



Università Degli Studi Di Salerno
Dipartimento di Informatica

Tesi di laurea magistrale in
SISTEMI VIRTUALI TRIDIMENSIONALI INTERATTIVI

Tecniche di tracking markerless per
sistemi di manutenzione ed assemblaggio
in Realtà Aumentata

Relatori

Ch.mo Prof. Andrea Francesco Abate
Dott. Fabio Narducci
Dott. Stefano Ricciardi

Candidato

Gino Farisano
Matr. 0522500248

Anno Accademico 2015/2016

© 2006 - GINO FARISANO

ALL RIGHTS RESERVED.

Tecniche di tracking markerless per sistemi di manutenzione ed assemblaggio in Realtà Aumentata

ABSTRACT

In questo lavoro di tesi è stato implementato un sistema di realtà aumentata utilizzabile per guidare l'utente nell'assemblaggio delle componenti di una scheda madre per computer. La soluzione proposta, utilizzabile in modalità hand held display oppure attraverso un HMD quale il Google Cardboard, sfrutta la videocamera dello smartphone come periferica di acquisizione; le potenzialità di calcolo del dispositivo sono, invece, utilizzate per effettuare il tracking della scena.

Il riconoscimento delle diverse componenti vviene con tecniche di tracciamento che non utilizzano marker (sistemi marker-less) capaci di sfruttare le caratteristiche naturali degli oggetti come features. Dopo aver testato diverse alternative software commerciali e non, si è deciso di implementare i prototipi AR realizzati con il framework Vuforia; in seguito ad un'attenta valutazione, si è scelto di utilizzare, per lo stesso target, innumerevoli "scansioni" dello stesso, da più angolazioni e punti di vista. Gli esperimenti condotti mostrano come la soluzione implementata sia più efficiente rispetto a quella che utilizza una singola "scansione": i tempi di tracciamento del bersaglio sono più che raddoppiati e le aumentazioni mostrate risultano, anche in condizioni sfavorevoli, sempre perfettamente allineate rispetto alla scena osservata.

I sistemi AR realizzati sono stati confrontati con una soluzione di manutenzione tradizionale, un manuale cartaceo. Dai test condotti su un campione di 18 soggetti

è emerso come i risultati misurati per il sistema AR hand held display siano migliori rispetto a quelli calcolati per le soluzioni HMD AR e il manuale cartaceo, sia in termini di errori commessi durante la procedura di manutenzione, sia per quanto riguarda i tempi di completamento dei task proposti. Dai risultati del NASA-TLX risulta, inoltre, come il carico di lavoro soggettivo percepito dagli utenti che indossano un HMD sia maggiore.

Indice

1	INTRODUZIONE	1
1.1	Main goal	2
1.2	Problem statement	2
1.3	Thesis Outline	2
2	LA REALTÀ AUMENTATA	4
2.1	Applicazioni	6
2.1.1	Medico	6
2.1.2	Annotazione e visualizzazione	7
2.1.3	Robot Path Planning	8
2.1.4	Intrattenimento	9
2.2	L'AR nel settore della manutenzione: riflessioni e implicazioni . .	10
2.3	Stato dell'arte	14
2.4	Augmented Reality Systems	18
2.4.1	Sistemi marker-less	23
2.4.1.1	Sistemi marker-less Sfm	24
2.4.1.2	Sistemi marker-less model based	27
2.5	Dispositivi di visualizzazione	29
2.6	Metriche per visori see-through video	31
3	SOLUZIONI SOFTWARE PER LA MODELLAZIONE DI UN SISTEMA AR	34
3.1	Blender	36

3.1.1	L'interfaccia di Blender	37
3.2	Unity3D	41
3.2.1	Interfaccia e altre features di Unity	41
3.2.2	Scripting	44
3.3	Sistemi commerciali AR basati su rilevamento ottico	45
3.3.1	Soluzioni AR marker based e marker-less a confronto . .	50
4	IMPLEMENTAZIONE	57
4.1	Object target di Vuforia	57
4.2	Scelte progettuali effettuate	59
4.3	Oggetti target utilizzati per la modellazione della scena	61
4.4	Unity3D e Vuforia: modellazione di una scena AR	61
4.5	Gestione delle occlusioni	65
4.6	User interface	66
4.7	Comparazione algoritmi single scan VS multi scan	69
5	SPERIMENTAZIONE: MANUALE TRADIZIONALE VS SISTEMI AR	71
5.1	Variables selection e hypothesis formulation	72
5.1.1	NASA-TLX	74
5.2	Creazione dei gruppi di test	74
5.3	Descrizione degli esperimenti	77
5.4	Risultati degli esperimenti	78
5.4.1	Ipotesi nulla H_0 , H_1 e H_2	80
5.4.1.1	Ipotesi nulla H_2 : multiple comparison	85
5.4.2	Valutazione dei questionari post experiment	86
6	CONCLUSIONI	88
APPENDICE A	MANUALE CARTACEO	90
APPENDICE B	QUESTIONARI E RISULTATI	112
REFERENCES		129

Elenco delle figure

2.0.1 Rappresentazione del "virtually continuum", immagine tratta da [84]	5
2.0.2 Proventi AR vs Proventi VR, immagine tratta da [13]	6
2.1.1 Differenti esempi di utilizzo delle realtà aumentata nel campo della chirurgia, immagine tratta da [90]	7
2.1.2 L'utente apre una porta inserendo un codice di sblocco su una "interfaccia aumentata", immagine tratta da [107]	8
2.1.3 Robot path planning: le linee virtuali mostrano il movimento previsto di un braccio robotico, immagine tratta da [35].	8
2.1.4 Smart Terrain di Vuforia	9
2.2.1 Assemblaggio di costruzioni presentata nel paper [105]	13
2.3.1 Applicazione AR per la stima di un crash test [59]	14
2.3.2 Assemblaggio della serratura di un'automobile, tratta da [96]	16
2.3.3 Assemblaggio di mobili attraverso la realtà aumentata, tratta da [115]	17
2.3.4 Feedback negativo da parte del sistema ITS presentato nel paper [108]	18
2.4.1 Sensore inerziale, InertialCube4 della Intersense	21
2.4.2 Il sistema di tracking meccanico CyberGlove utilizza 22 sensori per monitorare la posizione delle mani dell'utente che interagisce con gli oggetti virtuali	22

2.4.3	PTAM separa la fase di tracking e mapping in due thread, immagine tratta da [92]	25
2.4.4	Algoritmo FAST, immagine tratta da [16]	26
2.4.5	Tassonomia degli algoritmi model based, tratta da [77]	28
2.4.6	Workflow della metodologia Point Sampling, tratta da [77]	29
2.5.1	Principio di funzionamento e soluzione basata su HMD Optical See-Through, tratta da [35]	30
2.5.2	Principio di funzionamento e soluzione basata su HMD Video See-Through, tratta da [35]	31
2.6.1	VR box	33
3.0.1	Processo di modellazione di una mano a partire da un cubo	35
3.1.1	L'interfaccia di Blender	38
3.1.2	Render di alcuni modelli 3D realizzati con Blender	40
3.2.1	Interfaccia e pannelli Unity	43
3.3.1	Image target di Vuforia	48
3.3.2	ARMedia: selezione di features da un modello 3D e da un'immagine 2D	49
3.3.3	Distanza massima di riconoscimento del bersaglio, tratta da [102]	51
3.3.4	Massima inclinazione, tratta da [102]	52
3.3.5	Boxplot metrica 1,2,3,4	54
3.3.6	Barplot metrica 5,6,7,8	55
3.3.7	Classifica globale SDKs AR	56
4.2.1	Scansione dell'oggetto target da più angolazioni	59
4.2.2	Flow chart dell'algoritmo multimarker-less implementato	60
4.4.1	Struttura del progetto Unity3D realizzato	62
4.4.2	Parte dell'inspector dello script animante gli oggetti 3D virtuali	64
4.5.1	Corretta gestione delle occlusioni	65
4.6.1	Vista utente con head hand display e HMD see-through video	66
4.6.2	Vuforia virtual button	68

4.7.1 Algoritmo con più scansioni VS algoritmo con singola scansione	70
5.0.1 Assemblaggio delle componenti di una scheda madre	72
5.4.1 Alpha level	79
5.4.2 Tempi medi task	81
5.4.3 Errori medi task	82
5.4.4 NASA-TLX Workload	84
5.4.5 95% family-wise confidence level NASA-TLX	86
5.4.6 Livello di gradimento del sistema di manutenzione utilizzato e dell'AR	87

Acknowledgments

UN TRAGUARDO IMPORTANTE NON SI RAGGIUNGE MAI DA SOLI ma con l'aiuto di persone che ti supportano e incoraggiano durante il cammino.

Un doveroso ringraziamento va al professore Andrea Abate per le opportunità che mi ha offerto in quest'ultimo anno e mezzo, per avermi seguito durante l'intero progetto WISCH e successivamente in questa attività di tesi. Ringrazio infinitamente il dottor Fabio Narducci: sei stato per me non solo un ottimo relatore ma soprattutto un amico. Ringrazio inoltre il dottor Stefano Ricciardi per i consigli e i suggerimenti preziosi dispensati per la redazione di questa tesi.

Un ringraziamento di cuore va alla mia famiglia, la cosa per me più preziosa, che mi è stata sempre vicina nei momenti di difficoltà e di gioia: grazie mamma e papà per l'amore che mi dimostrate ogni giorno; spero di poter ricompensare al più presto i vostri sacrifici. Grazie ai miei zii, ai miei nonni, a mio fratello e sua moglie Paola, al piccolo Francesco, ai miei cugini, a Giovanni e Ada che mi hanno accolto nella loro casa come un figlio.

Grazie agli amici dell'università, in primis Daniele, Massimo e Nicolò con i quali ho condiviso questi ultimi tre anni di studio. Senza di voi sarebbe stato tutto più triste e noioso. Vi voglio bene. Ringrazio Luca, Michele, Angela e Paola del NeuRoNe Lab per essere stati sempre pronti ad aiutarmi. Grazie a chi ha condiviso con me un tratto di strada di questo lungo percorso: Savino Novelli, Luigi Cetraro, Giovanni Grano, Giuseppe Luciano, Silvia Bianco, Arianna Domenicone,

Giuseppe Valitutto e Roberto Fortunato. Ognuno di voi mi ha donato una parte di se e mi ha reso quello che sono oggi. Grazie infinite.

Infine ringrazio la persona che più di tutte è stata presente, che mi ha ascoltato, mi ha spronato, mi ha aiutato, ha gioito, ha pianto e sofferto per me. Senza di te sarebbe stato tutto più difficile e avrebbe avuto meno senso. Grazie Florinda. This is for you.

A FLORINDA

1

Introduzione

La lettura delle istruzioni è uno step fondamentale quando si deve operare su nuovi dispositivi. Mentre gli utenti leggono sempre meno i manuali, i progettisti stanno cercando di sostituirli con interfacce user friendly in modo che si possa fare a meno di essi. Negli ultimi decenni, inoltre, sono stati introdotti metodi più efficaci rispetto a quelli tradizionali, tra cui la digitalizzazione delle istruzioni corredate da animazioni e i video tutorial. Le informazioni presentate in questi sistemi, sebbene offrano all'utente un livello di comprensione maggiore, non coesistono con gli oggetti reali. L'utente è chiamato quindi a svolgere la funzione di medium tra il manuale e l'oggetto; la traduzione della "rappresentazione del mondo raffigurata nei manuali" è a lui delegata.

Alla luce delle osservazioni sopra esposte, utilizzando la realtà aumentata, in questo elaborato si descriverà uno strumento in grado di fondere nello stesso luogo le istruzioni e gli oggetti su cui operare.

1.1 MAIN GOAL

In questo lavoro di tesi è stato implementato un sistema di realtà aumentata utilizzabile per guidare l'utente nell'assemblaggio delle componenti di una scheda madre per computer. Lo studio qui presentato si propone, inoltre, di analizzare le differenze tra le soluzioni AR realizzate rispetto ad una metodologia tradizionale, il manuale cartaceo.

1.2 PROBLEM STATEMENT

Le domande a cui si cercherà di dare risposta in questo lavoro di tesi sono sintetizzate di seguito.

È possibile realizzare un sistema di manutenzione basato sulla realtà aumentata in grado di:

- Utilizzare la tecnologia marker-less
- Eseguire su un dispositivo mobile in modalità hand held display o con un HMD.
- Aiutare l'utente a completare più velocemente e con meno errori una procedura di manutenzione su una scheda madre.
- Ridurre il workload complessivo a cui l'utente è sottoposto se utilizza un sistema di manutenzione tradizionale.

1.3 THESIS OUTLINE

Nel secondo capitolo sarà presentata al lettore la realtà aumentata e i settori in cui quest'ultima può essere impiegata; per ogni possibile campo di applicazione saranno quindi citati alcuni degli ultimi lavori presentati in letteratura. La trattazione proseguirà con un recente stato dell'arte sull'utilizzo della realtà aumentata nel campo della manutenzione; particolare accento verrà posto sulle problematiche

esistenti in tale settore e sul perché l'AR potrebbe apportare dei benefici positivi in un contesto di manutenzione industriale. Il capitolo proseguirà con la descrizione degli algoritmi di tracking esistenti; maggiore accento sarà posto sui sistemi marker-less. Infine saranno presentati al lettore i dispositivi di visualizzazione utilizzati per fruire della realtà aumentata.

Nel terzo capitolo saranno introdotti gli strumenti software per realizzare un sistema di realtà aumentata, dai software di modellazione 3D ai motori grafici, fino ad arrivare alla descrizione degli SDKs utilizzati per realizzare efficientemente applicazioni AR. La trattazione proseguirà con un'analisi comparativa delle principali librerie AR commerciali, sia quelle basate su sistemi marker-based che marker-less.

Nel quarto capitolo sarà introdotta al lettore l'implementazione del prototipo AR realizzato, a partire dalla scelte progettuali sino ad arrivare alle descrizioni del codice redatto. Infine, saranno descritti i punti di forza della soluzione AR implementata.

Nell'ultimo capitolo si discuterà dei risultati sperimentali ottenuti confrontando le soluzioni hand held display AR e HMD AR presentate nel capitolo precedente con un sistema di manutenzione tradizionale, un manuale cartaceo. Infine, alla luce dei risultati ottenuti, sanno tratte le opportune conclusioni.

2

La realtà aumentata

LA REALTÀ AUMENTATA, termine coniato da Caudell e Mizzell nei primi anni novanta [43], consente di arricchire il mondo reale con elementi virtuali. Come sottolineato da Azuma nel primo survey sull'AR [35], gli elementi virtuali generati al computer devono coesistere con quelli reali e presentare, inoltre, le seguenti proprietà:

- essere interattivi
- essere aggiunti alla scena in real time
- essere allineati con gli altri oggetti nella scena.

È una variante della realtà virtuale, in cui l'utente è totalmente immerso in una realtà immaginaria, che non presenta alcun elemento reale; la realtà aumentata, invece, non isola l'utente dall'ambiente reale, anzi lo completa, in un mondo che è fatto

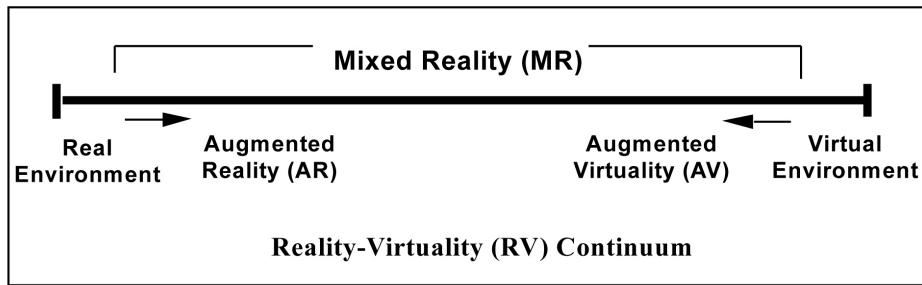


Figura 2.0.1: Rappresentazione del "virtually continuum", immagine tratta da [84].

contemporaneamente di oggetti reali e virtuali. Milgram e Kishino, tuttavia, non collocano realtà e virtualità in due universi separati, ma in un unico universo in cui è possibile passare gradualmente dall'uno all'altro senza soluzione di continuità, muovendosi nello spazio della mixed reality (Fig. 2.0.1.).

In seguito ai primi studi di Azuma, l'AR ha suscitato molto interesse nelle università e nei laboratori di ricerca e sono state tenute diverse conferenze sull'argomento, tra le quali: l'International Workshop and Symposium on Augmented reality e l'International Symposium on Mixed Reality [34].

Successivamente sono stati finanziati molti progetti di ricerca e, con il rilascio delle prime librerie software AR (basti pensare ad ARToolkit [71]), sono state realizzate le prime applicazioni di realtà aumentata.

Infine, nel corso degli ultimi 2-3 anni con l'avvento di smartphone di nuova generazione, la commercializzazione di Head Mounted Display (HMD) a poche decine (Google Cardboard) o centinaia (Oculus Gear VR) di euro, il rilascio di decine di libreria AR (Vuforia, AR-media, CraftAR, Metaio, IN2AR, ecc.) ha fatto il suo ingresso sul mercato un nuovo tipo di realtà aumentata. L'AR è esplosa ispirando non solo progetti di ricerca nelle università ma anche in aziende che hanno creduto nelle potenzialità di questo straordinario strumento.

Le previsioni per il futuro sono molto incoraggianti; la realtà aumentata, sebbene sia nata solo dopo quella virtuale, riuscirà ad affermarsi con più facilità perché

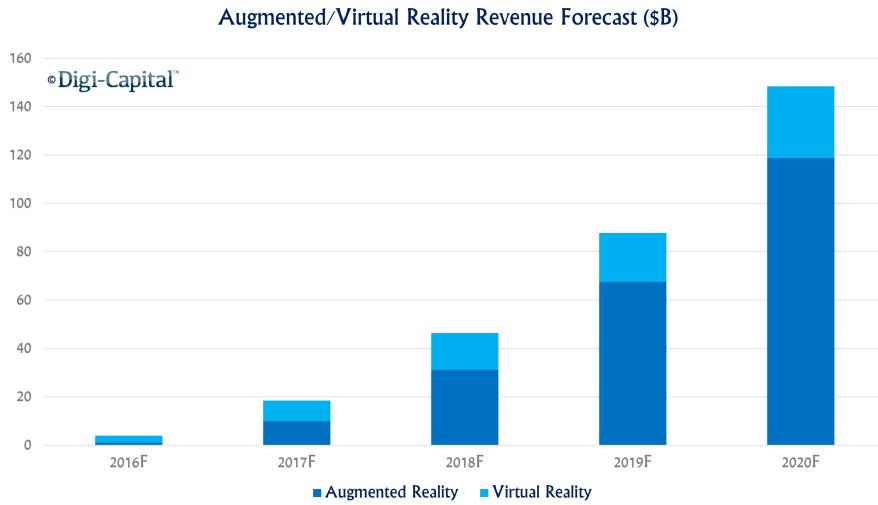


Figura 2.0.2: Proventi AR vs Proventi VR, immagine tratta da [13].

ha suscitato maggiore interesse tra il pubblico. A tal proposito, nell'immagine 2.0.2 sono mostrati i risultati delle previsioni della società Digi-Capital [13] sui proventi che l'AR e la VR apporteranno sino al 2020: il mercato AR sembra essere molto simile a quello di smartphone e tablet; i forecast di Digi-Capital per la realtà aumentata prevedono il raggiungimento di 90 miliardi di dollari complessivi entro il 2020, rispetto ai soli 30 miliardi ipotizzati per la realtà virtuale nel medesimo periodo.

2.1 APPLICAZIONI

2.1.1 MEDICO

Nel suo survey sulla realtà aumentata Azuma scrive "i medici possono utilizzare la realtà aumentata durante un intervento chirurgico. Dopo aver effettuato un'esonanza magnetica potrebbero utilizzare i dati raccolti ed avere, in real time, una visione a raggi X del paziente" [35]. A tal proposito in [90], al fine ridurre gli in-

terventi di chirurgia invasiva, è presentato un sistema in grado di "aumentare" le parti del corpo esterne ai pazienti con immagini endoscopiche (Fig. 2.1.1).

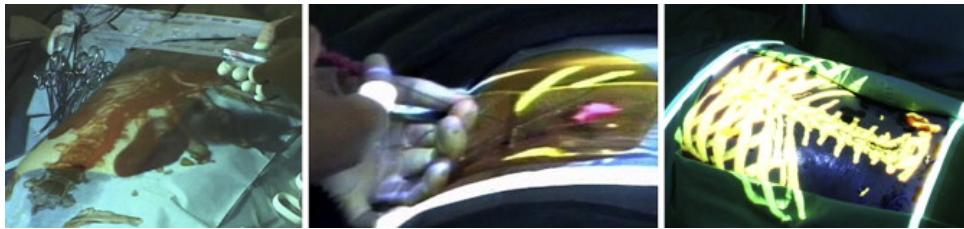


Figura 2.1.1: Differenti esempi di utilizzo delle realtà aumentata nel campo della chirurgia, immagine tratta da [90]

In [76] è stato realizzato un sistema di visualizzazione AR in cui le immagini autostereoscopiche sono aggiunte in real time sulla zona di interesse. Gli esperimenti hanno dimostrato che gli errori di registrazione sono millimetrici e il tempo per guidare un ago verso l'obiettivo sono ridotti del 75%. Ancora, in [68] la realtà aumentata e il leap motion sono utilizzati come strumento di training per gli interventi di laparoscopia.

2.1.2 ANNOTAZIONE E VISUALIZZAZIONE

La realtà aumentata è utilizzata in questo settore per annotare su oggetti e ambienti informazioni pubbliche o private [35]. Ad esempio, al Massachusetts Institute of Technology (MIT) è stato realizzato un prototipo che consente all'utente di aprire una porta inserendo un codice di sblocco su una "interfaccia aumentata" [66] (Fig. 2.1.2). Ancora, nel 1995, al centro europeo Computer-Industry Research Centre (ECRC), è stato realizzato un applicativo attraverso il quale l'utente, indicando la parte di un motore, visualizza il nome del componente desiderato [107].



Figura 2.1.2: L'utente apre una porta inserendo un codice di sblocco su una "interfaccia aumentata", immagine tratta da [107].

2.1.3 ROBOT PATH PLANNING

La realtà aumentata può essere utilizzata in questo settore per facilitare la programmazione di robot e, in particolare, nella pianificazione di percorsi *collision-free* [54] (Fig. 2.1.4). Applicazioni di esempio sono robot che pitturano [94] (i movimenti dell'utente sono convertiti in un programma per robot), training del personale che deve svolgere attività con strumentazione avanzata [38], efficiente disposizione di strumentazione in una stanza [59].

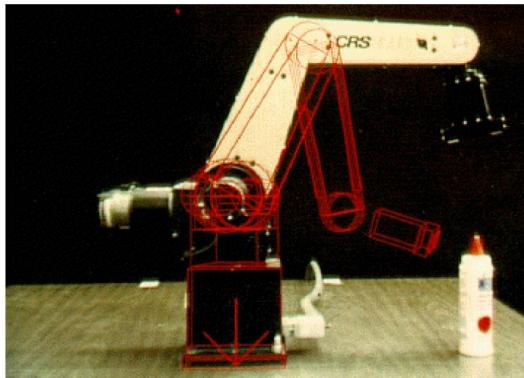


Figura 2.1.3: Robot path planning: le linee virtuali mostrano il movimento previsto di un braccio robotico, immagine tratta da [35].

2.1.4 INTRATTENIMENTO

L'AR nel settore dell'intrattenimento, soprattutto negli ultimi anni, ha permesso ai non addetti ai lavori di toccare con mano le potenzialità della realtà aumentata. Ad esempio, sistemi come Project Tango di Google o Smart Terrain di Vuforia (Fig. 2.1.4) consentono di analizzare l'ambiente circostante al fine di creare un ambiente virtuale che occlude gli oggetti presenti in una stanza. Queste nuove tecnologie aprono sicuramente infinite possibilità per applicazioni commerciali e di intrattenimento. Ancora, i ricercatori del Georgia Institute of Technology hanno realizzato l'applicazione AR Second life [1] che consente a persone reali di coesistere con personaggi virtuali in una sorta di vita parallela. Altre applicazioni, come ad esempio MSQRD [5] consentono di aggiungere in real time a foto o video oggetti virtuali (ad esempio baffi, cappelli, ecc.) o addirittura, scambiare i volti dei soggetti in un selfie.



Figura 2.1.4: Smart Terrain di Vuforia

2.2 L'AR NEL SETTORE DELLA MANUTENZIONE: RIFLESSIONI E IMPPLICAZIONI

Il mercato industriale consta generalmente di cinque fasi: progettazione, assemblaggio, inspezione, testing e manutenzione. La realtà aumentata può apportare il proprio contributo in ognuna di esse, in questa sezione tuttavia si affronteranno e approfondiranno le fasi dell'assemblaggio manuale e della manutenzione. È infatti importante sottolineare che, sebbene molte delle operazioni di assemblaggio e manutenzione oggi siano completamente automatizzate, alcune attività richiedono ancora manodopera umana.

Fino a qualche anno fa, la maggior parte delle operazioni in questi settori era condotta da personale specializzato che applicava procedure documentate in manuali cartacei. Negli ultimi anni il mercato, però, ha subito forti mutamenti e in linea con le previsioni di Browne, Sackett e Wortmann [112], ha introdotto nuove necessità, riassunte di seguito:

- Elevata varietà di prodotti
- Costi di produzione bassi
- Elevata velocità di produzione
- Alta customizzazione da parte del cliente

Per far fronte a questi mutamenti sono state introdotte nuove tecniche di produzione tra le quali la produzione agile [105]; essa è intesa come risorse e persone che possono essere scambiate velocemente, efficientemente e in modo imprevisto, così da far fronte alla variabilità e all'incertezza del mercato[86]. I sistemi di produzione dovrebbero essere quindi in grado di fornire capacità produttiva *on demand* e contemporaneamente, garantire un adeguato livello di specializzazione dei manutentori. Però, il training di operai esperti richiede mesi, se non anni, poiché le procedure di manutenzione esigono elevate capacità di problem solving. Inoltre, come detto in precedenza, i processi industriali evolvono rapidamente; non

è quindi possibile (ed economico) riformare il personale quando le procedure di manutenzione cambiano.

L'utilizzo della realtà aumentata sarebbe sicuramente una soluzione ideale a queste necessità e consentirebbe ai manutentori di ricevere informazioni in modo dinamico, così da realizzare, al minor costo, la produzione agile. È inoltre comunemente teorizzato come l'assistenza AR in compiti di assemblaggio e manutenzione aumenti la produttività, diminuendo la percentuale di errori e il carico cognitivo dovuto alla traduzione di istruzioni astratte dei manuali in attività concrete [105].

L'interpretazione e l'esecuzione delle istruzioni, che in alcuni casi rappresenta il 50% dell'intera procedura di manutenzione, possono infatti richiedere un notevole sforzo mentale che può diventare sempre più forte all'aumentare della difficoltà dei task da eseguire [88]. Inoltre, gli sforzi cognitivi richiesti sono iterativi poiché è necessario un continuo confronto tra l'informazione presente nei manuali e gli elementi reali; in più aumentano al crescere della complessità degli oggetti. Secondo Richardson [97] la complessità di quest'ultimi è influenzata da variabili quali ad esempio il numero di pattern ripetuti o il numero di fori, che possono accrescere la confusione e il tempo necessario per completare le operazioni di manutenzione richieste.

Il riconoscimento di un oggetto è inoltre influenzato dal suo orientamento; secondo Shepard e Metzler [101] maggiore è la differenza di orientamento tra un oggetto e la sua rappresentazione su un manuale, più lungo è il tempo di riconoscimento. Gli operatori potrebbero quindi imbattersi in assemblaggi complessi, come quelli che richiedono interventi su numerosi componenti (magari rotati rispetto alla loro rappresentazione sui manuali), con forme complesse, che facilmente creano ambiguità.

Le istruzioni per assemblare, manutenere o riparare un determinato componente, invece, potrebbero risultare più semplici da comprendere se fossero disponibili, non più come manuali di testo con immagini ma, piuttosto, come oggetti 3D sovrapposti alla scena reale. Infatti, questi ultimi potrebbero essere animati, rendendo più semplici le istruzioni da seguire e, contemporaneamente, evitando gli innumerevoli context switch dal manuale alla scena reale sulla quale operare.

A tal proposito in [64] è stato condotto uno studio per analizzare gli angoli di rotazione della testa e la velocità con cui essa ruota in tipiche operazioni di manutenzione all'interno di un carrarmato blindato militare. L'analisi condotta ha dimostrato che l'utilizzo di un'applicazione di realtà aumentata con un HMD riduce sensibilmente i movimenti della testa dei manutentori con conseguente beneficio nell'espletamento dei task. Ancora, uno studio condotto da Haines e i suoi collaboratori [61] indica che i piloti che fanno uso di visualizzatori head-up (HUP) per visualizzare informazioni di viaggio, muovono meno la testa e gli occhi rispetto a piloti che non ne fanno uso. Ne risulta una maggiore attenzione e una riduzione degli errori dovuti a distrazioni.

Inoltre, gli esseri umani tendono a memorizzare le informazioni in modo più efficace quando queste ultime sono "agganciate" ad un quadro di riferimento nel mondo reale. Demostene, oratore greco nato intorno al 384 a.c ad esempio, usando una strategia nota come "metodo di loci" memorizzava lunghi discorsi camminando all'interno della sua casa, associando ad ogni parte del discorso un oggetto della sua abitazione. Ancora, nel campo delle neuroscienze, differenti teorie suggeriscono una forte relazione tra la posizione spaziale e la memoria di lavoro.

A tal proposito, risultati positivi a favore dell'AR sono presentati in [105] dove gli esperimenti hanno dimostrato come la sovrapposizione di istruzioni 3D per l'esplorazione di una procedura di assemblaggio, riduca la percentuale degli errori fino all'82%. In particolare Tang e i suoi collaboratori hanno utilizzato per la sperimentazione delle costruzioni per bambini (Fig. 2.2.1) al fine di azzerare l'errore inputabile al così detto "fattore esperienza".

I risultati, generalizzabili per qualsiasi procedura di assemblaggio, hanno dimostrato che l'AR riduce sensibilmente il carico mentale a cui l'utente è sottoposto; alcune attività cognitive sono completamente delegate al sistema.

Ricollegandoci a quanto detto in precedenza, infatti, quando il manutentore sposta l'attenzione dal manuale cartaceo alla scena dove operare deve memorizzare le operazioni da compiere (la posizione e l'orientamento degli oggetti) nella memoria a breve termine. Un sistema AR elimina queste richieste per sovrapposizioni.



