloccante (asincrono), in quanto il processo coinvolge differenti *threads* che, terminato il proprio compito, non rimangono in attesa, ma possono assumere in carico altri *task* semilavorati da altri *threads*.

6.4.2 Sistema AR video-based

Sebbene i campi di applicazione della MR siano molto numerosi ed atti ad eclissare, secondo alcuni, totalmente l'impiego della Realtà Virtuale in un futuro prossimo [7], essi hanno finora utilizzato solamente una parte dell'ampio potenziale della MR. Infatti, la ricerca è stata fino a questo momento focalizzata sulla generazione e visualizzazione di oggetti grafici 2D o 3D da sovrapporre al mondo reale visto dall'utente attraverso i *display see-through*, senza considerare la possibilità di inserire, all'interno del FOV di tali dispositivi, dei *frames* video collimati con gli oggetti degli ambienti reali.

Se si considera il fatto che l'82% dell'intero traffico di Internet sarà video dipendente entro il 2022 [94] e che la MR sta diventando sempre più popolare a velocità sbalorditiva [95], la ricerca sullo *streaming* video verso tecnologie MR costituisce una strategia molto promettente, in grado di influenzare significativamente le presenti e future modalità di svago e di lavoro, specialmente in un mondo dove concetti come quello di *lockdown* o di distanziamento sociale stanno diventando una triste realtà. La pandemia di COVID-19 in corso sta colpendo la *supply chain* globale dei beni fisici e dei servizi, con il risultato di produrre la peggiore crisi economica dai tempi della seconda guerra mondiale. Per ovviare alle problematiche causate dal *lockdown*, molte aziende stanno cambiando in modo radicale il loro modo di produrre: le principali azioni intraprese vanno nelle due direzioni di:

- lavoro da remoto (telelavoro, più comunemente noto come lavoro da casa)
- *smart working* (ovvero trasformazione digitale delle industrie e dei servizi)

La ricerca in oggetto sullo *streaming* video verso tecnologie MR va a favore di entrambi i concetti: lo *streaming* video caratterizza il telelavoro; i dispositivi MR rappresentano una trasformazione digitale che comporta nuove tecnologie aventi lo scopo di migliorare sia le *performances*, che la soddisfazione nello svolgimento dei *task*.

L'integrazione delle due componenti, ovvero lo *streaming* video verso dispositivi MR, potrebbe produrre impatti positivi non solo in chiave di collaborazione remota, bensì anche in relazione al miglioramento del *workload* e della *situational awareness* degli operatori addetti allo svolgimento

di *task* in particolari contesti *safety-critical*. Ne sono esempio le torri di controllo del traffico aereo e gli elicotteri *search and rescue*, ambiti in cui gli operatori necessitano di vedere il mondo fisico che li circonda integrato con l'informazione digitale aggiuntiva (con esso registrata) rappresentata da oggetti grafici 2D/3D e *frames* video acquisiti da telecamere IR a lunga gittata collocate in prossimità o anche a distanza degli operatori. Entrambi i contesti sono stati scelti come casi di studio ed esposti nella parte finale del presente elaborato. Considerando sia le torri di controllo che gli elicotteri, la trattazione ha riguardato rispettivamente sia gli impianti di gestione/controllo che i mezzi di trasporto, le due macro-aree in cui ricadono non solo tutte le applicazioni AR relative al campo aeronautico, bensì all'intero settore dei trasporti (terrestri, marittimi, aerei), sia di carattere civile che militare.

La trattazione dei *frames* video ha spostato l'attenzione sull'acronimo "MR", che, come già visto, sta a indicare il termine di Realtà Mista. In particolare, nell'ambito di tale elaborato esso è stato utilizzato solamente a partire dall'inizio dell'attuale sottoparagrafo. Sebbene nella letteratura il termine MR sia largamente impiegato come sinonimo di AR, si è qui preferito l'utilizzo del primo acronimo. Ciò è dettato dal fatto che i *frames* video non hanno generalmente l'opportunità di essere trasparenti come invece è possibile per l'informazione digitale aggiuntiva rappresentata dagli oggetti grafici 2D/3D. Così, dal momento che, rispetto a quest'ultimi, i *frames* video per loro natura occludono maggiormente la visione sul mondo reale, l'ago della bilancia si sposta maggiormente verso il concetto di Realtà Virtuale (che occlude del tutto la visione sul mondo reale) e quindi, con essi, più che di un mondo reale "aumentato", si preferisce parlare di uno "misto" tra reale e virtuale.

Lo *streaming* video verso dispositivi HMD è stato finora oggetto di ricerca soprattutto nel campo della Realtà Virtuale. In tale ambito, lo sviluppo delle connessioni a banda larga, nonché della rete 5G, ha focalizzato l'indagine maggiormente verso i campi del *3D video streaming* e del *360° video streaming*.

In merito al primo, gli studi si sono concentrati principalmente sulle applicazioni mediche [17] e di intrattenimento [96].

Nel caso del *360° video streaming*, le immagini vengono inviate da remoto da una telecamera omnidirezionale al dispositivo HMD di Realtà Virtuale in tempo reale. Per ottenere esperienze immersive con livelli di qualità dei video accettabili, le strade da percorrere sono sostanzialmente due:

- implementare un ritardo di riproduzione dei pacchetti al ricevimento degli stessi per combattere il *jitter* causato dalla rete (diversi tempi di viaggio dei pacchetti dovuti ai ritardi di accodamento, elaborazione e trasmissione nei router attraversati sul percorso da origine a destinazione): più lungo è il ritardo stabilito, maggiori saranno i pacchetti arrivati in tempo per essere riprodotti (e quindi minori saranno le perdite di riproduzione), ma al tempo stesso minore sarà la caratteristica di *real-time* dello *stream* ricevuto (Fig.82);

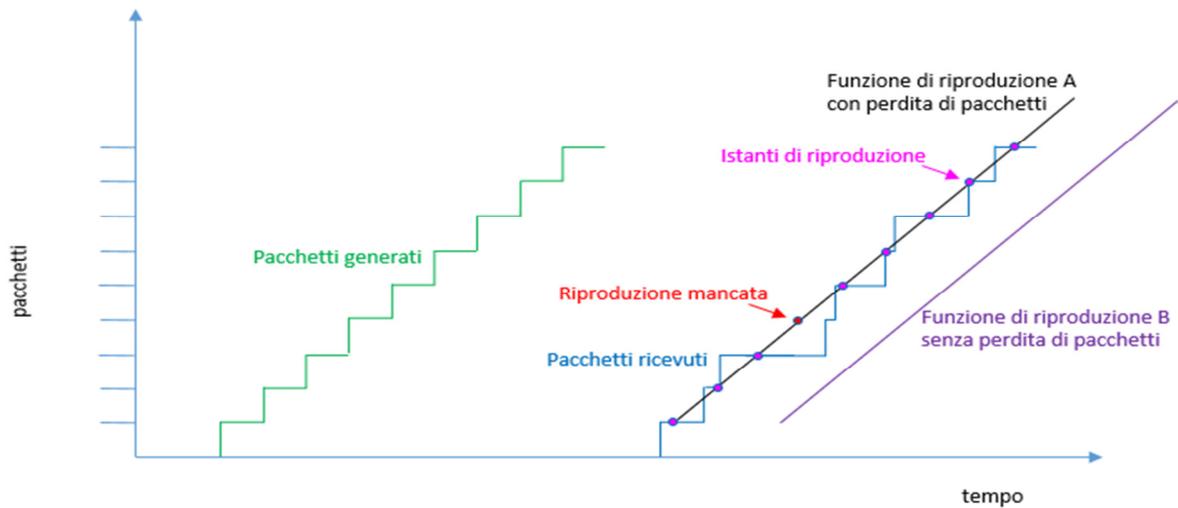


Fig.82: progettazione dei ritardi di riproduzione per combattere il jitter (diversi tempi di viaggio dei pacchetti) di rete.

- non ritardare la funzione di riproduzione, bensì anticipare quella dei pacchetti ricevuti (traslazione verso sinistra della funzione a gradini di colore blu in Fig.82). Tali soluzioni sono raggruppate con il termine di *interactive multi-view video streaming (IMVS) technique* [97] [98]. Esse studiano sostanzialmente come codificare l'immagine panoramica di 360° in *stream* multipli, cosicché all'utente in ricezione viene data la possibilità di abilitare o disabilitare gli stessi, minimizzando le interruzioni nell'esperienza di visione immersiva (Fig.83).

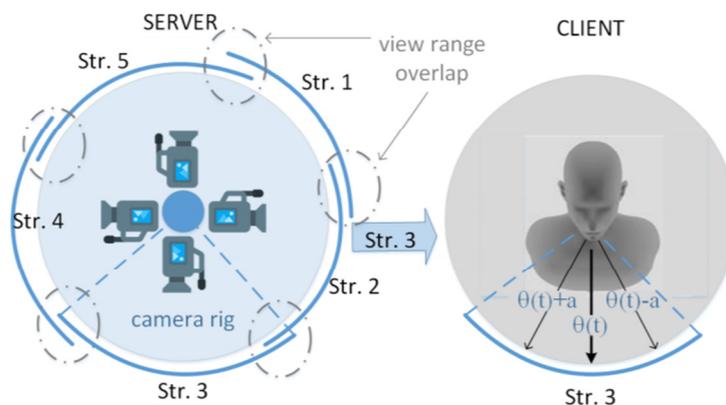


Fig.83: esempio di tecnica IMVS: durante lo streaming, il server abilita e trasmette i diversi stream in base all'angolo di rotazione della testa dell'utente $\theta(t)$ e del FOV $[\theta(t) - a, \theta(t) + a]$ al tempo t . Fonte [98].

Analogamente agli approcci descritti sopra, in ambito di tale elaborato la ricerca sullo *streaming* video verso tecnologie MR si è concentrata su trasmissioni multiple derivanti da quattro telecamere, ognuna caratterizzata da un FOV orizzontale di circa 90°: in tale modo, lo *stream* risultante è in grado di coprire tutti i 360° acquisiti da una telecamera omnidirezionale. Tuttavia, a differenza di ciò che è stato esposto precedentemente per i dispositivi HMD di Realtà Virtuale:

1. la funzione dei pacchetti ricevuti è parallela a quella di generazione degli stessi, in quanto le telecamere non sono collocate in remoto, bensì in prossimità dei MR HMD a cui inviano gli *stream* e quindi il *jitter* di rete può essere assunto costante;
2. non è stata qui applicata alcuna tecnica IMVS, cosicché l'utente in ricezione non abilita o disabilita in modo interattivo i diversi *stream*. Tale scelta è dipesa dal fatto che i casi di studio considerati potrebbero richiedere la presenza contemporanea di tutti gli *stream* nel FOV del MR HMD per massimizzare la *situational awareness*.

La ricerca sul video *streaming* verso i MR HMD condotta durante il terzo anno di studio ha permesso di concettualizzare e sviluppare una soluzione ibrida fra i due differenti approcci seguiti fino ad ora nella costruzione dei sistemi MR. Il primo passo da compiere nel processo di creazione degli stessi riguarda la decisione di come combinare il reale con il virtuale. Essa rappresenta un bivio a due direzioni: tecnologia MR ottica o video [1].

La prima basa il proprio funzionamento su combinatori ottici (Fig.84), che sono parzialmente:

- trasparenti (l'utente è in grado di vedere il mondo reale circostante)
- riflettenti (gli *overlays* virtuali generati dai *monitor* vengono riflessi dai combinatori).

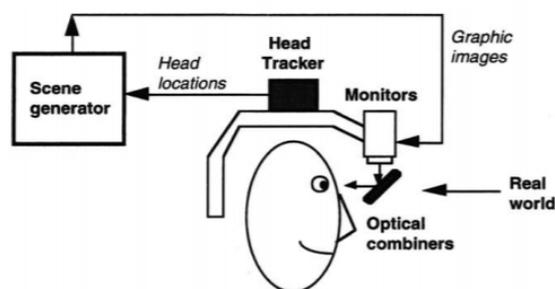


Fig.84: tecnologia MR ottica. Fonte [1].

D'altra parte, i dispositivi MR video non permettono all'utente la visione diretta del mondo reale, bensì mediata attraverso le immagini provenienti da telecamere che lo riprendono. Essi si dividono in due differenti categorie: *video-based see-through* e *monitor-based technology* (Fig.85).

