

POLITECNICO DI TORINO

Corso di Laurea Magistrale
Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

Controllo di un braccio robotico mediante una Brain Computer Interface



Relatori

Andrea Sanna
Federico Manuri
Francesco De Pace
Hannes Kaufmann

Candidato

Matteo Bosco

Anno Accademico 2021-2022

Desidero ringraziare coloro che mi hanno supportato durante questo percorso, la mia famiglia, i miei amici

Un sincero ringraziamento al professor Andrea Sanna che mi ha permesso di vivere un'esperienza preziosa

Ringrazio il ricercatore Federico Manuri per il supporto offerto durante il progetto di studio al Politecnico di Torino

Ringrazio il ricercatore Francesco De Pace per l'accoglienza e la disponibilità dimostrate durante la mia permanenza a Vienna

Ringrazio infine il professor Hannes Kaufmann per avermi ospitato e sostenuto all'Università tecnica di Vienna

No price is too high to pay for the privilege of owning yourself

- Friedrich Nietzsche

Abstract

Al fine di sviluppare un sistema per il controllo di un braccio robotico mediante una Brain Computer Interface (BCI) è stato impiegato il dispositivo NextMind: un'interfaccia neurale non invasiva di tipo reattivo basata sulla tecnologia EEG-SSVEP che permette di rilevare l'attività cerebrale generata dalla stimolazione della corteccia visiva tramite l'utilizzo di componenti visuali chiamati NeuroTag. È stato inizialmente condotto uno studio al Politecnico di Torino con lo scopo di analizzare il NextMind mediante l'integrazione dello stesso a tre differenti dispositivi di visualizzazione: un monitor, il Microsoft HoloLens 2 (visore per la realtà aumentata) e l'Oculus Rift (visore per la realtà virtuale). Successivamente, all'Università tecnica di Vienna, il NextMind è stato impiegato per la realizzazione del sistema finalizzato all'interazione con il braccio meccanico attraverso il solo uso della vista: risulta possibile, infatti, effettuare operazioni di selezione e posizionamento (pick and place) grazie all'utilizzo di una videocamera per il rilevamento degli oggetti e ad un'interfaccia utente proiettata sull'area di interesse. Sono stati, infine, condotti dei test utente per la valutazione del sistema presentando due differenti algoritmi di posizionamento dei NeuroTag: uno adattivo ed uno non adattivo. Sebbene entrambi gli algoritmi abbiano presentato un'ottima operabilità, quello adattivo si è dimostrato essere migliore e, in generale, il sistema è risultato essere solido ed efficace, proponendosi come punto di inizio per eventuali sviluppi futuri.

Indice

Elenco delle tabelle	IV
Elenco delle figure	V
Acronimi e Glossario	VII
1 Introduzione	1
1.1 Organizzazione dei capitoli	2
2 Stato dell'Arte	3
2.1 Interazione uomo-robot mediante BCI	3
2.1.1 Robotica di servizio.....	3
2.1.2 Robotica industriale	5
2.1.3 Robotica biomedica.....	6
2.1.4 Domotica	8
2.1.5 Robotica militare.....	9
2.1.6 Robotica d'intrattenimento	10
2.2 Dispositivo NextMind	11
2.3 Riepilogo	13
3 Tecnologie	15
3.1 Tecnologie Hardware	15
3.1.1 NextMind	15
3.1.2 Stäubli TX2-60	18
3.1.3 HoloLens 2	19
3.1.4 Oculus Rift	20
3.1.5 Gripper elettromagnetico.....	21
3.1.6 Proiettore laser	24
3.2 Tecnologie Software	25
3.2.1 Unity	25
3.2.2 Visual Studio	26
3.2.3 Visual Studio Code	26
3.2.4 MoveIt	27
3.2.5 Arduino IDE	28

4	Studio preliminare	29
4.1	Architettura dell'applicazione	29
4.1.1	Hub	29
4.1.2	Calibration	30
4.1.3	Performance task	32
4.1.4	Appearance task	34
4.2	Test	36
4.2.1	Valutazione degli esperti	36
4.2.2	Test di usabilità	37
4.3	Risultati	38
5	Sistema BCI-Robot	41
5.1	Infrastruttura del sistema	43
5.2	Rilevamento degli oggetti	47
5.2.1	Calibrazione videocamera	47
5.2.2	Identificazione oggetti	49
5.3	Architettura dell'applicazione	50
5.3.1	Calibrazione NextMind	50
5.3.2	Interfacce utente	53
5.4	Algoritmi di posizionamento	55
5.4.1	Algoritmo adattivo	55
5.4.2	Algoritmo non adattivo	61
5.5	Colori interfacce utente	62
6	Test e risultati	63
6.1	Test	63
6.1.1	Calibrazione	64
6.1.2	Test di usabilità	65
6.1.3	Compilazione questionari	67
6.2	Risultati	69
6.2.1	SUS	70
6.2.2	NASA-TLX	72
6.2.3	Single-Item Questionnaire	73
6.2.4	Prestazioni	74
7	Conclusioni e sviluppi futuri	79
7.1	Conclusioni	79
7.2	Sviluppi futuri	81
A	Codice Arduino gripper	85
B	Controlli sistema	87
C	Algoritmi di posizionamento	89
	Bibliografia e Sitografia	91

Elenco delle tabelle

2.1	Riepilogo dei principali progetti di HMI mediante BCI.....	13
4.1	Configurazioni testate nella performance task.....	36
4.2	Errori e tempi medi di ogni configurazione.....	38
4.3	Risultati del questionario SUS per ogni utente.....	39
6.1	Risultati analisi statistica del System Usability Score (SUS).....	71
6.2	Intervalli di interpretazione dei risultati NASA-TLX.....	72
6.3	Risultati analisi statistica del NASA-TLX	73
6.4	Risultati analisi statistica del Single-Item Questionnaire.....	74
6.5	Risultati analisi statistica dei tempi di selezionamento	76
6.6	Risultati analisi statistica degli errori	77
7.1	Riepilogo vantaggi e svantaggi dei due algoritmi progettati	81