Figura 2.2.1: Assemblaggio di costruzioni per bambini presentata nel paper [105]

zione spaziale, diminuendo così il carico cognitivo richiesto[105]. Ne conseguono benefici sia fisici che mentali. Nakanishi e Sato [87], ad esempio, hanno confrontato una procedura di manutenzione tradizionale e un sistema basato su un Retinal Imaging Display (RID). L'analisi della concentrazione di oxy-emoglobina negli emisferi celebrali degli utenti che utilizzano il RID evidenzia uno sforzo minore rispetto a quello calcolato per i sistemi di manutenzione tradizionale. Inoltre, l'analisi del battito cardiaco ha rivelato che l'uso del RID può ridurre lo stress. [105],[95],[83], invece, hanno misurato il carico mentale dell'utente utilizzando il *Task Load Index* della Nasa [63]. I risultati dei questionari indicano dei benefici in termini di frustrazione, sforzo mentale, fisico e temporale per gli utenti che utilizzano l'AR anzichè procedure tradizionali.

Concludendo, l'utilizzo della realtà aumentata nel campo della manutenzione potrebbe rivoluzionare completamente le modalità di espletamento delle procedure di riparazione tradizionali; in accordo con Neumann, infatti, quest'ultima è complementare ai processi cognitivi umani [88].

2.3 STATO DELL'ARTE

La realtà aumentata nel campo della manutenzione ha suscitato molto interesse tra i ricercatori. Tale interesse si riflette nella formazione di diversi consorzi di ricerca dedicati al topic [64]. Tra i più importanti ricordiamo:

ARVIKA [59]: è un progetto sponsorizzato dal ministro dell'educazione e ricerca tedesco il cui scopo è di implementare sistemi di realtà aumentata mobile nel campo industriale. L'obiettivo principale è di estendere l'AR in "realità tradizionali" (e non solo in applicazioni di nicchia) pertanto, gli studi del progetto si sono concentrati in particolare sulla praticità e l'applicabilità della realtà aumentata in azienda; la maggior parte dei prototipi AR precedentemente sviluppati erano, infatti, troppo ingombranti per essere integrati con successo in ambienti di lavoro industriali. I ricercatori hanno condotto pertanto test di usabilità per valutare gli aspetti ergonomici, l'hardware e il software richiesti dagli applicativi AR. Un esempio di prototipo realizzato con il progetto ARVIKA è un'applicazione per la nota casa automobilistica tedesca Volkswagen che consente di confrontare i risultati stimati (calcolati a PC) di un crash test con le deformazioni avvenute in seguito a un incidente reale "simulato" (Fig. 2.3.1.). Altri prototipi includono: un'app per la realizzazione di cablaggi complessi, un'app di assistenza su una macchina di produzione, un'app marker based per pianificare l'organizzazione di macchinari in una stanza, ecc.

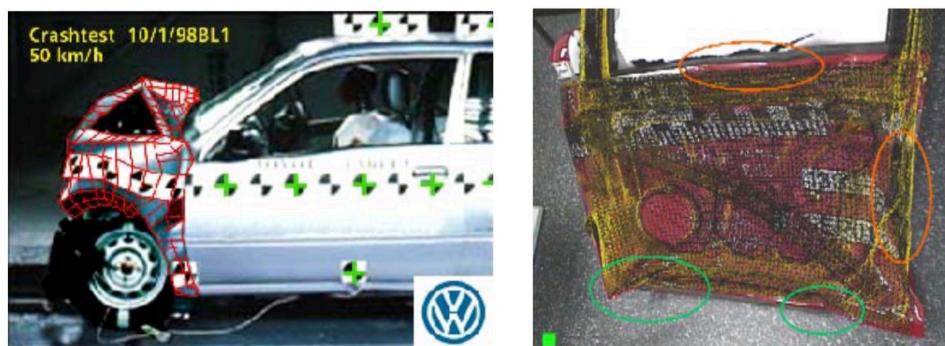


Figura 2.3.1: Applicazione AR per la stima di un crash test [59]

ARTESAS [74]: lo scopo del consorzio è l'esplorazione e il testing di tecnologie legate alla realtà aumentata al fine di diffondere i risultati ottenuti al mercato industriale. Il progetto, basato sui risultati di ARVIKA, ha approfondito le seguenti tematiche: (1) algoritmi di tracking in ambienti industriali (in primis sistemi marker-less), (2) dispositivi AR user-friendly, (3) sistemi di *authoring* per la creazione di istruzioni basate sulla realtà aumentata.

Oltre all'interesse dei consorzi di ricerca sopra citati, l'AR nel settore della manutenzione è stato oggetto di studio in ambienti universitari e in laboratori di ricerca e sviluppo nelle aziende.

Nel 1993, il gruppo di Steven Feiner della Colombia University [55] ha costruito un'applicazione per la manutenzione di una stampante laser. All'azienda Boeing hanno sviluppato un sistema basato sull'AR per guidare un tecnico nella realizzazione del cablaggio del sistema elettrico di un aeroplano [35]. Ancora, nel 1998, Reiners e i suoi collaboratori [96] hanno realizzato un sistema AR che assiste i manutentori nell'assamblaggio di una portiera di un automobile (Fig. 2.3.2). Il prototipo realizzato, utilizzante un HMD e dei marker quadrati, è stato mostrato al pubblico alla fiera di Hannover nel 1998 e ha suscitato molto interesse in questa nuova area. Altri studi simili hanno fornito risultati positivi a favore dell'AR in svariati domini applicativi: montaggio di mobili [115] (Fig. 2.3.3), assemblaggio di strumenti medici [91], manutenzione del motore di un aereo [65], ecc.

Ancora, Baird e Barfiel [36] hanno realizzato uno dei primi studi sull'efficacia della realtà aumentata nell'assemblaggio dei componenti di una scheda madre per computer. In particolare, hanno confrontato le prestazioni di 15 utenti utilizzando supporti didattici differenti: un manuale di carta tradizionale, un manuale su schermo LCD e, infine, un sistema AR basato su HMD see-through video e uno basato su un sistema see-through ottico. I risultati degli esperimenti hanno dimostrato l'efficacia dell'utilizzo dei sistemi AR rispetto agli strumenti tradizionali, sia per quanto riguarda i tempi di completamento dei task che il numero di errori commessi; il dispositivo see-through video, inoltre, si è dimostrato più efficace rispetto alla soluzione see-through ottica.

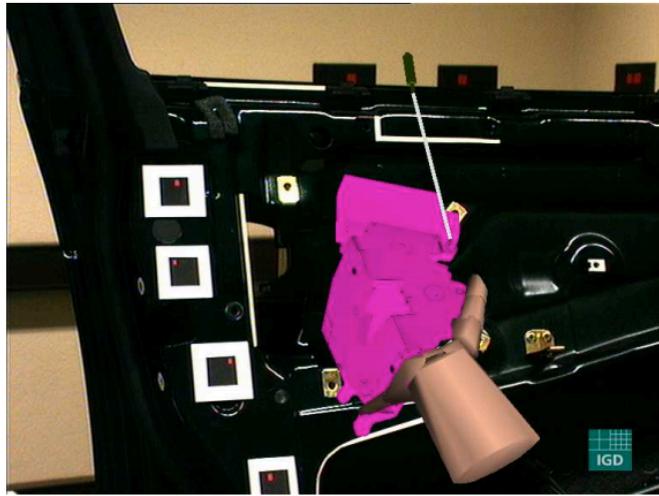


Figura 2.3.2: Assemblaggio della serratura di un'automobile, tratta da [96]

Una sperimentazione simile [36], anche in questo caso riferita a una procedura di assemblaggio su una scheda madre, dimostra come i tempi di completamento siano nettamente inferiori con soluzioni AR. I risultati ottenuti sottolineano come la deviazione standard di quest'ultimi sia più bassa per soluzioni AR testate (HMD see-through video e see-through ottiche) rispetto alle soluzioni non-AR.

Anche in [105], i risultati sperimentali dimostrano che, sebbene una soluzione AR non apporti benefici temporali nell'espletamento dei task, riduce sensibilmente il numero di errori commessi durante la procedura di manutenzione oggetto di studio. Tali statistiche sono in linea con i risultati presentati in [64], i quali confermano che l'AR non apporta alcun beneficio in termini temporali nell'espletamento dei task proposti sebbene diminuisca il numero di errori commessi.

Altri lavori riguardanti l'utilizzo della realtà aumentata per l'assemblaggio di una scheda madre sono presentati in [108] e in [95]. Nel primo, gli autori studiano l'efficacia dell'AR combinata ad un sistema ITS (Intelligent Tutoring System); gli studi mostrano in particolare come l'utilizzo di un sistema intelligente migliori le performance degli utenti fino al 30% se confrontati con sistemi AR tradizionali.¹.

¹Un sistema ITS è in grado di fornire all'utente istruzioni personalizzate e feedback (Fig. 2.3.4).

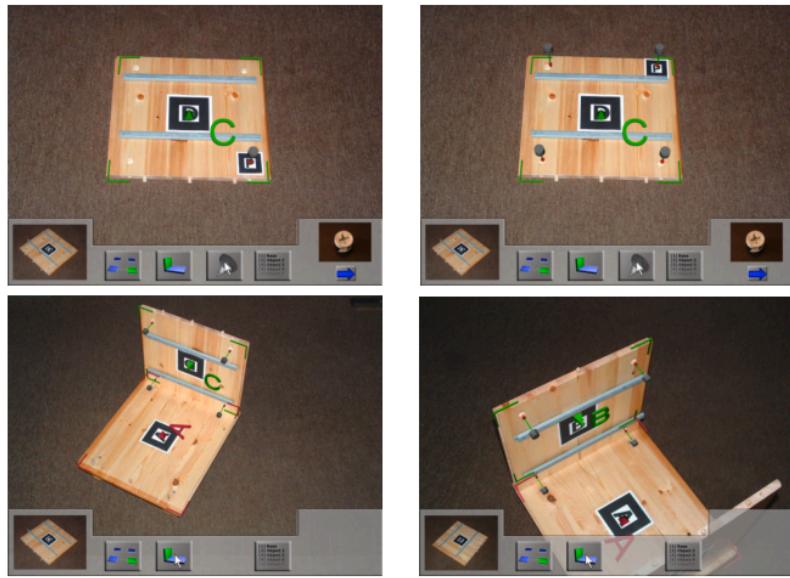


Figura 2.3.3: Assemblaggio di mobili attraverso la realtà aumentata, tratta da [115]

Nel secondo lavoro, invece, è valutata l'efficacia di una sistema AR basata su monitor LCD tradizionale. Dagli studi condotti emerge come la soluzione proposta sia "industrialmente accettabile"; secondo gli autori, infatti, soluzioni AR basate su HMD sono ancora in fase prototipale, costose, ingombranti e non adatte ad essere utilizzate per lunghi periodi di tempo.

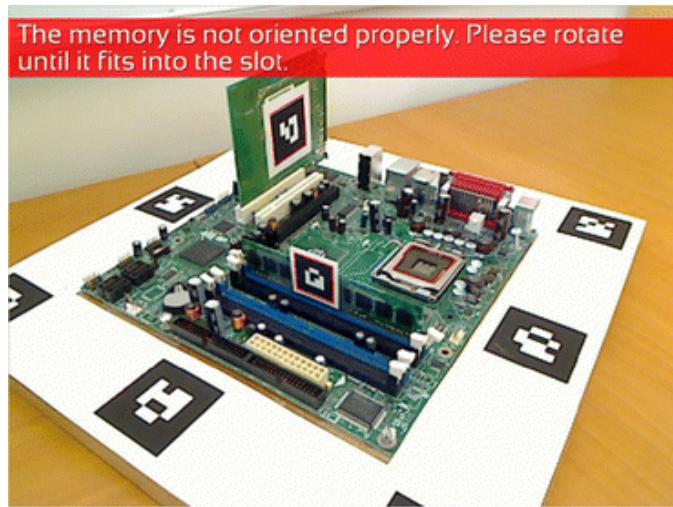


Figura 2.3.4: Feedback negativo da parte del sistema ITS presentato nel paper [108]

2.4 AUGMENTED REALITY SYSTEMS

Idealmente, un sistema AR dovrebbe essere in grado di proporre all'utente un ambiente in cui gli elementi reali e virtuali siano indistinguibili. Gli elementi generati al computer dovrebbero quindi trovarsi sempre nella giusta posizione e nella giusta scala rispetto al punto di vista dell'osservatore. Nella VR, invece, tale prerequisito è meno forte. Pensiamo ad esempio ad un utente che indossa un HMD e visualizza la sua mano (rappresentata virtualmente) immerso in un ambiente totalmente immaginario; se la mano virtuale e quella reale non sono esattamente nella stessa posizione (c'è un errore di 5 mm ad esempio) l'utente difficilmente lo intuirà. L'errore sarà invece più evidente in un contesto AR: se le due mani non sono perfettamente allineate la discrepanza sarà visiva [35].

Altri parametri che rendono un sistema AR “un buon sistema” sono: una corretta gestione delle occlusioni, l’adattamento cromatico degli oggetti virtuali in base alle condizioni della luce e la gestione delle ombre generate da quest’ultimi. Tali problemi, sebbene secondari, non sono banali e la comunità scientifica sta lavorando molto per risolverli, al fine di rendere l’esperienza AR sempre più coinvolgente.

Ricollegandoci a quanto detto in precedenza, il posizionamento degli oggetti virtuali, chiamato registrazione, è risolto effettuando il tracking dell’ambiente; in altre parole, il sistema AR deve ”seguire” i movimenti dell’utente e comprendere dove esso si trova rispetto agli oggetti reali nella scena.

In base alla tecnologia utilizzata per il rilevamento è possibile distinguere differenti tipologie di tracking [114]:

- Inerziale
- Tracking basato su GPS
- Magnetico
- Ultrasonico
- Meccanico
- Ottico

Ognuno di questi sistemi presenta, ovviamente, sia vantaggi che svantaggi ed è impossibile definire quale sia il migliore in assoluto. La scelta del sistema di tracking da adottare deve pertanto seguire i requisiti richiesti dall’applicativo che si intende realizzare; è inoltre importante valutare le caratteristiche dell’ambiente in cui si deve operare. A tal proposito nella tabella 2.4.1, tratta da [42], sono riportate, per ogni tipologia di tracking, le specifiche soddisfatte.

I sensori inerziali, quali ad esempio l’InertialCube4 della Intersense (Fig. 2.3.2) oppure quelli integrati negli smartphone di ultima generazione, sono adatti per applicazioni di simulazione, di training e per il motion capture, ecc. In particolare, essi utilizzano accelerometri e giroscopi per rilevare rispettivamente la velocità lineare e angolare [111]. Sebbene siano molto precisi, però, in poco tempo ”vanno alla deriva”.

La tecnologia GPS (Global Positioning System) è adatta esclusivamente per utilizzzi all’aperto; attraverso una rete dedicata di satelliti artificiali in orbita, ove vi sia

Technology	Range (m)	Setup time (hr)	Precision (mm)	Time (s)	Environment
Optical: marker-based	10	0	10	∞	in/out
Optical: markerless	50	0-1	10	∞	in/out
Optical: outside-in	10	10	10	∞	in/out
Optical: inside-out	50	0-1	10	∞	in/out
GPS	∞	0	5000	∞	out
WiFi	100	10	1000	∞	in/out
Magnetic	1	1	1	∞	in/out
Ultrasound	10	1	10	∞	in
Inertial	1	0	1	10	in/out
Hybrid	30	10	1	∞	in/out
UWB	10-300	10	500	∞	in
RFID: active	20-200	when needed	500	∞	in/out
RFID: passive	0.05	when needed	500	∞	in/out

Tabella 2.4.1: Comparazione delle più comuni tecniche di tracking. Range: grandezza della regione da "tracciare". Setup: tempo di preparazione e calibrazione. Precisione: accuratezza del tracking. Time: tempo massimo prima che i dati del tracking diventino spuri. Environment: luogo in cui il tracker può essere utilizzato: dentro o fuori. Tabella tratta da [42]

un contatto privo di ostacoli con almeno quattro satelliti visibili, fornisce informazioni sulle coordinate geografiche del ricevitore[4]. Per la realizzazione di sistemi AR, tuttavia, il solo GPS, che fornisce informazioni esclusivamente sulla posizione, non può essere utilizzato come tracker; infatti, quest'ultimo necessita di una bussola per fornire l'orientamento dell'utente rispetto alla scena osservata. In [103], ad esempio, con un sistema ibrido utilizzante il GPS e un sistema inerziale, è descritta un'app AR per ricreare in loco la rappresentazione virtuale di un castello del tredicesimo secolo.

Le tecniche di tracking magnetiche, come si evince dalla tabella 2.4.1, sono utilizzabili sia all'interno che all'esterno e sono molto precise anche se hanno un limitato range di azione. Quest'ultime utilizzano un trasmettitore che emette un campo elettromagnetico, nel range degli 8-14 KHz [37], che è intercettato da un ricevitore; come si evince in [3], i sensori possono essere incorporati in molti materiali senza comprometterne la capacità di ricezione. Tuttavia, questi sistemi di tracking sono altamente sensibili alle interferenze da metalli ferrosi e campi magnetici soprattutto in ambienti in cui le potenziali fonti di interferenza non sono stazionarie. Pertanto, in accordo con [65], tale tecnologia non è utilizzabile in contesti di

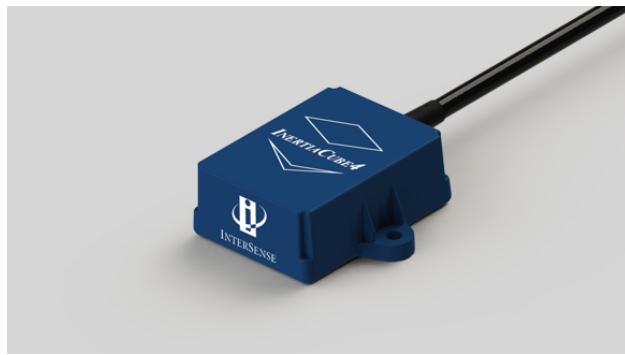


Figura 2.4.1: Sensore inerziale, InertialCube4 della Intersense

manutenzione.

Il tracking ultrasonico calcola la posizione misurando il tempo che un segnale sonoro impiega a raggiungere un microfono. Questi sistemi, i primi ad essere utilizzati nell'AR [65], sono molto precisi e poco costosi; tuttavia, al pari dei dispositivi di tracking magnetici, sono molto suscettibili a interferenze sonore esterne.

Il tracking meccanico utilizza collegamenti meccanici a potenziometri e altri dispositivi elettronici per determinare la posizione e l'orientamento del sistema da tracciare. Quest'ultimi forniscono una stima molto affidabile della posizione nonostante limitino molto la libertà di movimento dell'utilizzatore (Fig. 2.4.2.).

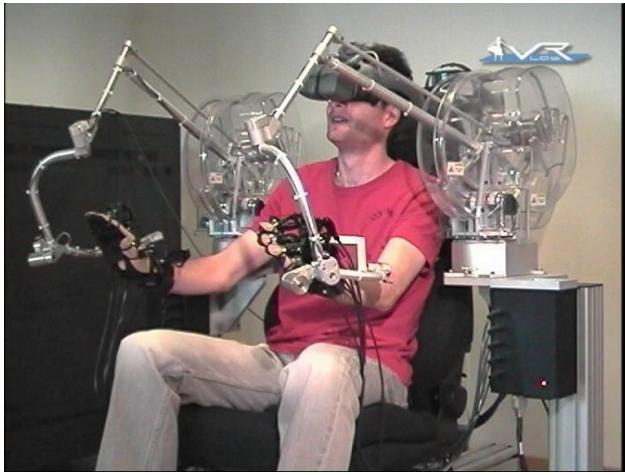


Figura 2.4.2: Il sistema di tracking meccanico CyberGlove utilizza 22 sensori per monitorare la posizione delle mani dell'utente che interagisce con gli oggetti virtuali

L'ultima classe di sistemi di tracking è quella ottica. Questi sono utilizzati sempre più spesso poiché sono molto economici e forniscono una buona accuratezza e robustezza. Esistono due tipologie di sistemi ottici: marker based e marker-less [77]. Per acquisire la scena usano una o due camere (rispettivamente sistema mono e sistema stereo); in questo caso, affinchè gli elementi virtuali siano correttamente allineati alla scena reale, è necessario stimare la posa della camera per ogni fotogramma acquisito [60].

Un sistema marker based è definito come una libreria di marker, tipicamente figure quadrate bianche e nere, e un algoritmo per riconoscere ed effettuare il tracking di quest'ultime. L'utilizzo del bianco e nero in particolare offre il miglior contrasto rispetto all'ambiente circostante e quindi un riconoscimento più immediato. Da quando è stato introdotto il primo sistema basato su marker, sono stati condotti numerosi studi sull'aspetto che questi devono presentare e sulle tecniche per riconoscerli. Argomenti di ricerca vertono su forma, dimensione, cornice dei marker, algoritmi per il riconoscimento, ecc.: il sistema NaturalPoint OptiTrack ad esempio utilizza marker retroriflettenti illuminati da raggi ad infrarossi; un sistema simile, presentato in [93] utilizza marker invisibili disegnati con una penna fluorescente.

scente; ancora, in [29] è descritto un sistema in cui un insieme di marker sono stati posizionati su un treppiede in modo da poter essere visti da qualsiasi angolazione. Tuttavia, benché i sistemi marker-based aumentino la robustezza e riducano la potenza di calcolo richiesta per il tracking, possono essere "invasivi" ed occludere parti dell'oggetto da tracciare; inoltre richiedono che la scena sia "preparata" prima di essere aumentata. Pertanto, la ricerca scientifica nel campo dei sistemi ottici si sta muovendo nella direzione di tecnologie che utilizzano le caratteristiche naturali degli oggetti come features, rendendo possibili i cosiddetti sistemi marker-less. Tale argomento, molto rilevante per la natura di questo elaborato, sarà approfondito separatamente nel paragrafo successivo.

2.4.1 SISTEMI MARKER-LESS

Come detto in precedenza, i sistemi di tracking marker-less utilizzano le caratteristiche naturali degli oggetti come features; per "aumentare la realtà" non c'è quindi bisogno di preparare la scena con marker "intrusivi" o utilizzare strumentazione particolare (come ad esempio sensori inerziali, meccanici, ecc.). Tali sistemi, infatti, utilizzano esclusivamente una o più telecamere per estrarre features rilevanti dall'ambiente da tracciare; queste caratteristiche sono elaborate per stimare la posa degli oggetti presenti nella scena [22].

Il tracking delle caratteristiche naturali degli oggetti è un'operazione molto complessa e richiede un carico computazionale molto pesante. Le componenti da tracciare, infatti, a differenza dei marker (che hanno forma e dimensioni note) possono essere tridimensionali ed avere forme molto articolate.

I sistemi di tracking marker-less possono essere classificati in due gruppi: model based e Structure from motion (Sfm) based. Nei primi, la conoscenza del mondo reale è "conservata" in un modello 3D che è utilizzato per stimare la posa della camera [77]. I secondi, invece, sono in grado di costruire contemporaneamente una mappa 3D della scena mentre effettuano il tracking della posizione e dell'orientamento della camera [52].

2.4.1.1 SISTEMI MARKER-LESS SfM

In letteratura questa classe di algoritmi, meglio conosciuta come Simultaneus Localization and Mapping (SLAM), è senza dubbio uno dei problemi più importanti nel settore della robotica². Infatti, se un robot vuole orientarsi e muoversi liberamente in un ambiente sconosciuto (senza sbattere contro un muro) deve aggiornare una mappa dell'ambiente esplorato man mano che visita posti nuovi e, contemporaneamente, mantenere traccia di se stesso in essa; per fare ciò, ovviamente, il robot deve essere equipaggiato con sistemi sensoriali che possono includere laser, sonar, GPS, tecnologie ad infrarossi e quant'altro [33]. Nel campo della realtà aumentata, però, dove gli obiettivi sono diversi (anche se simili), si utilizzano solamente telecamere per effettuare il tracking della scena; in questo caso si parla pertanto di visual SLAM.

In un primo periodo, tutti gli sforzi della comunità scientifica si sono concentrati sullo sviluppo di sistemi visual SLAM che necessitavano di due camere per acquisire la scena, così da avere una visione tridimensionale della realtà osservata [49] [48] [69]. Successivamente, con la pubblicazione del paper [47], in cui è stato descritto uno dei primi sistemi SLAM monoculari, la comunità scientifica ha prodotto molti risultati significativi nella direzione dei sistemi SLAM che utilizzano una sola camera. Tra i più rilevanti è inevitabile non citare:

MonoSLAM [47][50]: utilizzando una sola camera consente di creare online una mappa sparsa (ma persistente) di features in modo probabilistico. Quest'ultima è inizializzata all'avvio del sistema ed evolve dinamicamente utilizzando i filtri di Kalmar; le stime probabilistiche della camera e delle caratteristiche osservate sono pertanto aggiornate frame dopo frame (alcune features possono anche essere eliminate).

PTAM [72]: Parallel Tracking and Mapping propone una soluzione semplice,

²SLAM e Sfm sono spesso utilizzati come sinonimi, tuttavia, mentre SLAM è un algoritmo online, Sfm lavora offline (può richiedere ore di elaborazione). Un tipico problema di SfM è, ad esempio, il seguente: data una grande collezione di foto di una singola struttura esterna (come il Colosseo), costruire un modello 3D [106].

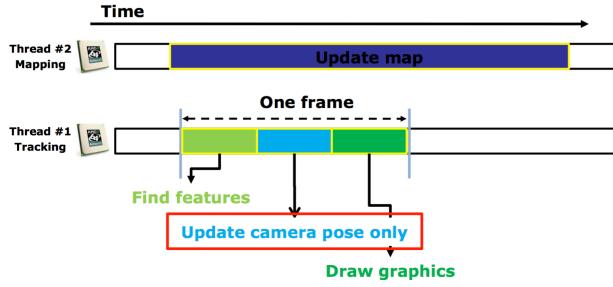


Figura 2.4.3: MonoSLAM, per ogni frame, deve aggiornare sia la posa della camera che la mappa dell'ambiente tracciato. PTAM, invece, effettua l'operazione di mapping (operazione molto costosa) in un altro thread.

efficace e a basso costo del problema SLAM, infatti, dopo due anni dalla pubblicazione del paper "ufficiale", gli stessi autori hanno presentato una versione di PTAM eseguibile su uno smartphone iPhone [73].

Piuttosto che basarsi su modelli 3D dell'ambiente da tracciare, PTAM utilizza l'algoritmo FAST [99] (molto veloce ed "economico" computazionalmente) per estrarre le features dai frame dello stream video (Fig. 2.4.4.). PTAM, inoltre, effettua la fase di localization e di mapping in parallelo (Fig. 2.4.3.). Questo sistema SLAM è stato installato, testato e le seguenti considerazioni sono state tratte: l'acquisizione delle features di interesse avviene tempestivamente (altri sistemi quali Vuforia object target e Wikitude SLAM per la stessa area di lavoro impiegano diversi minuti), le features acquisite sono tantissime, la qualità del tracking è peggiore rispetto a quella di Vuforia object target e migliore rispetto a quella di altre soluzioni SLAM testate (Wikitude e Kudan).

DTAM[89]: a differenza dei sistemi SLAM descritti sinora, DTAM non si basa su algoritmi di *feature extraction* (quali ad esempio FAST) ma utilizza ogni pixel delle immagini acquisite per produrre un modello denso della scena tracciata; in particolare, DTAM utilizza le mappe di profondità (depth map) dei keyframe selezionati dall'algoritmo al fine di produrre un "mosaico" della scena con milioni di vertici. Il modello denso è quindi utilizzato per stimare i movimenti della camera. Il sistema è, inoltre, altamente parallelizzabile in quanto può utilizzare le migliaia

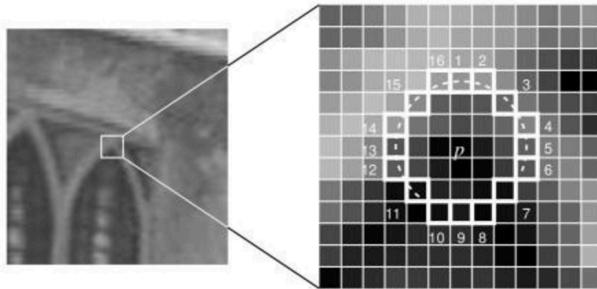


Figura 2.4.4: FAST analizza la luminosità dell’immagine intorno ad un presunto punto chiave. Per capire se tale punto è effettivamente un keypoint, l’algoritmo esamina un cerchio di 16 pixel attorno ad esso. Se è possibile trovare un arco di punti contigui (con luminosità simile), di lunghezza maggiore a $\frac{3}{4}$ del perimetro del cerchio, allora quest’ultimo è un keypoint FAST.

di unità elaborative delle moderne GPU. Ne consegue un tracking robusto ed affidabile se confrontato a sistemi quali PTAM [14]; tuttavia, l’algoritmo non è utilizzabile su sistemi mobile. Purtroppo non è stato possibile testare questo sistema perché gli autori non ne hanno rilasciato il codice sorgente. Un’implementazione open di DTAM [58], invece, è stata realizzata per un Google Summer Of Code. Questa soluzione è stata installata e testata su un PC con GPU compatibile con CUDA, però, lavora solo off-line con una sequenza di immagini predefinite.

SVO[57]: acronimo di Fast Semi-Direct Monocular Visual Odometry, è un algoritmo SLAM che utilizza l’intensità del colore per tracciare il movimento di un oggetto in ambienti outdoor. Utilizza modelli probabilistici per costruire depth map semi dense, permettendo di risparmiare molto tempo nel calcolo della ”stima del movimento”. Gli svantaggi di questo metodo sono la necessità di avere un insieme molto grande di dati per l’inizializzazione dei parametri del modello probabilistico utilizzato; SVO è inoltre molto sensibile alle condizioni di illuminazione.

LSD-SLAM [53]: è un algoritmo SLAM che, come SVO, converte i frame delle stream video in mappe di profondità semi dense. In particolare, utilizzando la funzione di regressione di Huber, quest’ultime sono sovrapposte creando il modello 3D utilizzato per il tracking. Il vantaggio principale di LSD-SLAM rispetto ai metodi basati su caratteristiche, è la possibilità di generare una ricostruzione com-

pleta della scena, che contiene texture non rilevabili da algoritmi che si basano su features.

ORB-SLAM : l'algoritmo in questione utilizza lo scheletro di PTAM (con delle ottimizzazioni) sostituendo il descrittore FAST con ORB[100]. Quest'ultimo, definito dagli autori "un'alternativa efficiente a SIFT o SURF³", è essenzialmente una fusione dell'algoritmo FAST e BRIEF [41] (un features point descriptor basato su stringhe binarie) che offre prestazioni elevate nell'estrazione e nel matching di features. Il sistema SLAM risultante, è pertanto, molto performante; i risultati sperimentali dimostrano che non solo funziona meglio di PTAM, ma anche di sistemi SLAM densi.