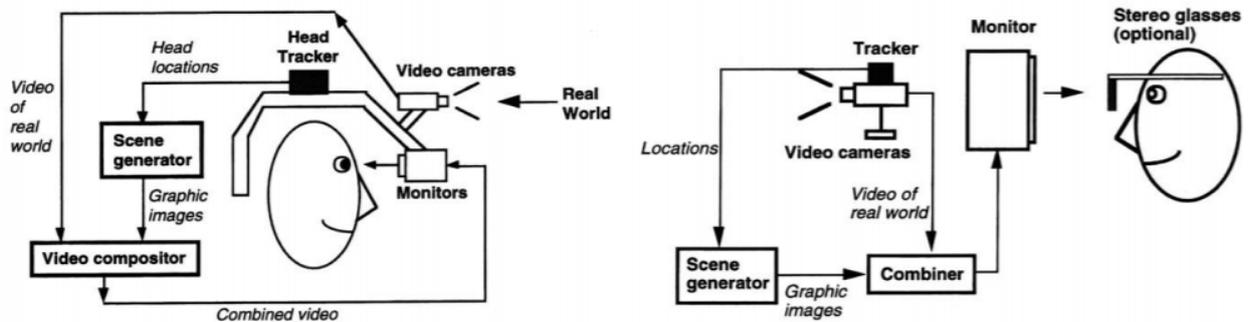


Fig.85: *video-based see-through technology* (a sinistra) e *monitor-based technology* (a destra). Fonte [1].

Sebbene tutte le applicazioni MR finora proposte seguano una direzione piuttosto che l'altra, la possibilità di integrare più *stream* video in tempo reale all'interno del *layer* virtuale di una tecnologia MR ottica permette di colmare il *gap* di ricerca rappresentato dall'assenza di soluzioni ibride derivanti dalla fusione delle differenti funzionalità che caratterizzano entrambe le categorie distinte (tecnologia MR ottica e video) in cui vengono suddivisi tutti i dispositivi MR (Fig.86).

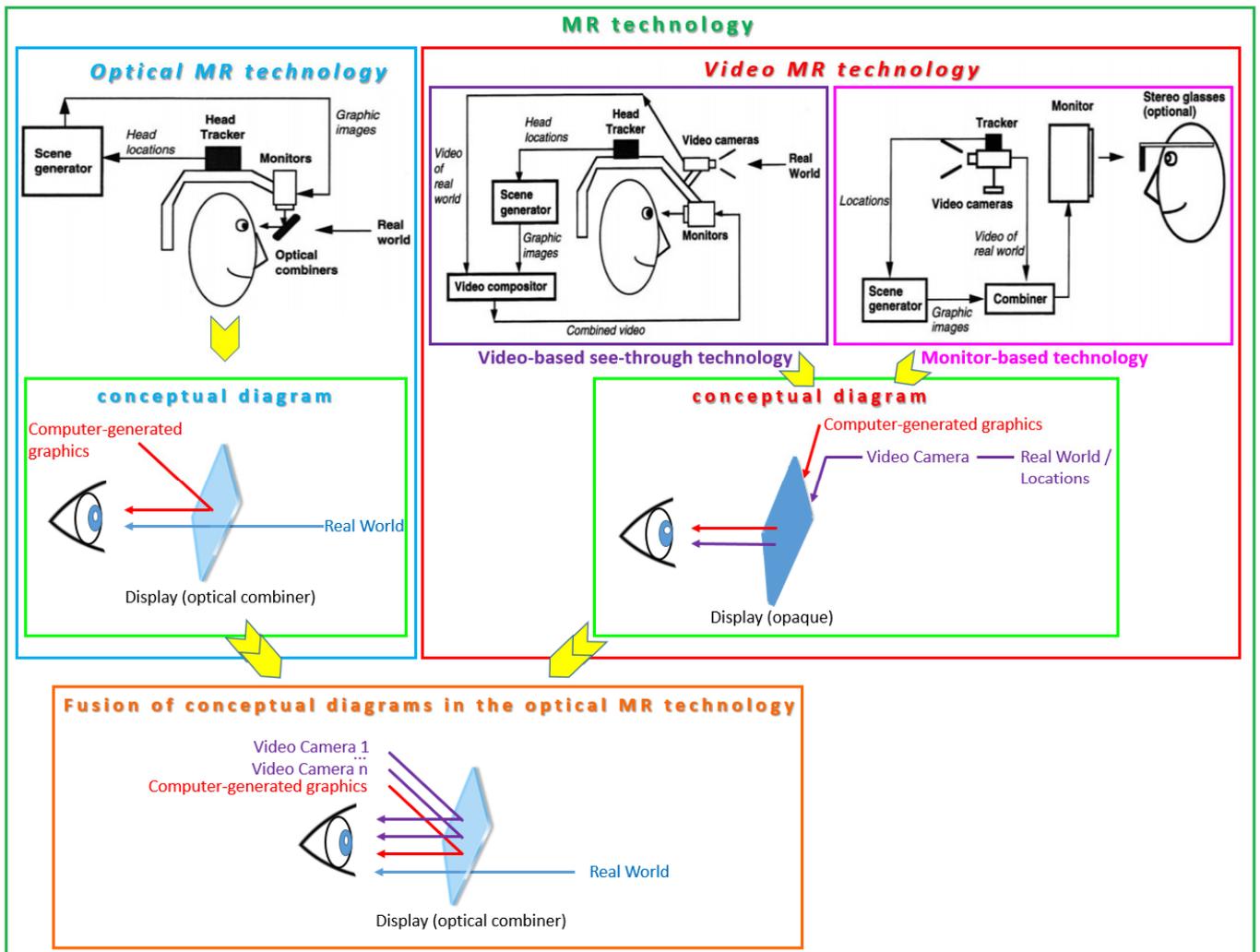


Fig.86: la soluzione ibrida fra tecnologia MR ottica e video.

Inoltre, l'integrazione degli *stream* video in tempo reale nelle tecnologie MR ottiche apre la strada verso nuove e complesse sfide di registrazione: invece di concepire *frames* video statici, ovvero in posizioni fisse del FOV (come avviene nel caso della tecnologia MR video), essi possono essere progettati come dinamici e registrati (collimati) con gli oggetti del mondo reale attorno all'utente. In questo caso le risorse computazionali dei sistemi MR non devono essere solamente indirizzate a fornire un'efficiente visione dei *frames* video, bensì occorre che esse siano anche dedicate all'allineamento degli stessi con gli oggetti reali del mondo fisico visto dall'utente attraverso il *display* della tecnologia MR ottica.

L'integrazione di *stream* video multipli all'interno di un sistema MR è stata implementata mediante la piattaforma di test riportata in Fig.87.

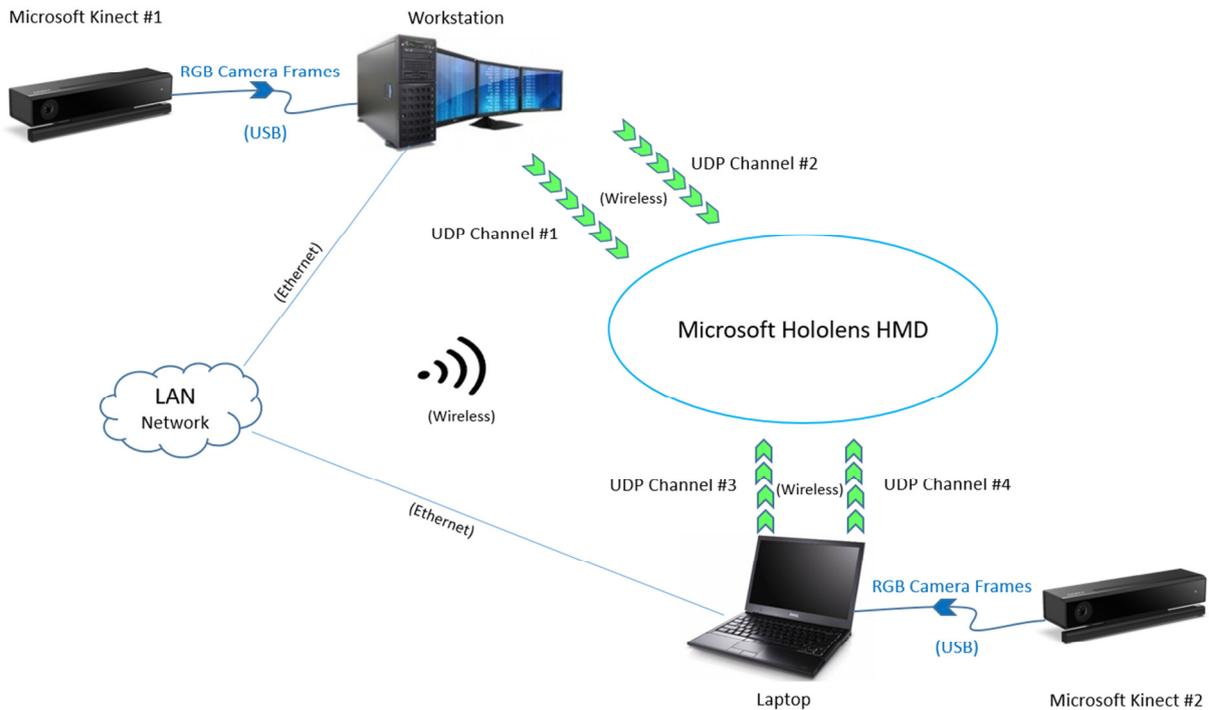


Fig.87: architettura della piattaforma utilizzata per lo streaming video verso tecnologie MR HMD.

La motivazione che ha mosso la realizzazione di tale architettura è consistita nel riprodurre la soluzione ibrida concettualizzata in Fig.86. In particolare, quattro *stream* video sono generati mediante videocamere RGB dei dispositivi USB Microsoft Kinect e inviati al visore Hololens attraverso una *workstation* ed un *laptop* (*server*). Il numero degli *stream* non è stato scelto a caso: dal momento che ogni videocamera Kinect è caratterizzata da un FOV orizzontale di circa 90° (Fig.34), lo *stream* risultante è in grado di coprire un angolo di 360°. Ciò permette di focalizzare l'indagine sulla trattazione dello *streaming* video catturato da una ipotetica telecamera omnidirezionale ed inviato verso un MR HMD. Sebbene la piattaforma sia caratterizzata da quattro *stream*, solamente due videocamere Kinect sono state incluse nell'architettura. Infatti, ogni immagine catturata da ciascuna di esse viene convogliata dai *server* in due canali *wireless* differenti ed è poi inviata all'Hololens (*client*). In questo modo viene simulata l'architettura che coinvolge quattro differenti videocamere USB connesse ad altrettanti *server*. L'idea di considerare solo una *workstation* generante quattro *stream* verso il *client* è stata abbandonata al fine di evitare di sovraccaricare eccessivamente le *performances* dei *server*. Al contrario, lo scenario con i due *server* non desta tali preoccupazioni, in quanto gli esperimenti hanno mostrato che il collo di bottiglia è sempre collocato sul lato client: le *performances* della seconda versione del dispositivo Hololens nel processare i quattro *stream* vengono più che raddoppiate rispetto a quelle relative alla prima versione rilasciata nel 2016. Essendo tutte le componenti della piattaforma incluse nella stessa *Local Area Network* (LAN), gli *stream* catturati dalle telecamere non mirano a fornire all'HMD delle informazioni su ambienti collocati in remoto, bensì dettagli sul mondo attorno all'utente: lo

scopo di tale architettura è quello di simulare il comportamento di piattaforme in cui il visore indossato dall'operatore migliora le sue capacità di contestualizzare dati derivanti da telecamere in grado di cogliere dettagli in condizioni di scarsa visibilità oppure a lunga distanza. Questo è il contesto dei casi di studio delle torri di controllo aeroportuali e degli operatori *search and rescue* a bordo degli elicotteri a cui tale elaborato fa riferimento.

I risultati ottenuti dagli esperimenti sul *multi-streaming* video verso i dispositivi MR HMD sono basati sull'impiego della piattaforma di Fig.87. Di seguito si riportano le specifiche tecniche di ciascuna componente, eccetto che per i dispositivi Kinect e Hololens, le cui caratteristiche sono già state mostrate precedentemente, rispettivamente in Fig.34 e Fig.69:

- *workstation*:
 - CPU: Intel® Core i5-650 3.20G 1333MHz 4T\Dual core\4MB cache;
 - Memoria: 8GB DDR3 PC3-10600;
 - Sistema operativo: Windows 10;

- *laptop*:
 - CPU: 3rd gen Intel Core i7-3840QM Processor (2.80GHz Turbo, 8MB);
 - Memoria: 32GB (4x8GB) 1600MHz DDR3;
 - Sistema operativo: Windows 10.