Elenco delle figure

2.1	Partecipante affetta da tetraplegia nell'intento di bere autonomamente.....	4
2.2	Sistema di comunicazione tra paziente e robot mediante una BCI.....	5
2.3	Collaborazione tra operatore e cobot mediante BCI.....	5
2.4	Sistema di collaborazione con interfaccia in AR.....	6
2.5	Configurazione dell'esperimento con monitoraggio dell'attenzione.....	7
2.6	BMSCI per pazienti quadriplegici (sx) e paraplegici (dx).....	7
2.7	Sistema di impianto domotico controllato mediante BCI.....	8
2.8	Sistema per il controllo di uno sciame di droni tramite BCI.....	9
2.9	Integrazione di una BCI nel gioco degli scacchi.....	10
2.10	Mindflex della Mattel.....	10
2.11	Test dell'interfaccia duale rBCI-pBCI.....	11
2.12	Sistema per casa intelligente controllato mediante NextMind.....	12
3.1	Esempio di BCI non invasiva.....	15
3.2	Sensore NextMind basato su tecnologia EEG-SSVEP.....	17
3.3	Esempio di NeuroTag.....	17
3.4	Robot Stäubli TX2-60.....	18
3.5	Microsoft HoloLens 2.....	19
3.6	Oculus Rift e controller Oculus Touch.....	20
3.7	Circuito elettronico del gripper elettromagnetico.....	21
3.8	MCU Arduino Micro.....	22
3.9	Modulo Bluetooth HC-05.....	22
3.10	Elettromagnete 12V.....	23
3.11	Relè a 4 canali.....	23
3.12	Proiettore laser LG HF80LA.....	24
3.13	Interfaccia utente Unity.....	25
3.14	Interfaccia utente Visual Studio.....	26
3.15	Interfaccia utente MoveIt.....	27
3.16	Interfaccia utente Arduino IDE.....	28
4.1	Interfaccia utente Hub.....	30
4.2	Pannello di calibrazione per il controllo della qualità dei contatti.....	31
4.3	NeuroTag utilizzato per la calibrazione del NextMind.....	31
4.4	Schermata di configurazione della performance task.....	32
4.5	Performance task.....	33
4.6	Schermata risultati della performance task.....	33

4.7	Schermata di configurazione dell'appearance task	34
4.8	Possibili personalizzazioni dei NeuroTag.....	35
4.9	Appearance task	35
4.10	Tester durante l'utilizzo dei tre dispositivi di visualizzazione	37
4.11	Configurazioni 2 (sinistra) e 5 (destra) della performance task	39
5.1	Robot, area di deposito e interfaccia utente del sistema	41
5.2	Schema del sistema BCI-Robot	42
5.3	Architettura hardware del sistema	43
5.4	Diagramma di flusso del sistema.....	45
5.5	Grafo della rete ROS.....	46
5.6	Oggetti utilizzati per i test del sistema.....	47
5.7	Calibrazione prospettica della videocamera	48
5.8	Calibrazione cromatica della videocamera.....	48
5.9	Interfaccia utente per il rilevamento degli oggetti	49
5.10	Interfaccia utente per il controllo qualità dei contatti	50
5.11	NeuroTag utilizzato durante la calibrazione del NextMind	51
5.12	Punteggio riassuntivo ottenuto durante la fase di calibrazione.....	52
5.13	Regione della UI adibita alla visualizzazione delle icone informative.....	52
5.14	Interfaccia utente della scena Unity integrante l'algoritmo adattivo.....	53
5.15	Aree di posizionamento algoritmo adattivo (sinistra) e non adattivo (destra)	54
5.16	Interfaccia utente per il debug dell'algoritmo adattivo.....	54
5.17	Fasi di creazione della matrice dell'algoritmo adattivo.....	56
5.18	Tecnica del raycasting per il calcolo delle occlusioni.....	57
5.19	Ordine di computazione dei bounding box	57
5.20	Rette congiungenti i centri dei bounding box alla posizione dell'utente	58
5.21	Albero quadramentale generato dalle celle valide della matrice.....	59
5.22	Matrice finale completa di NeuroTag	59
5.23	Raggi di controllo per la verifica della soluzione	60
5.24	Soluzione adattiva osservata dall'utente (sinistra) e proiettata (destra)	60
5.25	Soluzione non adattiva osservata dall'utente (sinistra) e proiettata (destra) .	61
5.26	Cerchio cromatico e codici dei colori utilizzati per l'interfaccia utente.....	62
5.27	Palette di colori associata agli oggetti da sinistra a destra.....	62
6.1	Grafico occorrenze età del gruppo di studio	63
6.2	Diagramma riassuntivo della struttura dei test utente	64
6.3	Partecipante del test durante il processo di calibrazione	65
6.4	Configurazioni e sequenze di selezione dei test di usabilità	66
6.5	Questionario NASA Task Load Index (NASA-TLX)	68
6.6	Distribuzione non normale (sinistra) e normale (destra).....	69
6.7	Box plot simmetrico (sinistra) e asimmetrico (destra)	70
6.8	Punteggi del System Usability Score (SUS).....	70
6.9	Scala di interpretazione dei risultati SUS.....	71
6.10	Punteggi del NASA-TLX	72
6.11	Punteggi del Single-Item Questionnaire.....	74
6.12	Tempi di selezione dei NeuroTag e tempi complessivi delle prove	75
6.13	Errori complessivi rilevati durante le prove	76

Acronimi

aBCI Active Brain Computer Interface
AR Augmented Reality
BCI Brain Computer Interface
BLE Bluetooth Low Energy
BMI Brain Machine Interface
BMSCI Brain-Machine-Spinal-Cord Interface
CPS Cyber-Physical System
DARPA Defense Advanced Research Projects Agency
DLP Digital Light Processing
DOF Degree Of Freedom
ERP Event-Related Potential
fMRI functional Magnetic Resonance Imaging
HCI Human-Computer Interaction
HDMI High-Definition Multimedia Interface
HMD Head-Mounted Display
HMI Human-Machine Interaction
HPU Holographic Processing Unit
HSV Hue Saturation Value
HTML HyperText Markup Language
IA Intelligenza Artificiale
IDE Integrated Development Environment

IK Inverse Kinematics
IoT Internet of Things
LED Light-Emitting Diode
MCU MicroController Unit
MI-BCI Motor Imagery-based Brain Computer Interface
MR Mixed Reality
MRTK Mixed Reality Toolkit
NASA National Aeronautics and Space Administration
NIRS Near InfraRed Spectroscopy
OLED Organic Light-Emitting Diode
pBCI Passive Brain Computer Interface
PET Positron Emission Tomography
PWM Pulse Width Modulation
rBCI Reactive Brain Computer Interface
RGB Red Green Blue
ROS Robot Operating System
SCI Spinal Cord Injury
SDK Software Development Kit
SMR Sensory Motor Rhythms
SNR Signal to Noise Ratio
SSVEP Steady State Visually Evoked Potentials
SUS System Usability Score
TLX Task Load Index
TOF Time Of Flight
UART Universal Asynchronous Receiver-Transmitter
UI User Interface
URL Uniform Resource Locator