2.4.1.2 SISTEMI MARKER-LESS MODEL BASED

Gli algoritmi di tracking model based possono essere classificati in due categorie: recursive tracking e tracking by detection. (Fig. 2.4.5.). In particolare, i primi utilizzano le pose della camera calcolate precedentemente per stimare la posa corrente; i secondi, invece, si basano esclusivamente sulle misurazioni dell'ultimo frame per stimare la posa attuale della camera.

Prendendo in considerazione il tipo di funzione utilizzata per il monitoraggio, inoltre, le tecniche model based possono essere ulteriormente classificate in[77]:

edge based: la posa della camera è calcolata effettuando il matching tra gli archi di un modello 3D (una rappresentazione virtuale dell'oggetto da tracciare) e gli stessi presenti nell'oggetto reale. Per quanto concerne i metodi ricorsivi, la stima degli archi è effettuata campionando dei punti consecutivi lungo i bordi del modello 3D; quest'ultimi sono quindi confrontati con gli archi calcolati dai frame acquisiti dalla camera. La posa della camera è pertanto aggiornata minimizzando la distanza tra le proiezioni dei punti campionati nel modello 3D e i match trovati nelle immagini acquisite (Fig. 2.4.6.) [113]. Per quanto concerne i metodi tracking by detection, invece, i metodi edge based, chiamati view based, confrontano

³SIFT e SURF sono due descrittori per estrarre features da immagini. Da qualche anno quest'ultimi sono stati brevettati, pertanto bisogna pagare per utilizzarli.

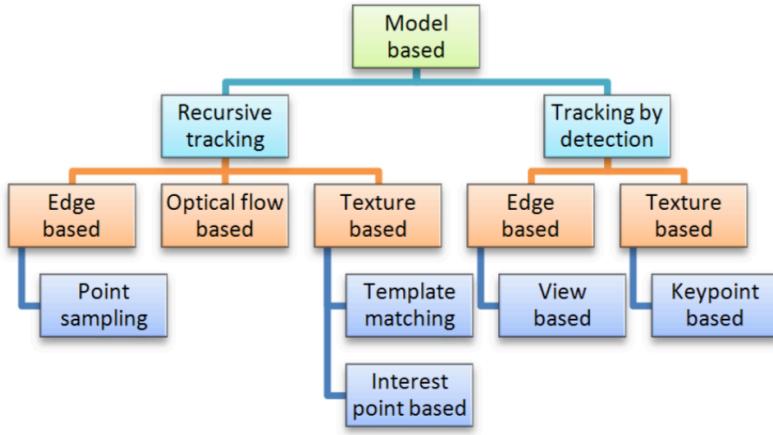


Figura 2.4.5: Tassonomia degli algoritmi model based, tratta da [77].

il frame corrente con viste 2D dell'oggetto target inquadrato da diverse posizioni e punti di vista.

Optical flow based: sfrutta le informazioni temporali estratte dal movimento per tracciare gli oggetti nella scena [109].

Texture based: questo metodo considera le texture presenti negli oggetti per effettuare il tracking. Nelle tecniche di tracking ricorsive, tali metodi possono essere ulteriormente classificati in: template matching e interest point based. Il primo metodo utilizza un modello di approssimazione per stimare il movimento dell'oggetto di cui effettuare il tracking[70]; il secondo, invece, stima la posa della camera utilizzando alcune features dell'oggetto da tracciare. In particolare, le caratteristiche estratte dal frame corrente sono confrontate con le features di keyframes selezionate offline. Per quanto concerne i metodi appartenenti alla classe tracking by detection, invece, i metodi texture based sono chiamati keypoint based. In questo caso, utilizzando dei descrittori quali SIFT, SURF, FAST, sono confrontate le features estratte dal frame corrente con quelle ottenute da una fase di training dell'oggetto da tracciare.

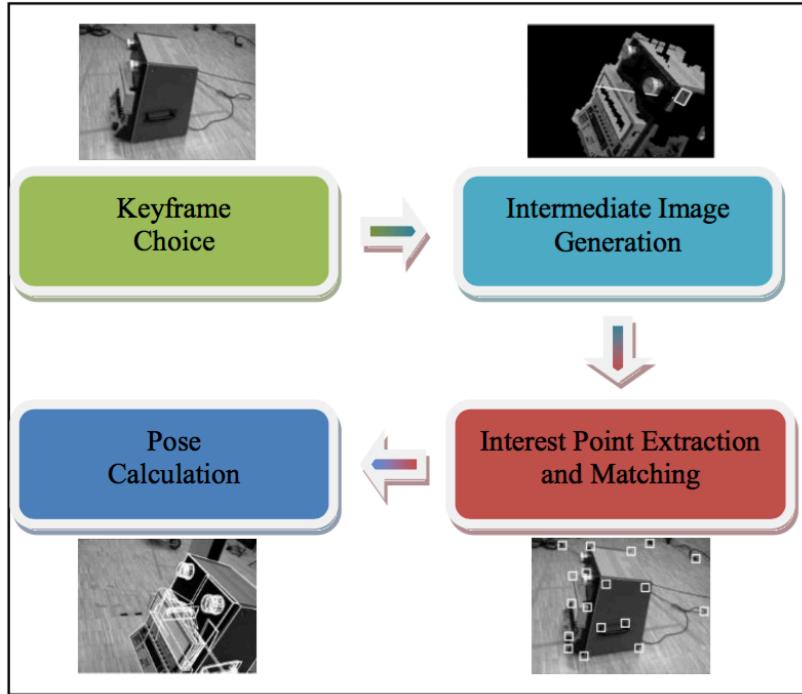


Figura 2.4.6: Workflow della metodologia Point Sampling, tratta da [77].

2.5 DISPOSITIVI DI VISUALIZZAZIONE

Si contano numerosi e diversi dispositivi di visualizzazione utilizzabili in sistemi di realtà aumentata. Si va da un semplice computer dotato di webcam [95], ai display portatili o hand held display (ad esempio i cellulari), o ancora ai visori HMD (Head Mounted Displays).

In particolare, il primo HMD fu realizzato nella seconda metà degli anni '60 da Ivan Sutherland e i suoi collaboratori. Quest'ultimo mostrava un ambiente virtuale formato da stanze tridimensionali e comprendeva un head-tracking per tracciare i movimenti della testa dell'utente [104].

Oggi giorno, per la realtà aumentata, sono disponibili in commercio due tipi di visori, chiamati optical see-through e video see-through. Tali dispositivi, indossabili dall'utente, hanno il vantaggio di fornire una sensazione di immersione nell'ambiente AR e, contemporaneamente, permettono di avere le mani libere per

agire sulla scena.

Un visore see-through ottico, molto simile ad un Head Up Display utilizzato dai piloti degli aerei dell'aviazione militare, impiega un divisore di fascio ottico composto da uno specchio traslucido posto davanti agli occhi dell'utilizzatore [35]; quest'ultimo trasmette la luce in una direzione e contemporaneamente la riflette nell'altra (Fig. 2.5.1.). Si tratta quindi di tecnologie parzialmente trasmittenti poiché, guardando attraverso la lente, si può vedere l'immagine virtuale sovrapposta a quella reale. Lo svantaggio dei combinatori ottici è, tuttavia, quella di ridurre l'intensità della luce della scena reale; infatti, attraverso tali display, è possibile percepire circa solo il 30% della luce reale.

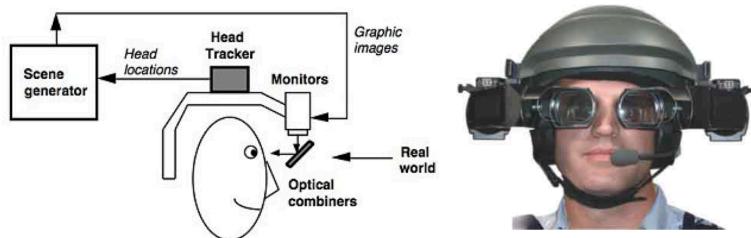


Figura 2.5.1: Principio di funzionamento e soluzione basata su HMD Optical See-Through, tratta da [35].

I visori see-through video usano, invece, una o due telecamere per acquisire le immagini del mondo reale; queste sono inviate ad un video compositore che le proietta sui display dell'HMD (uno per ciascun occhio), arricchite da elementi virtuali utilizzati per "aumentare" la scena reale.

Vantaggi e svantaggi di una soluzione rispetto all'altra non mancano affatto. I migliori risultati in termini di complessità degli oggetti virtuali sovrapposti al mondo reale si ottengono per mezzo dei visori video see-through[64]. Tuttavia, rispetto a quelli ottici, quest'ultimi richiedono sistemi di messa a fuoco automatica al fine di garantire la nitidezza del punto osservato dall'utente [34]. Inoltre, i tempi di risposta, ovviamente inferiori rispetto a quelli dell'occhio umano, rendono l'utilizzo

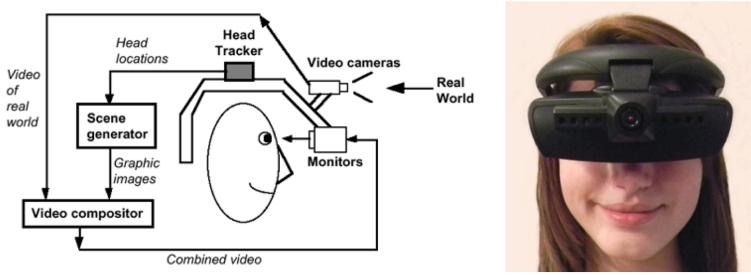


Figura 2.5.2: Principio di funzionamento e soluzione basata su HMD Video See-Through, tratta da [35].

di questi sistemi poco confortevole. I visori ottici, invece, non risentono di questo problema anche se la qualità degli oggetti virtuali aumentati non è paragonabile con i risultati ottenuti dai visori video see-through; la trasparenza delle lenti, infatti, impedisce agli elementi virtuali di essere correttamente visualizzati dall’utente.

Altri dispositivi di visualizzazione, sicuramente meno utilizzati rispetto a quelli citati, sono i virtual retinal display e i projecton display [34]. Nei primi, i contenuti virtuali sono proiettati, attraverso dei laser, direttamente negli occhi dell’utilizzatore; i secondi, invece, utilizzando dei proiettori 3D mostrano l’aumentazione direttamente sugli oggetti presenti nella scena. In quest’ultimo caso, non è quindi necessario utilizzare particolari visori per visualizzare le aumentazioni.

2.6 METRICHE PER VISORI SEE-THROUGH VIDEO

Il lavoro della telecamera in un sistema see-through video è quello di acquisire la scena dal punto di vista dell’utente; successivamente, il video in tempo reale è elaborato ed aumentato con contenuti grafici (modelli 3D e informazioni testuali).

Tra i parametri più importanti da considerare in un sistema di tipo see-through video si ricordano:

- **Risoluzione:** indica il numero di punti elementari presenti in uno spazio lineare (un centimetro, un pollice, ecc.). Come descritto in [30], al fine di garantire una buona sensazione di immersione, un HMD dovrebbe avere

una risoluzione di 1000x1000 pixel per occhio. Purtroppo, il dispositivo di visualizzazione utilizzato in questo lavoro di tesi (un Nexus 5) ha una risoluzione di (1920x1080)/2 pixel per occhio; il requisito della risoluzione non è pertanto soddisfatto.

- **Field Of View (FOV):** è il numero di gradi in angolo visivo mantenendo gli occhi fermi. Quello umano è di circa 135° mentre quello delle camere è calcolato nel modo seguente[24]:

$$\alpha = 2 \arctan \frac{d}{2f}$$

In particolare, d è la dimensione del diametro del sensore; come descritto in [17] tale valore si calcola a partire dal formato di quest'ultimo . La lunghezza focale (f), invece, è una caratteristica della lente e rappresenta la distanza tra la superficie del sensore o della pellicola e il centro della lente dell'obiettivo con il fuoco impostato all'infinito; questo valore è in genere fornito dal produttore.

La camera utilizzata in questo lavoro di tesi (quella di un Nexus 5), in base alle caratteristiche fornite dal produttore ⁴, ha un FOV di 70°. Anche in questo caso il valore ottenuto è inferiore a quello consigliato in [30] (80°). Durante gli esperimenti condotti si è cercato di aumentare il FOV della camera montando delle fish eye lens, però, l'immagine acquisita era troppo distorta per poter essere riconosciuta efficientemente dall'algoritmo di tracking utilizzato.

- **Frame rate:** è la frequenza di cattura o riproduzione dei fotogrammi che compongono un filmato. Il frame rate minimo di un video, affinché l'apparato visivo umano non percepisca sfarfallii e artefatti, si attesta sui 29,5 fps (30 Hz), ma può anche essere superiore se gli oggetti si muovono velocemente. Il sistema AR descritto nel quarto capitolo di questo elaborato, in base

⁴La camera del Nexus 5 ha un sensore di 1/3.2" (diagonale di 5.68 mm) e una lunghezza focale di 4 mm.

al surriscaldamento del dispositivo e alle difficoltà che l'algoritmo di tracking ha nel tracciare gli oggetti nella scena, funziona a 15-30 FPS. Anche in questo caso quando il frame rate percepito è troppo basso e non si gode di una buona esperienza AR.

Alla luce di quanto detto, il sistema utilizzato, un Nexus 5 inserito in un VR Box 2.6.1, non soddisfa i requisiti minimi per fornire una soddisfacente esperienza AR. HMD di fascia superiore avrebbero offerto sicuramente una sensazione di immersione maggiore, però, il loro costo è molto elevato (i visori Sensics, ad esempio, costano fino a decine di migliaia di euro). Concludendo, la soluzione see-through video utilizzata in questo lavoro di tesi, è, a livello prestazionale, comparabile con un sistema quale l'Oculus Rift DK2 equipaggiato con una buona camera. Un HMD quale l'EmaginZ800 fornisce, invece, una sensazione di immersione peggiore rispetto a quella implementata.



Figura 2.6.1: VR box: lo smartphone può essere inserito all'interno del visore ed essere utilizzato come dispositivo see-throuh video.

3

Soluzioni software per la modellazione di un sistema AR

LA REALIZZAZIONE DI UN SISTEMA AR consta generalmente di più step, ognuno dei quali è realizzato attraverso differenti software. I più importanti sono:

- **Software di modellazione:** consentono di creare oggetti complessi a partire da forme elementari quali, ad esempio, cubi, sfere ecc. In particolare, un modello 3D è realizzato applicando ad una primitiva solida elementare delle trasformazioni geometriche (scaling, rotazioni, ecc.) oppure dei modificatori. E' possibile moltiplicare le facce di un oggetto ed estruderle, combinare più forme tra loro (operazioni booleane), smussare gli angoli di un poligono, eliminare da un oggetto vertici, archi, ecc. In sintesi, applicando in modo

iterativo e non distruttivo diverse funzioni presenti nei software di modellazione è possibile ottenere, a partire da forme elementari, oggetti complessi e articolati (Fig. 3.0.1.). Attraverso degli store dedicati è possibile, inoltre, acquistare dei modelli 3D direttamente utilizzabili nei propri progetti.

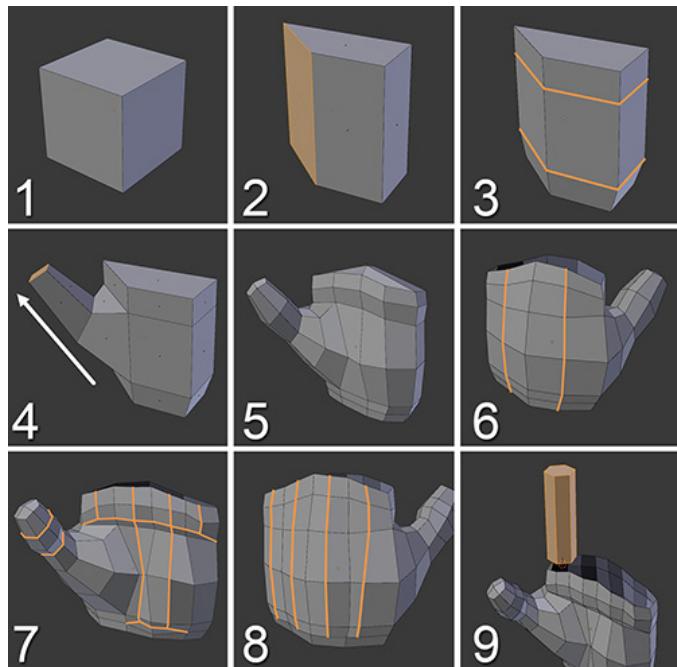


Figura 3.0.1: Processo di modellazione di una mano a partire da un cubo

- **Motori grafici:** è il nucleo software di un applicativo grafico in quanto fornisce le tecnologie di base tipicamente utilizzate per sviluppare un video-gioco o applicazioni di realtà virtuale ed aumentata. In particolare queste includono: un motore di rendering, un motore fisico o rilevatore di collisioni, suono, scripting, animazioni, intelligenza artificiale, networking, ecc. I motori grafici sono costruiti nella maggior parte dei casi su delle API grafiche, come Direct3D o OpenGL, che offrono uno stack software per interfacciarsi con la GPU. Spesso sono anche usate librerie di basso livello, come DirectX, SDL e OpenAL, le quali offrono un accesso diretto ad altro hardware (ad esempio mouse, tastiera, joystick, scheda di rete, e scheda audio)

[6]. Il tutto è fornito generalmente all'interno di un ambiente di sviluppo integrato (IDE); tutte le funzionalità sono pertanto "a portata di click".

- **AR SDKs (Software Development Kits)** : sono delle librerie software integrabili nei motori grafici, in ambienti di sviluppo mobile (Android, iOS, Windows Mobile) oppure utilizzabili con linguaggi web (pensiamo ad esempio a JavaScript). In alcuni casi, si veda Wikitude SDK o Vuforia SDK, tali librerie sono in grado di "interfacciarsi" direttamente con alcuni visori see through video e ottici. Utilizzando un SDK AR l'utente non deve preoccuparsi di reimplementare gli algoritmi di detection e di tracking che sono alla base di un sistema AR; quest'ultimi sono incapsulati, infatti, all'interno della libreria fornita.

Eccezione fatta per i software di modellazione, che devono essere necessariamente utilizzati per creare gli oggetti virtuali, i motori grafici e gli SDKs AR non sono indispensabili nella realizzazione di un sistema di realtà aumentata. In alcuni progetti di ricerca, dove l'obiettivo è studiare nuovi algoritmi di detection e di tracking, si utilizzano esclusivamente linguaggi di programmazione di basso livello (quali il C e il C++). In questi casi, le API grafiche come Direct3D o OpenGL, sono "chiamate" senza ricorrere ad alcun middleware; anche se difficili da migrare (soprattutto su sistemi mobile), le applicazioni in questione sono molto performanti. Il codice sorgente di algoritmi "innovativi", però, non sempre è disponibile al pubblico.

3.1 BLENDER

Gli oggetti 3D integrati nel sistema AR realizzato per questo progetto di tesi sono stati modellati con Blender.

Mentre tutti gli altri programmi di modellazione sono a pagamento, Blender è open source e gratuito; inoltre, a meno che non si debbano realizzare progetti da milioni di dollari, non ha nulla da invidiare a programmi molto costosi come 3ds Max e Maya. A tal proposito, i modelli utilizzati in questo progetto sono molto

semplici e utilizzano un sottoinsieme limitato delle potenzialità offerte dal software di modellazione utilizzato.

Blender [79] [32] è un software open-source, free e multipiattaforma che viene utilizzato non solo per la modellazione, ma anche per l'animazione, il compositing e il rendering di immagini tridimensionali. Nasce nel 1995 all'interno di uno tra i più importanti studi d'animazione europei: la società olandese NeoGeo. Dopo un'attenta valutazione, la compagnia decise che lo strumento 3D utilizzato fino a quel momento era troppo scomodo da mantenere e sviluppare, pertanto, dopo travagliate vicende, nel 2002 Blender diventò un progetto Open Source. Lo sviluppo oggi è portato avanti da un estesa community di volontari in tutto il mondo, guidata dal creatore originale, Ton Roosendaal.

Oggi Blender è un programma molto flessibile che può essere facilmente compreso e utilizzato da chi si avvicina per la prima volta al mondo della modellazione 3D, ma offre strumenti complessi anche per i più esperti. Esso è in grado di rappresentare diversi fenomeni fisici tra i quali: stato gassoso e liquido della materia e gravità. Inoltre offre delle primitive geometriche per modellare i diversi oggetti della scena, insieme a degli strumenti di manipolazione ed editing che consentono di realizzare qualsiasi forma si voglia.

3.1.1 L'INTERFACCIA DI BLENDER

L'interfaccia di Blender si presenta all'utente così come mostrata in figura 3.1.1; quest'ultima è disegnata completamente in OpenGL, pertanto tutte le finestre possono essere spostate, ingrandite/rimpicciolate e duplicate.

Gli strumenti evidenziati in rosso (Fig. 3.1.1.) consentono di manipolare forma e comportamento degli oggetti presenti nella scena. La sezione Transform, ad esempio, consente di muovere/rotare/scalare gli oggetti lungo gli assi x, y e z. In quest'area sono inoltre presenti la sezione Create (consente di creare delle primitive elementari), la sezione Animation (consente di gestire le animazioni degli oggetti 3D), la sezione Physics (consente di simulare fenomeni fisici reali), ecc.

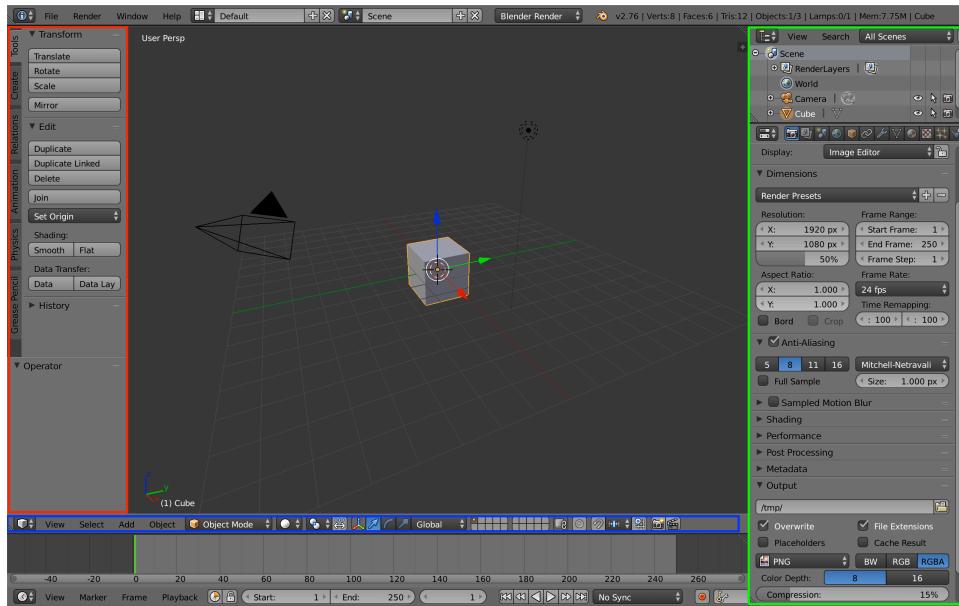


Figura 3.1.1: L'interfaccia di Blender

Gli strumenti evidenziati in verde (Fig. 3.1.1.) consentono di accedere a differenti funzionalità tra cui la gestione dei materiali (colore e texture degli oggetti) e del rendering. Con il termine rendering si intende il processo di calcolo della luce che interagisce sugli oggetti e i loro rispettivi materiali. In Blender gli strumenti per calcolare i rendering si chiamano Motori e sono raggruppabili in due categorie [18]: Biased e Unbiased. I motori Biased (utilizzati di default in Blender) calcolano la luce basandosi sull'approssimazione visiva; l'algoritmo calcola le dimensioni degli oggetti, la distanza dalle luci e ne ricava ombre e illuminazione. Questo tipo di soluzione tuttavia viene definita non fisicamente corretta, perché non calcola gli infiniti rimbalzi della luce sugli oggetti. Il motore Unbiased (in Blender chiamato Cycles), invece, calcola gli infiniti rimbalzi che ogni fonte luminosa produce; il risultato è quindi un immagine granulosa migliorata man mano che il processo di rendering avanza. Tra gli strumenti presenti in questa finestra (riferirsi a tal proposito all'area evidenziata in verde) è inoltre presente il pulsante per accedere ai modificatori di Blender; questi ultimi sono degli algoritmi che possono essere applicati ad ogni oggetto e consentono di realizzare degli "effetti" particolari in

maniera automatica. Tra i più utilizzati in questo lavoro è inevitabile non citare[2]:

- Mirror: consente di realizzare degli oggetti simmetricamente uguali. In poche parole, modificando la metà di un modello, automaticamente (e simmetricamente) si modifica anche l'altra metà.
- Array: crea copie (con vertici adiacenti) dello stesso oggetto.
- Boolean: effettua operazioni geometriche di unione, intersezione e differenza geometrica tra due modelli.
- Decimate: riduce il numero di poligoni di un oggetto.
- Solidify: aggiunge spessore alla mesh di un oggetto.
- Subdivision Surface: leviga le superfici attraverso l'interpolazione geometrica.
- Triangulate: converte tutte le facce in triangoli.

Il cuore del programma di modellazione è, infine, la 3D View (evidenziata in blue nella figura 3.1.1), dove si eseguono la maggior parte delle operazioni per modellare un oggetto. Qui troviamo diversi strumenti come pivot, layer, selettore di facce, archi e vertici e quant'altro. In questa sezione è possibile, inoltre, scegliere se lavorare in Object Mode o in Edit Mode: la prima modalità è usata per manipolare gli oggetti nella loro interezza, mentre la seconda per modificare la geometria di quest'ultimi. Molte delle funzionalità presenti in questa modalità si sono rivelate davvero utili nella modellazione degli oggetti 3D realizzati per il sistema AR proposto (Fig. 3.1.2.), pertanto si riportano di seguito alcune operazioni tipicamente eseguite in Edit Mode[2]:

- Selezione e creazione di vertici, archi e facce.
- Translazione, rotazione, scaling e cancellazione di vertici, archi e facce.
- Divisione delle facce di un oggetto.
- Estrusione delle facce di un oggetto.
- "Ritaglio" di geometrie particolari all'interno di una faccia (si veda strumento Knife).

Le possibilità offerte da Blender sono dunque infinite; l'utilizzo di questo sottosistema di funzionalità unite al potere dei modificatori consente di creare modelli anche molto complessi e articolati. Vi sono tuttavia tante altre funzionalità che non sono stati utilizzate in questo lavoro; per una panoramica globale delle possibilità offerte da questo straordinario software di modellazione (e non solo) rimandiamo il lettore alla guida ufficiale di Blender.

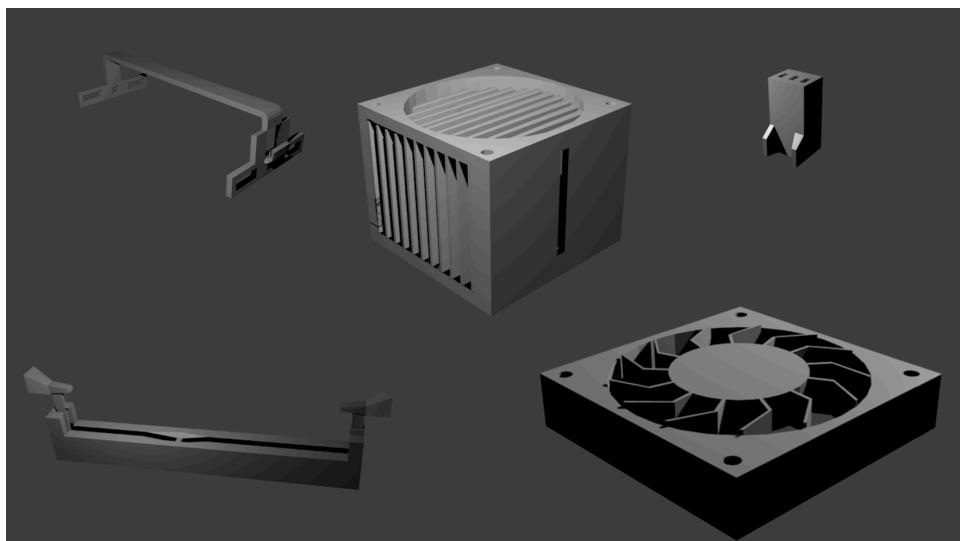


Figura 3.1.2: Render di alcuni modelli 3D realizzati con Blender

3.2 UNITY3D

Unity3D (o semplicemente Unity) è un engine per lo sviluppo di videogiochi che dal 2005 (anno del suo primo rilascio) è diventato sempre più popolare, soprattutto fra gli sviluppatori mobile, fino a potersi definire l'engine per videogames più usato al mondo [19]. L'ambiente di sviluppo, infatti, gira sia su Microsoft Windows che su OS X e i giochi che produce possono essere eseguiti su Windows, Mac, Linux, Xbox 360, PlayStation 3, PlayStation Vita, Wii, iPad, iPhone, Android e, dal 2014 anche su Playstation 4, Xbox One e Wii U[26].

Unity3D è inoltre l'engine grafico preferito da molti sistemi di realtà aumentata commerciali; in generale, importando nel proprio progetto un unitypackage (una raccolta di script, librerie esterne, ecc.) si hanno a disposizione tutti gli strumenti necessari per modellare una scena AR. In particolare, Unity consente di posizionare in maniera molto intuitiva gli oggetti virtuali rispetto ai target (gli oggetti reali) e si ha sempre una corrispondenza visiva di ciò che si ha rispetto a ciò che si vorrebbe. Tale paradigma è normalmente più difficile da "concretizzare" in altri ambienti quali, ad esempio, Android Studio e XCode; in questi casi la modellazione di una scena AR avviene in maniera prevalentemente "procedurale".

Nella prossima sezione si introdurrà al lettore Unity3D. Affrontare nello specifico il funzionamento e i tecnicismi del game engine esula dagli obiettivi di questo elaborato, pertanto l'overview proposta servirà solamente a fornire al lettore l'organizzazione delle interfacce e le basi per comprendere il funzionamento del programma. La seguente sezione può quindi essere considerata un indice degli argomenti da approfondire nei tutorial ufficiali [7].

3.2.1 INTERFACCIA E ALTRE FEATURES DI UNITY

Unity offre un'interfaccia formata da vari pannelli ridimensionabili che possono essere spostati a piacimento secondo le proprie esigenze operative.