La piattaforma sopra descritta rappresenta un particolare tipo di implementazione in un largo *set* di architetture possibili. Nulla vieta di scegliere altre soluzioni che integrano differenti tecnologie per ciascuno dei principali tre componenti su cui è basato l'intero processo del *multi-streaming* video in tempo reale, ovvero videocamere, MR HMD e LAN. Infatti, la trattazione è incentrata su di un efficiente metodo di frammentazione dei pacchetti UDP per la trasmissione di *stream* video in tempo reale che opera a livello di *layer di trasporto* della pila dei protocolli TCP/IP su cui si basa il funzionamento della rete (Fig.88), indipendentemente dalla particolare tecnologia considerata all'interno della LAN.

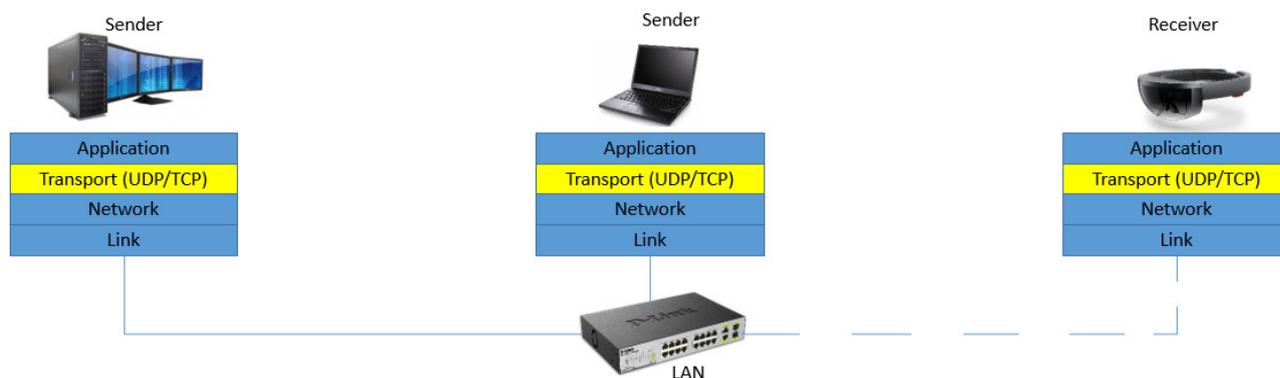


Fig.88: layer di trasporto della pila dei protocolli TCP/IP.

Al livello di *layer* di trasporto, in confronto al protocollo TCP, UDP è maggiormente adatto per la trasmissione di *stream* video in tempo reale, in quanto la comunicazione gode di maggiore rapidità. Laddove si ha a che fare con applicazioni *time-sensitive* (come per esempio lo *streaming* video o VoIP in tempo reale), l'affidabilità di TCP non rappresenta la soluzione migliore, in quanto, in questi casi si predilige la velocità di UDP, anche a costo di qualche pacchetto perso [99]. Oltre al fatto di essere caratterizzati dallo *streaming* video in tempo reale e quindi *time-sensitive*, i casi di studio delle torri di controllo aeroportuali e degli operatori *search and rescue* a bordo degli elicotteri sono anche *safety-critical*. Da ciò si comprende come in tali contesti il fattore tempo nel trasferimento dati sia vitale, spostando ancora di più la preferenza per l'impiego del protocollo UDP.

Tuttavia, la rapidità del protocollo UDP si paga a costo di una limitazione nella dimensione dei suoi pacchetti: essa è vincolata dal valore massimo di 65507 *bytes*. Ciò significa che non è possibile codificare molti *pixels* in un singolo pacchetto UDP e da ciò segue che la risoluzione delle immagini non è in grado di raggiungere valori elevati. Al contrario, il protocollo TCP non è affetto da dimensioni massime dei pacchetti e quindi mediante esso è possibile codificare un quantitativo qualunque di dati. Ciononostante, TCP è lento. Considerando la velocità come un *hard constraint*, ovvero un vincolo che deve essere necessariamente soddisfatto da ogni possibile approccio alla soluzione, la questione fondamentale consiste nel determinare *come implementare un trasferimento immagini ad alta risoluzione con il protocollo UDP caratterizzato da una elevata rapidità di trasmissione dati*. La chiave di volta per risolvere il problema è insita nel processo di suddivisione del quantitativo totale di *pixels* di ogni immagine che è necessario codificare in più pacchetti UDP. Si consideri la seguente analogia. Un gruppo (immagine) di numerosi passeggeri (*pixels*) deve essere trasferito da Londra (videocamera RGB del Kinect) a Roma (Hololens). Non è possibile utilizzare un aereo di dimensioni medie, come per esempio un Airbus 320 (pacchetto UDP), perché esso non è in grado di contenere tutto il gruppo. Invece di impiegare un aereo con maggiore capacità, per esempio un Airbus 330 (pacchetto TCP), il gruppo di turisti viene suddiviso in due Airbus 320 (due pacchetti UDP) ed è ricomposto a destinazione (Roma).

La problematica del trasferimento di immagini ad alta risoluzione mediante UDP sembrerebbe quindi risolta mediante la suddivisione dell'informazione legata a ciascuna immagine in più pacchetti. Tuttavia, ciò non basta in quanto UDP soffre di un altro inconveniente, ovvero di essere un protocollo disordinato: l'ultimo pacchetto inviato potrebbe arrivare a destinazione prima di altri inviati precedentemente e quindi non è possibile prevedere l'ordine di arrivo dei pacchetti, in quanto il protocollo UDP non fornisce alcun controllo (per questo motivo è più rapido di TCP). Da ciò segue che la suddivisione dei *pixels* di una immagine in più pacchetti comporta necessariamente un processo di riordino dei *pixels* a destinazione, dove l'immagine deve essere ricomposta come un *puzzle*. La chiave di volta sta nell'evitare di suddividere la risoluzione di un *frame* in un ampio numero di pacchetti, ovvero rinunciare a considerare risoluzioni molto alte, in quanto in tale caso l'implementazione TCP potrebbe risultare più rapida di UDP, a causa dell'ingente lavoro svolto dall'algoritmo di riordino a destinazione, il quale non necessita invece di essere implementato in TCP. Tuttavia, il problema delle alte risoluzioni non ricade nella casistica della telecamera RGB del dispositivo Kinect, caratterizzato da 1920x1080 *pixels*. Sotto tali condizioni, integrando il processo di riordino a destinazione (Hololens), il protocollo UDP continua ad essere molto più rapido di TCP. Così, ricevuto il pacchetto da uno dei quattro canali UDP, il client decodifica, processa e salva l'informazione utile (*payload*) contenuta nel pacchetto, senza bloccare, con tali operazioni, il processo di ricezione dei dati sullo stesso canale e sugli altri tre. Ciò è stato implementato mediante programmazione asincrona *multithreading* (vista in 6.4.1). Infine, ricevuto il numero totale di *pixels* caratterizzante ogni frame catturato dalla videocamera RGB del dispositivo Kinect, il client Hololens li riordina opportunamente e ricompono l'immagine originale.

Riassumendo, il processo del *multi-streaming* video in tempo reale da una ipotetica telecamera omnidirezionale (simulata mediante quattro *stream* catturati da videocamere RGB Kinect con FOV di circa 90°) verso il dispositivo Microsoft Hololens è stato implementato attraverso le tre seguenti procedure fondamentali:

1. suddivisione dei *pixels* di ogni *frame* catturato dalla videocamera in pacchetti UDP multipli per superare le problematiche di limitazione del *payload* degli stessi;
2. uso della programmazione asincrona *multithreading* sul lato Hololens (*client*) per gestire tutte le operazioni (ricezione, decodifica, processamento e salvataggio dell'informazione) in modo parallelo (*multithreading*) e non bloccante (asincrono), così da ottimizzare i tempi computazionali (si veda 6.4.1);
3. implementazione del processo di riordino dei *pixels* a destinazione atto a ricomporre l'immagine originale, necessario per cautelarsi contro l'arrivo disordinato dei pacchetti UDP.

Il risultato dell'integrazione dei quattro *stream* video all'interno del visore Microsoft Hololens è mostrato in Fig.89, dove, per semplificare la rappresentazione, essi sono catturati da un solo Kinect, diversamente da quanto considerato nell'architettura di Fig.87. Nell'immagine sottostante catturata dall'interno dell'Hololens, visore e sensore sono posti l'uno di fronte all'altro ed i quattro *stream* sono posizionati agli angoli del FOV del dispositivo HMD.



Fig.89: integrazione di quattro stream video in tempo reale all'interno di HoloLens inviati dalla videocamera RGB del Kinect

I test condotti in laboratorio mediante l'architettura considerata in Fig.87 hanno permesso di valutare il processo del *multi-streaming* video in tempo reale focalizzando l'attenzione sui seguenti due aspetti principali che caratterizzano la valutazione di ogni implementazione:

- qualità video;
- *latency* (ritardo).

La qualità video è a sua volta condizionata da due fattori fondamentali:

- risoluzione: numero di *pixels* caratterizzanti ciascuna immagine;
- *frame rate*: numero di immagini processate al secondo, ovvero *frames per second* (fps).

Maggiore è la risoluzione ed il *frame rate*, superiore sarà il livello di dettaglio dell'immagine e più popolata sarà la sequenza dei *frames* (ovvero minore sarà la capacità dell'occhio umano di distinguere l'avvicinarsi delle immagini) rispettivamente.

Per quanto riguarda la *latency*, essa coincide con il lasso di tempo che intercorre tra l'istante in cui la telecamera cattura l'immagine e quello in cui quest'ultima è mostrata all'utente mediante il dispositivo di visualizzazione.

Nei test condotti in laboratorio, le *performances* dell'Hololens (fps) e la *latency* (ms) sono state valutate in funzione della variazione della risoluzione codificata dai *server*. In altre parole:

1. la videocamera RGB di ogni sensore Kinect acquisisce i *frames* 30 volte al secondo ad una risoluzione di 1920x1080 *pixels*;
2. i *server* (la *workstation* ed il *laptop*):
 - ✓ variano in modo continuo la risoluzione dei *frames* dei quattro *stream* dello stesso quantitativo di *pixels*;
 - ✓ codificano i *frames* mediante il formato di compressione JPG;
 - ✓ inviano i *frames* all'Hololens;
3. l'Hololens:
 - ✓ riceve i quattro *stream*;
 - ✓ decodifica i *frames*;
 - ✓ presenta la successione di immagini (video) dei quattro *stream* su *layers* virtuali collocati nell'ambiente reale attorno all'utente;
4. valutazione del *multi-streaming* video in tempo reale da videocamere ad Hololens mediante monitoraggio degli effetti sulle *performances* del visore (fps) e sulla *latency* (ms);

Mentre le *performances* dell'Hololens vengono calcolate mediante un processo di computazione che coinvolge solamente il visore, per la *latency* non vige lo stesso discorso. Infatti, dal momento che essa coincide con il lasso di tempo che intercorre tra l'istante di acquisizione dell'immagine e quello in cui quest'ultima è mostrata all'utente sull'Hololens, il processo di computazione della *latency* coinvolge più attori ed in particolare più processori con cicli di *clock* differenti e quindi con sistemi di riferimento del tempo differenti. Non è quindi possibile calcolare la *latency* come differenza tra il tempo di presentazione dell'immagine calcolato da Hololens e quello di acquisizione della stessa computato dal *server*, in quanto quest'ultimo e visore non sono caratterizzati da un identico sistema di riferimento del tempo. Si potrebbe quindi pensare di calcolare la *latency* riferendosi ad un orologio atomico sul *web*, facendo in modo che sia i *server* che Hololens consultino tale arbitro imparziale. Tuttavia, anche questo metodo è destinato a fallire a causa del *jitter* di rete (diversi tempi di viaggio dei pacchetti dovuti ai ritardi di accodamento, elaborazione e trasmissione nei router attraversati nel *web*). L'unica soluzione plausibile alla problematica del calcolo della *latency* sta quindi nel computarla rispetto al sistema di riferimento temporale relativo ad un unico processore, quello dei *server*, cronometrando il tempo che intercorre tra l'istante di acquisizione dell'immagine e quello di ricezione del messaggio di *acknowledgement* (ACK) inviato da parte di Hololens: la *latency* sarà data dalla differenza tra tale tempo cronometrato e il tempo di transito del messaggio ACK da Hololens a *server* (Fig.90), quest'ultimo assunto

costante (perché tale è il *jitter* di rete nella LAN) e noto (calcolato da parte di Hololens cronometrando il lasso di tempo che intercorre tra l'invio e la ricezione del messaggio che transita verso il *router* e torna indietro).