USB Universal Serial Bus

UWP Universal Windows Platform

UX User Experience

VEP Visual Evoked Potential

VR Virtual Reality

XR Extended Reality

Glossario

cobot Collaborative robot

ECoG Elettrocorticografia

EEG Elettroencefalografia

MEG Magnetoencefalografia

Capitolo 1

Introduzione

Negli ultimi decenni l'interazione uomo-robot è divenuta sempre più importante nel panorama tecnico scientifico. Questo è dovuto prevalentemente alle recenti innovazioni tecnologiche e all'avvento dell'industrializzazione che ha descritto un rapido cambiamento dalla manodopera umana all'ausilio o talvolta addirittura alla completa surrogazione da parte di macchinari. Oggigiorno molteplici incarichi in cui è necessario un valore aggiunto (come precisione, forza o rapidità di esecuzione) sono svolti dai cosiddetti cobot, ovvero dei robot costruiti per collaborare con il personale umano.

Alla base di questa tesi di laurea vi è lo studio dell'interazione tra uomo e macchina mediante l'utilizzo di una Brain Computer Interface, ovvero un'interfaccia neurale in grado di rilevare e decodificare gli impulsi elettrici prodotti dal cervello. L'ambito applicativo delle interfacce neurali come dispositivi di comunicazione tra uomo e robot risulta essere estremamente vasto, coinvolgendo molteplici settori tra i quali emerge in particolare la robotica di servizio. Da alcuni anni, infatti, l'utilizzo di robot collaborativi a servizio di singoli individui è stato l'oggetto di numerosi studi atti a migliorare lo stile di vita di persone con limitazioni motorie o difficoltà di linguaggio.

L'interfaccia neurale utilizzata per lo studio è il NextMind: una BCI non invasiva di tipo reattivo basata sugli Steady State Visually Evoked Potentials (SSVEP) che utilizza la tecnologia EEG per rilevare l'attività cerebrale generata dalla stimolazione della corteccia visiva. Con lo scopo di analizzarne il funzionamento, è stato condotto uno studio preliminare al Politecnico di Torino in collaborazione con il docente e relatore Andrea Sanna ed il ricercatore post-dottorato Federico Manuri, definendo delle regole per lo sviluppo di un'interfaccia utente (UI) per un'applicazione che integri l'utilizzo del dispositivo. Un altro importante proposito del progetto è stato quello di valutare se le prestazioni e l'usabilità della BCI variassero in funzione del dispositivo di visualizzazione: per lo studio sono stati impiegati un monitor, un HMD-AR (Microsoft HoloLens 2) ed un HMD-VR (Oculus Rift).

Infine, all'Università tecnica di Vienna è stato progettato e sviluppato un sistema in collaborazione con il docente Hannes Kaufmann ed il ricercatore post-dottorato Francesco De Pace. Il sistema in oggetto permette, tramite l'esclusivo utilizzo del NextMind, il controllo di un braccio robotico per operazioni di selezione e posizionamento (pick and place). Consiste nel rilevamento di oggetti posizionati in uno spazio di lavoro raggiungibile

dal cobot intorno ai quali vengono visualizzati tramite proiezione laser i bounding box¹ associati a dei NeuroTag² per permettere la selezione tramite l'interfaccia neurale. Una volta selezionato un NeuroTag, il braccio meccanico provvede ad effettuare un'operazione di raccolta dell'oggetto associato e di spostamento dello stesso in un'area di deposito.

L'obiettivo principale del progetto risulta essere quello di studiare un sistema che possa permettere ad utenti con limitazioni nei movimenti o nella parola di interagire con oggetti circostanti mediante l'ausilio di un robot collaborativo. L'aspetto innovativo del sistema si può individuare nell'interfaccia utente in AR proiettata che, al giorno d'oggi, non è presente in alcuno studio relativo alla HCI mediante Brain Computer Interface. Sebbene la finalità si identifichi nella robotica di servizio, un ulteriore utilizzo del sistema progettato potrebbe essere l'impiego dello stesso in ambienti industriali al fine di fornire agli operatori un metodo di selezione adoperabile, ad esempio, per fornire istruzioni ad un cobot in una catena di produzione per l'assemblaggio di componentistica. Essendo possibile rilevare gli oggetti tramite una videocamera, si potrebbero scansionare gli eventuali componenti forniti in ordine non predefinito e gli operatori potrebbero selezionarli senza l'esigenza di utilizzo delle mani (le quali potrebbero essere sporche, occupate o l'operatore potrebbe indossare dei guanti ingombranti inibendo la possibilità di digitare su una tastiera o su uno schermo) o di comandi vocali (ipotizzando un ambiente industriale eccessivamente rumoroso a causa di macchinari).

1.1 Organizzazione dei capitoli

Nel capitolo 2 sarà analizzato lo stato dell'arte per quanto riguarda l'interazione uomo-robot mediante le interfacce neurali e, successivamente, saranno trattati in particolare progetti nei quali è stato utilizzato il dispositivo NextMind come principale strumento di interazione tra uomo e macchina.

Nel capitolo 3 saranno introdotte e dettagliate le tecnologie hardware e software utilizzate per lo studio preliminare e per lo sviluppo del sistema in oggetto.

Nel capitolo 4 sarà descritto lo studio preliminare del dispositivo NextMind svolto al Politecnico di Torino per analizzare l'interfaccia neurale ed individuare le differenze in termini di prestazioni ed usabilità al variare dei dispositivi di visualizzazione.

Nel capitolo 5 saranno presentati architettura e funzionamento del sistema progettato e sviluppato all'Università tecnica di Vienna per il controllo di un braccio robotico mediante una Brain Computer Interface.

Nel capitolo 6 saranno descritti i test condotti al fine di analizzare il sistema in termini di interazione uomo-robot e, successivamente, saranno commentati i risultati ottenuti.

Nel capitolo 7 saranno evidenziate le conclusioni complessive dello studio e, infine, saranno proposti eventuali sviluppi futuri per il miglioramento del sistema.