L'editor prevede le seguenti finestre:

- **Scene:** la scene view è la finestra che rappresenta la scena attualmente aperta: è possibile muovere, spostare e rotare gli oggetti virtuali in un ambiente a tre dimensioni. Per la realizzazione del prototipo AR discusso in questa tesi, la scene view si è rivelata particolarmente utile per il posizionamento dei contenuti virtuali sugli oggetti target; in Play Mode, spostando i contenuti virtuali nella scene view è stato possibile visualizzare la posizione che essi presentavano nella game view (vedi punto successivo).

Oltre agli oggetti “tangibili”, la scene view permette la visualizzazione i cosiddetti Gizmo, oggetti che non hanno una rappresentazione fisica e che vengono visualizzati come delle icone.

- **GameView:** mostra il risultato finale dell'applicazione realizzata; nel caso di un'applicazione AR, in particolare, i contenuti visualizzati in questa finestra sono lo stram video di una camera e i contenuti virtuali che aumentano la realtà.
- **Hierarchy:** presenta tutti gli oggetti nella scena; quest'ultimi, detti GameObject, possono essere telecamere, luci, modelli 3D, prefabs, ecc.
- **Project:** è una finestra contenente tutte le risorse necessarie per sviluppare un applicativo, tra cui modelli 3D, effetti sonori, texture, ecc. A differenza dei contenuti presenti nella finestra Hierarchy, gli oggetti dalla project view non hanno un ruolo attivo all'interno dell'applicativo che si intende realizzare; solo se spostati nell'Hierarchy quest'ultimi saranno parte integrante del risultato finale mostrato all'utente.
- **Inspector:** questa finestra visualizza i componenti e le proprietà dei GameObject selezionati nella Hierarchy. In particolare, nel progetto realizzato per questa tesi, la Hierarchy è stata utilizzata per settare i valori di input di alcuni oggetti presenti nella scena; ad esempio per gestire le animazioni dei modelli virtuali è necessario fornire una posizione spaziale di inizio/fine, l'asse sul quale l'oggetto sarà animato, ecc.

Di seguito è mostrato uno screenshot dell'interfaccia Unity. La scene view è evidenziata in giallo, la game view in lilla, l'hierarchy view in rosso, la project view in verde e l'inspector in blu.

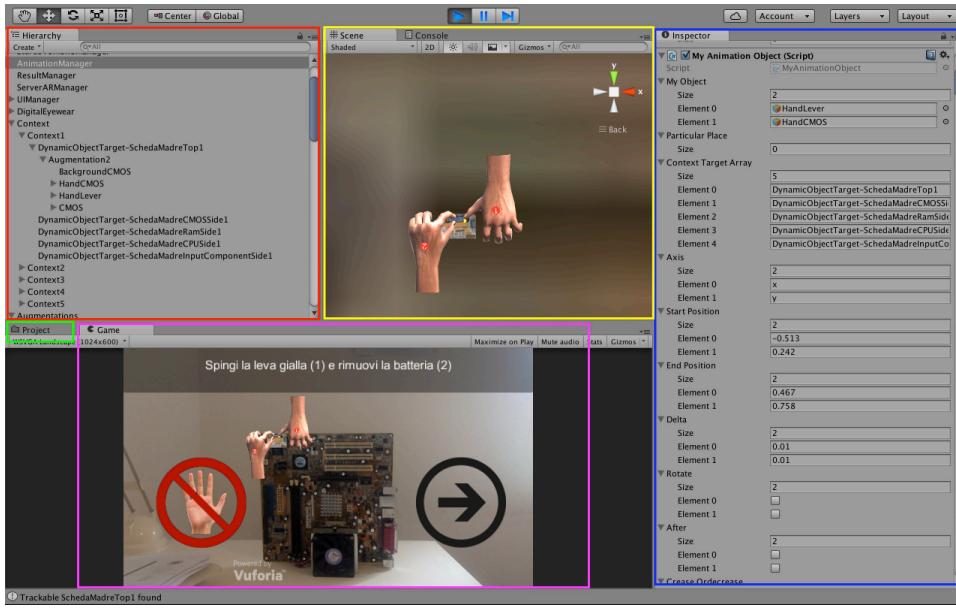


Figura 3.2.1: Interfaccia e pannelli Unity.

Altre features di Unity3D, di fondamentale importanza per la creazione di un gioco o di un sistema AR sono:

- **I Game Object:** tutto ciò che viene visualizzato a schermo in Unity è un GameObject; lo è ad esempio un modello che aumenterà la scena AR, una camera che riconoscerà i targer nella scena, le luci, gli elementi della user interface e quant'altro. Il comportamento di ogni Game Object può essere modificato attraverso i parametri pubblici mostrati dall'oggetto correntemente selezionato nell'inspector oppure attraverso degli script. In particolare, attraverso la funzione GetComponent si può ottenere l'istanza di un qualsiasi Game Object e modificarne il comportamento a runtime: è possibile associare altri script (AddComponent) oppure distrutteggiare quelli esistenti.

ti (DestroyComponent), renderizzare una mesh, disattivare componenti, ecc.

- **Parent-Children** : in Unity è possibile collegare il comportamento di un oggetto ad un altro creando una delle relazioni padre-figlio. Queste relazioni possono estendersi anche su più livelli.
- **Prefab** : Unity utilizza il meccanismo dei prefab, asset salvati nel progetto, riutilizzabili in tutte le scene. Per creare un prefab basta creare un GameObject nella Hierarchy, aggiungere proprietà e componenti e trascinarlo nel pannello Project. Ogni volta che il prefab verrà trascinato nella scena verrà quindi creata un'istanza di quell'oggetto: tutte le modifiche apportate al prefab nel pannello Project, verranno applicate a tutte le istanze del prefab nella scena del gioco. A titolo esemplificativo, molti degli SDKs AR presentati nella sezione 3.3 utilizzano il meccanismo dei Prefab fornire all'utente "oggetti già pronti" da usare nei propri progetti AR.

3.2.2 SCRIPTING

In Unity, per aggiungere interattività agli oggetti bisogna creare degli script e collegarli ai Game Object in scena. In breve, gli script in Unity sono classi che ereditano da Monobehavior, una classe che identifica una categoria di oggetti (fisici, ma anche virtuali) che hanno tutti le stesse proprietà. Ogni script pertanto sovrascrive automaticamente i metodi:

- **Start**: funzione chiamata all'inizio del ciclo di vita di un oggetto, solo se l'oggetto è abilitato. Serve a inizializzare le variabili.
- **Update**: funzione chiamata ad ogni frame, prima del rendering. Viene utilizzata soprattutto per aggiornare la posizione di un oggetto o per controllare l'input.

3.3 SISTEMI COMMERCIALI AR BASATI SU RILEVAMENTO OTTICO

Il potenziale della realtà aumentata è stato notato da molte aziende che hanno iniziato a sviluppare engines AR per il grande pubblico. Gli SDKs disponibili oggi sono tantissimi, altri stanno nascendo [25], altri ancora, invece, non sono più disponibili.

In questo lavoro sono state testate le seguenti soluzioni AR commerciali:

ARToolKit[10]: è una libreria open-source (il progetto è scaricabile da GitHub) scritta in linguaggio C e C++; è utilizzata per creare applicazioni di realtà aumentata robuste ed affidabili sia in campo accademico che commerciale[80] [81]. Attualmente supporta il riconoscimento dei markers (compreso il multimarker), dei codici a barre 2D e il natural feature tracking. La libreria è compilata per tutte le piattaforme: Mac OS X, PC, Linux, Android, iOS ed è disponibile il plugin gratuito per Unity3D Pro; inoltre è utilizzabile sia con sistemi see-through ottici che video. Per gli esperimenti condotti sono stati utilizzati i sample ARSimple (app marker based per android) e ARMulti (app multi-marker based per Android) scaricabili da [11].

IN2AR[20]: è un engine di realtà aumentata multipiattaforma che consente all'utente di riconoscere ed aumentare immagini bidimensionali; è disponibile anche un plugin per Unity3D. L'esempio di prova è stato scaricato dal play Store Android.

CraftAR[12]: è un insieme di tools rivolti principalmente alle aziende che vogliono utilizzare l'AR per pubblicizzare i propri prodotti. Dall'app web CraftAR è possibile modellare una scena AR direttamente dal browser; sono disponibili inoltre gli SDK per Android e iOS (vi sono anche le estensioni per PhoneGap e Cordova), un plugin per Unity e un servizio REST. Il sample dell'SDK in questione è stato scaricato dal Play Store Android.

Wikitude[27]: permette di creare applicazioni di realtà aumentata pensate soprattutto per contesti outdoor; uno dei punti di forza dell'SDK è infatti la possibilità di sfruttare il tracking basato sulla posizione (utilizzando il GPS, "la posizione web" e la bussola). Oltre al tracking location based Wikitude consente

di aumentare immagini planari e, da ottobre 2015, l'azienda utilizza SLAM per il tracking di oggetti e piccole scene [28]. La libreria è compilata per le piattaforme Android e iOS (sono disponibile anche plugings per Cordova, Titanium e Xamarin) ed è disponibile un plugin gratuito per Unity3D; inoltre supporta soluzioni see-through ottiche e video quali Epson Maverio, Google Glasses e Vuzix. Per gli esperimenti marker based e marker-less condotti è stato utilizzato il sample Wikitude scaricabile dal Play Store Android.

Kudan[21]: è un engine di realtà aumentata disponibile per piattaforme Android e iOS che consente all'utente di riconoscere ed aumentare immagini bidimensionali; da gennaio del 2016 anche Kudan utilizza SLAM come tecnologia marker-less. Per gli esperimenti marker based e marker-less condotti è stato utilizzato il sample Kudan scaricabile dal Play Store Android.

Experenti[15]: è una società italiana nata da una collaborazione tra l'Università di Padova e Mentis, società di consulenza strategica. Permette di riconoscere e aumentare immagini planari. Per gli esperimenti marker based condotti è stata utilizzata il sample Experenti scaricabile dal Play Store Android.

Vuforia: era un progetto della Qualcomm Inc., azienda operante nel settore delle telecomunicazioni tra i primi 20 produttori di semiconduttori al mondo; nel novembre del 2015 è stata acquisita dalla PTC Inc. A differenza di quanto accaduto per Metaio e 13th Lab (vedi sotto), però, l'SDK di Vuforia è ancora disponibile al pubblico. Le opportunità AR offerte dall'azienda statunitense sono variegate e sono fruibili attraverso dispositivi Android ed iOS per cui sono rilasciati i rispettivi SDKs e il plugin per Unity. Vuforia supporta inoltre differenti dispositivi see-through, sia ottici che video, tra i quali: Epson Moverio BT-200, ODG R-6 R-7, Samsung Gear VR e Google Cardboard; a marzo 2016 PTC Inc. ha annunciato che Vuforia supporterà anche Microsoft HoloLens. L'SDK presenta le seguenti features:

- Riconoscimento di immagini planari: utilizza come target delle immagini bidimensionali dalle quali sono estratte le caratteristiche utilizzate per il riconoscimento. Ad ogni immagine è attribuito un voto (da 0 a 5) in base

alle features che essa possiede; immagini con basso contrasto oppure con “motivi ripetuti” non sono facilmente riconoscibili.

- Multi target e Cylinder target: consentono rispettivamente di riconoscere una o più immagini in una disposizione geometrica definita e immagini arrotolate in un cilindro o un cono.
- Frame marker: Vuforia può utilizzare 512 marker, chiamati Frame Marker, distribuiti in un archivio all'interno della cartella assets (presente in qualsiasi SDK Vuforia).
- User Generated Targets (UGT): sono image targets create a runtime dall'utente. In poche parole, l'utente può scattare una foto all'ambiente reale e subito dopo tracciarlo ed aumentarlo.
- Object Recognition (OR): consente di rilevare ed effettuare il tracking di oggetti 3D. Tale feature sarà approfondita dettagliatamente nel capitolo successivo dato che l'object target di Vuforia è stato scelto come sistema marker-less per l'implementazione del prototipo AR realizzato.

Di seguito sono riportati due sistemi commerciali, entrambi con funzionalità markerless, ormai non più in vendita, nonostante fossero molto apprezzati dal pubblico per la loro oggettiva affidabilità e robustezza.

Metaio: era una startup tedesca nata come un progetto della Volkswagen; rappresentava uno dei framework più completi nell'ambito della realtà aumentata, infatti, era compatibile con Unity, funzionava su sistemi operativi Android, iOS, Windows e utilizzava sia sistemi marker based che marker-less. In particolare, l'SDK consentiva di utilizzare ID Marker, Picture Marker (basati su immagini) e LLA marker [31] (Latitude/Longitude/Altitude Marker), speciali marcatori che integrano informazioni geografiche; tra le altre cose Metaio è stata una delle prime aziende ad utilizzare la tecnologia SLAM. L'Apple, sempre più interessata alla realtà aumentata, ha acquistato la startup tedesca, pertanto, l'SDK Metaio non è più in vendita.

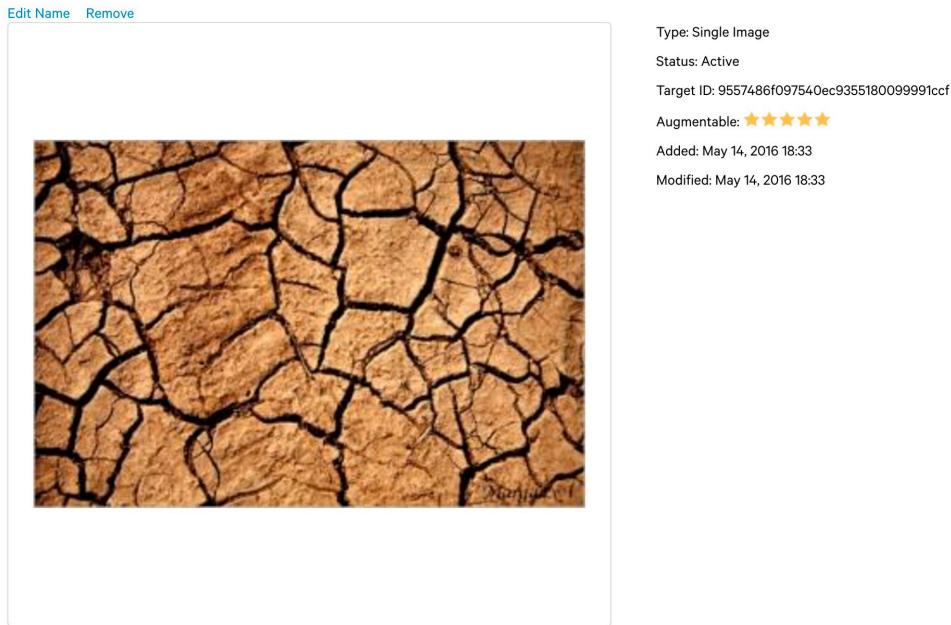


Figura 3.3.1: Image target di Vuforia.

13th Lab: era una piccola azienda svedese che ha sviluppato un SDK chiamato Pointcloud, utilizzabile sia su iOS che su Android e compatibile con Unity 3D. Anche questa piccola realtà, il cui punto di forza era SLAM, nel dicembre del 2014 è stata acquisita. In particolare Facebook inc. intende utilizzare le potenzialità della tecnologia visual SLAM sviluppata da 13h Lab sull'Oculus Rift, la cui società è stata anch'essa acquisita da Facebook (<https://www.facebook.com/zuck/posts/10101319050523971>).

Altre soluzioni AR che non presentano una licenza free sono:

ARmedia[8]: fornisce un ambiente per creare, modellare soluzioni marker-based e marker-less per piattaforme iOS, Android, Windows, OS X; è inoltre presente un plugin per Unity3D. Per quanto concerne la modellazione di un sistema marker-less, gli steps da seguire sono i seguenti:

- Utilizzando un'apposita app, scattare delle foto all'oggetto 3D da utilizzare come marker. Come riportato sul sito di ARmedia tipicamente sono neces-

sarie dalle 30 alle 50 per avere un buon risultato; inoltre alcuni scatti devono presentare parti comuni.

- Caricare le foto sulla piattaforma online di ARMedia ed associare a quest'ultime un modello 3D dell'oggetto scansionato (utilizzando ad esempio Autodesk's 123D Catch).
- Selezionare dei punti rappresentativi dal modello 3D e dalle immagini 2D facendo in modo che questi coincidano (Fig. 3.3.2.).

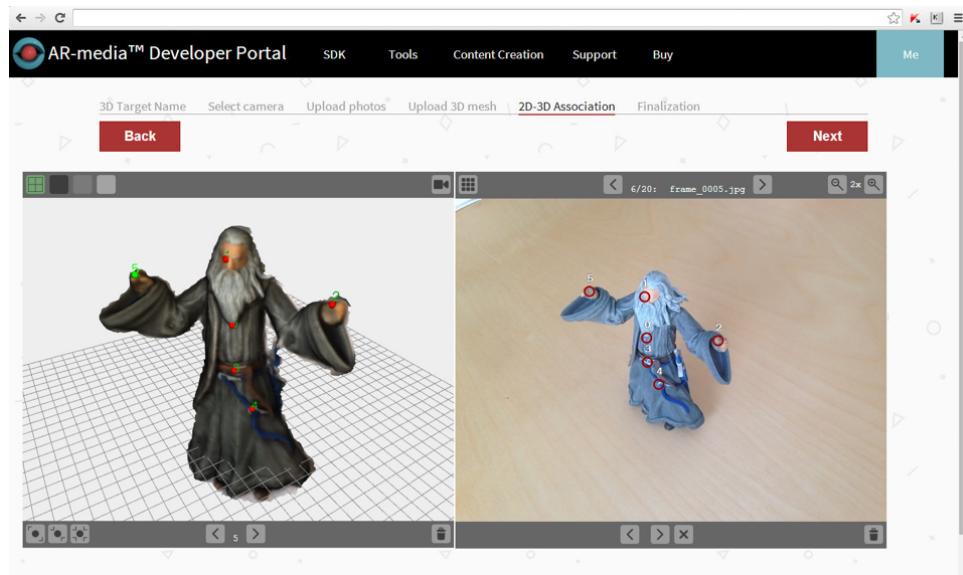


Figura 3.3.2: ARMedia: selezione di features da un modello 3D e da un'immagine 2D tipica dei sistemi Interest Point Based.

In linea con gli obiettivi di questa tesi sarebbe stato desiderabile implementare un sistema AR marker-less utilizzando l'SDK in questione; però, tale feature è a pagamento. Il sample marker based testato invece è stato scaricato da [9].

Project Tango[23]: è un SDK commercializzato da Google composto da un tablet Android equipaggiato con GPU Cuda e dotato di una camera grandangolare, una camera stereo, un sensore timestamping, sensori e proiettori ad infrarossi

e uno stack software. Il kit completo è venduto ad un prezzo di 512 € e consente agli sviluppatori di effettuare tipiche operazioni di Simultaneous Localization and Mapping, tra cui: motion tracking per stimare il movimento del dispositivo nello spazio, area learning per apprendere le caratteristiche dell'ambiente esplorato e depth perception per comprendere qual è la distanza degli oggetti osservati dalla camera. Project Tango può utilizzare APIs C, Java ed è disponibile anche un plugin per Unity3D. Anche se non è strettamente legato al progetto Tango è importante sottolineare come Google dal 2009 stia lavorando ad un progetto che consente ad un'automobile di guidare autonomamente utilizzando, tra le altre tecnologie, SLAM.

3.3.1 SOLUZIONI AR MARKER BASED E MARKER-LESS A CONFRONTO

In questa sezione si discuterà dei risultati ottenuti confrontando le performance di differenti SDKs commerciali AR; l'obiettivo della sperimentazione è stato duplice:

- Scegliere un SDKs per implementare un prototipo AR marker-less.
- Comprendere quanto le soluzioni marker-less siano deficitarie rispetto a quelle marker based.

La sperimentazione è stata svolta effettuando delle misurazioni oggettive e nelle medesime condizioni di luce artificiale. Per quanto concerne le soluzioni marker based i target utilizzati sono quelli forniti nei rispettivi sample; questi sono stati stampati e mostrati all'AR camera senza modificare la loro dimensione e il loro aspect ratio originale. Per le soluzioni marker-less, invece, il target utilizzato è la scheda madre mostrata nell'immagine 5.0.1; in particolare quest'ultima è stata scansionata in posizione orizzontale rispetto al piano di lavoro.

La camera utilizzata per tutti gli esperimenti è quella incorporata in un LG Nexus 5 (8.0 megapixels).

Prendendo spunto da [102] e da [56], si è testato:

1 - Distanza massima di riconoscimento del bersaglio: l'oggetto target (immagine 2D o oggetto 3D) inizialmente è posto ad una distanza di 100 cm dalla ca-

mera inquadrante la scena; finchè il bersaglio non è riconosciuto, la camera avanza di volta in volta di 5 cm finchè il riconoscimento avviene (Fig. 3.3.3.).

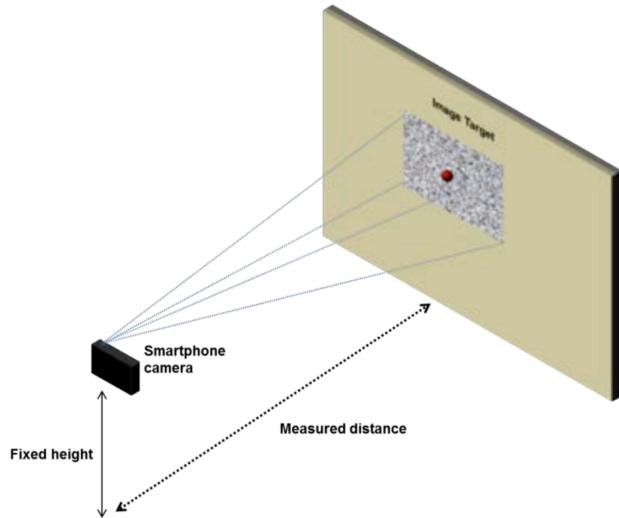


Figura 3.3.3: Distanza massima di riconoscimento del bersaglio, tratta da [102].

2 - Distanza massima tracking: quando l'oggetto target è stato riconosciuto, la camera indietreggia finchè il bersaglio non è più tracciato.

3 - Massima occlusione: a distanza fissata uguale per tutti gli algoritmi testati (50 cm), il target è occluso finchè il tracking del bersaglio fallisce.

4 - Massima inclinazione: a distanza fissata uguale per tutti gli algoritmi testati (50 cm), l'angolo che il target crea con il piano orizzontale è incrementato finchè il tracking del bersaglio avviene (Fig. 3.3.4.). L'inclinazione iniziale del target è 0° e raggiunge massimo 90° (vista frontale); la misurazione dell'angolo è stata effettuata con un goniometro.

5 - Qualità del tracking con camera fluttuante: a distanza fissata uguale per tutti gli algoritmi testati (50 cm), la camera è mossa ad una velocità sempre crescente. La qualità del tracking è valutata assegnando un punteggio da 0 a 6 (0: insufficiente, 6: eccellente) ad ogni algoritmo testato.

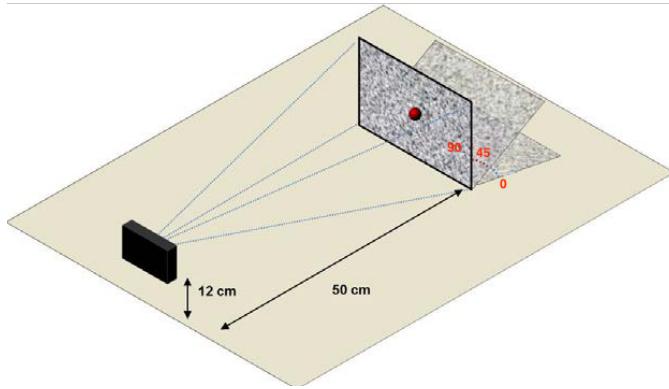


Figura 3.3.4: Massima inclinazione, tratta da [102].

6 - Falsi negativi: l'aumentazione non appare quando il target è visibile. La valutazione è stata effettuata assegnando un punteggio in una scala compresa da 0 a 6 (0: nessun falso negativo riscontrato , 6: molti falsi negativi).

7 - Falsi positivi: l'aumentazione appare anche quando il target non è visibile. La valutazione è stata effettuata assegnando un punteggio in una scala compresa da 0 a 6 (0: nessun falso positivo riscontrato , 6: molti falsi positivi).

8 - Tempo necessario per la detection: a distanza fissata uguale per tutti gli algoritmi testati (50 cm) con target a 90°, è stimato il tempo necessario all'algoritmo per riconoscere il bersaglio.

Per ogni metrica analizzata sono riportati di seguito i risultati ottenuti e alcune note appuntate durante gli esperimenti:

SDK sample	Max dist. recognition	Max dist. tracking	Max occlusion	Min angle	Tracking quality	False negative	False positive	Time
ARToolKit ARSimple	100 cm	220 cm	1/10	20°	5	0	3	0 sec
ARToolKit ARMulti	55 cm	70 cm	5/6	30°	3	0	3	2 sec
ARMedia	100 cm	165 cm	1/3	45°	2	0	0	1 sec
IN2AR	55 cm	140 cm	1/2	30°	3	0	2	0 sec
CraftAR	30 cm	180 cm	1/3	30°	3	0	0	2 sec
Wikitude Image Target	30 cm	150 cm	1/2	25°	3	0	0	0 sec
Wikitude SLAM	30 cm	40 cm	1/2	70°	1	5	5	0 sec
Kudan Image Target	95 cm	150 cm	3/4	20°	2	0	0	3 sec
Kudan SLAM	100 cm	300 cm	1	0°	4	6	6	0 sec
Experenti	45 cm	185 cm	3/4	35°	3	0	0	0 sec
Vuforia Image Target	90 cm	190 cm	5/6	25°	6	0	0	0 sec
Vuforia Frame Marker	75 cm	100 cm	2/3	35°	3	0	0	4 sec
Vuforia UGT	100 cm	120 cm	2/3	60°	3	0	0	0 sec
Vuforia OR	65 cm	280 cm	3/4	10°	4	0	0	1 sec

Tabella 3.3.1: Comparazione SDKs AR

Appunti degli esperimenti:

- ARToolKit ARSimple e ARMulti mostrano spesso l'aumentazione quando è inquadrata una superficie con contorni neri.
- IN2AR a volte blocca l'aumentazione anche quando il target non è visibile alla camera.
- Kudan SLAM presenta diverse problematiche tra le quali: la detection avviene solamente quando il bersaglio è riconosciuto la prima volta; quando il tracking fallisce l'aumentazione continua ad essere visibile.
- Wikitude SLAM continua a mostrare l'aumentazione quando il target non è più visibile alla camera; inoltre la detection del bersaglio fallisce quasi sempre e il tracking di quest'ultimo è molto instabile.
- Il tracking di Kudan Image Target "va a scatti".

I risultati ottenuti sono stati normalizzati rispetto al massimo (ad esempio metrica 1) e rispetto al minimo (ad esempio metrica 4) in modo che mostrassero dei valori nell'intervallo [0-1] dove 0 rappresenta un cattivo risultato e 1 il miglior risultato.

Nelle figure 3.3.5 e 3.3.6 sono graficati i risultati delle misurazioni effettuate; si noti come per una questione di leggibilità dei grafici le prime quattro metriche siano mostrate con dei boxplot mentre le ultime 4 con dei barplot; in quest'ultimo caso i valori ottenuti sono stati ordinati.

Dall'analisi condotta emerge come le soluzioni marker based siano decisamente migliori di quelle marker-less.

Eccezione fatta per Vuforia Object Target, che non presenta problematiche particolarmenente gravi, le altre soluzioni SLAM mostrano pesanti deficit: come detto precedentemente Kudan non è in grado di rieffettuare il tracking del target quando questo fallisce; solo al primo avvio (con il target puntato dalla camera) l'algoritmo in

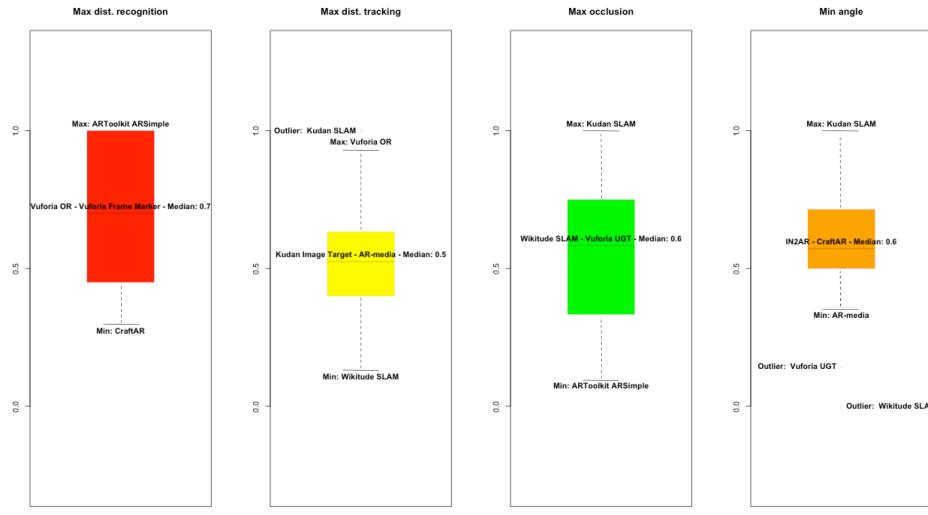


Figura 3.3.5: Boxplot metrica 1,2 e 3.

questione riesce a tracciare (anche molto bene) l'oggetto target. Wikitude SLAM, invece, pecca sotto ogni punto di vista. Al primo avvio, dopo aver spostato la camera di pochi centimetri il tracking fallisce; solo ritornando nella stessa posizione la scena viene tracciata di nuovo.

Vuforia Object Target è, invece, molto robusto ed affidabile: ottiene ottimi risultati nei test 2 e 4 e medi risultati nel test 3; leggermente sotto la media sono invece i risultati nel test 1. Unica grande pecca dell'algoritmo è un elevato tempo di attesa prima che la detection avvenga.

Per quanto concerne i sistemi marker-based le soluzioni ARToolkit sono vincenti e perdenti allo stesso tempo: ARSimple ottiene dei pessimi risultati per la metrica 3 e 8 (vedi appunti esperimenti) ottimi, invece, quelli relativi alle metriche 1,2,5; ARMulti gestisce bene le occlusioni (metrica 3) mentre non riesce a riconoscere e a tracciare bene i bersagli su lunghe distanze. La qualità del tracking è media e, come per ARSimple, presenta delle problematiche per i falsi positivi.