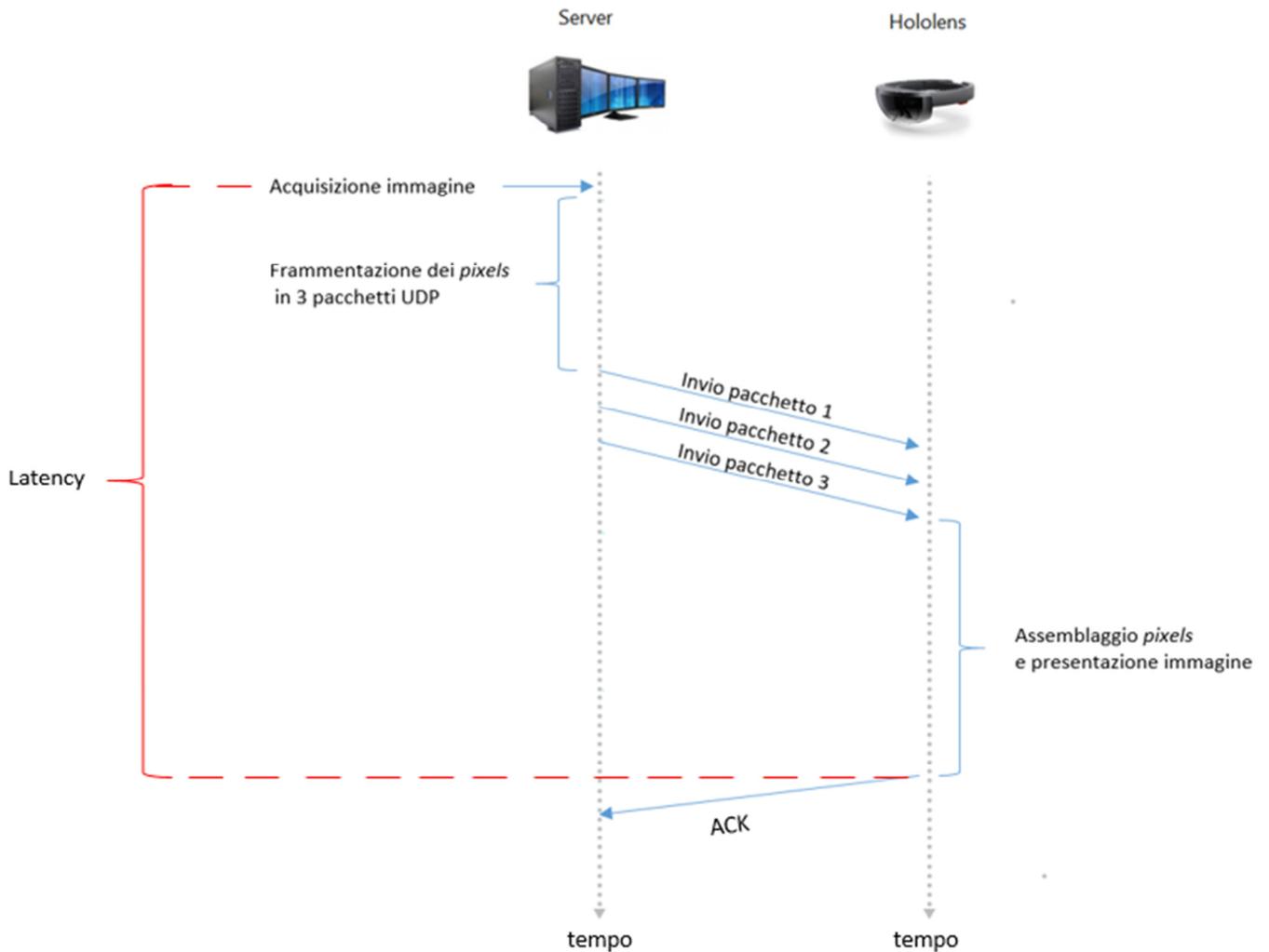


Fig.90: metodo di computazione della latency.

L'implementazione del *multistreaming* verso Hololens è stata caratterizzata dal formato di compressione delle immagini JPG. Esso rappresenta un buon compromesso tra i formati DXT1 e H.264 nel caso il dispositivo Hololens debba ricevere immagini con risoluzioni non troppo elevate [100]. Dagli esperimenti si è evinto che, incrementando parallelamente la risoluzione di ognuno dei quattro *stream* dello stesso quantitativo di *pixels*, le *performances* di Hololens e la *latency* si mantengono stabili rispettivamente a 60 fps ed a 200 ms fino a risoluzioni di 500 *pixels*. Al di sopra

di tale valore il *frame rate* e la *latency* peggiorano drasticamente. Infatti, per risoluzioni maggiori a 500 *pixels*, Hololens non è più in grado di processare le immagini in arrivo, cosicché molti *frames* in *input* al visore vengono scartati, perché il protocollo UDP non è caratterizzato da alcun *buffer* in entrata. Da ciò segue che lo *streaming* dei quattro video assume un andamento a scatti (*jerk*). Al peggiorare delle *performances* di Hololens, i processi di decodifica dei dati e di *rendering* del visore risentono dell'impatto, cosicché i tempi per effettuare tali operazioni si allungano, condizionando in modo negativo la *latency* totale che intercorre tra l'istante di acquisizione dell'immagine da parte del *server* e la presentazione della stessa su Hololens.

Nel prossimo capitolo conclusivo il sistema AR video-based sarà applicato ai due contesti *safety-critical* considerati come casi di studio nell'ambito del corso di dottorato: le torri di controllo aeroportuali e gli operatori *search and rescue* a bordo degli elicotteri.

7 Applicazione del Sistema AR video-based ai casi di studio: torri di controllo aeroportuali ed operatori *search and rescue* a bordo degli elicotteri

La ricerca sul *multi-streaming* video in tempo reale verso gli MR HMD può produrre un impatto positivo su numerosi campi di applicazione nell'industria e nei servizi. A scopo dimostrativo sono stati elaborati due casi di studio attraverso i quali si vogliono evidenziare i potenziali benefici prodotti in alcuni contesti *safety-critical* ed in particolare in applicazioni relative alle torri di controllo aeroportuali ed in applicazioni che coinvolgono operatori di servizi *search and rescue* collocati a bordo di elicotteri di soccorso.

L'implementazione degli AR *overlays* in entrambi i casi di studio ha comportato l'integrazione di ogni tipologia di informazione digitale comunicata per mezzo dei protocolli di scambio dati UDP/TCP: stringhe di testo, *textures*, *stream* video multipli in tempo reale ed oggetti grafici 2D/3D vengono trasmessi e generati sulla base di una connessione *wireless*.

7.1 Caso di studio n° 1 - Torri di controllo aeroportuali

Insieme agli AR *overlays* caratterizzanti l'ambito aeroportuale di cui si è molto discusso nei capitoli precedenti in relazione al progetto RETINA (Fig.73) ed al *sistema AR adattivo e collaborativo* (Fig.79), il *multistreaming* video in tempo reale verso dispositivi HMD apporterebbe degli impatti fortemente positivi sugli operatori incaricati a monitorare il traffico aereo dalle torri di controllo. Si pensi ad integrare quattro *stream* derivanti da altrettante videocamere IR con capacità di visione a lunga distanza poste ad esempio in cima alla torre di controllo ed orientate in modo da fornire una panoramica a 360° sull'aeroporto: in condizioni notturne, di maltempo, di scarsa visibilità, o in caso di strutture ed edifici occludenti la vista del controllore *ground* (avente la responsabilità sul traffico aereo in movimento nell'area dei parcheggi e sulle vie di circolazione), quest'ultimo sarebbe supportato da una tecnologia in grado di aumentare fortemente la *situational awareness* e quindi la sicurezza nella gestione delle operazioni di movimentazione degli aeromobili sul *ground* aeroportuale, contesto in cui avviene la grande maggioranza degli incidenti aerei.

Analogamente, gli stessi impatti positivi riguarderebbero il controllore *tower*, incaricato a gestire le operazioni sulle piste di volo. L'applicazione implementata è descritta in Fig.91, in cui una videocamera IR con capacità di visione a lunga distanza segue un aereo in atterraggio ed invia lo *stream* in tempo reale al dispositivo HMD indossato dal controllore *tower*.

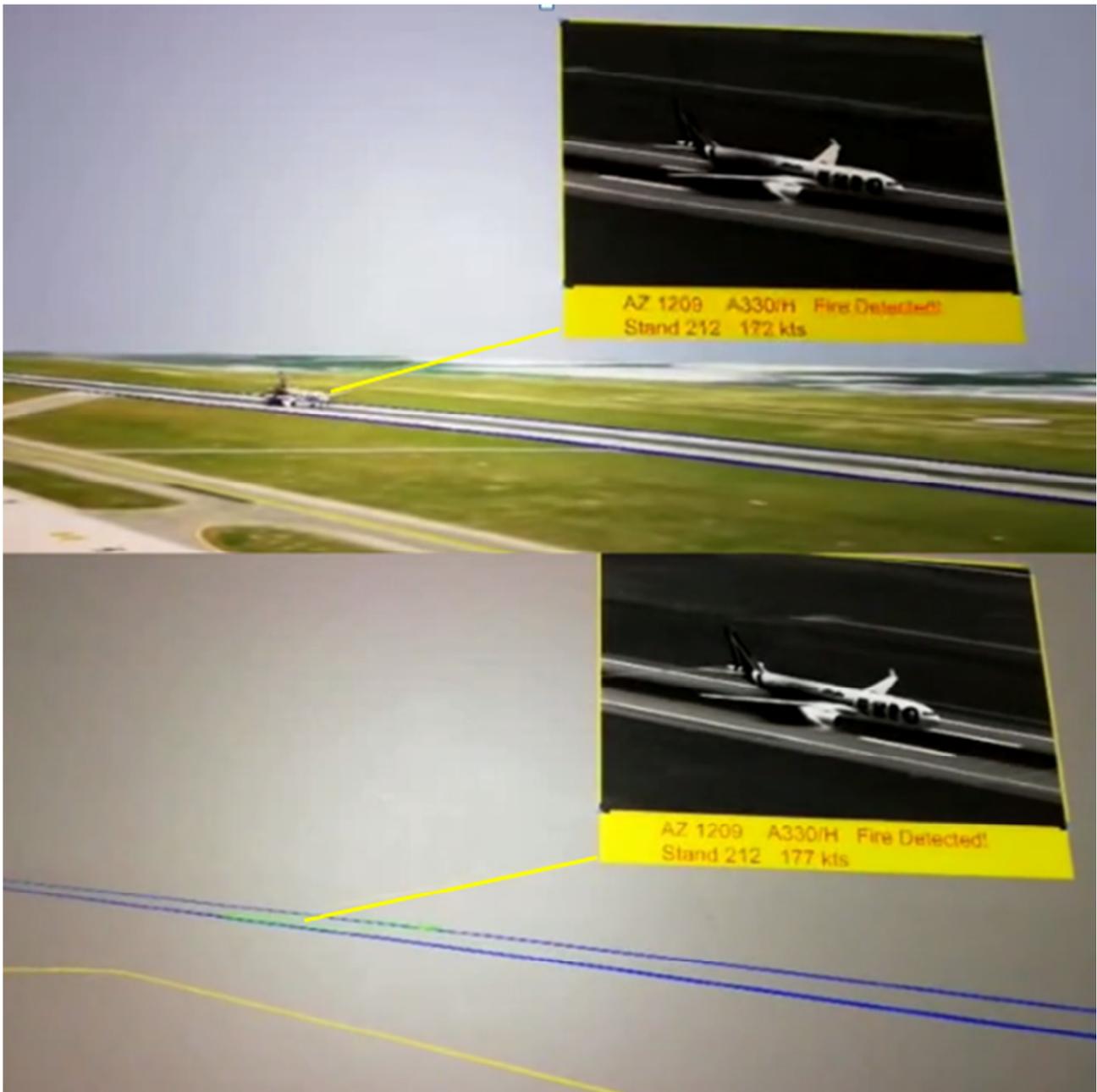


Fig.91: streaming video in tempo reale al dispositivo HMD indossato dal controllore tower. L'atterraggio è monitorato in modo continuo in ogni condizione di visibilità. In caso di nebbia (parte sottostante), il controllore è in grado di accorgersi immediatamente della generazione dell'incendio ed allertare prontamente i mezzi di primo soccorso.