¹Rettangoli con area minima che permettono la delimitazione degli oggetti

²Componenti utilizzati per il corretto funzionamento del NextMind

Capitolo 2

Stato dell'Arte

Questo capitolo sarà dedicato all'analisi dello stato dell'arte per quanto riguarda l'interazione uomo-robot mediante le Brain Computer Interface (BCI). Successivamente verranno trattati in particolare progetti nei quali è stato utilizzato il dispositivo NextMind come principale strumento di interazione tra uomo e macchina.

2.1 Interazione uomo-robot mediante BCI

L'interazione uomo-robot è un'area di ricerca multidisciplinare con lo specifico obiettivo di ottimizzare il modo in cui utenti e macchine interagiscono. L'origine della Human-Computer Interaction (HCI) risale agli inizi degli anni ottanta, periodo in cui i calcolatori hanno iniziato ad essere ampiamente diffusi nelle famiglie e negli uffici rendendo necessario lo studio di un'interazione efficace ed intuitiva per permettere di elaborare testi ed unità numeriche. Oggi la ricerca si focalizza prevalentemente sulla progettazione di interfacce interattive per migliorare sempre di più l'esperienza utente, talvolta tramite l'utilizzo di tecnologie innovative. Esistono diversi ambiti in cui la robotica si è evoluta a tal punto da poter essere controllata dal cervello umano mediante l'utilizzo di BCI, a seguire i più importanti.

2.1.1 Robotica di servizio

Sicuramente la maggiore utenza delle interfacce neurali (BCI) è rappresentata da persone che dispongono di invalidità fisiche dal momento che i dispositivi in questione permettono di creare un canale di comunicazione diretto con il cervello senza il coinvolgimento di processi motori.

Uno dei primi esempi di impiego di una BCI applicata alla robotica di servizio è rappresentato da uno studio [1] pubblicato nel 2012 nel quale due persone affette da tetraplegia¹ hanno imparato a muovere un cursore di un computer e a controllare dispositivi fisici mediante l'utilizzo di una MI-BCI (Motor Imagery-based Brain Computer Interface).

¹Paralisi che coinvolge gli arti e il torso

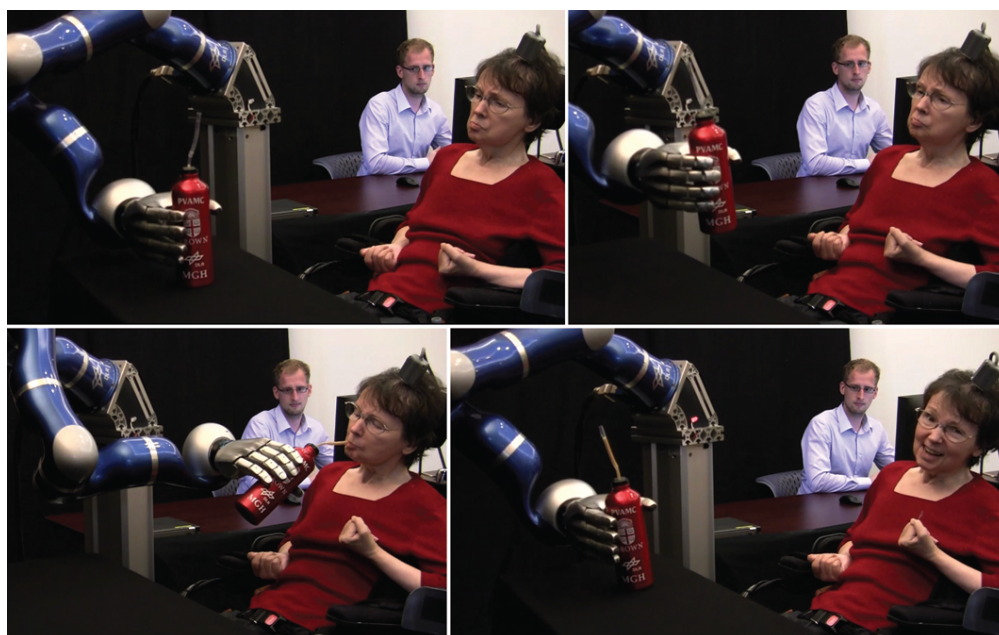


Figura 2.1. Partecipante affetta da tetraplegia nell'intento di bere autonomamente

I due partecipanti dello studio (una donna di 58 anni e un uomo di 65 anni) soffrivano entrambi di tetraplegia e anartria² e ad entrambi furono impiantate delle interfacce neurali invasive nella parte di cervello atta al controllo della mano dominante: nella donna, ritratta in figura 2.1, 5 anni prima l'inizio degli studi (nel 2005), nell'uomo 5 mesi prima (nel 2010). Attraverso l'utilizzo del DLR Light-Weight Robot III, un braccio meccanico in grado di riprodurre azioni complesse, entrambi i partecipanti sono riusciti ad integrare nella loro quotidianità attività che altrimenti sarebbero state impossibili.

Un sistema simile è stato sviluppato più recentemente nell'Istituto Italiano di Tecnologia di Genova [2] nel quale è stato integrato anche un dispositivo oculometrico (eye tracker).

Il progetto a seguire [3], invece, utilizza una BCI non invasiva (basata su P300) in cooperazione con un robot umanoide e si pone lo scopo di interpretare le risposte del paziente alle domande poste dall'automa. In questo modo il sistema, schematizzato in figura 2.2, permette al paziente di esprimersi attraverso il solo ausilio della vista: le domande sono comunicate in maniera verbale dal robot mentre le risposte sono visualizzate su uno schermo e il paziente può selezionarle semplicemente guardandole. Il sistema, per come è stato progettato, permette al paziente anche di interagire con altre persone poiché il robot, nel momento in cui comunica verbalmente, instaura un'interazione semi-autonoma con le persone circostanti. L'automa, quindi, agisce sia da agente con cui comunicare che da telepresenza del paziente.