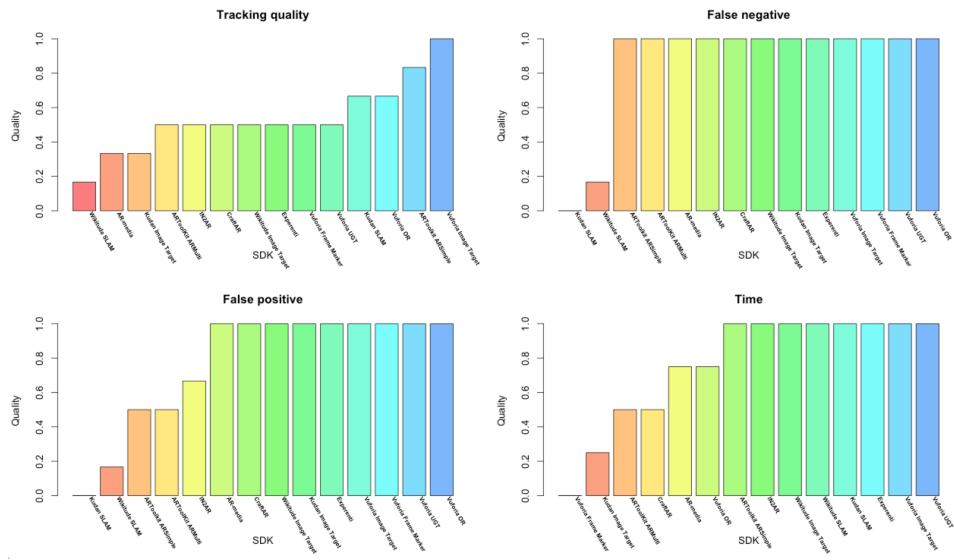


Figura 3.3.6: Barplot metrica 5,6,7,8.

Anche altre soluzioni marker based mostrano sia dei buoni e ottimi risultati che dei deficit:

- ARMedia non traccia bene la scena (metrica 5) e non gestisce in maniera efficiente le occlusioni; gli altri risultati sono medi (metrica 2 e 4) e ottimi (metrica 1).
- CraftAR riconosce il target ad una distanza molto ravvicinata (metrica 1); gli altri risultati sono invece medi.
- IN2AR presenta dei deficit sia per quanto riguarda i falsi positivi (metrica 8); altri risultati sono medio-bassi (metrica 1 e 3) e medi. Buono invece il risultato ottenuto per la metrica 4.
- Kudan Image Target non traccia molto bene il target e lo riconosce dopo molto tempo (metrica 8). Altri risultati sono medi ed altri ottimi (metrica 1 e 4).
- Experenti e Wikitude peccano sulla metrica 1. Gli altri risultati sono medi.

- Vuforia UGT ha delle difficoltà a riconoscere il target quando è leggermente inclinato. Gli altri risultati sono medi.

Vuforia Image Target mostra invece degli ottimi risultati (metrica 3 e 5) e dei buoni risultati (metrica 2). Gli altri sono medi; a differenza delle altre soluzioni marker-based non vi sono tuttavia deficit gravi.

Concludendo, Vuforia Image target e Object target sono le migliori soluzioni di realtà aumentata presenti oggigiorno sul mercato; gli altri sistemi testati, infatti, hanno almeno un deficit grave che potrebbe pregiudicare considerevolmente l'esperienza AR dell'utente.

A tal proposito si riporta di seguito un grafico mostrante la classifica globale delle soluzioni AR analizzate.

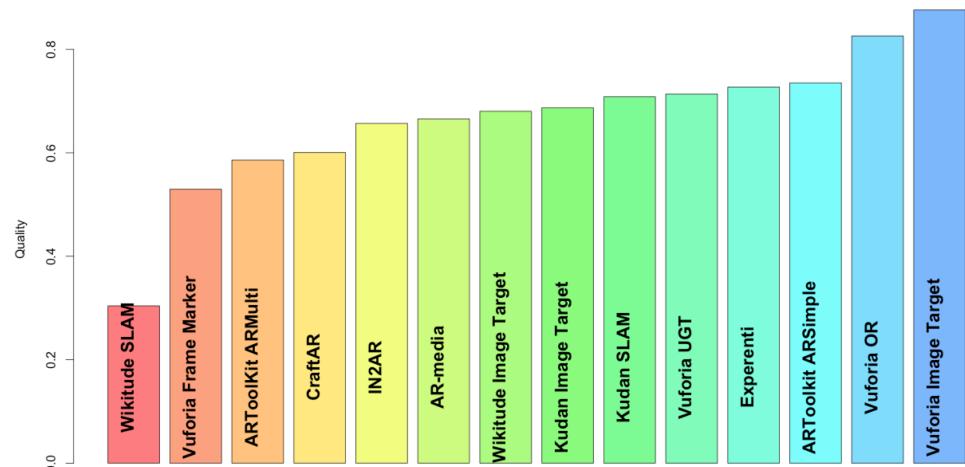


Figura 3.3.7: Classifica globale SDKs AR

4

Implementazione

In questo capitolo sarà introdotta l'implementazione del sistema AR realizzato per questo progetto di tesi. In linea con i risultati ottenuti dall'analisi comparativa descritta nel capitolo precedente, quest'ultima utilizza l'SDK Vuforia come motore per le aumentazioni.

4.1 OBJECT TARGET DI VUFORIA

Object target consente di rilevare ed effettuare il tracking di oggetti 3D estraendo da quest'ultimi natural features tracking; Vuforia tuttavia non ha reso pubbliche le informazioni sull'algoritmo utilizzato per il tracking, né sui descrittori per l'estrazione delle features.

Come riportato sul sito di Vuforia, Object Target può essere utilizzata con giocattoli (miniature di personaggi) e altri prodotti di consumo. Grandi oggetti come

case o autovetture, invece, non possono essere aumentati; in particolare, object target è ottimizzato per "oggetti che possono stare su un tavolo" e che presentano le seguenti caratteristiche:

- Rigidità
- Essere "contrast-based features"
- Non essere eccessivamente riflettenti alla luce

Scelto il target di interesse, così come descritto per ARMedia, l'utente deve procedere alla scansione di quest'ultimo con un'apposita applicazione; gli steps fondamentali da seguire sono riassunti di seguito:

- Scaricare l'app Object Scanner dal sito Internet di Vuforia e installarla sul proprio smartphone (Vuforia consiglia il Nexus 5 e il Galaxy S5 per ottimizzare i risultati).
- Stampare l'"Object Scanning Target", un foglio con un asse cartesiano utilizzato per posizionare l'oggetto 3D da scannerizzare; in particolare il target deve essere posizionato a partire dalla coordinata (0,0,0) affinchè l'intero oggetto sia scannerizzato.
- "Scansionare" l'oggetto target avendo cura di catturare quante più features possibili.

Terminata la procedura descritta, l'utente deve importare il file ottenuto nel target manager online, dove le features estratte dal target sono processate. Infine è possibile scaricare "un black box package" pronto all'uso da utilizzare con l'SDK fornito.

Come descritto nel capitolo precedente, l'implementazione del prototipo AR realizzato per questa tesi utilizza Unity3D.

4.2 SCELTE PROGETTUALI EFFETTUATE

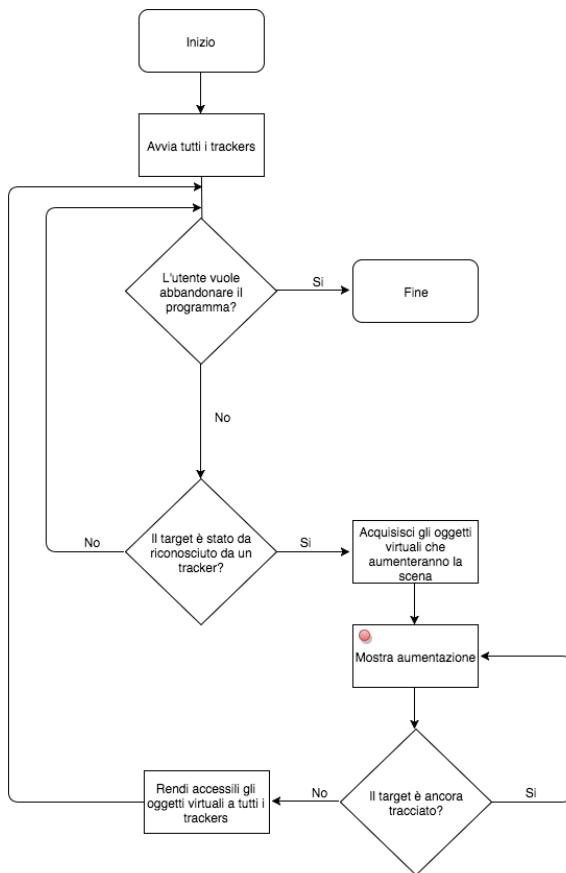
Come sottolineato nel capitolo 3 di questa tesi, Object Target di Vuforia è molto robusto e affidabile. Tuttavia, in un "contesto reale", quest'ultimo presenta delle limitazioni; infatti, se l'oggetto target è inquadrato dalla camera da un punto di vista differente da quello da cui è stato "scansionato", il riconoscimento del bersaglio è molto difficile. Ciò avviene poichè l'algoritmo di feature selection utilizzato acquisisce le caratteristiche del target solo per quell'angolazione. Altre caratteristiche, sebbene siano catturate, risultano meno in risalto.



Figura 4.2.1: Scansione dell'oggetto target da più angolazioni

A tal proposito si è pensato di scansionare il medesimo bersaglio da più punti di vista (Fig. 4.2.1.) permettendo così al tracker di Vuforia di riconoscere la scena con maggiore tempestività.

L'idea implementata è schematizzata nel flow chart mostrato in figura 4.2.2.



L'aumentazione è mostrata dal tracker che ha riconosciuto l'oggetto target per primo

Figura 4.2.2: Flow chart dell'algoritmo multimarker-less implementato. Si noti come vi siano tanti tracker quante sono le "scansioni" effettuate dell'oggetto target; questi ultimi accedono in maniera esclusiva ad una risorsa condivisa in comune.

4.3 OGGETTI TARGET UTILIZZATI PER LA MODELLAZIONE DELLA SCENA

NA

La scena AR modellata si compone di:

- una scheda madre per computer "scannerizzata" da 5 posizioni differenti. (Fig. 4.2.1.).
- Due banchetti di ram, uno sulla scheda madre ed un altro posizionato sul piano di lavoro, scansionati da un verso e dall'altro.
- Una scheda VGA.
- Una scheda ethernet.
- Una CPU.

Gli ultimi 3 oggetti sono stati acquisiti da un'unica angolazione perchè, normalmente, sono sempre orientati nella stessa direzione.

Tutti i target citati sono mostrati nell'immagine 5.0.1.

4.4 UNITY3D E VUFORIA: MODELLAZIONE DI UNA SCENA AR

Il codice scritto per questo progetto di tesi è stato redatto seguendo le best practices suggerite per i linguaggi orientati agli oggetti; pertanto, si è cercato di minimizzare l'accoppiamento delle classi e massimizzare la coesione di quest'ultime.

A tal proposito, per ogni funzionalità è stato realizzato il rispettivo "Manager"; inoltre, la gestione della UI è stata sempre separata dalla parte funzionale del programma.

La struttura generale del progetto Unity3D realizzato è mostrata nella figura 4.4.1; di seguito è invece riportata la descrizione di tutti i "Manager" realizzati:

ContextManager: è il cuore del programma attraverso il quale è orchestrata la procedura descritta nel flow chart 4.2.2. Esso presenta, per ogni oggetto 3D

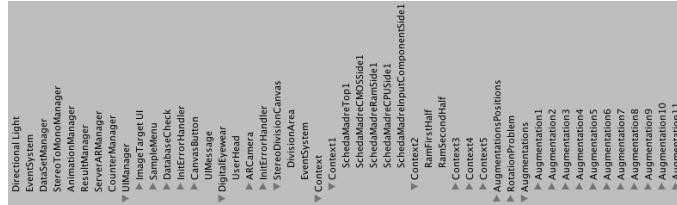


Figura 4.4.1: Struttura del progetto Unity3D realizzato

da tracciare, un GameObject contenente lo script DynamicDatasetLoader, il cui compito è di gestire l'avvio e lo stop del dataset di ciascun target scansionato e l'inizializzazione di tutti i trackers. Questi sono degli oggetti creati dinamicamente a cui è associato lo script MyTrackableEventHandler, responsabile di mostrare/disattivare le aumentazioni. In particolare, MyTrackableEventHandler (presentante tante istanze quanti sono i target) esegue i seguenti compiti:

- Quando un particolare oggetto è tracciato preleva gli oggetti 3D dal contenitore "Augmentations", contenente le aumentazioni di tutte le scene.
 - In accordo alle informazioni contenute nel contenitore "Positions", inizializza l'aumentazione prelevata al passo precedente posizionandola sul target tracciato. È importante sottolineare infatti che ogni oggetto virtuale presenta, per ogni lato tracciato, una scala e una posizione differente.
 - Renderizza gli oggetti virtuali in modo che questi siano mostrati alla camera.
 - Setta a true la variabile statica booleana `isBlockGlobal[CURRENTSCENE]` che rende inaccessibile il metodo "OnTrackableStateChanged" (chiamato automaticamente da Vuforia ogni qual volta un target è riconosciuto) a tutti gli altri trackers.
 - Quando il target non è più tracciato riporta gli oggetti virtuali nel contenitore "Augmentations" e rende accessibile a tutti gli altri trackers il controllo delle aumentazioni.

Un altro script molto importante in ContextManager è LookComponent. Esso consente di forzare lo stop di un tracker quando questo diventa molto instabile;

infatti, se la camera passa da un lato all’altro del target (e il lato tracciato continua ad essere sempre lo stesso) l’aumentazione mostrata potrebbe essere leggermente distorta. Per ogni tracker è stato quindi fissato un threshold rispetto all’ottimo (la camera si trova nella stessa posizione in cui si trovava quando l’oggetto è stato scansionato); se questo è oltrepassato il tracker è interrotto. La presenza di altre scansioni assicura che l’oggetto 3D sarà riconosciuto dall’altro tracker; questo approccio migliora considerevolmente l’esperienza AR utente poichè le aumentazioni mostrate sono sempre precisamente allineate con il target.

DataSetManager: a runtime, consente di effettuare il loading e la disattivazione dei dataset target. A tal proposito è importante sottolineare come l’applicativo realizzato modelli 22 scene AR differenti.

StereoToMonoManager: consente di effettuare lo switching dalla modalità binoculare a quella monoculare e viceversa. In particolare la gestione della modalità “visualizzazione stereo” è stata effettuata utilizzando l’SDK Google Cardboard; quest’ultimo utilizza due camere separate per inquadrare la scena.

AnimationManger: consente di animare gli oggetti virtuali presenti nella scena. Gli script ad esso associati permettono di configurare dall’inspector le varie proprietà delle animazioni, tra cui: oggetti coinvolti, velocità con cui questi devono essere animati, ecc (Fig. 4.4.2.). Si è scelto di non utilizzare gli strumenti di animazione interni a Unity 3D perchè in futuro l’applicazione in questione potrebbe prelevare i dati per animare gli oggetti virtuali da remoto; tale operazione sarebbe stata più complessa se avesse utilizzato gli strumenti interni di Unity.

CounterManager: consente di gestire il passaggio da una scena AR all’altra.

UIManager: fornisce gli strumenti necessari per interagire con la UI; quest’ultima è stata implementata utilizzando Canvas e labels. Per approfondire le scelte progettuali inerenti alla UI riferirsi alla sezione 4.6.

ServerARManager: consente all’applicazione di essere ”telecomandata” da remoto, infatti, lo script associato al manager utilizza un listener Tcp per stabilire una connessione con il client (un’app Andddroid).

Result manager: è utilizzato per tracciare i risultati dell’analisi condotta; in particolare sono salvati in un file, frame per frame, le seguenti informazioni: scena

corrente, lato dell'oggetto riconosciuto, accelerazione della camera lungo gli assi, ecc.

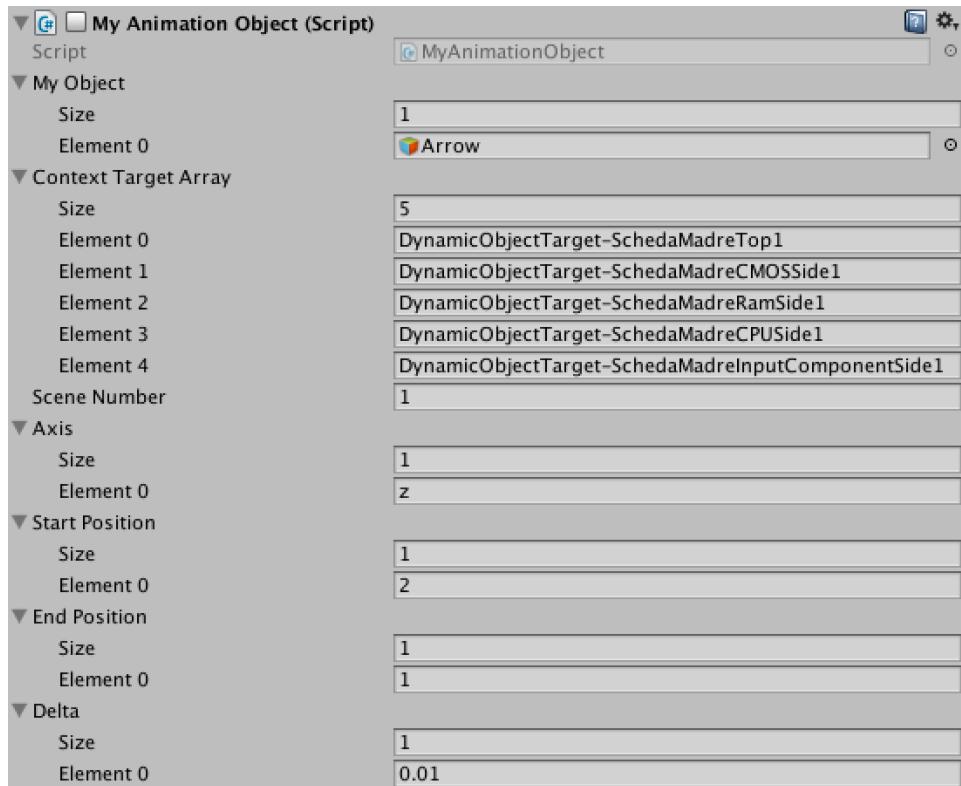


Figura 4.4.2: Parte dell'inspector dello script animante gli oggetti 3D virtuali. Nell'esempio mostrato, se il sistema è in un dato stato e "la camera visualizza una parte dell'oggetto scheda madre", lo script anima la freccia da Start Position fino ad End Position lungo l'asse z.

4.5 GESTIONE DELLE OCCLUSIONI

Quando si modella un'applicazione di realtà aumentata gli oggetti 3D virtuali dovrebbero essere mostrati solo quando non sono coperti da oggetti 3D reali; come mostrato nell'immagine (Fig. 4.5.1.), se non si utilizzano le giuste precauzioni, i contenuti aumentati sono mostrati anche quando non dovrebbero essere visibili.

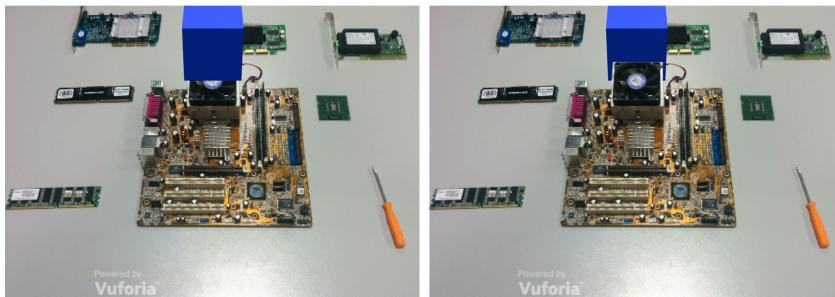


Figura 4.5.1: I contenuti aumentati dovrebbero essere mostrati soltanto se non sono occlusi da oggetti 3D reali. Nell'immagine a sinistra, il cubo virtuale dietro alla scheda madre è renderizzato "sopra" gli oggetti reali. Nell'immagine a destra la visualizzazione dei contenuti virtuali avviene correttamente.

Per questo motivo in tutte le scene AR realizzate in questo lavoro è stato necessario creare dei modelli di occlusione che permettessero di mascherare le parti degli oggetti reali sui quali, erroneamente, era mostrato il contenuto aumentatato. A quest'ultimi è stato aggiunto un particore Shader in grado di capire se l'oggetto virtuale era, rispetto al punto di vista dell'utente, posizionato "davanti" o "dietro" il target reale.

Per maggiori dettagli sull'utilizzo delle depth masking si rimanda l'utente alla pagina ufficiale di Vuforia.

4.6 USER INTERFACE

Il sistema AR implementato per questa tesi può essere utilizzato in modalità hand held display oppure attraverso un HMD see-through video. Entrambe le metodologie sono state testate e valutate; i risultati sono mostrati nel capitolo 5 di questo elaborato.

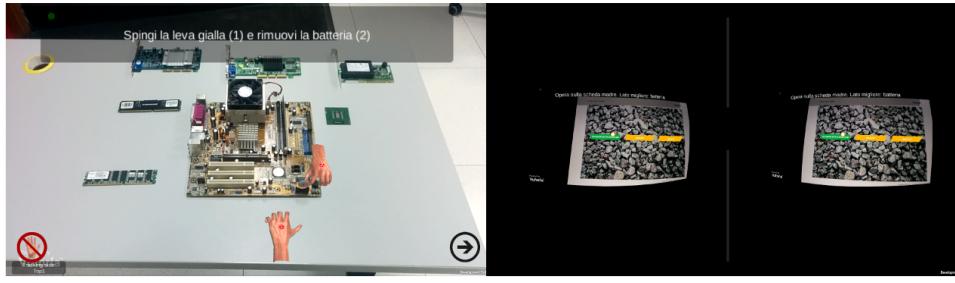


Figura 4.6.1: Vista utente con head hand display (immagine a sinistra) e HMD see-through video (immagine a destra). Nel secondo caso l’utente sta interagendo con gli strumenti della UI.

Per quanto concerne la modalità di visualizzazione hand held display, come mostrato nella figura 4.6.1 (parte sinistra) quest’ultima presenta due pulsanti: uno per spostarsi alla scena successiva ed un altro per non renderizzare le mani virtuali nell’aumentazione mostrata. L’utente, che ha in mano lo smartphone, può agire sui pulsanti della UI premendoli con i polpastrelli.

Se si indossa un HMD, invece, non si ha la possibilità di premere con le mani i pulsanti presenti sul display dell’HMD perché sono posti d’avanti al viso. Pertanto, per consentire all’utente di interagire con la UI sono state implementate e testate tre differenti soluzioni:

- Soluzione marker based utilizzante pulsanti virtuali: l’utente, occludendo una specifica parte di un marker può simulare la pressione di un pulsante fisico (Fig. 4.6.2.). Tale metodologia, presente nell’SDK Vuforia, può fornire molti falsi positivi e compromettere negativamente l’esperienza AR

dell’utente; per questo motivo tale sistema non è consigliato per sistemi AR critici.

- Soluzione gaze based utilizzante marcatori: l’utente, guardando in una specifica zona di un marker (per un tempo fissato) può simulare la pressione di un pulsante; questo sistema, molto utilizzato in giochi di sparatorie, usa strutture Unity3D quali Raycast per comprendere se la camera (in questo caso l’AR camera) collide con qualche oggetto virtuale nella scena. Un tale approccio, sebbene sia molto accurato, costringe l’utente a innumerevoli context switch dalla scena operativa alla UI.
- Soluzione magnet based: l’utente può simulare la pressione di un tasto agendo su una calamita posta al lato del visore; in particolare, il campo magnetico generata da quest’ultima può essere percepito dallo smartphone (dotato anch’esso di un magnete). Questa feature tuttavia non è presente in tutti i visori.

Per condurre alcuni degli esperimenti descritti nel capitolo 5 di questa elaborato, inoltre, è stato implementato un sistema in grado di telecomandare la soluzione HMD AR in base alle indicazioni dell’utente testato. Il lavoro svolto per questa tesi, infatti, non vuole studiare interfacce efficienti da utilizzare in applicazioni di realtà aumentata.

Per maggiori dettagli circa le modalità con le quali sono stati condotti i test dei sistemi proposti riferirsi al capitolo successivo.



Figura 4.6.2: Vuforia virtual button

4.7 COMPARAZIONE ALGORITMI SINGLE SCAN VS MULTI SCAN

In questa sezione si discuterà dei risultati sperimentali ottenuti confrontando il sistema marker based single scan con quello multi scan. Come è stato più volte sottolineato, la seconda soluzione apporta sostanzialmente due miglioramenti alla prima, in particolare:

Le aumentazioni mostrate sono maggiormente allineate e poco distorte, infatti, il target è tracciato sempre in condizioni favorevoli.

Un sistema multi scan permette di effettuare il tracking del target per un periodo più lungo. In altre parole, se si utilizza il sistema multi scan, i lassi di tempo in cui il target non è tracciato sono ridotti. Per stimare statisticamente l'entità dei miglioramenti si è proceduto come segue: è stato realizzato un video di 15 minuti che ha ripreso il target da tutte le angolazioni; in due round, utilizzando una camera virtuale (Cam Twist), è stato dato in pasto il video ottenuto alle due soluzioni da confrontare. Come si evince dalla figura 4.7.1, se si utilizza una sola scansione il target è tracciato per meno tempo; i punti neri (l'algoritmo non ha tracciato il target) sono 2.21 volte superiori nel plot "One side single scan". Nel plot "More side scan algorithm" si vede inoltre come le "mancanze" di "Top side" (scansione del target dall'alto) siano compensate dagli altri trackers disponibili.

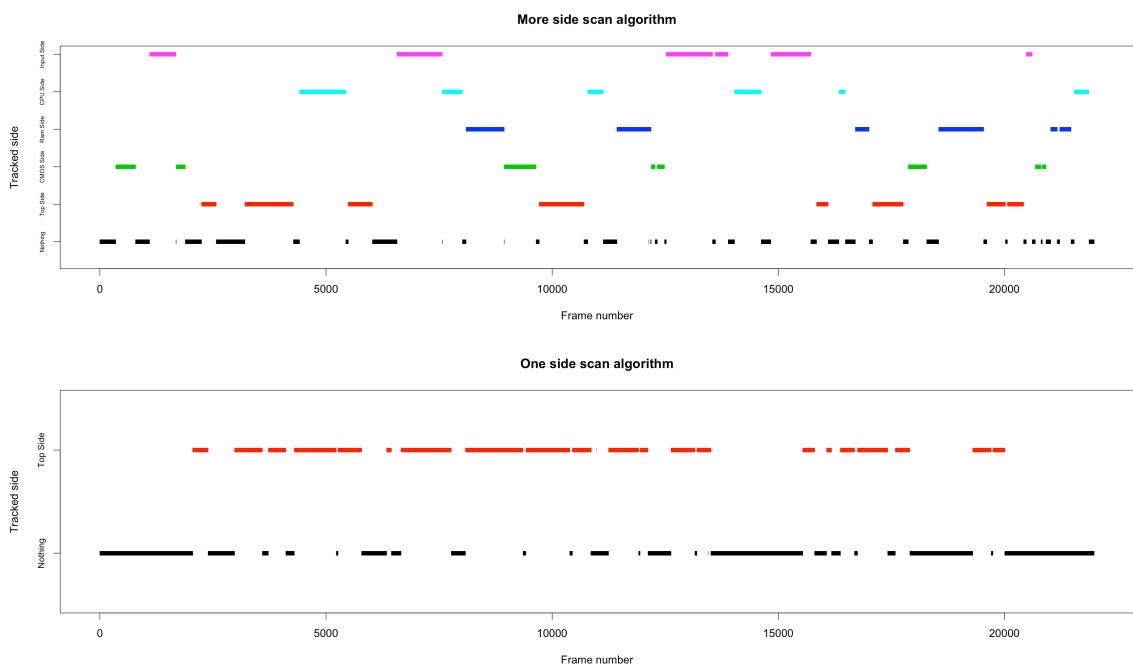


Figura 4.7.1: Algoritmo con più scansioni VS algoritmo con singola scansione

5

Sperimentazione: manuale tradizionale VS sistemi AR

In questa sezione si discuterà dei risultati sperimentali ottenuti confrontando le soluzioni hand held display AR e HMD AR presentate nel capitolo precedente con un sistema di manutenzione tradizionale: un manuale cartaceo. Quest'ultimo è stato redatto utilizzando le medesime istruzioni testuali e i medesimi modelli 3D utilizzati nei sistemi AR; nel manuale cartaceo, tuttavia, gli oggetti 3D non sono animati.

L'ambiente di test è presentato nell'immagine 5.0.1 mentre le operazioni da svolgere sulle varie componenti sono riportate nell'appendice A di questo lavoro. La sperimentazione effettuata è stata condotta seguendo le guide lines presentate in [110].

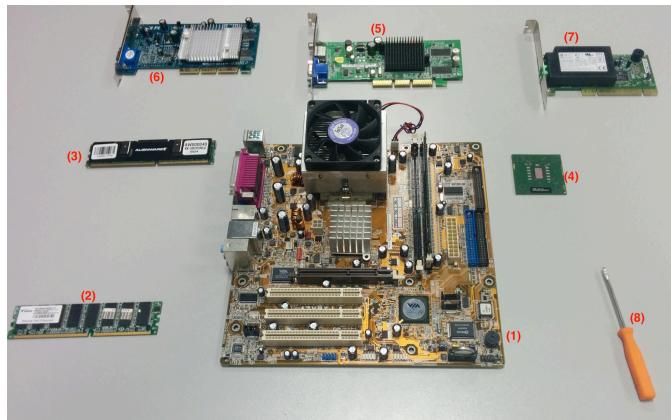


Figura 5.0.1: Assemblaggio delle componenti di una scheda madre (1). Oggetti riconosciuti dal sistema AR implementato: Ram (2), CPU (4), scheda VGA (5), scheda ethernet (7). Altri oggetti utilizzati per confondere l'utente: Ram (3), scheda VGA (6). Il cacciavite (8), invece, è stato utilizzato per compiere alcune operazioni di manutenzione richieste.

5.1 VARIABLES SELECTION E HIPOTHESIS FORMULATION

L'obiettivo di questo lavoro è fornire una risposta alle seguenti domande: un sistema di manutenzione basato sulla realtà aumentata può apportare dei benefici nell'espletamento di una procedura di assemblaggio/disassemblaggio di componenti? In che misura è migliore/peggiore rispetto ad un sistema di manutenzione tradizionale (in questo caso un manuale cartaceo)?