In particolare, il focus è concentrato su una situazione di incendio generatasi a causa di un motore in panne durante una procedura di atterraggio nell'aeroporto di Bologna in due differenti scenari: buone condizioni di visibilità e nebbia. Lo *stream* video in tempo reale inviato dalla videocamera al dispositivo HMD potrebbe migliorare fortemente la *situational awareness* del controllore *tower*, permettendo allo stesso di monitorare l'atterraggio in ogni istante (ovvero in tempo continuo, senza alcuna perdita di informazione tra due tracce radar successive) e condizione di visibilità. In modo specifico, il controllore *tower* è in grado di accorgersi immediatamente della generazione dell'incendio ed allertare prontamente i mezzi di primo soccorso; al contrario, senza lo *streaming*

video in tempo reale, in condizioni di nebbia il controllore *tower* è completamente impossibilitato ad accorgersi dello scaturirsi dell'incendio perché l'operatore non è in grado di vedere nulla al di fuori dei vetri della torre di controllo. Gli AR *overlays* considerati sull'Hololens e sovrapposti sull'ambiente virtuale visualizzato sullo schermo *Wall* sono costituiti da:

- ILS (verde),
- pista (blu),
- percorso verso il parcheggio (giallo),
- superficie di ingombro dell'aereo (verde),
- *label* con le informazioni significative relative all'aereo,
- connettore *label*-posizione (giallo),
- video *streaming* in tempo reale dalla telecamera IR con capacità di visione a lunga distanza.

In particolare, la sola presenza dell'ultimo overlay soprariportato avrebbe potuto evitare il catastrofico disastro aereo del volo 8303 avvenuto nel mese di maggio nei dintorni dell'aeroporto di Karachi. Dalle prime investigazioni, risulta che l'incidente è avvenuto per la congiunzione di due cause:

- problema con l'estrazione del carrello,
- altitudine troppo elevata durante la procedura di atterraggio.

Senza avvertire il controllore del problema al carrello, il pilota ha dichiarato di voler continuare con la procedura di atterraggio, sebbene troppo alto sul sentiero di discesa. Giunto in pista, l'aereo ha raschiato entrambi i motori per l'impatto con la stessa e poi ha abortito la procedura di atterraggio perché è arrivato troppo lungo sulla pista. Quindi è risalito in quota, ma i motori, ormai ovalizzati, hanno ceduto entrambi. Si immagina ora se il controllore fosse stato in grado di accorgersi della mancata estrazione del carrello durante la discesa dell'aereo in atterraggio. Sicuramente sarebbe andato tutto diversamente, perché avrebbe comandato la procedura di *Go-Around* mediante la quale l'aereo sarebbe stato costretto ad abortire l'attuale procedura di atterraggio, in quanto non si può far atterrare un aereo con un problema al carrello senza preparare ed attivare le opportune misure di emergenza, come:

- l'intervento dei vigili del fuoco
- l'innaffiamento della pista
- lo scarico del carburante da parte dell'aereo.

Ebbene, lo *streaming* video in tempo reale verso l'HMD avrebbe dato la possibilità al controllore di accorgersi del problema col carrello e quindi di evitare quella catastrofe. Infatti, l'operatore *tower* avrebbe comandato la procedura di *Go-Around*. Poi, nel successivo tentativo di atterraggio, il pilota si sarebbe potuto allineare correttamente con il sentiero di discesa ILS e quindi, senza arrivare lungo

sulla pista, non avrebbe riportato l'aereo in quota dopo aver ovalizzato i motori. A tal proposito è stata implementata una applicazione in cui sono stati riportati:

- tutti i dettagli dell'incidente (reperiti dal programma TV “*TheFlightChannel*”)
- una simulazione con alto grado di fedeltà (modello aeroportuale di Karachi georeferenziato e torre di controllo correttamente posizionata) di come si sarebbe potuta evitare la catastrofe mediante l'utilizzo dello *streaming* video in tempo reale verso la tecnologia MR HMD (Fig.92).

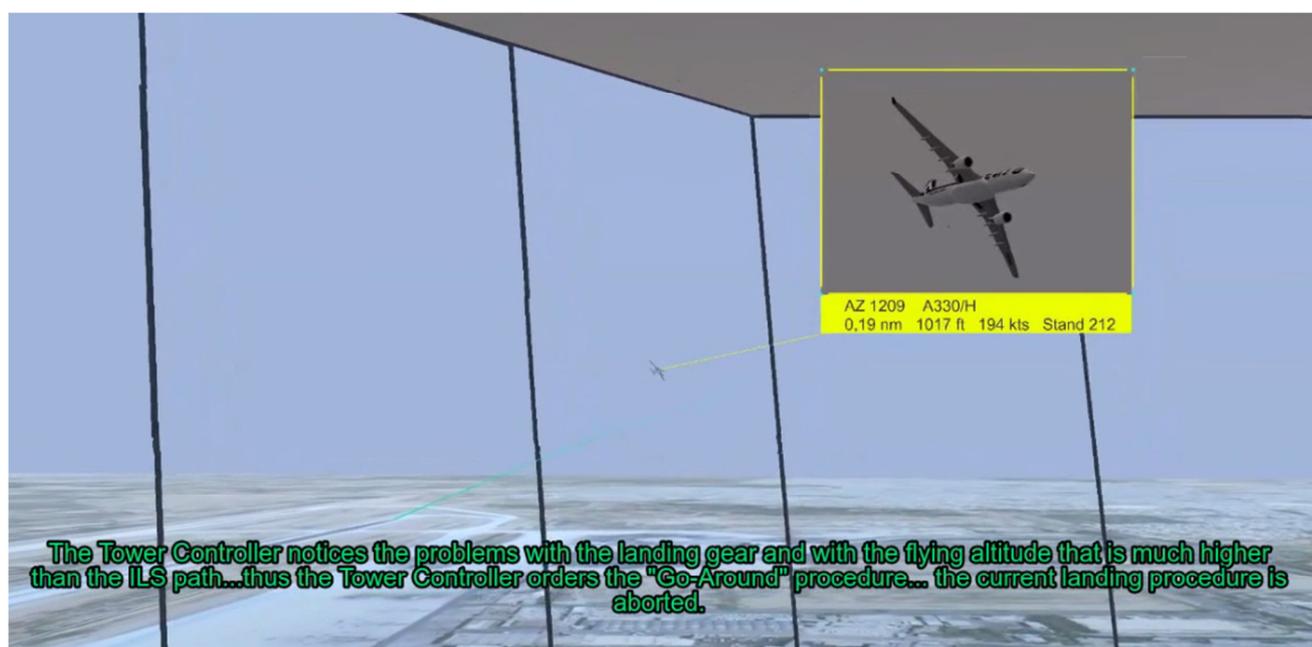


Fig.92: lo streaming video in tempo reale verso l'HMD avrebbe potuto dare al controllore la possibilità di accorgersi del problema col carrello e quindi di evitare il disastro aereo del volo 8303 a Karachi.

Per concludere la trattazione di tale caso di studio, è bene notare che lo *streaming* video in tempo reale verso i MR HMD non solo, come dimostrato, permetterebbe di migliorare significativamente la *situational awareness* e quindi la sicurezza nella gestione dei voli da parte di entrambi i controllori *ground* e *tower*, ma, grazie al monitoraggio continuo degli aeromobili (contrariamente a quanto avviene attualmente con perdita di informazioni tra due tracce radar successive), apporterebbe anche degli impatti fortemente positivi nell'ottimizzazione dell'uso della pista in caso di operazioni miste (partenze ed arrivi sulla stessa pista) oppure in condizioni di emergenza.

7.2 Caso di studio n° 2 - Operatori *search and rescue* a bordo degli elicotteri

Senza perdita di generalità, il caso di studio contempla una emergenza in mare con una persona da salvare. Le due ipotesi principali sono le seguenti (Fig.93):

- 1) sono note le coordinate GPS dell'uomo in mare;
- 2) l'elicottero dispone di una telecamera IR con capacità di visione a lunga distanza ed un sistema avanzato di stabilizzazione ottica dell'immagine; tale dispositivo è orientato ad ogni istante verso la posizione della persona da salvare.

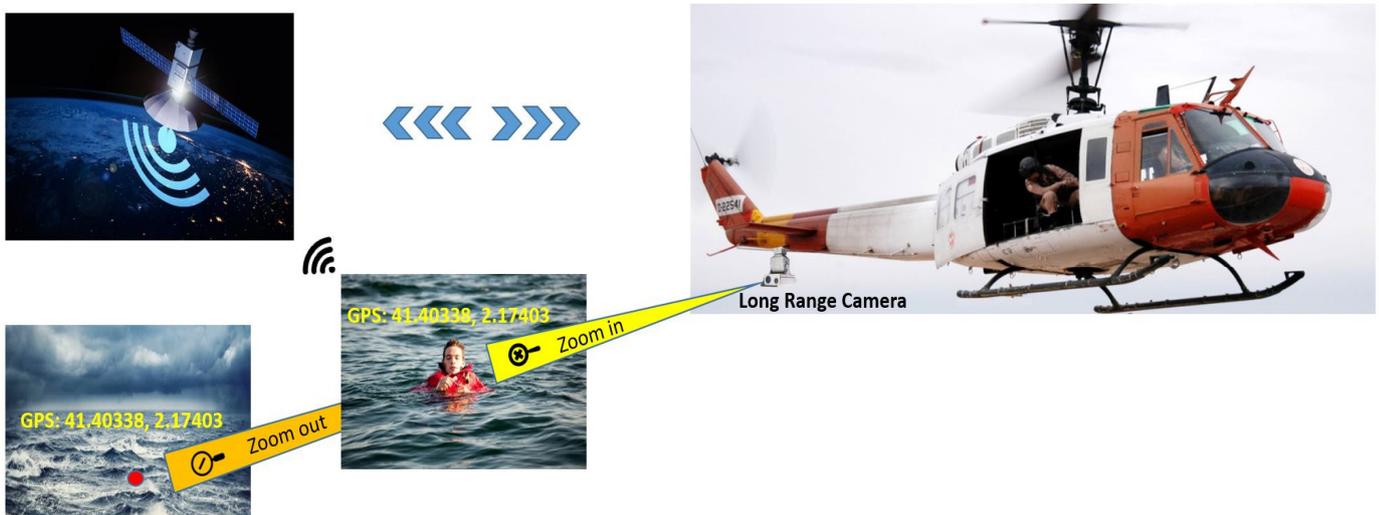


Fig.93: le due ipotesi su cui si basa il caso di studio relativo agli operatori *search and rescue* a bordo degli elicotteri.

Tali ipotesi permettono di estendere il caso di studio ad altri contesti simili, come per esempio quello dei mezzi di primo soccorso (Polizia, Croce Rossa, Vigili del Fuoco) o quello della Difesa. In quest'ultimo, l'assunzione della persona da salvare può essere facilmente rimpiazzata dalla presenza di un *target* rappresentante una minaccia o un nemico.

La principale motivazione alla base dello studio è insita nel processo di evoluzione dei *task* degli operatori *search and rescue*. Con l'introduzione degli schermi fissi dentro agli elicotteri, lo sguardo dell'operatore è stato distolto dall'ambiente reale fuori dall'elicottero (*head-out view*), dal momento che la maggior parte dell'informazione è attualmente disponibile sui *monitor* e *display* collocati all'interno della cabina (*head-in view*). Da ciò segue che tali dispositivi di visualizzazione incrementano il *workload* e riducono la *situational awareness* degli operatori, obbligandoli

continuamente a trasferire il loro sguardo dall'equipaggiamento di bordo all'ambiente reale fuori dall'elicottero e viceversa, al fine di contestualizzare nel mondo fisico l'informazione presentata dai dispositivi di visualizzazione (Fig.94).