²Incapacità o difficoltà di articolazione dei suoni

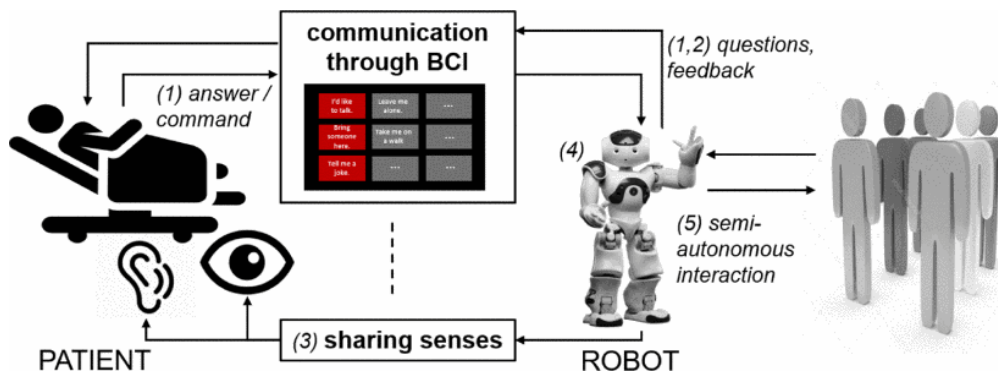


Figura 2.2. Sistema di comunicazione tra paziente e robot mediante una BCI

Un altro esempio di come una BCI possa essere utilizzata a supporto di pazienti con una mobilità limitata è l'utilizzo in collaborazione con un robot in grado di muoversi nello spazio circostante e di rilevare e manipolare oggetti [4].

2.1.2 Robotica industriale

Con l'avvento dell'industria 4.0, ovvero la nuova rivoluzione industriale incentrata principalmente sull'integrazione di nuove tecnologie per la creazione di sistemi ciberfisici (CPS) [5], il combinare forza lavoro umana e meccanica è divenuto uno dei principali obiettivi di ricerca per incrementare produttività e sicurezza nei settori industriali.

Come si vede in figura 2.3, un recente progetto dell'Università degli Studi di Pavia [6] ha permesso ad un operatore, mediante una BCI basata su SSVEP, di collaborare con un braccio meccanico in diverse fasi di un processo di assemblaggio di un profilo in alluminio.

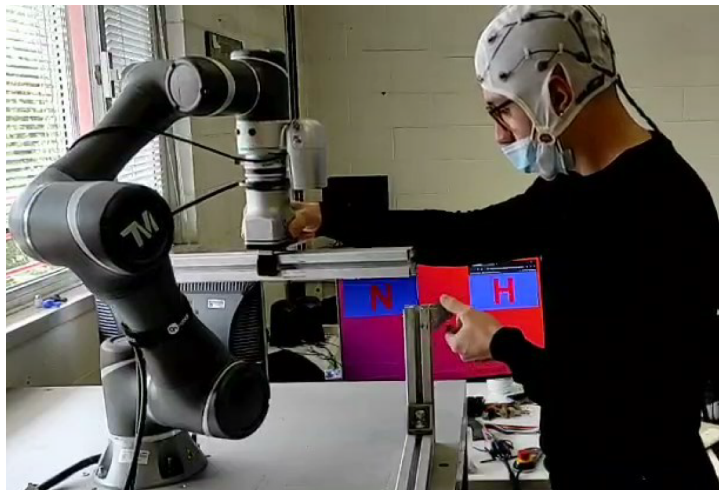


Figura 2.3. Collaborazione tra operatore e cobot mediante BCI

Applicazioni simili si possono trovare anche in studi precedenti [7][8][9][10] fino addirittura ad arrivare al 2000 con un celebre esperimento nel quale viene insegnato ad una scimmia a controllare un braccio meccanico utilizzando una BCI invasiva [11]. Di particolare interesse è un progetto del Politecnico di Torino [12] nel quale al controllo di un braccio robotico mediante BCI basata su SSVEP viene integrata la realtà aumentata (AR). Indossando il Microsoft HoloLens 2, dispositivo in figure 2.4 e 3.5 che permette di visualizzare in sovrapposizione al mondo reale degli oggetti virtuali, è infatti possibile per l'utente fornire istruzioni al robot durante l'assemblaggio di una morsa meccanica potendo visionare e manipolare nel contempo i singoli componenti.

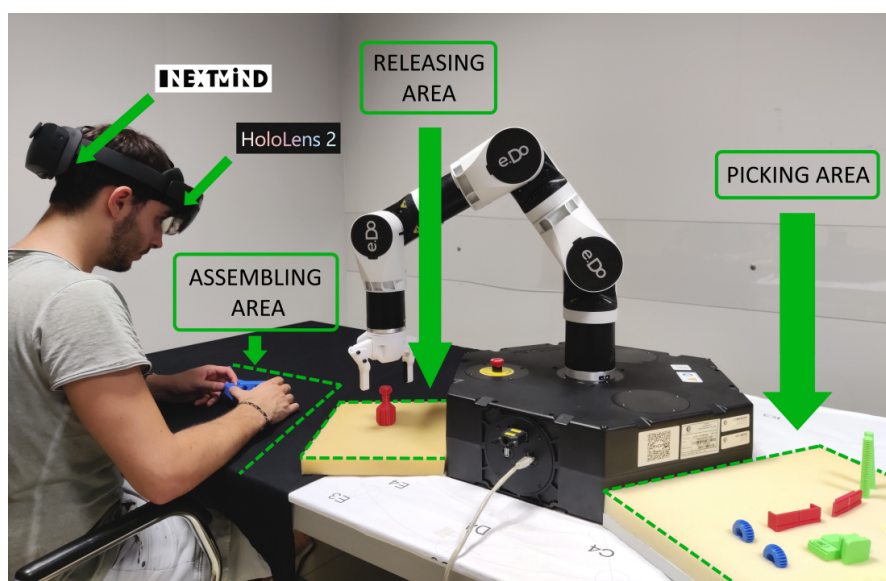


Figura 2.4. Sistema di collaborazione con interfaccia in AR

2.1.3 Robotica biomedica

La chirurgia robotica è sempre più in fase di espansione [13] grazie ai molti vantaggi in termini di precisione e affidabilità dei robot con i quali operazioni di elevata complessità si traducono in semplici ed accessibili. L'utilizzo di una BCI per il controllo di questi robot può mettere a disposizione dei chirurghi un metodo di selezione o di controllo diverso dalle mani che, durante un'operazione, è importante restino libere e sterili.

Le interfacce neurali possono, inoltre, essere utilizzate per il monitoraggio dell'attenzione, impiego approfondito da alcuni ricercatori dell'Istituto Italiano di Tecnologia di Genova che hanno condotto uno studio [14] utilizzando una BCI per incrementare le prestazioni e la sicurezza di operazioni chirurgiche monitorando la concentrazione degli utenti durante un'ablazione laser. La configurazione dell'esperimento è consultabile in figura 2.5.