Le variabili indipendenti, quindi, sono i tre differenti sistemi di manutenzione:

- Hand held display AR
- HMD see-through video AR
- Manuale cartaceo tradizionale

Le variabili dipendenti, invece, sono:

- Il tempo impiegato per eseguire i task di manutenzione richiesti
- Il numero di errori commessi

- Il carico di lavoro (workload) soggettivo a cui l'utente è sottoposto durante l'espletamento della procedura

Per quanto concerne la prima variabile dipendente oggetto di studio il tempo è stato misurato con un cronometro ed espresso in secondi. Il numero di errori commessi, invece, è valutato per ogni task in accordo al template presentato nell'appendice B di questo lavoro; in particolare ogni task può essere suddiviso in più sottotask, pertanto, se l'utente ha completato la procedura parzialmente (ad esempio scena 20) oppure non ha seguito esattamente le istruzioni proposte da sistema (ad esempio scena 7), il punteggio assegnato è inferiore ad 1. Il carico di lavoro soggettivo, infine, utilizza i risultati del questionario NASA TLX (riferirsi alla sezione 5.1.1 per maggiori dettagli).

Formalmente le ipotesi nulle da invalidare sono le seguenti:

- Ipotesi nulla H_0 . Non c'è alcuna differenza nel tempo impiegato a completare una procedura di manutenzione se l'utente utilizza un sistema cartaceo tradizionale, una soluzione AR hand held display o una soluzione AR basata su HMD see-through video.

$$H_0: Error(ARHHD)=Error(ARHMD)=Error(PaperManual).$$

Ipotesi alternativa: $Error(ARHHD) \neq Error(ARHMD) \neq Error(PaperManual)$.

- Ipotesi nulla H_1 Non c'è alcuna differenza nel numero di errori commessi durante i tasks se l'utente utilizza un sistema cartaceo tradizionale, una soluzione AR hand held display o una soluzione AR basata su HMD see-through video.

$$H_1: Time(ARHHD)=Time(ARHMD)=Time(PaperManual).$$

Ipotesi alternativa: $Time(ARHHD) \neq Time(ARHMD) \neq Time(PaperManual)$.

- Ipotesi nulla H_2 . Non c'è alcuna differenza nel carico di lavoro soggettivo richiesto se l'utente utilizza un sistema cartaceo tradizionale, una soluzione AR hand held display o una soluzione AR basata su HMD see-through video.

$H_2: Workload(ARHHD) = workload(ARHMD) = workload(PaperManual)$.

Ipotesi alternativa: $workload(ARHHD) \neq workload(ARHMD) \neq workload(PaperManual)$.

5.1.1 NASA-TLX

Il NASA-TLX, sviluppato dalla National Aeronautics and Space Administration (NASA), è uno strumento di valutazione del carico di lavoro soggettivo richiesto per l'espletamento di un task [63]. È somministrato ai partecipanti sotto forma di questionario; all'utente è richiesto di autovalutare la sua prestazione considerando sei diverse metriche: richiesta mentale, richiesta fisica, richiesta temporale, prestazione, sforzo e frustrazione (Table. 5.1.1). La somministrazione prevede due fasi: nella prima, per ognuna delle sei metriche, l'utente valuta in una scala Likert che va da 1 a 20 il carico di lavoro richiesto dal task eseguito; nella seconda, invece, l'intervistato "pesa" l'importanza di ogni metrica scegliendo per 15 volte una delle due alternative proposte dal sistema; in poche parole l'utente sceglie quale metrica ha un maggior peso per il task eseguito. Ad esempio, tutti i questionari compilati per questo progetto hanno evidenziato, per tutti e i tre sistemi di manutenzione analizzati, come una procedura di assemblaggio di componenti di una scheda madre richieda un basso sforzo fisico; valori più alti sono mostrati, invece, per le altre cinque metriche del NASA-TLX.

Il risultato del questionario in questione fornisce un valore compreso nell'intervallo 0-100 dove 0 indica un carico di lavoro percepito molto basso e 100 molto alto.

In questo lavoro è stata utilizzata una versione web based del NASA-TLX scritta da Keith Vertanen [39].

5.2 CREAZIONE DEI GRUPPI DI TEST

L'analisi condotta è stata svolta in un quasi-experiment [110] [45] poichè il campione di soggetti che ha eseguito i test non è stato scelto in modo casuale; era formato, infatti, da meno di 20 persone e un'assegnazione randomica dei soggetti ad ogni gruppo avrebbe creato, molto probabilmente, degli squilibri tra le diverse

Nome	Estremi	Descrizione
RICHIESTA MENTALE - Mental Demand	Bassa/Alta	Quanta attività mentale e percettiva era richiesta (es., pensare, decidere, calcolare, ricordare, osservare, cercare, ecc.)? Il compito era facile o difficile, semplice o complesso, impegnativo o leggero?
RICHIESTA FISICA - Physical Demand	Bassa/Alta	Quanta attività fisica era richiesta (es. spingere, tirare, girare, controllare, attivare, ecc.)? Il compito era facile o impegnativo, lento o rapido, leggero o pesante, riposante o faticoso?
RICHIESTA TEMPORALE - Temporal Demand	Bassa/Alta	Quanta pressione temporale hai avvertito a causa della frequenza o del ritmo con cui i compiti, o le fasi del compito, si susseguivano? Il ritmo era lento e tranquillo o rapido e frenetico?
SFORZO - Effort	Basso/Alta	Quanto hai dovuto impegnarti (mentalmente e fisicamente) per raggiungere il tuo livello di prestazione?
PRESTAZIONE - Performance	Buona/scarsa	Quanto pensi di aver raggiunto gli obiettivi del compito stabiliti dallo sperimentatore (o da te stesso)? Quanto sei soddisfatto della tua prestazione nel raggiungere questi obiettivi?
LIVELLO DI FRUSTRAZIONE - Frustration	Basso/Alto	Durante il compito, quanto ti sei sentito incerto, scoraggiato, irritato, stressato e infastidito rispetto a sicuro, gratificato, appagato, rilassato e soddisfatto?

Tabella 5.1.1: Definizioni delle scale di valutazione del NASA-TLX

classi oggetto di studio. In particolare, prima di ogni esperimento, è stato sottoposto ad ogni candidato un questionario in modo da quantificare il suo livello di confidenza con le componenti di una scheda madre; in base al punteggio ottenuto, il soggetto intervistato è stato quindi assegnato ad un gruppo piuttosto che ad un altro.

Pertanto, i gruppi creati, ognuno formato da 6 persone, presentano le seguenti proprietà:

- Tutti i soggetti partecipanti all'esperimento hanno un'età compresa tra i 20 e i 30 anni e sono tutti studenti di discipline scientifiche all'università degli studi di Salerno.
- Ogni gruppo è formato in egual misura da donne e uomini; è infatti comunemente noto come le donne sia normalmente meno brave degli uomini in attività manuali di assemblaggio.
- Ogni gruppo presenta in "egual" misura da persone con molta esperienza (valori 5-6), con media esperienza (valori 3-4) e con poca o nulla esperienza (valori 0-2) nell'assemblaggio delle componenti di una scheda madre. A tal proposito nella tabella 5.2.1 sono riportate, per ogni gruppo creato, le statistiche per la metrica "Quanto sei bravo ad assemblare i componenti di una scheda madre?". Come si evince da quest'ultima, la differenza tra le medie e le deviazioni standard dei punteggi ottenuti in ogni gruppo è sempre inferiore a 0.5. Per maggiori dettagli circa le scale utilizzate nei questionari riferirsi all'appendice B di questa tesi.

Infine, per quanto concerne la variabile "Quando assembli un componente segui le istruzioni?" si è cercato di gestire la variabilità di tale fattore invitando ogni utente a seguire alla lettera le istruzioni presentate dal sistema di manutenzione utilizzato nel test. Come descritto nel paragrafo successivo, eventuali "leggerezze" dell'utente sono state corrette durante l'esecuzione dei test 1 e 2 dell'esperimento; i risultati ottenuti in tali task, quindi, non sono stati inseriti nelle statistiche finali.

Method	Mean	Standard deviation
Manuale cartaceo	3.4	1.14
AR handheld display	3.3	1.63
AR HMD	3.16	1.17

Tabella 5.2.1: Experiece factory

5.3 DESCRIZIONE DEGLI ESPERIMENTI

Tutti gli esperimenti sono stati condotti nel medesimo luogo e con le medesime condizioni di luce artificiale. Entrambi i sistemi AR realizzati utilizzano un Nexus 5 con fotocamera da 8.0 megapixel; la soluzione HMD AR utilizza inoltre il VR box presentato nell'immagine 2.6.1. Per evitare il surriscaldamento eccessivo dello smartphone, ogni test è stato diviso in due parti: nella prima sono stati eseguiti i primi 10 task e nella seconda i rimanenti. Dal testing del sistema AR realizzato, infatti, è emerso che la qualità del tracking è inversamente proporzionale al surriscaldamento del dispositivo; normalmente i primi lags cominciano a manifestarsi dopo una ventina di minuti dal lancio dell'applicativo. Ovviamente, per una questione di equità, anche i test effettuati con il manuale tradizionale sono stati divisi in due parti.

Il cambio di scena, per ognuno dei sistemi testati è stato gestito così come descritto di seguito:

- Manuale tradizionale: l'utente, dopo aver completato un task fa cenno all'operatore di voler eseguire il task successivo. Poi, autonomamente, consulta il manuale riportante le informazioni per eseguire il task successivo.
- Hand held display: l'utente, quando ha completato un task, autonomamente clicca il pulsante "Next scene".
- HMD AR: l'utente, dopo aver completato un task, fa cenno all'operatore di voler eseguire il task successivo. L'operatore, utilizzando un'applicazione che si collega al sistema AR (vedi sezione 4.6), fa avanzare lo scene counter. L'utente può inoltre chiedere all'operatore di nascondere/mostrare l'aumentazione; in molti casi, infatti, gli oggetti virtuali occludono completamente gli oggetti reali.

Ciò premesso, ad ogni partecipante all'esperimento è stata spiegata la motivazione per la quale si stava conducendo il suddetto test; successivamente, è stato

mostrato all’utente il funzionamento del sistema proposto e l’utilizzo della UI. Pertanto, per tutti treatment , le scene 1-2 (vedi appendice A) non sono state oggetto di studio; quest’ultime hanno consentito all’utente di familiarizzare con le interfacce del sistema. Ove necessario quindi sono stati corretti gli utilizzi impropri della soluzione da testare.

Infine ad ogni partecipante è stato detto di completare la procedura di manutenzione richiesta avendo cura di minimizzare il tempo di esecuzione e il numero di errori commessi.

5.4 RISULTATI DEGLI ESPERIMENTI

In questa sezione saranno descritti i risultati degli esperimenti condotti. L’elaborazione dei dati è stata effettuata con il software statistico R; quest’ultimo offre degli strumenti molto potenti per graficare i risultati e, inoltre, presenta moltissimi algoritmi di utilità (soprattutto statistici) già implementati. Tra quelli utilizzati in questo lavoro è inevitabile non citare:

L’analisi della varianza (Anova = Analysis of variance)[82][98][40]: è una procedura statistica che consente di verificare se ci siano delle differenze tra più gruppi di dati. L’ipotesi alla base di tale metodo è che dati G gruppi, sia possibile scomporre la varianza in due componenti: varianza interna ai gruppi (anche detta varianza Within) e varianza tra i gruppi (varianza Between). Informalmente, se i risultati osservati sono omogenei all’interno dello stesso gruppo ed eterogenei tra gruppi differenti è ragionevole affermare che le metodologie osservate sono differenti tra loro. L’Anova, implementata in R nella funzione *oav*[51], richiede che i dati dei vari gruppi in esame seguano la distribuzione gaussiana (ipotesi di normalità) e che la loro variabilità sia la medesima (ipotesi di omoschedasticità). In questa analisi, come suggerito in [40], è stato condotto il test di Shapiro e Wilk per verificare la normalità e il test di Barlett per verificare l’ipotesi di omoschedasticità; le funzioni R utilizzate per tali test sono rispettivamente *shapiro.test* e *bartlett.test*.

Metodo di Kruskal-Wallis[75]: è il corrispondente non parametrico dell’analisi della varianza ed è utilizzato per verificare l’uguaglianza delle mediane di diversi

gruppi; si usa, come tutti gli altri metodi non parametrici, quando non può essere assunta una distribuzione normale del campione osservato. In questo lavoro, nonostante i dati soddisfino l'ipotesi di normalità, si è scelto di confermare i risultati dell'ANOVA con tale metodologia; infatti, quando il campione a disposizione è molto piccolo (come in questo caso) è possibile che i risultati di un test parametrico non siano affidabili[40][85][67].

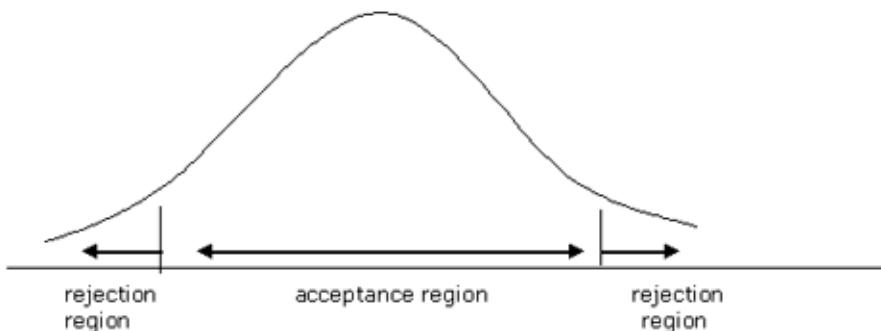


Figura 5.4.1: Alpha level: alpha level più piccoli riducono la grandezza della regione critica. Le ipotesi nulle formulate sono pertanto più difficili da invalidare.

Applicando le metodologie sopra descritte, se dal confronto dei risultati ottenuti si evince un livello di significatività (alpha level) minore o uguale a 0.05 (5 %) l'ipotesi nulla è rifiutata. In questo caso, ciò equivale a dire che almeno uno dei sistemi di manutenzione oggetto di studio si colloca nella regione critica: in media, i risultati delle misurazioni osservate sono "molto diverse" rispetto a quelle calcolate per gli altri metodi. Un alpha level fissato a 0.05 è stato utilizzato in [64] e in [105]; come in questo lavoro, anche nei suddetti, sono stati comparati differenti sistemi di manutenzione. Confronti più rigidi utilizzano invece un livello di significatività pari a 0.01.

In base alle ipotesi nulle formulate nella sezione 5.1, l'analisi condotta ha prodotto i risultati riportati di seguito.

5.4.1 IPOTESI NULLA H_0 , H_1 E H_2

Ipotesi nulla H_0 . Come riportato nel grafico 5.4.2 i soggetti che hanno utilizzato il manuale tradizionale hanno prodotto i tempi migliori e contemporaneamente i peggiori. In particolare il boxplot di colore rosso evidenzia una variabilità molto alta nei tempi medi di completamento: quest'ultimi sono compresi tra 35 secondi e 60 secondi. Infatti, gli utenti che hanno confidenza con una scheda madre e le sue componenti "pagano" un prezzo relativamente alto se utilizzano un sistema AR; le informazioni riportate nel manuale cartaceo sono sufficienti a comprendere dove operare e cosa fare. Gli utenti meno pratici, invece, traggono giovamento dalle informazioni aggiuntive che un sistema AR apporta. In generale la soluzione di manutenzione tradizionale risulta vincente per i task di localizzazione dei componenti esterni (scena 3, 11, 18 e 21); le schede da montare sulla scheda madre (RAM, VGA, ethernet), infatti, sono molto semplici da distinguere l'una dall'altra in quanto presentano caratteristiche distintive molto nette. I sistemi basati sull'AR richiedono invece che l'utente scansioni ogni componente esterno prima che il sistema gli suggerisca qual è quello coinvolto nel task corrente. Come per i task "semplici" (ad esempio scena 6), l'overhead da pagare con l'AR è troppo alto. Per i task complessi, invece, nei sistemi AR (soprattutto HHD AR) le animazioni suggeriscono all'utente come effettuare le operazioni di manutenzione richieste: ne conseguono tempi di completamento più brevi. Inoltre, come evidenziato nei boxplot verde e blu (Fig. 5.4.2) la variabilità interna ai gruppi AR è molto bassa. La media ottenuta per HHD è tuttavia simile (anche se leggermente più bassa) rispetto a quella ottenuta con il sistema di manutenzione tradizionale.

Per quanto riguarda i risultati ottenuti con i sistemi AR, la soluzione HHD produce tempi decisamente migliori rispetto a quelli ottenuti dagli esperimenti con l'HMD. L'utilizzo di quest'ultimo risulta molto invasivo e poco pratico: indossando un visore la maggior parte degli utenti ha lamentato una scarsa percezione dello spazio affermando che l'orientamento era più tattile che visivo; per completare i task più complessi, infatti, molti utenti hanno tolto il visore. Soluzioni HMD basate con acquisizione stereo avrebbero sicuramente prodotto risultati migliori; la

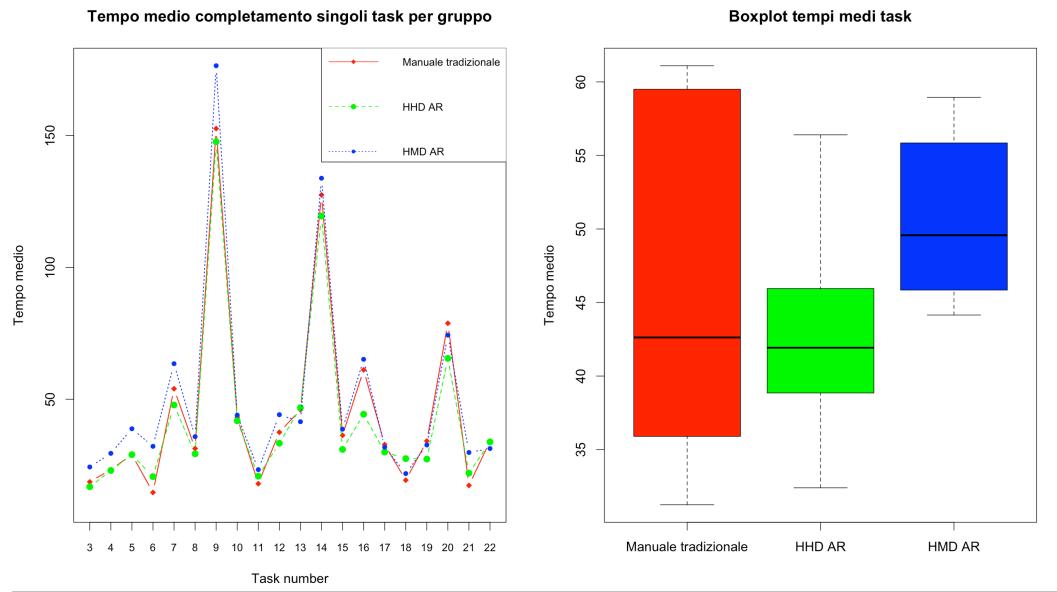


Figura 5.4.2: Tempi medi task: nel grafico a sinistra la media è calcolata, per ogni gruppo, per il medesimo task; nel secondo grafico la media è calcolata, per ogni gruppo, su tutti i task.

percezione della profondità sarebbe stata sicuramente migliore.

Alla luce delle osservazioni fatte, prima di procedere con il test dell'ANOVA, per ogni utente, si è proceduto ad effettuare la media dei tempi di completamento prodotti in ogni task. Fissando un alpha level a 0.05 come criterio di significatività, l'analisi della varianza non ha prodotto risultati statisticamente rilevanti:

c'è il 35% di probabilità che l'ipotesi nulla sia vera; sabbene risultino delle differenze tra le diverse metodologie di manutenzione quest'ultime non sono abbastanza forti da invalidare l'ipotesi nulla H_0 . Risultati simili sono stati ottenuti con il metodo di Kruskal-Wallis; in questo caso, il p-value ottenuto è 0.29.

Ipotesi nulla H_1 .

Come si evince dal grafico 5.4.3 molti dei task proposti erano molto semplici; indifferentemente dal metodo di manutenzione utilizzato, non sono stati commessi errori (task 3, 5, 8, ecc.). Per quanto concerne l'esecuzione dei task che richiedevano di effettuare le operazioni in un ordine prestabilito (scena 4, 7 e 16), invece,

le soluzioni AR hanno ottenuto i migliori risultati. Le aumentazioni, infatti, si spostano quando il punto di vista dell'utente cambia; questo facilita enormemente la compresione delle operazioni di manutenzione richieste. Utilizzando il manuale tradizionale, al contrario, alcuni utenti hanno avuto delle difficoltà nel comprendere quale fosse l'ordine giusto di esecuzione dei task; la scheda madre, infatti, non sempre era orientata nella posizione che presentava nel manuale tradizionale.

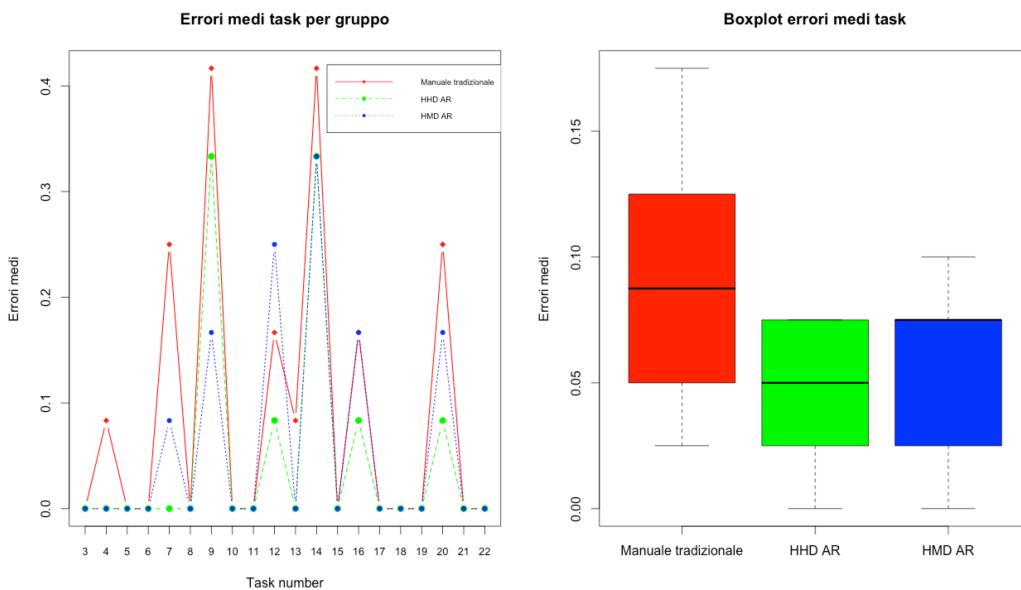


Figura 5.4.3: Errori medi task: nel grafico a sinistra la media è calcolata, per ogni gruppo, per il medesimo task; nel secondo grafico la media è calcolata, per ogni gruppo, su tutti i task.

Anche per quanto riguarda l'esecuzione di task complessi (scena 9, 14 e 20) le soluzioni AR based ottengono i risultati migliori. Come sottolineato precedentemente, le animazioni sovrapposte al target reale aiutano l'utente a comprendere come effettuare determinate operazioni, si pensi ad esempio alla scena 9: "Sgancia i fermi che bloccano il dissipatore. Aiutati con la mano (esercita pressione) e il cacciavite (spinge il fermo verso il basso (1) e lo ruota (2))". Senza l'ausilio della realtà aumentata alcuni utenti non sono stati in grado di completare il task in questione.

Risultati positivi a favore dell'AR sono stati ottenuti anche per il task 20; in particolare quest'ultimo richiedeva di agire su componenti molto piccole della scheda madre. Basandosi sulle informazioni contenute nel manuale tradizionale alcuni utenti hanno avuto delle difficoltà a completare la procedura di manutenzione proposta.

Alla luce delle osservazione fatte, prima di procedere con il test dell'ANOVA, per ogni utente, si è proceduto ad effettuare la media degli errori commessi in tutti task. Fissando un alpha level a 0.05 come criterio di significatività, l'analisi della varianza non ha prodotto risultati statisticamente rilevanti:

c'è il 21% di probabilità che l'ipotesi nulla sia vera; sabbene risultino delle differenze tra le diverse metodologie di manutenzione quest'ultime non sono abbastanza forti da invalidare l'ipotesi nulla H₁.

Risultati simili sono stati ottenuti con il metodo di Kruskal-Wallis; in questo caso, il p-value calcolato è 0.36.

Ipotesi nulla H₂

Come si evince dal grafico 5.4.4 il carico di lavoro soggettivo percepito dagli utenti, per tutti gli indicatori del NASA-TLX, è maggiore quando si utilizza un Head Mounted Display. Numerosi studi hanno evidenziato alcuni dei disturbi causati dall'utilizzo di visori see-through video [44][62][78], tra i quali: nausea, mal di testa, senso di smarrimento, ecc.; inoltre, questi sintomi possono essere accompagnati da disagi in termini di prestazioni, sia fisiche che mentali e persistere anche dopo l'utilizzo dell'HMD. Gli esperimenti condotti in questo lavoro hanno confermato tali risultati evidenziando come soluzioni see-through video non siano ancora mature per essere impiegate in ambienti operativi. Confrontando i risultati sperimentali ottenuti emerge:

- le soluzioni manuale tradizionale e HHD AR non differiscono in termini di sforzo percepito dall'utente; anche i pesi attribuiti ad ogni metrica sono prosciughi uguali. A tal proposito fare riferimento al valore di W su ogni barra del plot in figura 5.4.4 e alla seconda parte del NASA-TLX discusso nella sezione 5.1.1. Come detto in precedenza, invece, valori che si discostano molto dalla media sono stati ottenuti dalla soluzione HMD AR; in questo

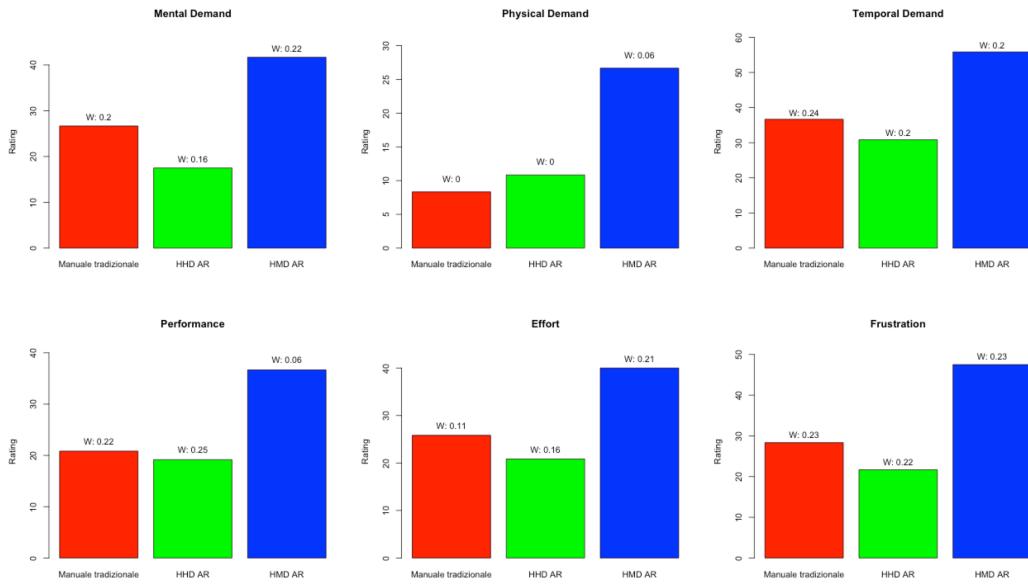


Figura 5.4.4: Workload complessivo misurato per ognuno dei sistemi di manutenzione oggetto di studio

caso i soggetti che hanno preso parte alla sperimentazione hanno attribuito poco valore alla metrica Performance.

- La metrica Physical demand ha un peso uguale a zero per il manuale tradizionale e la soluzione HHD AR; per quanto concerne il sistema HMD AR, invece, alcuni utenti hanno dato importanza a questo indicatore; infatti, dopo la sessione di esperimenti alcuni soggetti hanno avvertito un lieve mal di testa dopo aver completato i task proposti.

Alla luce delle osservazioni fatte, anche in questo caso, i tre sistemi di manutenzione sono stati confrontati attraverso l'analisi della varianza; fissando un alpha level uguale a 0.05 come criterio di significatività, in questo caso, sia l'ANOVA test che il Kruskal-Wallis test hanno prodotto risultati statisticamente rilevanti. I p-value risultanti sono rispettivamente 0.003 e 0.01, pertanto, è possibile rifiutare l'ipotesi nulla H_0 ed affermare: si evidenziano delle differenze nel carico di lavoro

soggettivo percepito tra un sistema di manutenzione tradizionale, una soluzione AR hand held display e una soluzione AR basata su HMD see-through video.

5.4.1.1 IPOTESI NULLA H₂: MULTIPLE COMPARISON

L'ANOVA e il Kruskal-Wallis test rispondono alla domanda: ci sono delle differenze significative tra le medie dei diversi gruppi osservati? Tuttavia tali metodi non forniscono alcuna informazione su come i dati si differenziano tra loro. A tal proposito, se l'ipotesi nulla è rigettata, bisogna procedere a un confronto a coppie tra i diversi gruppi oggetti di studio, i cosiddetti post-hoc experiments [46].

Come consigliato in [40] si è utilizzato "il test delle differenze oneste di Tukey", implementato nel pacchetto *multicomp* in R. La funzione utilizzata (*glht*) effettua un test di ipotesi lineare generale rispetto ad una funzione lineare di interesse *lincfct*, che si ottiene dalla matrice dei contrasti parametrici *mcp* calcolata sui tre fattori utilizzati per questo confronto. I confronti multipli hanno prodotto i seguenti risultati:

- Non vi è alcuna differenza statisticamente significativa tra il workload misurato per il manuale cartaceo e quello misurato per la soluzione AR HHD. Il p-value risultante è, infatti, 0.54.
- C'è una differenza statisticamente significativa tra il workload misurato con il manuale cartaceo e quello misurato per la soluzione AR HMD. Il p-value risultante è, infatti, 0.02.
- C'è una differenza statisticamente significativa tra il workload misurato con per il sistema AR HHD e quello misurato per la soluzione AR HMD. Il p-value risultante è, infatti, 0.003.

Le differenze tra i diversi gruppi oggetto di studio è inoltre evidenziata graficamente nel plot mostrato di seguito. Solo il confronto tra i valori misurati per il manuale cartaceo e il sistema AR HHD "intersecano" il centro della funzione lineare utilizzata.