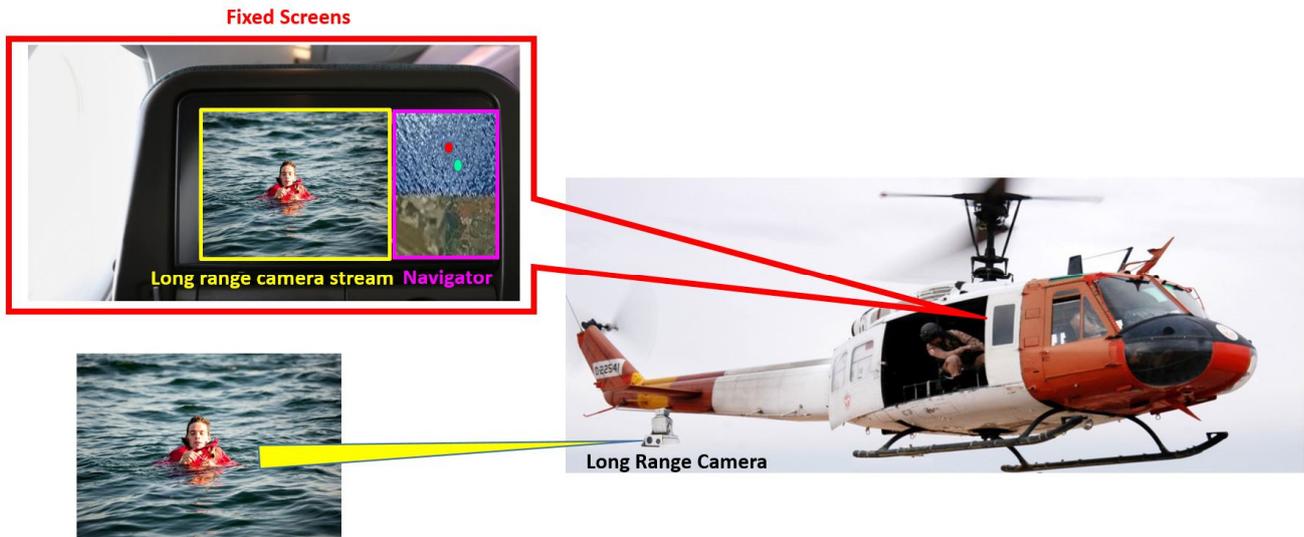


Fig.94: scenario operativo attuale - uso di monitor e display collocati all'interno dell'elicottero

D'altro canto, i MR HMD offrono l'opportunità di trasferire tutta l'informazione utile per lo svolgimento dei *task* dai dispositivi fissi collocati dentro all'elicottero verso l'ambiente esterno, mediante oggetti grafici 2D o 3D e *frames* video collimati con il mondo reale attorno al velivolo. Sovrapponendo l'informazione digitale direttamente sull'ambiente reale, i MR HMD permettono di processare entrambi in modo simultaneo, eliminando la necessità di connettere mentalmente i dati digitali al mondo fisico. In questo modo, consentendo all'operatore di focalizzare totalmente la propria attenzione sull'ambiente reale, i MR HMD potrebbero abbattere il carico cognitivo ed aumentare la *situational awareness* nei *task* di monitoraggio, soprattutto in quelle condizioni in cui le capacità visive naturali sono limitate dal maltempo, dalla distanza o da elementi occludenti. Una possibile soluzione di scelta e di configurazione degli AR *overlays*, la cui fattibilità necessita ancora di essere opportunamente valutata dagli *stakeholders* del settore, è riportata in Fig.95 ed è costituita da:

- mirino registrato con la posizione della persona da salvare;
- altitudine e distanza dal target;
- *stream* video in tempo reale proveniente dalla *long range camera* ed ancorato al mirino;

- *stream* video in tempo reale dal navigatore di bordo.

In aggiunta ai suddetti overlays, potrebbero anche essere considerate degli indicatori digitali (freccie) per invitare l'operatore ad orientare la propria attenzione verso la posizione della persona da salvare.

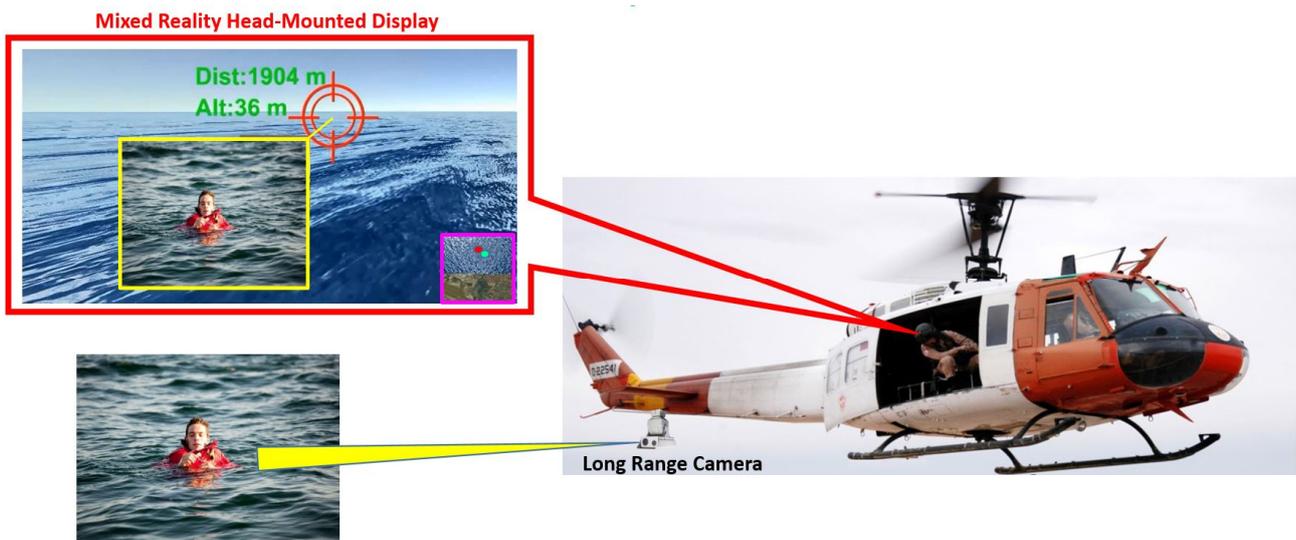


Fig.95: scenario operativo di progetto - migrazione dell'informazione digitale dai monitor fissi verso i MR HMD.

Come visto in 6.4.2, la tecnologia Hololens non riesce a processare alte risoluzioni (superiori a 500 *pixels*) quando l'applicazione coinvolge *stream* multipli (quattro o più *stream*). Ciononostante, questo fatto non rappresenta una problematica per il caso di studio in esame: le immagini acquisite dalla *long range camera* dell'elicottero possono essere codificate ed inviate al visore ad una risoluzione che è appena sufficiente a permettere all'operatore di essere pienamente consapevole sugli sviluppi della situazione. A tale scopo 500 *pixels* possono bastare e ciò è confermato dal fatto che molte *long range camera* con capacità di rilevamento della presenza umana basano il proprio funzionamento su tali valori di risoluzione. Tali risultati incoraggianti hanno permesso l'implementazione della piattaforma di validazione in ambiente di laboratorio. Essa è caratterizzata dall'integrazione di due applicazioni: una atta a funzionare sul sistema CAVE e l'altra su Hololens. Il primo fornisce l'ambiente immersivo di Realtà Virtuale che replica con un alto grado di fedeltà l'ambiente reale di cabina di elicottero in cui opera l'operatore *search and rescue*. Così, lo *scenario operativo attuale* con uso di *monitor* e *display* collocati all'interno dell'elicottero è simulato mediante il sistema CAVE (Fig.96).



Fig.96: scenario operativo attuale simulato sul CAVE

D'altra parte, analogamente al progetto RETINA, Hololens abilita la sovrapposizione dei MR *overlays* sull'ambiente virtuale immersivo del CAVE, cosicché lo *scenario operativo di progetto* (rimozione dei *monitor* di cabina e trasferimento della relativa informazione sul dispositivo HMD) è simulato mediante l'integrazione tra l'Hololens (rappresentante la tecnologia HMD indossata dagli operatori *search and rescue*) e CAVE (Fig.97).

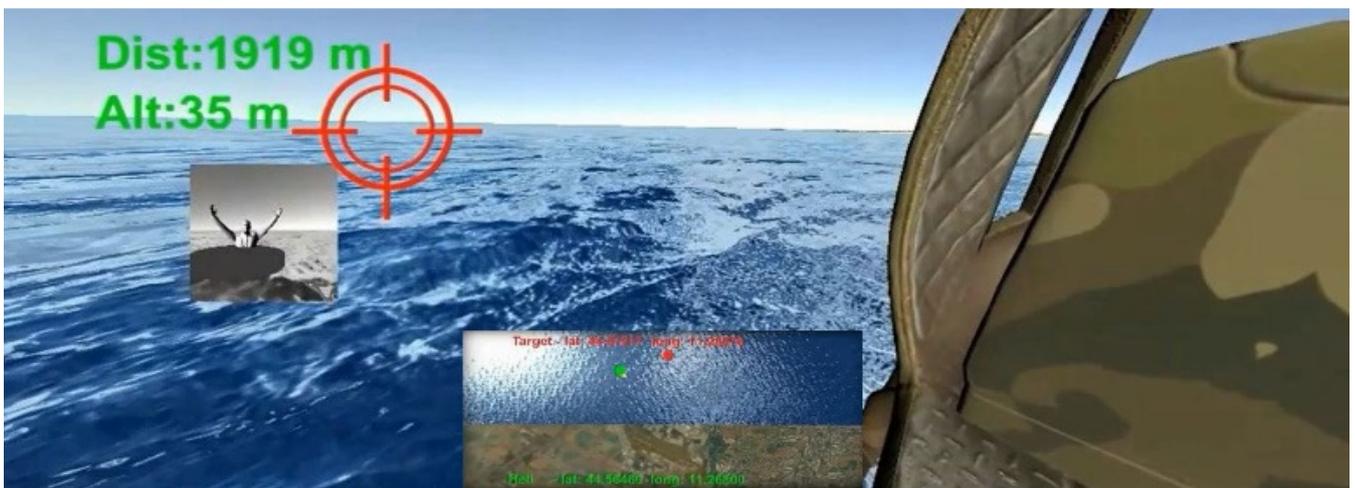


Fig.97: scenario operativo di progetto simulato mediante integrazione CAVE-Hololens.

8 Risultati

Il risultato finale dell'elaborato di tesi è rappresentato principalmente da un *Sistema AR Adattivo, Collaborativo e Video-Based*. Esso ha un carattere flessibile in quanto può essere generalizzato e facilmente esteso a numerosi campi di applicazione nell'industria e nei servizi. Al fine di dimostrare il valore aggiunto di un sistema di questo tipo, sono stati sviluppati, a livello simulativo, due casi di studio, uno relativo alle torri di controllo aeroportuali e l'altro riguardante la riprogettazione dell'interfaccia uomo macchina degli operatori *search and rescue* a bordo degli elicotteri di soccorso. Oltre a questi due esempi, entrambi nel campo aeronautico, questo tipo di architettura AR potrebbe essere sperimentata in altri contesti operativi nel settore dei trasporti in generale, per migliorare le prestazioni in termini di sicurezza ed efficienza, di interfaccia uomo macchina sia di comando e controllo, sia di guida e navigazione.

Tale tipologia di applicazione è stata selezionata proprio perché il suo carattere *safety-critical*, in cui il fallimento delle strategie può causare la perdita di vite umane, mira ad aiutare a comprendere le potenzialità dello studio. Per massimizzare le prestazioni e minimizzare i rischi, si è pensato ai due aspetti quali:

- *la creazione di un ambiente di Realtà Aumentata che consentisse di lavorare all'interno di un team e favorisse la suddivisione di task, responsabilità e workload diversi fra utenti con differenti obiettivi operativi;*
- *lo sviluppo di un'interfaccia orientata al miglioramento delle human performances in termini di riduzioni del carico di lavoro (workload), incremento della situational awareness e di una migliore accessibilità all'informazione utile per eseguire i task, presentata esattamente nel momento in cui si ha necessità di essa.*

Il sistema AR sviluppato risulta:

- *adattivo* (gli operatori condividono lo stesso ambiente, ma sperimentano differenti visualizzazioni AR: uno stesso visore può essere programmato per fornire automaticamente differenti *set* di *overlays* a seconda del *task* svolto, delle fasi operative, nonché delle condizioni ambientali)
- e
- *collaborativo* (lo scambio di informazioni avviene fra tutte le componenti e fra tutti i livelli – fra gli operatori, nonché fra gli stessi ed i loro supervisori, quest'ultimi aventi il compito di ottimizzare le strategie di *teamwork* in tempo reale).