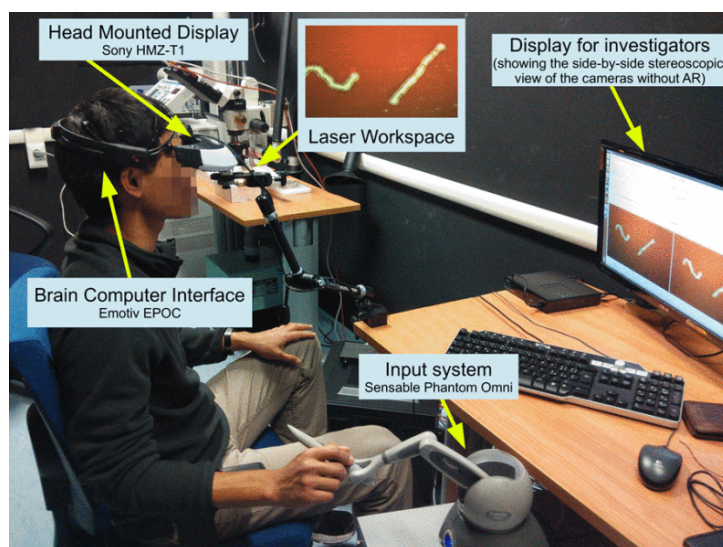


Figura 2.5. Configurazione dell'esperimento con monitoraggio dell'attenzione

Un'altra applicazione rilevante per quanto concerne la robotica in ambito medico è sicuramente l'utilizzo di esoscheletri o robot impiegati per la riabilitazione di pazienti con difficoltà motorie. Con la diffusione di BCI non invasive, infatti, sono diversi gli studi che si pongono come obiettivo quello di analizzare i percorsi riabilitativi di pazienti servendosi di interfacce neurali [15], anche integrando tecnologie immersive come la realtà virtuale (VR) [16]. Un esempio di notevole rilevanza, riportato in figura 2.6, è il seguente studio sulla neuroriabilitazione [17] nel quale viene analizzato l'utilizzo di Brain-Machine-Spinal-Cord Interface (BMSCI) e neuroprotesi su pazienti con lesioni al midollo spinale (SCI).

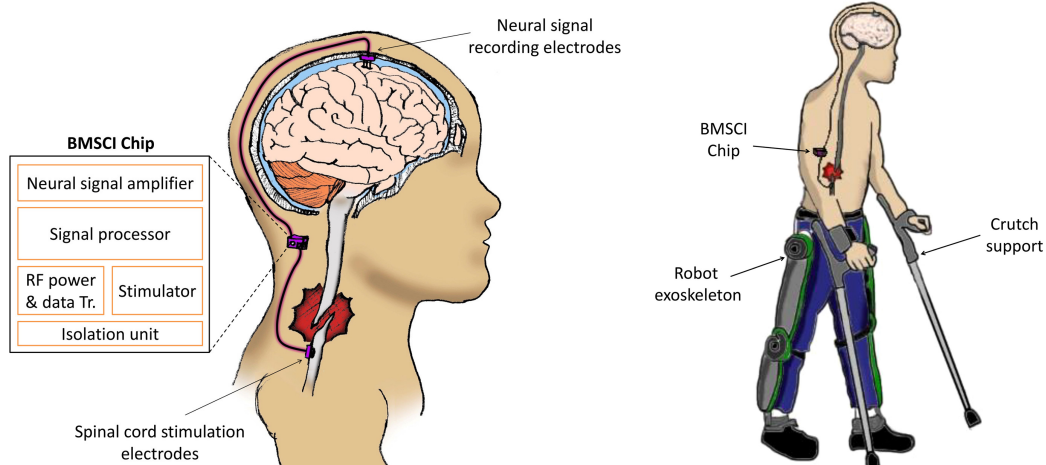


Figura 2.6. BMSCI per pazienti quadriplici (sx) e paraplegici (dx)

2.1.4 Domotica

L'utilizzo di oggetti intelligenti nelle case di tutto il mondo sta diventando gradualmente quotidianità per milioni di persone: è l'era dell'IoT. Spesso gli oggetti capaci di connettersi con il mondo esterno sono controllati da usuali dispositivi come cellulari o particolari telecomandi, facilmente reperibili e di basso costo. Con l'evolversi degli studi sulle BCI sono nati anche progetti nei quali la modalità di controllo degli oggetti appartenenti all'IoT si è evoluta ed identificata nelle interfacce neurali [18]. Molteplici sono i vantaggi di utilizzare una BCI come metodo di controllo della propria casa intelligente, soprattutto per gli utenti i quali presentano difficoltà motorie.

Un esempio di progetto per integrare le interfacce neurali con il mondo IoT consiste nell'equipaggiare i dispositivi intelligenti con delle scatole che fungono da indicatori [19]. Il sistema in questione, schematizzato in figura 2.7, è un impianto ibrido in quanto presenta un'unità cablata (unità computer-controller-indicatori) e un'unità wireless (unità BCI-computer).

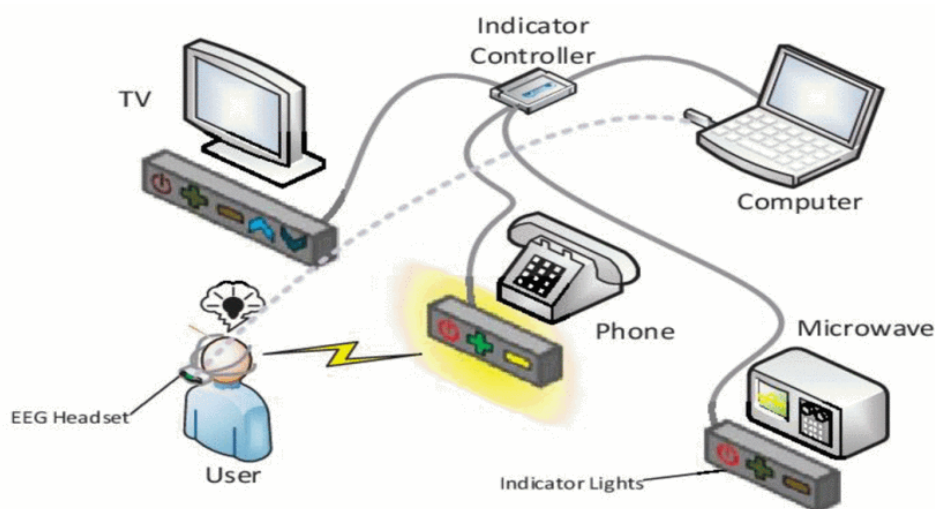


Figura 2.7. Sistema di impianto domotico controllato mediante BCI

Un'impostazione differente può essere data al sistema se, invece di utilizzare degli indicatori fisici, viene utilizzato uno schermo per visualizzare l'interfaccia necessaria al controllo degli oggetti intelligenti. In questo caso tutte le funzionalità del sistema risultano essere accessibili in un unico posto, grande vantaggio per utenti completamente impossibilitati al movimento [20].