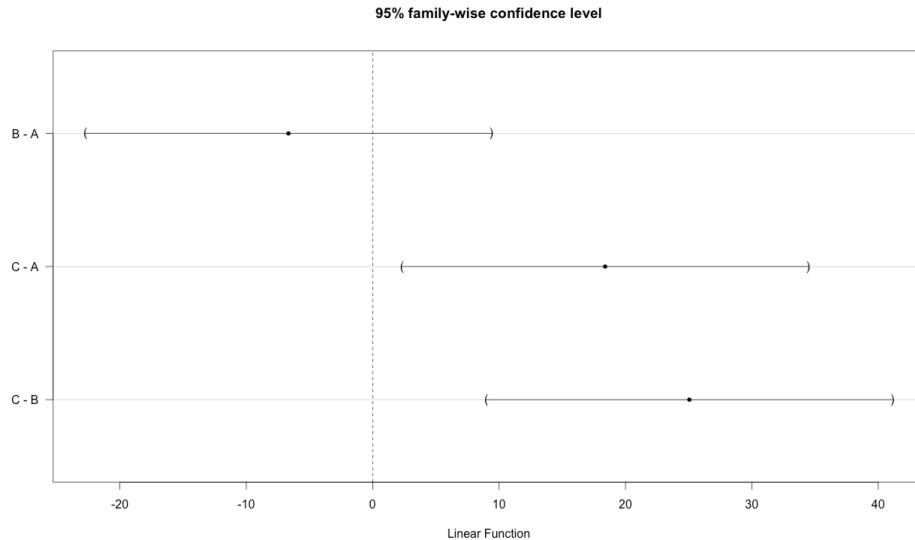


Figura 5.4.5: 95% family-wise confidence level NASA-TLX: Il manuale cartaceo presenta la label "A", il sistema AR HHD la label "B" e il sistema AR HMD la label "C".

5.4.2 VALUTAZIONE DEI QUESTIONARI POST EXPERIMENT

Al termine degli esperimenti è stato chiesto ad ogni utente di compilare un questionario per capire cosa ne pensasse del sistema di manutenzione utilizzato durante il test.

Agli utenti che avevano condotto l'esperimento con il manuale cartaceo sono state inoltre mostrate la soluzioni AR implementate.

L'analisi dei questionari ha prodotto i risultati schematizzati in figura 5.4.6.

Ulteriori risultati tabellari sono riportati nell'appendice B di questo elaborato.

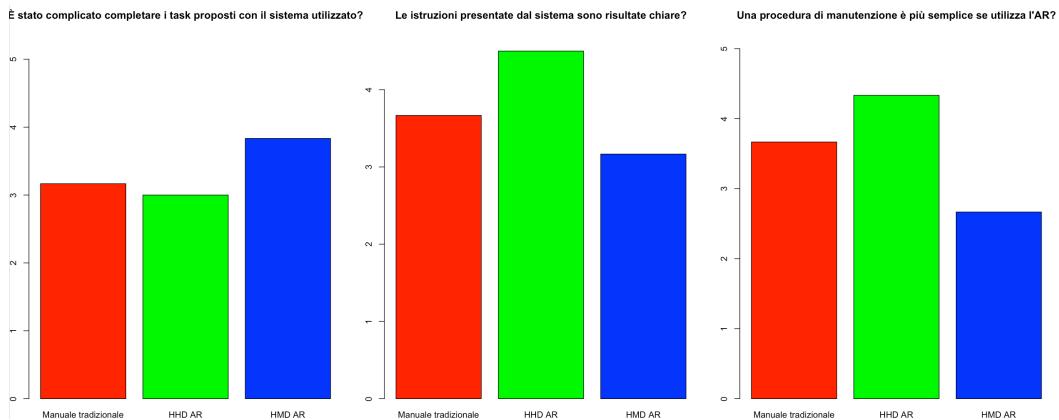


Figura 5.4.6: Livello di gradimento del sistema di manutenzione utilizzato e dell'AR. In quest'ultimo caso, anche gli utenti che hanno utilizzato il manuale cartaceo hanno espresso il proprio giudizio sui sistemi di realtà aumentata.

6

Conclusioni

Dai risultati sperimentali è emerso come gli utenti valutino positivamente la soluzione AR hand held display; anche se i risultati ottenuti non sono statisticamente rilevanti, i test condotti hanno mostrato che il sistema AR in questione possa apportare dei benefici positivi nell'espletamento di alcune procedure di manutenzione. Alcuni task, difficili da comprendere con il manuale cartaceo, sono eseguiti in tempi più brevi e con meno errori quando è utilizzata una soluzione AR di tipo hand held display.

Il prototipo AR utilizzante un Head Mounted Display, invece, non è stato accolto positivamente. Dagli esperimenti condotti è emerso che, non solo tale sistema non apporta benefici positivi nell'esecuzione di operazioni di manutenzione, ma aumentata considerevolmente il workload percepito dagli utenti. Utilizzando un HMD di qualità superiore, in accordo ai risultati presentati in [64] e in [105] si sarebbero ottenuti sicuramente delle statistiche migliori. In questi lavori tuttavia, i

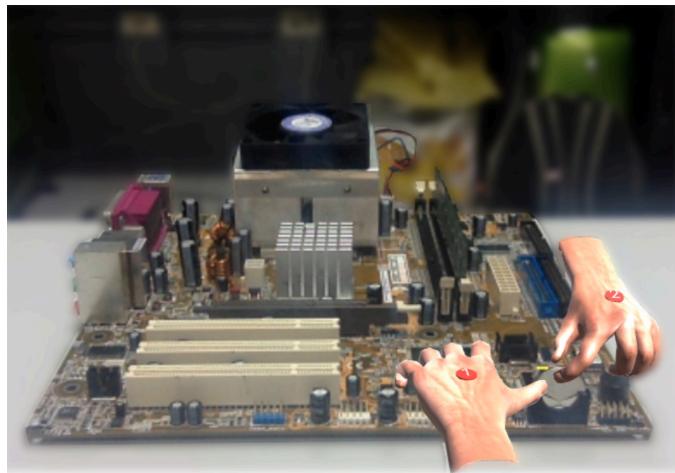
sistemi di manutenzione tradizionali non sono stati confrontati con soluzioni AR HHD. Ulteriori studi sono necessari per comprendere se, rispetto a sistemi HMD AR di fascia non consumer (sia soluzioni ottiche che video see-through), i sistemi HHD siano ancora la soluzioni vincenti.

Concludendo, si è fortemente convinti che al crescere della potenza di calcolo dei dispositivi mobile e dei personal computer le soluzioni marker-less dense (come ad esempio DTAM e LSD-SLAM) possano essere impiegate efficientemente in applicazioni di realtà aumentata in real time.

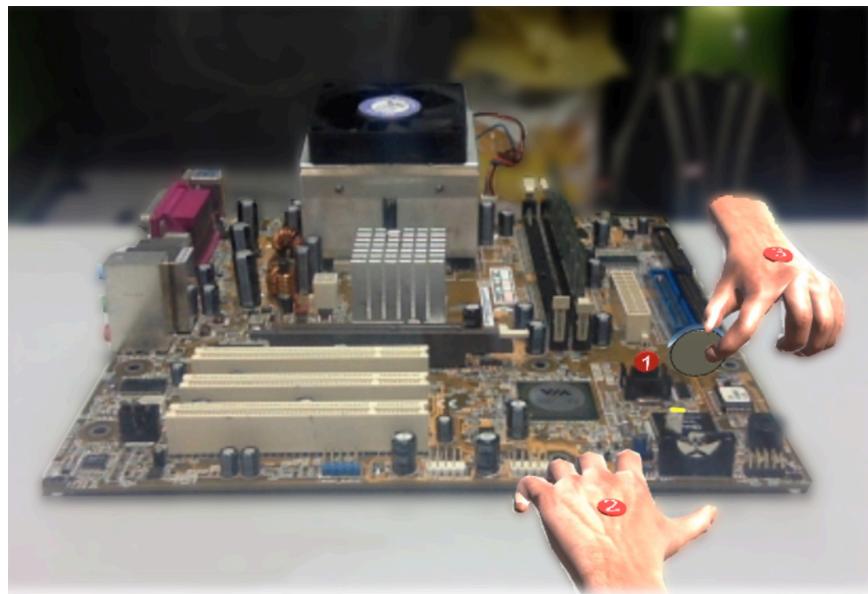
A

Manuale cartaceo

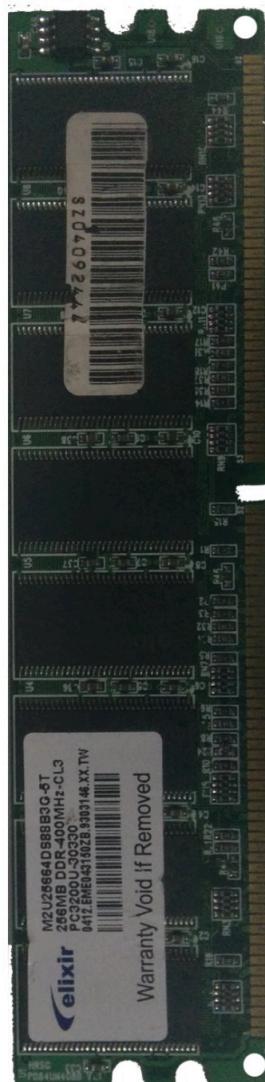
Scena 1: spingi la leva gialla (1) e rimuovi la batteria (2).



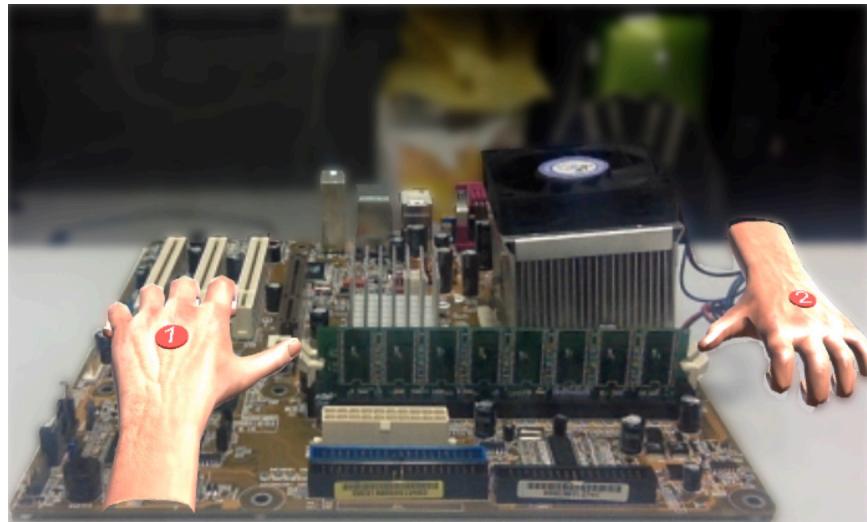
Scena 2: ruota la batteria rimossa al passo precedente (1), spingi la leva gialla (2), reinserisci la batteria nello slot (3). Attenzione al verso di reinserimento!



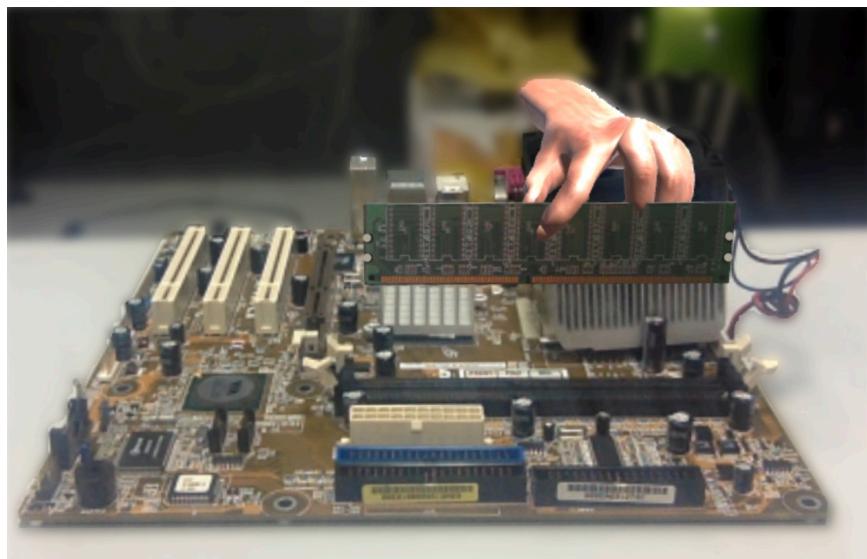
Scena 3: individua la RAM, ti servirà successivamente. Quindi vai alla prossima scena.



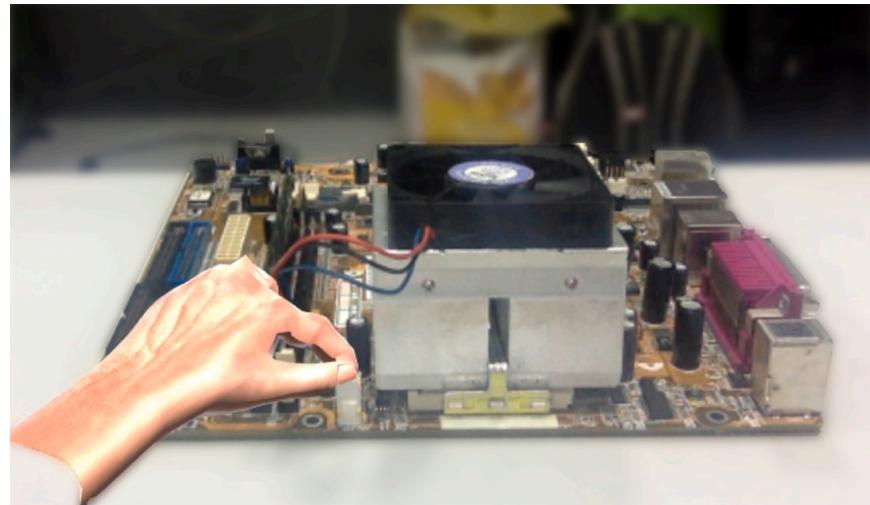
Scena 4: spingi le leve nell'ordine indicato.



Scena 5: Rimuovi la RAM nello slot sulla scheda madre (1). Poi, prendi la RAM suggerita dal sistema precedentemente e montala nello slot da cui hai rimosso l'altra RAM (2)



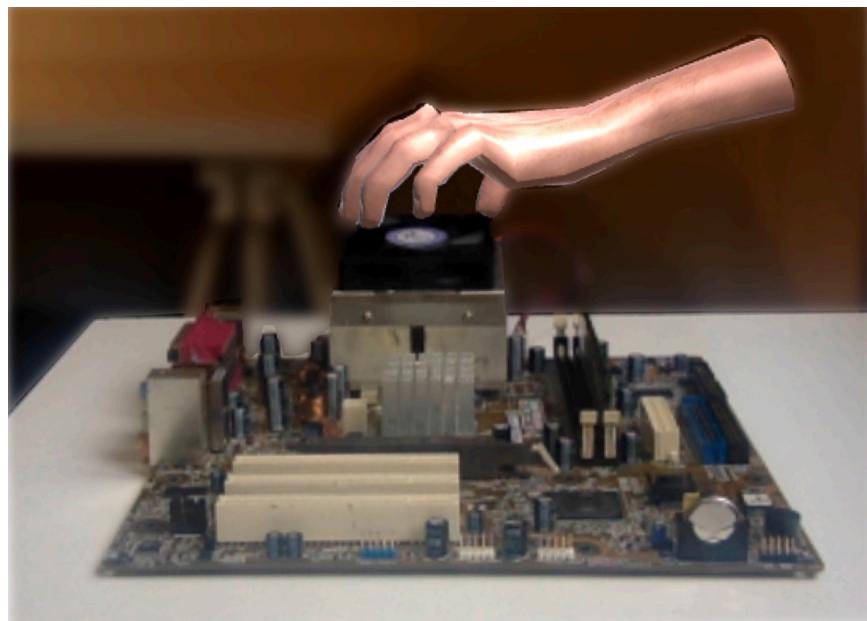
Scena 6: rimuovi lo spinotto.



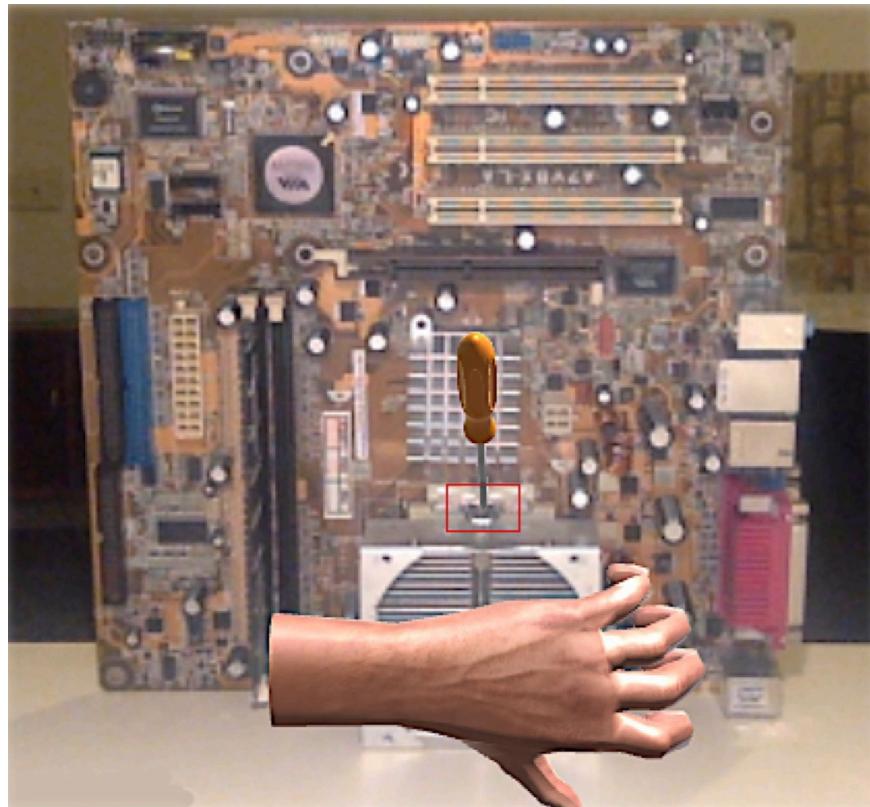
Scena 7: svita le viti nell'ordine indicato.



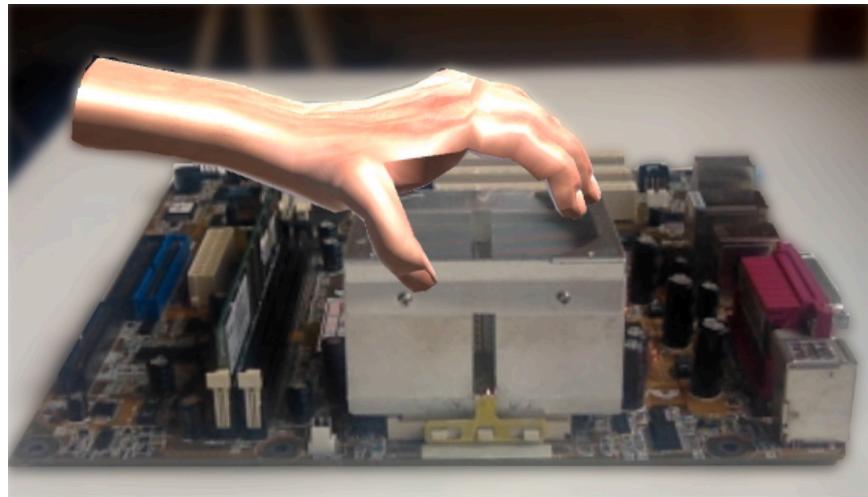
Scena 8: rimuovi la ventola.



Scena 9: Sgancia i fermi che bloccano il dissipatore. Aiutati con la mano (esercita pressione) e il cacciavite (spinge il fermo verso il basso (1) e lo ruota (2)).

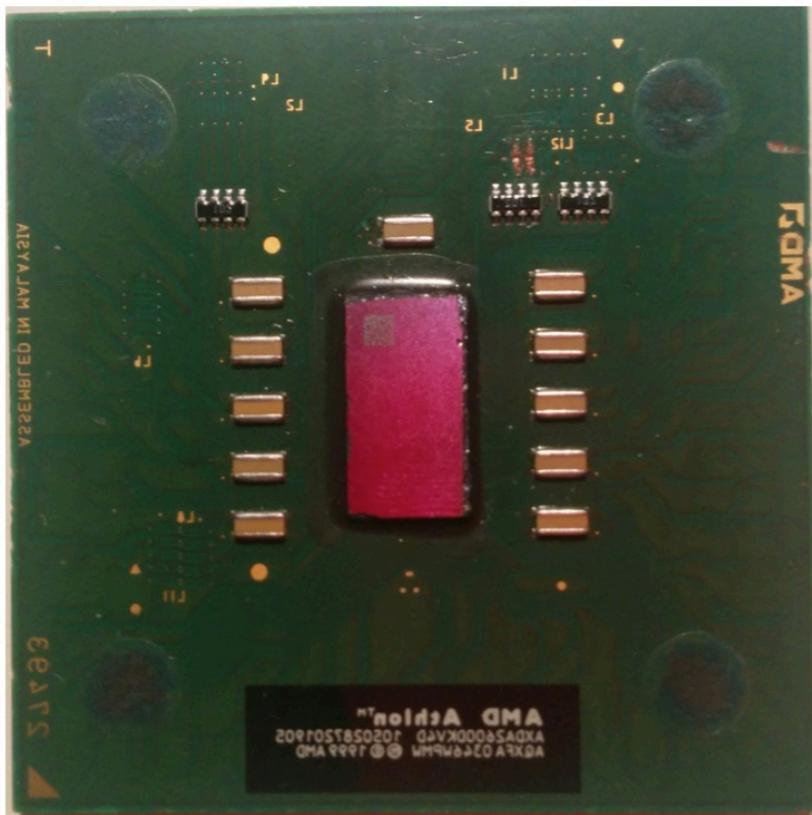


Scena 10: ruota leggermente il dissipatore (1) così da sganciare il fermo esterno (evidenziato in giallo). Poi rimuovi il dissipatore (2).

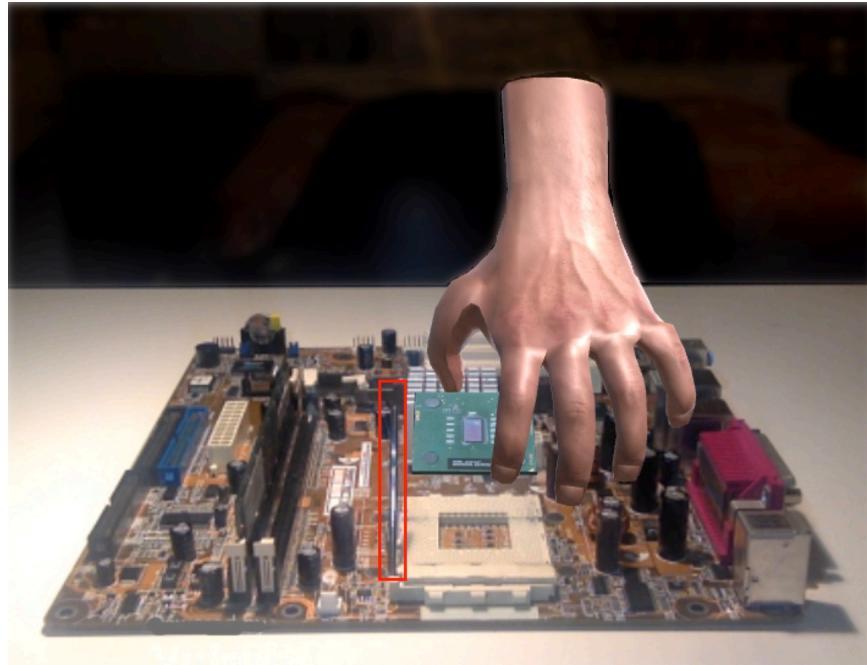


Scena 11: individua la CPU suggerita dal sistema (ti servirà successivamente).

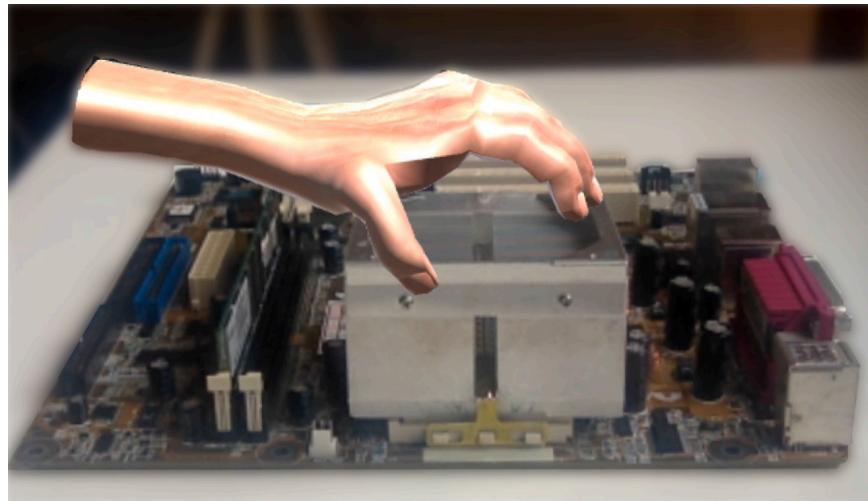
Quindi vai alla prossima scena.



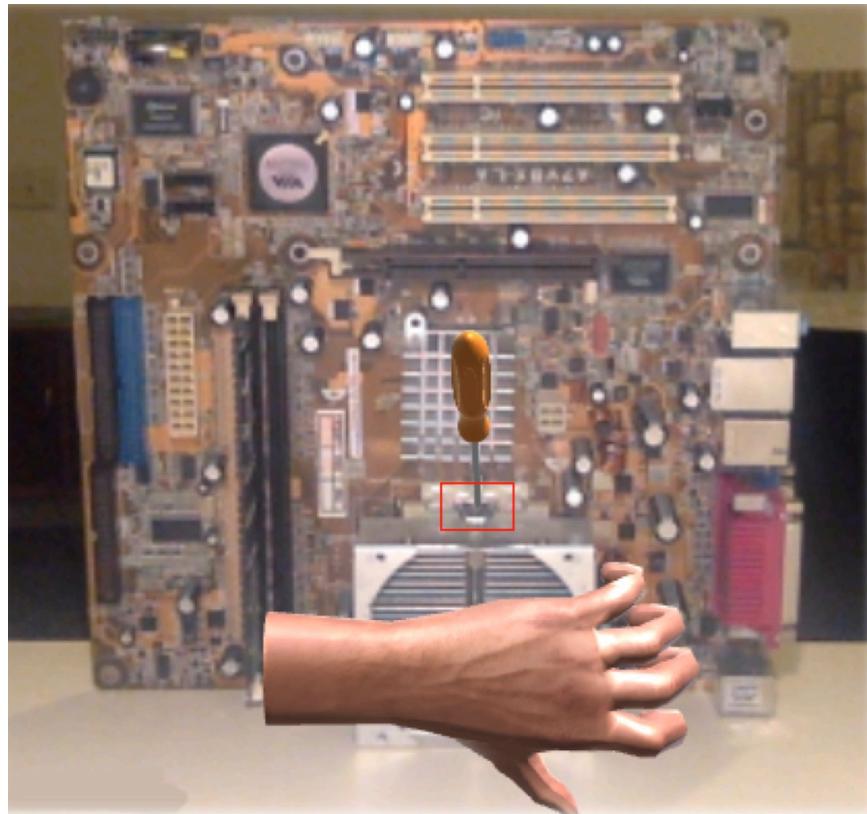
Scena 12: porta la leva verso l'alto (1) e inserisci la CPU presa precedentemente (2). Poi, riporta la leva verso il basso (3). Attenzione: la CPU non si aggancia, però, assicurati che sia centrata nello slot.



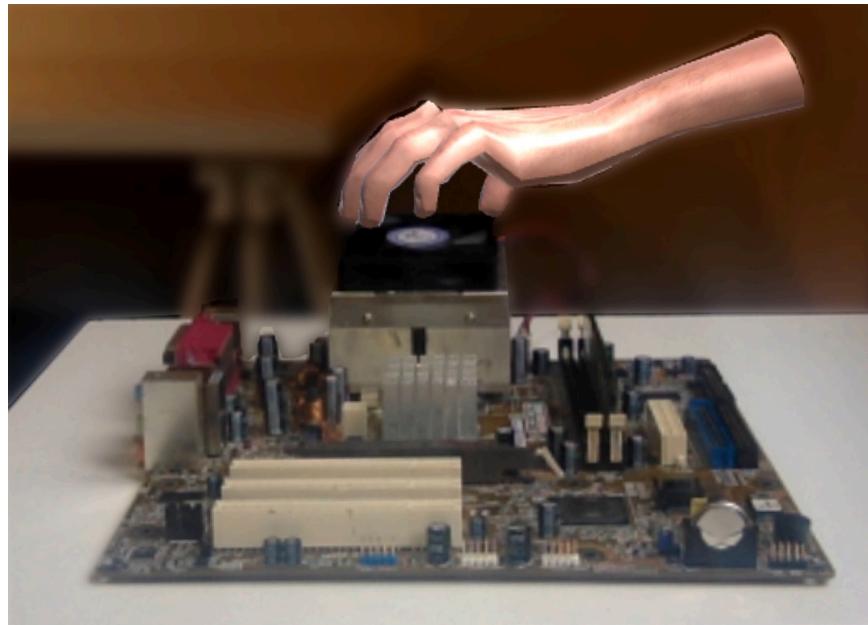
Scena 13: porta il dissipatore nella posizione indicata (1). Poi ruotalo leggermente in modo che il fermo (evidenziato in giallo) si agganci (2).



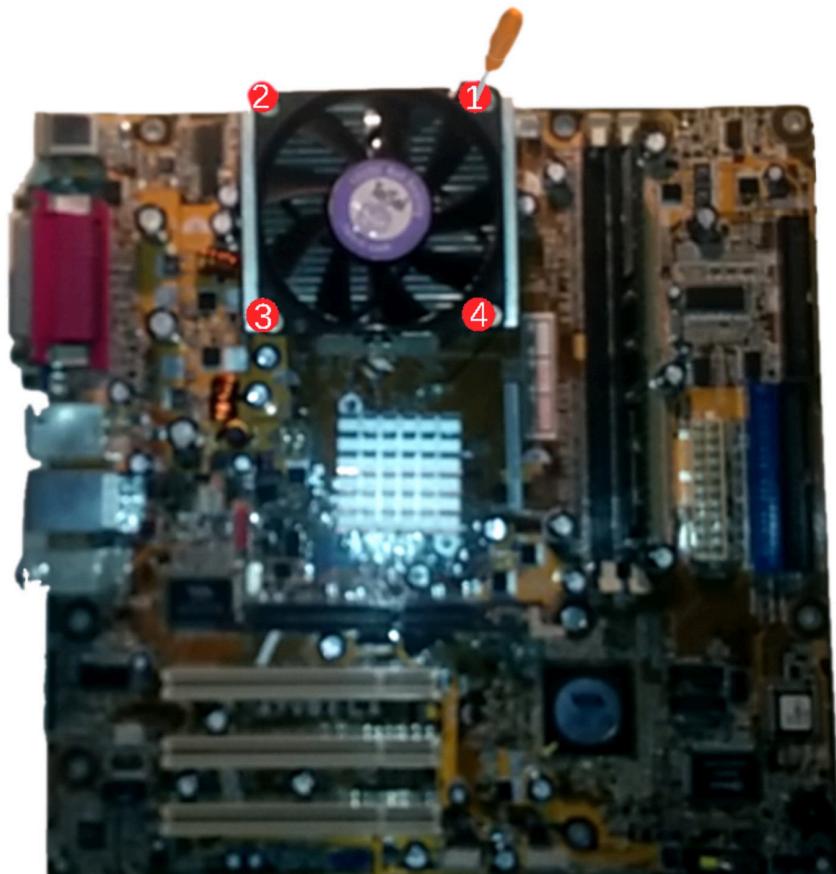
Scena 14: fissa il dissipatore sulla scheda. Aiutati con la mano (esercita pressione) e il cacciavite (spinge il fermo verso il basso (1) e lo ruota (2)).