L'obiettivo di concepire un sistema che fosse efficiente anche dal punto di vista dai Fattori Umani (*human factors*) ha portato la ricerca ad esplorare la direzione dell'integrazione del *multistreaming* video in tempo reale all'interno di *layers* virtuali di sistemi AR ottici, ovvero visori *see-through* che

permettono all'utente di sovrapporre l'informazione digitale aggiuntiva al mondo reale che ha attorno. E' stata proposta una soluzione ibrida derivante dalla fusione delle differenti funzionalità che caratterizzano entrambe le categorie distinte (tecnologia AR ottica e video) in cui vengono suddivisi tutti i dispositivi AR. Al fine di ciò, è stato concettualizzato, sviluppato ed implementato un efficiente metodo di comunicazione basato sulla frammentazione dei pacchetti del protocollo UDP. Dai risultati degli esperimenti condotti sul dispositivo Microsoft HoloLens HMD in ambiente di laboratorio si è evinto che, sebbene prontamente reattivo in termini di *latency*, il visore non è in grado di processare alte risoluzioni nel caso di *stream* video multipli ad esso inviati. Infatti, incrementando parallelamente la risoluzione di ognuno dei quattro *stream* (lo *stream* risultante è in grado di coprire un angolo di 360°, permettendo di focalizzare l'indagine sulla trattazione dello *streaming* video catturato da una ipotetica telecamera omnidirezionale) dello stesso quantitativo di *pixels*, le *performances* di HoloLens e la *latency* si mantengono stabili rispettivamente a 60 fps ed a 200 ms fino a risoluzioni di 500 *pixels*. Al di sopra di tale valore il *frame rate* e la *latency* peggiorano drasticamente. Ciononostante, questo fatto non rappresenta una problematica per i casi di studio esaminati nel presente elaborato: le immagini acquisite dalla *long range camera* della torre di controllo o dell'elicottero possono essere codificate ed inviate al visore ad una risoluzione che è appena sufficiente a permettere all'operatore di essere pienamente consapevole sugli sviluppi della situazione in atto. D'altronde, molte *long range camera* con capacità di rilevamento della presenza umana che sono in commercio basano il proprio funzionamento su valori di risoluzione intorno ai 500 *pixels*.

9 Conclusioni e sviluppi futuri

Lo scopo generale della ricerca è stato quello di concettualizzare, sviluppare e testare nuovi concetti di interfacce AR per le applicazioni nei settori dell'industria e dei servizi, tenendo conto della rapida evoluzione dei sistemi e dei dispositivi dedicati a tale tecnologia che sta diventando sempre più matura e popolare.

La ricerca condotta ha permesso di:

- 1) realizzare applicazioni di tracking dell'utente basate su di una interfaccia astratta in grado di acquisire dati da qualunque hardware di tracking.
- 2) Implementare efficienti processi di scambio dati tra dispositivi AR, abilitando in tal modo la condivisione delle informazioni e la collaborazione tra utenti.
- 3) Ideare, sviluppare e testare un nuovo concetto di interfaccia AR: l'integrazione del *multistreaming* video in tempo reale all'interno dei *layers* virtuali di un Sistema AR ottico. Tale filone di ricerca innovativo ha permesso di introdurre una soluzione ibrida integrante le differenti funzionalità dei due diversi approcci finora seguiti nella costruzione di ogni sistema AR, rappresentati dalla tecnologia ottica o quella video.
- 4) Testare efficientemente il concetto del *multistreaming video in tempo reale verso i sistemi AR ottici* nei due casi di studio simulati in ambiente di laboratorio: torri di controllo aeroportuali ed operatori *search and rescue* a bordo degli elicotteri.

Per concludere, la ricerca condotta ha prodotto come risultato un efficiente *Sistema AR Adattivo, Collaborativo e Video-Based*, in cui:

- gli utenti condividono lo stesso ambiente, ma sperimentano differenti visualizzazioni AR: uno stesso visore può essere programmato per fornire automaticamente differenti *set* di *overlays* a seconda del *task* svolto, delle fasi operative, nonché delle condizioni ambientali (*Adattivo*);
- lo scambio di informazioni avviene fra tutte le componenti e fra tutti i livelli – fra gli operatori, nonché fra gli stessi ed i loro supervisori, quest'ultimi aventi il compito di ottimizzare le strategie di *teamwork* in tempo reale (*Collaborativo*);
- la tecnologia AR ottica è abilitata alla ricezione di *stream* video multipli in tempo reale da differenti videocamere e/o *display* (“*Video-Based*”).

Quest'ultima funzionalità è stata concettualizzata, sviluppata ed implementata mediante un efficiente metodo di comunicazione basato sulla frammentazione dei pacchetti del protocollo UDP. Dai risultati degli esperimenti condotti sul dispositivo Microsoft HoloLens HMD in ambiente di

laboratorio si è evinto che, sebbene prontamente reattivo in termini di *latency*, il visore non è in grado di processare alte risoluzioni nel caso di *stream* video multipli ad esso inviati. Ciononostante, questo fatto non rappresenta una problematica per i casi di studio esaminati nel presente elaborato: le immagini acquisite dalla *long range camera* della torre di controllo o dell'elicottero possono essere codificate ed inviate al visore ad una risoluzione che è appena sufficiente a permettere all'operatore di essere pienamente consapevole sugli sviluppi della situazione in atto.

Il risultante *Sistema AR Adattivo, Collaborativo e Video-Based* potrebbe produrre un impatto positivo in numerosi settori dell'industria e dei servizi, non solo su quelli che adottano il paradigma della *collaborazione remota*. I seguenti contesti sembrano essere al momento quelli più promettenti:

- *centri di controllo nel campo dei trasporti e della security, personale di primo soccorso, search and rescue e Difesa*: uso di tecnologia AR per incrementare la *situational awareness* degli operatori, nonché l'efficienza e la *safety/security* dell'intero sistema;
- *collaborazione remota*: consultazione e/o condivisione di video su richiesta in relazione alle modalità di svolgimento di un *task*, specialmente in ambienti non facilmente accessibili (*Oil & Gas* ed aerospazio);
- *training*: simulazione di *task* reali;
- *chirurgia*: fornitura di AR *overlays* per rendere maggiormente efficiente l'operato del chirurgo, eventualmente assistito da una *equipe* in remoto mediante *streaming* video in tempo reale;
- *healthcare security*: monitoraggio e controllo di ambienti affollati mediante sistemi AR atti a misurare la temperatura corporea al fine di evitare la diffusione di virus.

Per quanto riguarda gli sviluppi futuri, ulteriori studi arricchiranno la promettente ricerca sul *multistreaming* video in tempo reale, andando a suddividere la *latency* complessiva nelle sue molteplici componenti (acquisizione immagine, codifica, trasmissione, decodifica, *rendering*), al fine di investigare ed ottimizzare il flusso dati lungo l'intero percorso tra l'acquisizione dell'immagine e la proiezione della stessa sul *display* del visore, considerando anche l'impatto dei diversi formati di compressione delle immagini (DXT1, H.264, etc.).

Inoltre, una volta stabiliti definitivamente gli AR *overlays* da adottare (mediante sessioni di *brainstorming* con gli *stakeholders* del settore), la piattaforma di test implementata per il caso degli operatori *search and rescue* a bordo degli elicotteri potrà essere validata, similmente a ciò che è stato fatto per il Progetto RETINA, attraverso sperimentazioni di laboratorio eseguite mediante una campagna di simulazioni *human-in-the-loop* atta a coinvolgere gli operatori *search and rescue*. Lo scopo è quello di confrontare

- lo scenario attuale, caratterizzato dalla presenza di *monitor* e *display* fissi collocati all'interno della cabina (*head-in view*),

con

- lo scenario di progetto, dove tutta l'informazione digitale viene migrata sui dispositivi MR HMD indossati dagli operatori *search and rescue* (*head-out view*).

L'obiettivo finale consisterà nel dimostrare l'impatto positivo apportato dai visori MR in termini di registrazione degli AR *overlays* (allineamento tra l'informazione digitale e gli oggetti del mondo fisico reale attorno all'utente) e di *human factors*. Il *multistreaming* video in tempo reale verso i dispositivi HMD potrebbe apportare dei forti miglioramenti in termini di *workload* e *situational awareness* degli operatori nei *task* di monitoraggio di vasti ambienti, soprattutto in quelle condizioni in cui le capacità visive naturali sono limitate dal maltempo, dalla distanza o da elementi occludenti.

10 Bibliografia

- [1] Azuma, R. "A Survey of Augmented Reality" *Presence: Teleoperators and Virtual Environments* 06.04 (1997): 355–385.
- [2] Milgram, P., Kishino F. "A taxonomy of mixed reality visual displays". *IEICE TRANSACTIONS on Information and Systems* 77.12 (1994): 1321-1329.
- [3] Peterson, Stephen, and Ella Pinska, "Human Performance with simulated Collimation in Transparent Projection Screens" in *Proceedings of the Second International Conference on Research in Air Transportation (ICRAT)*, 2006.
- [4] O. Bimber and R. Raskar, *Spatial Augmented Reality: Merging Real and Virtual Worlds*. Natick, MA, USA: A. K. Peters, Ltd., 2005.
- [5] C. Anthes, R. J. Garcia-Hernandez, M. Wiedemann, and D. Kranzlmuller, "State of the art of virtual reality technology" 2016, pp. 1–19.
- [6] N. Masotti, F. De Crescenzo, and S. Bagassi, "Augmented Reality in the Control Tower: A Rendering Pipeline for Multiple Head-Trackable Head-up Displays" *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, Springer International Publishing, 2016.
- [7] Jon Peddie. "Augmented Reality: Where We Will All Live." Springer International Publishing, 2017.
- [8] "A Manager's Guide to Augmented Reality". *Harvard Business Review* - 2017.
- [9] Wickens, C. D., Hollands, J. G., Banbury, S., & Parasuraman, R. "Engineering psychology and human performance". Psychology Press, 2015.
- [10] Sheridan, Thomas B. "Humans and automation: System design and research issues". *Human Factors and Ergonomics Society*, 2002.
- [11] Merlino, Massimo, and Ilze Spröge. 2017. "The Augmented Supply Chain." In *Procedia Engineering*, 178:308–18. Elsevier Ltd. doi:10.1016/j.proeng.2017.01.053.
- [12] Caudell, T. P., & Mizell, D.W.. (1992, January 7–10). *Augmented reality: An application of head-up display technology to manual manufacturing processes, research. & technology* Boeing Computer Services, Seattle, WA, USA, System Sciences, 1992.
- [13] Feiner, S.K., Seligmann, D.D., "Cutaways and ghosting: satisfying visibility constraints in dynamic 3D illustrations". *The Visual Computer* 8, 292–302 (1992). <https://doi.org/10.1007/BF01897116>.
- [14] C. Sandor, A. Cunningham, A. Dey and V. Mattila, "An Augmented Reality X-Ray system based on visual saliency," 2010 IEEE International Symposium on Mixed and Augmented Reality, Seoul, 2010, pp. 27-36, doi: 10.1109/ISMAR.2010.5643547.
- [15] S. Zollmann, D. Kalkofen, E. Mendez, and G. Reitmayr, "Image-based ghostings for single layer occlusions in augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2010)*. IEEE, Oct. 2010, pp. 19–26.

- [16] S. Zollmann et al., "Image-based x-ray visualization techniques for spatial understanding in outdoor augmented reality," in Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design. ACM, 2014.
- [17] Livatino, Salvatore, et al. "Stereoscopic visualization and 3-D technologies in medical endoscopic teleoperation", IEEE Transactions on Industrial Electronics 62.1 (2014): 525-535.
- [18] Feng, Chuan, Jerzy W. Rozenblit, and Allan J. Hamilton. "A computerized assessment to compare the impact of standard, stereoscopic, and high-definition laparoscopic monitor displays on surgical technique", Surgical endoscopy 24.11 (2010): 2743-2748.
- [19] Lee, K. Augmented Reality in Education and Training. TECHTRENDS TECH TRENDS 56, 13–21 (2012). <https://doi.org/10.1007/s11528-012-0559-3>.
- [20] Kesim, M. Tumerkan and Yasin Ozarslan. "Augmented Reality in Education: Current Technologies and the Potential for Education." Procedia - Social and Behavioral Sciences 47 (2012): 297-302.
- [21] Hsin-KaiWu, Silvia Wen-Yu Lee, Hsin-YiChang, Jyh-ChongLiang, "Current status, opportunities and challenges of augmented reality in education", Computers & Education, Volume 62, March 2013, Pages 41-49.
- [22] F. De Crescenzo, M. Fantini, F. Persiani, L. Di Stefano, P. Azzari and S. Salti, "Augmented Reality for Aircraft Maintenance Training and Operations Support," in IEEE Computer Graphics and Applications, vol. 31, no. 1, pp. 96-101, Jan.-Feb. 2011.
- [23] «Form Smart Swim Goggles,» [Online]. Available: <https://www.formswim.com/>.
- [24] Endsley, Mica R. "Toward a theory of situation awareness in dynamic systems". Human factors, 1995, 37(1), 32-64.
- [25] Silva, A., Ferreira, E., Laforge, N., Carvalho, S., "The use of surveillance data in augmented reality system to assist control tower operations"; IEEE, 3rd Experiment International Conference, 2015.
- [26] Bagassi, S., De Crescenzo, F., Lucchi, F., Persiani, F. "Innovation in man machine interfaces: use of 3D conformal symbols in the design of future HUDs". 28th International Congress of the Aeronautical Sciences, Brisbane, Australia, Vol. 23, Sept. 2012.
- [27] Spitzer, C.R., "The avionics handbook", CRC Press, 2001, pp 542.
- [28] Peinecke, N., Schmerwitz, S., Doehler, H.-U., Lueken, T.: Review on Conformal Displays: More Than Highway in the Sky. Optical Engineering, 56 (5). Society of Photo-optical Instrumentation Engineers (SPIE). 2017.
- [29] Viertler, F. X.: Visual Augmentation for Rotorcraft Pilots in Degraded Visual Environment (Doctoral dissertation, Technische Universität München). 2017.
- [30] Tran, T.H., Behrend, F., Fuenning, N., Arango, A.: Single Pilot Operations with AR-Glasses using Microsoft HoloLens, IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), Sept. 2018, DOI: 10.1109/DASC.2018. 8569261.
- [31] «F-35: Under the Helmet of the World's Most Advanced Fighter,» [Online]. Available: <https://www.aviationtoday.com/2018/08/24/f-35-helmet-worlds-advanced->