2.1.5 Robotica militare

La robotica militare è uno degli ambiti meno sviluppati per quanto riguarda l'utilizzo di BCI ma, parallelamente, è un ambito in cui il potenziale d'uso è di rilevanza non trascurabile. Già dal 2013 la DARPA, ovvero l'agenzia governativa del Dipartimento della Difesa degli Stati Uniti incaricata dello sviluppo di nuove tecnologie per uso militare, ha iniziato a supportare la ricerca in campo neuroscientifico [21] anche attraverso progetti come la U.S. BRAIN Initiative. Scopo di questi progetti è, oltre allo studio attraverso interfacce neurali del cervello umano in ambito biomedico e militare, l'integrazione di BCI come equipaggiamento per missioni militari [22].

Risalente al 2012 è, invece, lo studio dell'attività del cervello delle sentinelle militari durante i turni di guardia mediante l'utilizzo di una BCI non invasiva basata su SSVEP [23]. Scopo dell'esperimento è il monitorare l'attività cerebrale per identificare e prevenire eventuali colpi di sonno durante il periodo di vedetta. Durante l'osservazione sono stati collocati due vettori di 3 LED ciascuno posizionati simmetricamente a 45° dall'asse di simmetria del volto del soggetto e ad una distanza di 10cm. I diodi, in primo piano rispetto all'ambiente che misurava una luminanza di $150\text{cd}/\text{m}^2$, sono stati fatti lampeggiare per la durata dei test alle frequenze di 15Hz e 20Hz e con una luminanza di $4000\text{cd}/\text{m}^2$.

Un'altra applicazione che negli ultimi anni ha suscitato interesse nel settore militare è l'utilizzo di interfacce neurali per il controllo di droni [24][25]. Come si evince dalla rappresentazione di figura 2.8, l'idea è quella di sfruttare gli impulsi cerebrali per permettere il controllo anche di sciame di droni (3 o più) con scopo di ispezione oppure incarichi militari collaborativi.

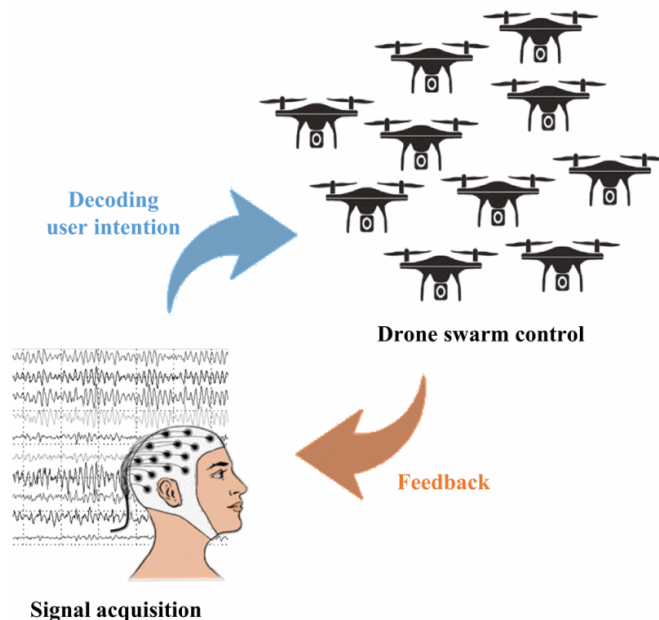


Figura 2.8. Sistema per il controllo di uno sciame di droni tramite BCI

2.1.6 Robotica d'intrattenimento

Il settore dell'intrattenimento, per quanto riguarda le interfacce neurali, non risulta essere un settore particolarmente maturo. Esistono tuttavia esempi in letteratura [26] di integrazioni di BCI a giochi a turni mediante l'utilizzo di bracci meccanici come, mostrato con intento esemplificativo in figura 2.9, gli scacchi. Risultano comunque nebulosi i vantaggi dell'utilizzo di interfacce neurali se non per utenti con difficoltà motorie.



Figura 2.9. Integrazione di una BCI nel gioco degli scacchi

Un gioco che al lancio nell'autunno del 2009 suscitò particolare interesse è Mindflex poiché integrava l'utilizzo di una BCI, motivo per cui apparve decisamente innovativo [27]. Lo scopo del gioco raffigurato in figura 2.10 è quello di concentrarsi per guidare una pallina attraverso un percorso ad ostacoli personalizzabile. I sensori indossati dall'utente, infatti, permettono di controllare una piccola ventola situata sotto la pallina che le consente di levitare in aria: più l'utente si concentra, più la pallina acquisisce quota.



Figura 2.10. Mindflex della Mattel

2.2 Dispositivo NextMind

Il NextMind è una rBCI non invasiva che utilizza la tecnologia EEG basata su SSVEP per rilevare l'attività cerebrale dalla corteccia visiva a seguito di stimoli visuali. Commercializzata nel 2020, l'interfaccia neurale è uscita dal mercato il 23 Marzo 2022 a seguito dell'acquisto della compagnia da parte della società americana Snap Inc.. In questo frangente di tempo il dispositivo si è distinto per essere, oltre che di particolare affidabilità e precisione [28], una più che valida BCI ad un costo accessibile a molti.