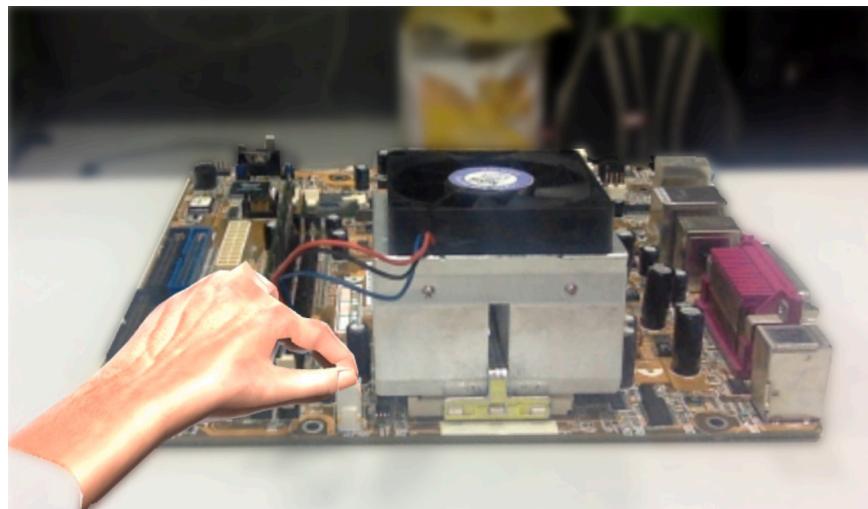
Scena 15: porta la ventola nella posizione indicata.



Scena 16: avvita le viti nell'ordine indicato.



Scena 17: inserisci lo spinotto. Attenzione: le sporgenze gialle sono rivolte all'esterno della scheda.



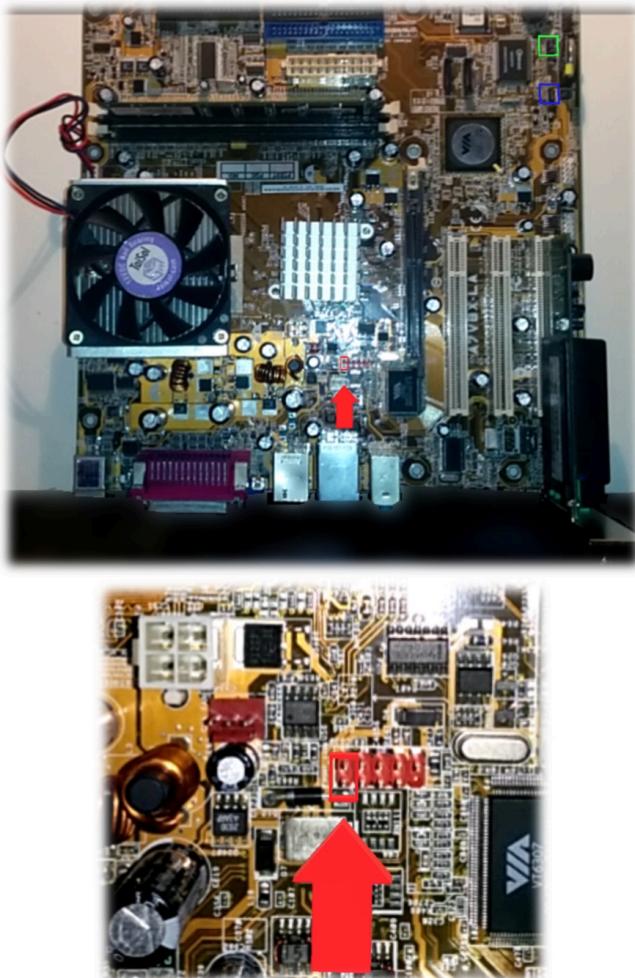
Scena 18: individua la scheda ethernet suggerita dal sistema (ti servirà successivamente). Quindi vai alla prossima scena.



Scena 19: inserisci la scheda ethernet presa precedentemente nello slot suggerito.



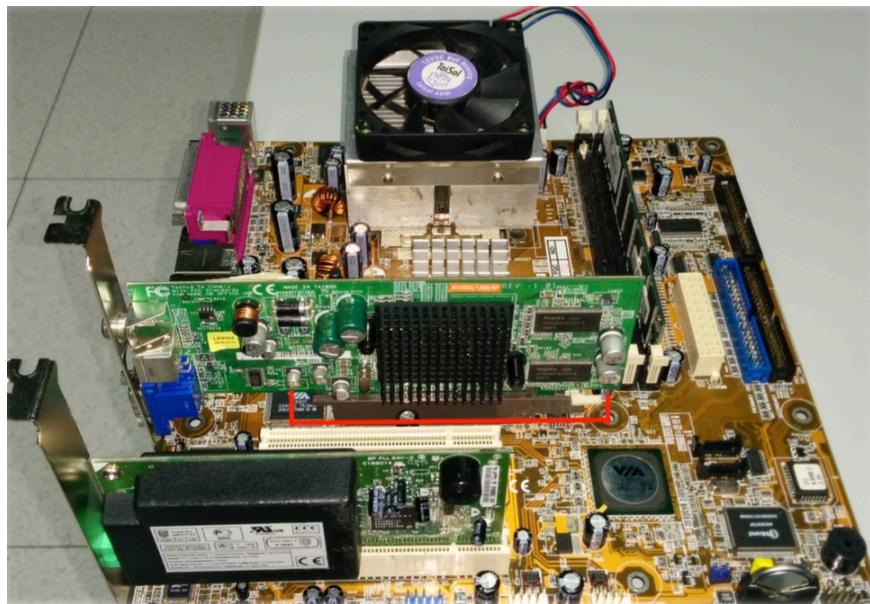
Scena 20: rimuovi il piedino evidenziato in verde (1). Poi, sposta il piedino evidenziato in blu nella posizione indicata dalla freccia (2).



Scena 21: prendi la scheda VGA suggerita dal sistema (ti servirà successivamente).



Scena 22: inserisci la scheda VGA presa precedentemente nello slot suggerito



B

Questionari e risultati

Questionario pre-esperimento

I dati sono organizzati per righe (anche nelle tabelle sottostanti se non è specificato diversamente): i primi 6 utenti hanno utilizzato il manuale tradizionale, i secondi hanno utilizzato il sistema HHD AR e i terzi il sistema HMD AR. La scala utilizzata nelle domande di seguito riportate va da 1 a 6.

1. Quanto sei bravo ad assemblare i componenti di una scheda madre?
2. Quando assembli un componente segui le istruzioni?
3. Ti piace la tecnologia? Quanto?

Sesso	Esperienza (1)	Istruzioni (2)	Tecnologia (3)
F	2	4	4
F	2	4	3
F	3	3	3
M	4	3	4
M	5	2	5
M	3	3	6
M	4	4	6
F	3	2	3
F	3	4	4
M	3	3	2
M	6	5	2
F	2	1	1
M	5	5	2
M	4	4	3
F	3	5	5
M	2	4	6
F	2	3	4
F	3	2	3

Tempi di completamento task espressi in secondi

Colonne: task number. La numerazione parte da 3 perchè i primi 2 task non sono stati oggetto di analisi. Riferirsi al capitolo 5 per maggiori dettagli.

3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
21	24	37	18	65	45	288	61	21	45	74	193	45	54	27	15	33	102	18	36
26	27	28	14	85	36	182	55	28	40	57	180	53	80	35	31	46	121	19	47
15	18	33	13	46	24	145	46	19	32	43	120	32	95	43	18	25	79	14	31
19	16	26	14	41	24	98	38	12	38	29	94	27	50	38	14	31	66	15	28
13	26	21	12	38	26	83	28	15	31	25	80	33	46	23	15	26	47	13	24
18	29	29	17	49	33	120	31	13	39	48	98	28	42	31	23	44	58	25	39
15	21	33	24	47	24	98	44	33	27	55	143	29	38	36	28	25	68	26	29
15	18	29	18	45	28	180	43	16	34	46	144	30	54	31	26	34	77	18	33
21	23	33	16	64	34	156	38	18	28	42	88	32	53	33	24	26	56	23	26
18	25	23	17	43	32	128	35	16	35	46	71	28	40	34	27	25	74	24	36
14	23	27	18	33	25	93	36	15	27	26	64	36	38	25	27	28	44	18	31
18	28	29	31	55	33	231	55	27	49	66	207	31	43	21	33	26	74	23	48
17	31	36	21	42	40	98	57	31	46	58	78	48	61	35	25	36	54	31	38
23	33	39	34	66	38	151	54	26	44	54	132	32	70	29	23	31	95	32	37
34	28	31	46	80	39	183	38	29	47	44	149	43	82	34	22	36	82	43	27
21	26	39	33	68	31	149	29	19	49	39	124	40	57	28	20	32	76	28	32
28	32	43	24	71	38	280	47	18	44	28	200	38	66	37	22	27	85	23	28
23	27	45	35	54	29	198	39	17	35	26	120	31	55	28	19	34	54	22	26

Errori commessi durante i task di manutenzione

Valori uguali a 1 indicano che il task è stato completato correttamente; valori uguali a 0.5 indicano che il task è stato completato con degli errori; valori uguali a 0 indicano che il task non è stato completato con successo.

NASA-TLX

Gli utenti sono organizzati per colonne: i primi sei hanno effettuato il test con il manuale tradizionale, i secondi sei con il sistema HHD AR e gli ultimi sei con il sistema HMD AR. Le prime sei righe indicano i rating ottenuti in ognuno dei sei indicatori previsti dal NASA-TLX (prima parte del test); la settima riga indica il punteggio globale ottenuto nel test; le restanti righe rappresentano i pesi degli indicatori (seconda parte del test). Per maggiori dettagli sul NASA-TLX riferirsi alla sezione 5.1.1 di questo elaborato.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
45	25	20	25	15	30	15	30	15	5	15	25	35	60	45	30	50	30
10	5	5	10	15	5	10	15	10	5	10	15	60	15	25	15	10	35
65	35	20	25	35	40	20	45	20	10	30	60	25	70	60	50	85	45
30	30	15	5	15	30	5	30	15	10	25	30	20	25	55	20	70	30
45	40	5	15	15	35	10	45	10	15	20	25	45	45	40	20	50	40
45	35	20	20	20	30	20	30	10	20	15	35	55	55	30	60	45	40
49.33	33.66	17.33	16.66	20.33	33.66	13.66	36	14.33	12.33	21	33.66	45.33	56	43	44.66	57	35.33
0.26	0.13	0.2	0.26	0.13	0.2	0.13	0.13	0.13	0.2	0.2	0.2	0.2	0.2	0.06	0.26	0.26	0.33
0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0.06	0	0	0.13
0.26	0.33	0.26	0.13	0.2	0.26	0.26	0.33	0.2	0.13	0.13	0.13	0.13	0.26	0.26	0.2	0.2	0.13
0.06	0.13	0.33	0.33	0.33	0.13	0.33	0.26	0.33	0.2	0.26	0.13	0	0.06	0.06	0.06	0.06	0.13
0.13	0.13	0.06	0.06	0.06	0.2	0.06	0.06	0.13	0.26	0.26	0.2	0.26	0.2	0.26	0.2	0.2	0.13
0.26	0.26	0.13	0.26	0.26	0.2	0.2	0.26	0.2	0.2	0.13	0.33	0.2	0.26	0.26	0.26	0.26	0.13

Questionario post-esperimento

In questo caso gli utenti sono organizzati per righe. Il range di valori utilizzato nelle domande 1 e 2 è compreso da 1 a 6.

1. È stato complicato completare i task proposti con il sistema AR?
2. Le istruzioni presentate nel sistema AR sono risultate chiare?
3. Pensi che l'espletamento di una procedura di manutenzione/assemblaggio di un componente sia più efficace se utilizza la realtà aumentata? Se il punteggio ottenuto in questa risposta è ≤ 2 il significato dei valori nelle celle 5 e 6 della tabella sottostante è:
 - 1: È più facile seguire le istruzioni di un manuale cartaceo
 - 2: Ci vuole troppo tempo prima che il sistema mi dica cosa fare
 - 3: La realtà aumentata non è una tecnologia matura
 - 4: Il sistema è troppo lento
 - 5: È più facile seguire le istruzioni di un manuale cartaceo
 - 6: Non capisco bene come usare il sistema
 - 7: Le informazioni mostrate non si vedono bene
 - 8: Le informazioni scompaiono appena mi muovo.

Se il punteggio ottenuto in questa risposta è invece > 2 il significato dei valori nelle celle 5 e 6 è:

- 1: Le animazioni mi aiutano a completare i task più velocemente
- 2: I task da seguire sono meno ambigui se utilizzo la realtà aumentata
- 3: Posso effettuare uno zoom sul luogo dove operare
- 4: Posso osservare la scena da qualsiasi angolazione
- 5: Ho a disposizione più informazioni rispetto ad un manuale cartaceo.

1	2	3	4	5
4	3	4	1	3
3	3	4	1	2
3	4	4	1	2
3	3	2	3	5
3	4	5	1	2
3	5	3	2	3
3	4	4	1	2
3	3	3	2	3
3	3	4	1	4
3	6	5	4	5
3	5	4	1	2
3	6	6	1	3
3	3	3	2	4
4	4	3	1	3
4	3	2	3	6
3	2	3	1	4
5	3	2	1	4
4	4	3	1	2

Bibliografia

- [1] Augmented reality system uses firefly mv: Firefly mv application story. <https://www.ptgrey.com/case-study/id/10379>. Accessed: 2016-04-01.
- [2] Blender. <https://www.blender.org/>. Accessed: 2016-4-23.
- [3] About electromagnetics. <http://polhemus.com/applications/electromagnetics>. Accessed: 2016-04-20.
- [4] Sistema di posizionamento globale. https://it.wikipedia.org/wiki/Sistema_di_posizionamento_globale, note = Accessed: 2016-4-18.
- [5] Msqrdf company info. <http://msqrdf.me/about.html>. Accessed: 2016-04-10.
- [6] Motore grafico. https://it.wikipedia.org/wiki/Motore_grafico. Accessed: 2016-4-25.
- [7] Unity3d. <https://unity3d.com>. Accessed: 2016-4-26.
- [8] Armedia. <http://dev.inglobetechnologies.com/>, . Accessed: 2016-5-04.
- [9] Armedia sample android. http://www.inglobetechnologies.com/it/new_products/arplayer_android/info.php, . Accessed: 2016-5-05.
- [10] Artoolkit web site. <http://arToolkit.org/>, . Accessed: 2016-5-03.
- [11] Artoolkit sample android. http://arToolkit.org/documentation/doku.php?id=4.Android:android_examples, . Accessed: 2016-5-03.

- [12] Craft ar web site. <http://catchoom.com/>. Accessed: 2016-5-05.
- [13] Augmented/virtual reality to hit \$150 billion disrupting mobile by 2020. <http://www.digi-capital.com/news/2015/04/augmentedvirtual-reality-to-hit-150-billion-disrupting-mobile-by-2020/#.VyXTFa0LSRs>. Accessed: 2016-04-15.
- [14] "dtam: Dense tracking and mapping in real-time". <https://www.youtube.com/watch?v=Df9WhgibCQA&feature=youtu.be>. Accessed: 2016-4-21.
- [15] Experenti web site. <http://www.experenti.com/>. Accessed: 2016-5-06.
- [16] "fast algorithm for corner detection". http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html. Accessed: 2016-4-18.
- [17] La dimensione del sensore fotografico (fattore di crop). <http://www.fotografareindigitale.com/2014/08/la-dimensione-del-sensore-fotografico-fattore-di-crop/>. Accessed: 2016-5-25.
- [18] Guida blender. <http://www.html.it/guide/blender-3d-guida/>, . Accessed: 2016-4-23.
- [19] Guida unity3d. <http://www.html.it/guide/guida-unity-3d>, . Accessed: 2016-4-27.
- [20] Craft ar web site. <http://www.in2ar.com/>. Accessed: 2016-5-06.
- [21] Kudan web site. <https://www.kudan.eu/>. Accessed: 2016-5-06.
- [22] Real time pose estimation of a textured object. http://docs.opencv.org/3.1.0/dc/d2c/tutorial_real_time_pose.html#gsc.tab=o. Accessed: 2016-04-01.
- [23] Project tango web site. <https://developers.google.com/project-tango/>. Accessed: 2016-5-23.

- [24] Determining the focal length, image center, baseline and field of view measurements for point grey stereo vision cameras). <https://www.ptgrey.com/KB/10315>. Accessed: 2016-5-26.
- [25] The top 13 virtual reality and augmented reality startups to watch. <https://www.cbinsights.com/blog/virtual-and-augmented-reality-startups-to-watch/>. Accessed: 2016-4-30.
- [26] Unity (motore grafico). [https://it.wikipedia.org/wiki/Unity_\(motore_grafico\)](https://it.wikipedia.org/wiki/Unity_(motore_grafico)). Accessed: 2016-4-27.
- [27] Wikitude web site. <http://www.wikitude.com/>, . Accessed: 2016-5-06.
- [28] Wikitude 3d tracking (beta). <http://www.wikitude.com/blog-wikitude-3d-tracking-beta/>, . Accessed: 2016-5-06.
- [29] Andrea F Abate, Virginio Cantoni, Michele Nappi, Fabio Narducci, and Stefano Ricciardi. Hybrid multi-sensor tracking system for field-deployable mixed reality environment. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, pages 1450–1455. IEEE, 2015.
- [30] Michael Abrash. What vr could, should, and almost certainly will be within two years. *Steam Dev Days, Seattle*, 2014.
- [31] Dhiraj Amin and Sharvari Govilkar. Comparative study of augmented reality sdk's. *International Journal on Computational Sciences Applications (IJCSA)*, 5(1), 2015.
- [32] Simona Arma. Blender: il dietro le quinte, 2012.
- [33] Josep Aulinas, Yvan R Petillot, Joaquim Salvi, and Xavier Lladó. The slam problem: a survey. In *CCIA*, pages 363–371. Citeseer, 2008.
- [34] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47, 2001.
- [35] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.

- [36] Kevin M Baird and Woodrow Barfield. Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality*, 4(4):250–259, 1999.
- [37] Wolfgang Birkfellner, Johann Hummel, Emmanuel Wilson, and Kevin Cleary. Tracking devices. In *Image-Guided Interventions*, pages 23–44. Springer, 2008.
- [38] Rainer Bischoff and Arif Kazi. Perspectives on augmented reality based human-robot interaction with industrial robots. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 4, pages 3226–3231. IEEE, 2004.
- [39] Massimo Borelli. Nasa-tlx in html and javascript. <http://www.keithv.com/software/nasatlx/nasatlx.html>. Accessed: 2016-5-27.
- [40] Massimo Borelli. Anova e confronti multipli con r. 2014.
- [41] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.
- [42] Julie Carmigniani, Borko Furht, Marco Anisetti, Paolo Ceravolo, Ernesto Damiani, and Misa Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, 2011.
- [43] Thomas P Caudell and David W Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on*, volume 2, pages 659–669. IEEE, 1992.
- [44] Sue VG Cobb, Sarah Nichols, Amanda Ramsey, and John R Wilson. Virtual reality-induced symptoms and effects (vrise). *Presence*, 8(2):169–186, 1999.
- [45] Thomas D Cook. Quasi-experimental design. *Wiley Encyclopedia of Management*, 1979.
- [46] Michael J Crawley. Statistics: an introduction using r, 2005. Wiley, ISBN 0, 470(02298):1, 2005.

- [47] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.
- [48] Andrew J Davison and Nobuyuki Kita. 3d simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I-384. IEEE, 2001.
- [49] Andrew J Davison and David W Murray. *Mobile robot localisation using active vision*. Springer, 1998.
- [50] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1052–1067, 2007.
- [51] R Documentation. Fit an analysis of variance model. <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/aov.html>. Accessed: 2016-5-28.
- [52] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.
- [53] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.
- [54] HC Fang, Soh-Khim Ong, and Andrew YC Nee. Robot programming using augmented reality. In *CyberWorlds, 2009. CW'09. International Conference on*, pages 13–20. IEEE, 2009.
- [55] Steven Feiner, Blair Macintyre, and Dorée Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):53–62, 1993.
- [56] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 590–596. IEEE, 2005.

- [57] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 15–22. IEEE, 2014.
- [58] Paul C. Foster. Dense tracking and mapping (dtam) for opencv. <https://www.google-melange.com/gsoc/project/details/google/gsoc2014/anuranbaka/5757334940811264>. Accessed: 2016-5-25.
- [59] Wolfgang Friedrich, D Jahn, and L Schmidt. Arvika-augmented reality for development, production and service. In *ISMAR*, volume 2002, pages 3–4. Citeseer, 2002.
- [60] Yakup Genc, Sebastian Riedel, Fabrice Souvannavong, Cuneyt Akinlar, and Nassir Navab. Marker-less tracking for ar: A learning-based approach. In *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pages 295–304. IEEE, 2002.
- [61] Richard F Haines, Edith Fischer, and Toni A Price. Head-up transition behavior of pilots with and without head-up display in simulated low-visibility approaches. 1980.
- [62] Jukka Hakkinnen, Tero Vuori, and M Paakka. Postural stability and sickness symptoms after hmd use. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 147–152, 2002.
- [63] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.
- [64] Steven J Henderson and Steven Feiner. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 135–144. IEEE, 2009.
- [65] Steven J Henderson and Steven K Feiner. Augmented reality for maintenance and repair (armar). Technical report, DTIC Document, 2007.
- [66] Valentin Heun, Shunichi Kasahara, and Pattie Maes. Smarter objects: using ar technology to program physical objects and their interactions. In *CHI'13*

Extended Abstracts on Human Factors in Computing Systems, pages 961–966. ACM, 2013.

- [67] Dell Inc. How to analyze data with low quality or small samples, nonparametric statistics. <https://documents.software.dell.com/statistics/textbook/nonparametric-statistics>. Accessed: 2016-5-28.
- [68] Juan A Juanes, Juan J Gómez, Pedro D Peguero, and Pablo Ruisoto. Digital environment for movement control in surgical skill training. *Journal of Medical Systems*, 40(6):1–7, 2016.
- [69] Il-Kyun Jung and Simon Lacroix. High resolution terrain mapping using low altitude aerial stereo imagery. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 946–951. IEEE, 2003.
- [70] Frédéric Jurie and Michel Dhome. A simple and efficient template matching algorithm. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 544–549. IEEE, 2001.
- [71] Hirokazu Kato. Artoolkit. <http://www.hitl.washington.edu/artoolkit/>, 1999.
- [72] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.
- [73] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 83–86. IEEE, 2009.
- [74] Artesas Konsortium. Artesas—advanced augmented reality technologies for industrial service applications, 2007.
- [75] William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.
- [76] Hongen Liao, Takashi Inomata, Ichiro Sakuma, and Takeyoshi Dohi. 3-d augmented reality for mri-guided surgery using integral videography auto-stereoscopic image overlay. *Biomedical Engineering, IEEE Transactions on*, 57(6):1476–1486, 2010.

- [77] João Paulo Lima, Francisco Simões, Lucas Figueiredo, and Judith Kelner. Model based markerless 3d tracking applied to augmented reality. *Journal on 3D Interactive Systems*, 1, 2010.
- [78] Chiuhsiang Joe Lin, Hung-Jen Chen, Ping-Yun Cheng, and Tien-Lung Sun. Effects of displays on visually controlled task performance in three-dimensional virtual reality environment. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 25(5):523–533, 2015.
- [79] Marina Londei. Motori di rendering a confronto in blender: Teoria e applicazioni, 2015.
- [80] Julian Looser, Raphaël Grasset, Hartmut Seichter, and Mark Billinghurst. Osgart-a pragmatic approach to mr. In *Industrial Workshop at ISMAR*, volume 2006. Citeseer, 2006.
- [81] Pierre Malbezin, Wayne Piekarski, and Bruce H Thomas. Measuring artootkit accuracy in long distance tracking experiments. In *Augmented Reality Toolkit, The First IEEE International Workshop*, pages 2–pp. IEEE, 2002.
- [82] Department of Statistics Columbia University Martin A. Lindquist. One-way anova. <http://www.stat.columbia.edu/~martin/W2024/R3.pdf>. Accessed: 2016-5-27.
- [83] Zeljko Medenica, Andrew L Kun, Tim Paek, and Oskar Palinko. Augmented reality vs. street views: a driving simulator study comparing two emerging navigation aids. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 265–274. ACM, 2011.
- [84] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.
- [85] Minitab. Choosing between a nonparametric test and a parametric test. <http://blog.minitab.com/blog/adventures-in-statistics/choosing-between-a-nonparametric-test-and-a-parametric-test>. Accessed: 2016-5-28.
- [86] Paula M Monker. Search for agile manufacturing. *Manufacturing Engineering*, 113(5):40–43, 1994.

- [87] Miwa Nakanishi and Tomohiro Sato. Application of digital manuals with a retinal imaging display in manufacturing: behavioral, physiological, and psychological effects on workers. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 25(2):228–238, 2015.
- [88] Ulrich Neumann and Anthony Majoros. Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pages 4–11. IEEE, 1998.
- [89] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.
- [90] Stéphane Nicolau, Luc Soler, Didier Mutter, and Jacques Marescaux. Augmented reality in laparoscopic surgical oncology. *Surgical oncology*, 20(3):189–201, 2011.
- [91] Susanna Nilsson and Björn Johansson. Fun and usable: augmented reality instructions in a hospital setting. In *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining user interfaces*, pages 123–130. ACM, 2007.
- [92] Prof. Didier Stricker & Dr. Alain Pagani. "augmented reality, advanced slam, applications". http://ags.cs.uni-kl.de/fileadmin/inf_agc/3dcv-ws14-15/3DCV_lec13_SLAM_AR.pdf. Accessed: 2016-4-19.
- [93] Hanhoon Park and Jong-Il Park. Invisible marker tracking for ar. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 272–273. IEEE Computer Society, 2004.
- [94] Thomas Pettersen, John Pretlove, Charlotte Skourup, Torbjorn Engedal, and T Lkstad. Augmented reality for programming industrial robots. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 319. IEEE Computer Society, 2003.
- [95] Guido Maria Re, James Oliver, and Monica Bordegoni. Impact of monitor-based augmented reality for on-site industrial manual operations. *Cognition, Technology & Work*, pages 1–14, 2016.

- [96] Dirk Reiners, Didier Stricker, Gudrun Klinker, and Stefan Müller. Augmented reality for construction tasks: doorlock assembly. *Proc. IEEE and ACM IWAR*, 98(1):31–46, 1998.
- [97] Miles Richardson, Gary Jones, and Mark Torrance. Identifying the task variables that influence perceived object assembly complexity. *Ergonomics*, 47(9):945–964, 2004.
- [98] Ph.D. Robert I. Kabacoff. Anova. <http://www.statmethods.net/stats/anova.html>. Accessed: 2016-5-28.
- [99] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.
- [100] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.
- [101] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. 1971.
- [102] Alexandro Simonetti Ibañez and Josep Paredes Figueras. Vuforia v1. 5 sdk: Analysis and evaluation of capabilities. 2013.
- [103] Tor Egil Riegels Strand. Tracking for outdoor mobile augmented reality: Further development of the zion augmented reality application. 2008.
- [104] Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.
- [105] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80. ACM, 2003.
- [106] Postdoctoral Scholar at MIT’s Computer Science Tomasz Malisiewicz and Artificial Intelligence Laboratory. The future of real-time slam and "deep learning vs slam".

- [107] Mihran Tuceryan, Douglas S Gree, Ross T Whitaker, David E Breen, Chris Crampton, Eric Rose, and Klaus H Ahlers. Calibration requirements and procedures for a monitor-based augmented reality system. *Visualization and Computer Graphics, IEEE Transactions on*, 1(3):255–273, 1995.
- [108] Giles Westerfield, Antonija Mitrovic, and Mark Billinghurst. Intelligent augmented reality training for motherboard assembly. *International Journal of Artificial Intelligence in Education*, 25(1):157–172, 2015.
- [109] Christian Wiedemann, Markus Ulrich, and Carsten Steger. Recognition and tracking of 3d objects. In *Pattern Recognition*, pages 132–141. Springer, 2008.
- [110] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [111] Oliver J Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 14:1, 2007.
- [112] JC Wortmann, J Browne, and PJ Sackett. *The System of manufacturing: A prospective study*. Eindhoven University of Technology, Graduate School of Industrial Engineering and Management Science, 1993.
- [113] Harald Wuest, Florent Vial, and D Strieker. Adaptive line tracking with multiple hypotheses for augmented reality. In *Mixed and Augmented Reality, 2005. Proceedings. Fourth IEEE and ACM International Symposium on*, pages 62–69. IEEE, 2005.
- [114] Li Yi-bo, Kang Shao Peng, Qiao Zhi Hua, and Zhu Qiong. Development actuality and application of registration technology in augmented reality. In *Computational Intelligence and Design, 2008. ISCID'08. International Symposium on*, volume 2, pages 69–74. IEEE, 2008.
- [115] Jürgen Zauner, Michael Haller, Alexander Brandl, and Werner Hartmann. Authoring of a mixed reality assembly instructor for hierarchical structures. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, page 237. IEEE Computer Society, 2003.

Colophon

THIS THESIS WAS TYPESET using L^AT_EX, originally developed by Leslie Lamport and based on Donald Knuth's T_EX. The body text is set in 11 point Arno Pro, designed by Robert Slimbach in the style of book types from the Aldine Press in Venice, and issued by Adobe in 2007. A template, which can be used to format a PhD thesis with this look and feel, has been released under the permissive MIT (x11) license, and can be found online at github.com/suchow/ or from the author at suchow@post.harvard.edu.