- [47] «Whitepaper: Introduction to remote virtual tower,» [Online]. Available: https://www.frequentis.com/sites/default/files/support/2018-02/RVT_whitepaper.pdf.
- [48] M. Schmidt, M. Rudolph, B. Werther, N. Fürstenau, "Remote Airport Tower Operation with Augmented Vision Video Panorama HMI". published in: Proc. 2nd Int. Conf. Research in Air Transportation ICRAT 2006, June 24.28, Belgrade, Serbia, pp. 221-230.
- [49] M. Schmidt et al., "Remote airport traffic control center with augmented vision video panorama," 2009 IEEE/AIAA 28th Digital Avionics Systems Conference, Orlando, FL, 2009, pp. 4.E.2-1-4.E.2-15, doi: 10.1109/DASC.2009.5347479.
- [50] Fürstenau N. "Virtual and Remote Control Tower - Research, Design, Development and Validation". Springer, 2016, 336 pages.
- [51] Fürstenau N. "Design and Analysis of Virtual Airport Control Tower Systems and Remote Tower Centers". IFAC 2010, Valenciennes.
- [52] Fürstenau N., Mittendorf M., Ellis S.R. "Remote Towers: Videopanorama Frame Rate Requirements Derived from Visual Discrimination of Deceleration during Simulated Aircraft Landing". SESAR Innovation Days, 27th – 29th November 2012, Braunschweig.
- [53] J. López-Araquistain, I. Campaña, L. Bergesio and J. A. Besada, "Experimental prototype for remote tower systems design", Integrated Communications, Navigation and Surveillance Conference (ICNS), Herndon, VA, 2017, pp. 1D2-1-1D2-9, doi:10.1109/ICNSURV.2017.8011891.
- [54] Hagl M. et al. (2018) Augmented Reality in a Remote Tower Environment Based on VS/IR Fusion and Optical Tracking. In: Harris D. (eds) Engineering Psychology and Cognitive Ergonomics. EPCE 2018. Lecture Notes in Computer Science, vol 10906, Springer, Cham. https://doi.org/10.1007/978-3-319-91122-9_45.
- [55] E. R. Zorzal, A. Fernandes and B. Castro, "Using Augmented Reality to overlapping information in live airport cameras", 19th Symposium on Virtual and Augmented Reality (SVR), Curitiba, 2017, pp. 253-256, doi: 10.1109/SVR.2017.53.
- [56] «Drones and Augmented Reality -Powerful Tools when Disaster Strikes,» [Online]. Available: <https://www.suasnews.com/2016/10/drones-augmented-reality-powerful-tools-disaster-strikes/>.
- [57] R. Llasag, D. Marcillo, C. Grilo and C. Silva, "Human Detection for Search and Rescue Applications with UAVs and Mixed Reality Interfaces," 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1-6.
- [58] Huuskonen, Janna and Timo Oksanen. "Soil sampling with drones and augmented reality in precision agriculture." *Comput. Electron. Agric.* 154 (2018): 25-35.
- [59] «Collaborative Augmented Reality with Hands-Free Gesture Control for Remote Astronaut Training and Mentoring,» [Online]. Available: https://www.nasa.gov/directorates/spacetech/strg/nstrf2016/Collaborative_Augmented_Reality/.
- [60] R.G.S. Asthana, N.K. Sinha, "Computer Graphics For Scientists And Engineers", New Age International, 2007.

- [61] Qin, Zong, Yuning Zhang, Fang-Cheng Lin, Yi-Pai Huang and Han-Ping David Shieh. "A Review of Color Breakup Assessment for Field Sequential Color Display." *Information Display* archive 35 (2019): 13-43..
- [62] Hsu, C., Zhen, B., Qiu, W. et al. "Transparent displays enabled by resonant nanoparticle scattering". *Nat Commun* 5, 3152 (2014). <https://doi.org/10.1038/ncomms4152>.
- [63] Miles Hansard, Seungkyu Lee, Ouk Choi, Radu Horaud. *Time of Flight Cameras: Principles, Methods, and Applications*. Springer, pp.95, 2012, SpringerBriefs in Computer Science.
- [64] Helten, Thomas, Andreas Baak, Gaurav Bharaj, M. Müller, H. Seidel and C. Theobalt. "Personalization and Evaluation of a Real-Time Depth-Based Full Body Tracker." *2013 International Conference on 3D Vision (2013)*: 279-286.
- [65] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.
- [66] C. Cadena et al., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," in *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309-1332, Dec. 2016, doi: 10.1109/TRO.2016.2624754.
- [67] Schleicher D., Bergasa L.M., Ocaña M., Barea R., López E. (2009) Real-Time Hierarchical GPS Aided Visual SLAM on Urban Environments. In: Moreno-Díaz R., Pichler F., Quesada-Arencibia A. (eds) *Computer Aided Systems Theory - EUROCAST 2009*.
- [68] D. Schleicher, L. M. Bergasa, M. Ocana, R. Barea and M. E. Lopez, "Real-Time Hierarchical Outdoor SLAM Based on Stereovision and GPS Fusion," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 440-452, Sept. 2009.
- [69] Bhamidipati, Sriramya and Grace Xingxin Gao. "SLAM-based Integrity Monitoring Using GPS and Fish-eye Camera." *ArXiv abs/1910.02165* (2019).
- [70] Hening, Sebastian, C. Ippolito, K. Krishnakumar, V. Stepanyan and M. Teodorescu. "3D LiDAR SLAM Integration with GPS/INS for UAVs in Urban GPS-Degraded Environments." (2017).
- [71] Kiss-Illés, D.; Barrado, C.; Salamí, E. GPS-SLAM: An Augmentation of the ORB-SLAM Algorithm. *Sensors* 2019, 19, 4973.
- [72] Wang, L., C. Zhang and Ruigang Yang. "TofCut: Towards Robust Real-time Foreground Extraction using Time-of-flight Camera." (2016)..
- [73] He, Jun, Laura Balzano and J. Lui. "Online Robust Subspace Tracking from Partial Information." *ArXiv abs/1109.3827* (2011).
- [74] Y. Song, S. Noh, J. Yu, C. Park and B. Lee, "Background subtraction based on Gaussian mixture models using color and depth information," *The 2014 International Conference on Control, Automation and Information Sciences (ICCAIS 2014)*, Gwangju, 2014.
- [75] Maddalena, L.; Petrosino, A. Background Subtraction for Moving Object Detection in RGBD Data: A Survey. *J. Imaging* 2018, 4, 71..

- [76] F. Hafiz, A. A. Shafie, O. Khalifa and M. H. Ali, "Foreground segmentation-based human detection with shadow removal," International Conference on Computer and Communication Engineering (ICCCE'10), Kuala Lumpur, 2010, pp. 1-6.
- [77] J. Shotton et al., "Real-time human pose recognition in parts from single depth images," CVPR 2011, Providence, RI, 2011, pp. 1297-1304, doi: 10.1109/CVPR.2011.5995316.
- [78] Shingade, Ashish & Ghotkar, Archana. "Animation of 3D Human Model Using Markerless Motion Capture Applied To Sports". International Journal of Computer Graphics & Animation, 2014.
- [79] Christopoulou, E., & Xinogalos, S. (2017). Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices. International Journal of Serious Games, 4(4). <https://doi.org/10.17083/ijsg.v4i4.194>.
- [80] Bagassi, S., Lucchi, F., De Crescenzo, F., Piastra, S., "Design for comfort: Aircraft interiors design assessment through a human centered response model approach", 31st Congress of the International Council of the Aeronautical Sciences, ICAS 2018.
- [81] Marks, S., Javier E. Estevez and A. Connor. "Towards the Holodeck: Fully Immersive Virtual Reality Visualisation of Scientific and Engineering Data." IVCNZ '14 (2014).
- [82] Weiyuan Liu, "Natural user interface- next mainstream product user interface," 2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1, Yiwu, 2010, pp. 203-205, doi: 10.1109/CAIDCD.2010.5681374.
- [83] Kamel Boulos, M.N., Blanchard, B.J., Walker, C. et al. Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation. Int J Health Geogr 10, 45 (2011). <https://doi.org/10.1186/1476-072X-10-45>.
- [84] Norman Villaroman, Dale Rowe, and Bret Swan. 2011. Teaching natural user interaction using OpenNI and the Microsoft Kinect sensor. In Proceedings of the 2011 conference on Information technology education (SIGITE '11). Association for Computing Machinery.
- [85] G. Bruder, F. Steinicke and K. H. Hinrichs, "Arch-Explore: A natural user interface for immersive architectural walkthroughs," 2009 IEEE Symposium on 3D User Interfaces, Lafayette, LA, 2009, pp. 75-82, doi: 10.1109/3DUI.2009.4811208..
- [86] Daniel Wigdor and Dennis Wixon. 2011. Brave NUI World: Designing Natural User Interfaces for Touch and Gesture (1st. ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA..
- [87] Ahmed, Naveed and Hind Kharoub, Selma Manel Medjden, and Areej Alsaafin. "A Natural User Interface for 3D Animation Using Kinect," International Journal of Technology and Human Interaction (IJTHI) 16 (2020): 4, accessed (October 14, 2020).
- [88] Mortenson, Michael E. "Geometric modeling". 1997.
- [89] Choi, Ji-wung, Renwick Curry, and Gabriel Elkaim. "Path planning based on bézier curve for autonomous ground vehicles." Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008.
- [90] Yang, Kwangjin, and Salah Sukkarieh. "3D smooth path planning for a UAV in cluttered natural environments." 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE,

2008.

- [91] J. F. Kurose, K. W. Ross. "Computer Networking: A Top-Down Approach". Pearson Education - 2016 - pp. 864..
- [92] Churchill, E.F., Snowdon, D. Collaborative virtual environments: An introductory review of issues and systems (1998) *Virtual Reality*, 3 (1), pp. 3-15. doi: 10.1007/BF01409793.
- [93] Y. Shen, S.K. Ong, A.Y.C. Nee, "Augmented reality for collaborative product design and development", *Design Studies*, Volume 31, Issue 2, 2010, Pages 118-145, ISSN 0142-694X, <https://doi.org/10.1016/j.destud.2009.11.001>.
- [94] Cisco Annual Internet Report (2018–2023) - White paper - March 2020.
- [95] Augmented Reality and Virtual Reality Market Research Report - Allied Market Research - November 2019 - 230 pages.
- [96] K. Marty, P. Rajasekaran, Y. Sun and K. Fuchs, "A Low-cost Approach Towards Streaming 3D Videos of Large-scale Sport Events to Mixed Reality Headsets in Real-time," 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshop.
- [97] Cheung, G., Ortega, A., Cheung, N., : Interactive Streaming of Stored Multiview Video Using Redundant Frame Structures. *IEEE transactions on image processing*, Vol. 20, No. 3, March 2011.
- [98] Cheung, G., Zhi Liu, Zhiyou Ma., Jack Z. G. Tan, "Multi-Stream Switching For Interactive Virtual Reality Video Streaming", *IEEE International Conference on Image Processing (ICIP)*. September 2017.
- [99] CouhHD Video Streaming Protocols – White Paper 2013.
- [100] V. Kraft et al., "A comparison of streaming methods for the Microsoft HoloLens", *Deutsche Gesellschaft für Computer- und Roboterassistierte Chirurgie CURAC*, At Reutlingen, Germany, Volume: 18. September 2019.