Uno studio recente dell'ISAE-SUPAERO³ di Tolosa ha utilizzato il NextMind (BCI reattiva) congiuntamente ad un'interfaccia neurale passiva con lo scopo di creare un concetto innovativo di BCI duale e testarlo in uno scenario d'uso realistico [29][30]. Con il termine "duale" si intende un sistema che integra entrambe le componenti reattiva e passiva per migliorare l'interazione uomo-macchina (HCI). I test utente si sono svolti in un simulatore di volo e tutti i partecipanti sono stati scelti tra studenti e personale dell'ISAE-SUPAERO i quali hanno dovuto eseguire alcune liste di controllo e manovre aeronautiche. Facendo riferimento alla figura 2.11, a sinistra in rosso è possibile identificare la sezione in cui i partecipanti dovevano controllare delle spie attraverso il NextMind (rBCI) mentre in blu è possibile individuare un radar anticollisione che i partecipanti dovevano sorvegliare periodicamente per evitare che altri aerei (rappresentati da cerchi grigi) raggiungessero il centro, sinonimo di collisione. L'utilizzo della pBCI si riassume nel monitoraggio dell'attenzione del partecipante: vicino al radar è infatti presente una spia che assume il colore verde quando il pilota è concentrato sul radar e il colore giallo se l'attenzione è troppo bassa. In questo modo, se la pBCI dovesse rilevare che il pilota non è concentrato sul radar e il velivolo si trovasse in rotta di collisione con un altro aereo, il sistema sarebbe in grado di attivare autonomamente la manovra anticollisione.



Figura 2.11. Test dell'interfaccia duale rBCI-pBCI

I risultati dei test hanno dimostrato che tutti i partecipanti sono riusciti a pilotare l'aereo ed eseguire le liste di controllo con successo. Inoltre, sul totale di potenziali collisioni, quelle non rilevate dai piloti sono state correttamente compensate dal sistema pBCI.

³École nationale supérieure de l'aéronautique et de l'espace

Come precedentemente citato [12], il NextMind è stato utilizzato in un progetto del Politecnico di Torino come dispositivo di interazione tra utente e cobot mediante l'impiego del Microsoft HoloLens 2 (tecnologia AR). In questo modo è stato possibile studiare un sistema BCI-AR e comparare l'interazione tramite interfaccia neurale e l'utilizzo di gestures. Si cita anche il seguente studio [31].

Un altro recente utilizzo del dispositivo si può riscontrare nello sviluppo di un'applicazione per il controllo di una casa e sedia a rotelle intelligenti [32]. Sebbene il progetto non sia stato testato in un caso d'uso reale ma solo simulato in un ambiente virtuale creato attraverso il motore di gioco Unity 3D (figura 2.12), risulta essere di particolare rilevanza a causa dei diversi vantaggi del sistema quali l'essere di costo contenuto, completamente wireless, portatile, facile da usare e di precisione elevata. Sono state effettuate tre serie di test con un totale di 15 partecipanti aventi l'obiettivo di collaudare il sistema di navigazione nell'ambiente virtuale. I test sono consistiti prevalentemente nell'instradamento della sedia a rotelle e nell'interazione con dispositivi intelligenti (figura 2.12 in basso) tramite un'interfaccia visualizzata su schermo; tutti i partecipanti sono riusciti a completare gli incarichi con un elevato tasso di successo.

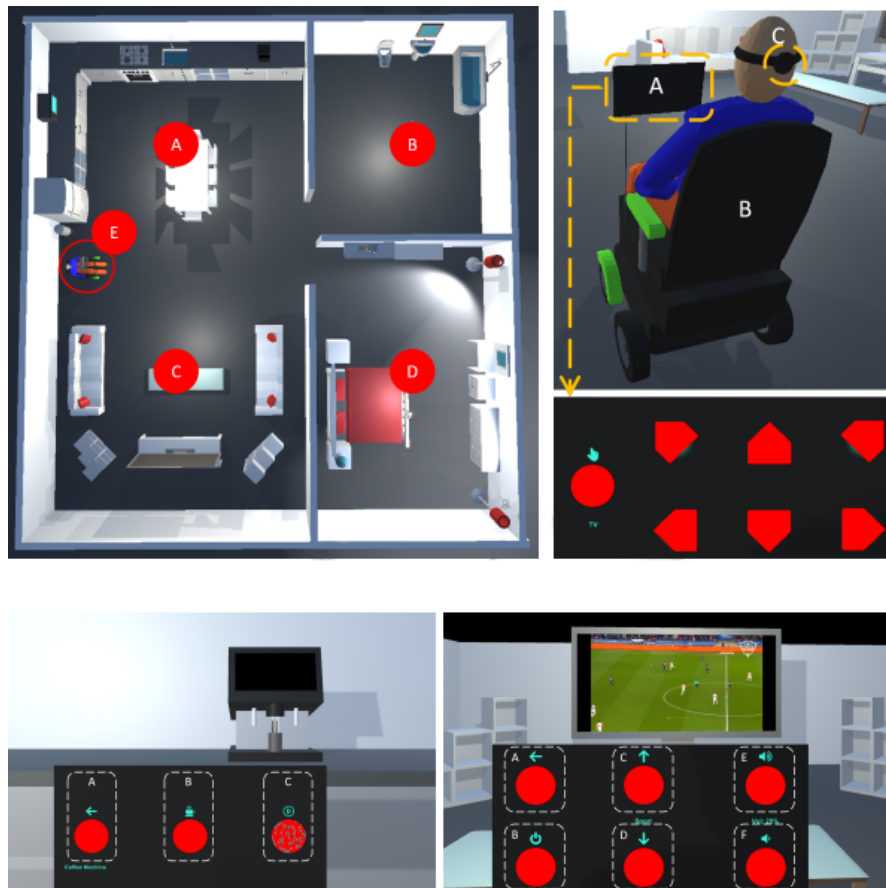


Figura 2.12. Sistema per casa intelligente controllato mediante NextMind

Infine il NextMind, oltre ad essere stato utilizzato in ambiti scientifici tra cui anche quello biomedico [33] per la riabilitazione di pazienti attraverso l'utilizzo di giochi a scopo formativo, è stato utilizzato anche in ambito artistico [34] per la realizzazione di una performance in Giappone.

2.3 Riepilogo

Come precedentemente affermato, sono individuabili diversi ambiti nei quali svariate tipologie di interfacce neurali sono state utilizzate come sistemi di interazione tra uomo e robot. In tabella 2.1 è presente un riepilogo dei principali progetti individuati in letteratura.

<i>Ambito</i>	<i>Tecnologia</i>	<i>Obiettivo</i>
Robotica di servizio	MI-BCI	Controllo di un braccio robotico
	P300 BCI	Interazione con robot umanoide
Robotica industriale	SSVEP-EEG-BCI	Assemblaggio industriale
	SSVEP-EEG-BCI	Controllo di un braccio robotico
Robotica biomedica	EEG-pBCI	Monitoraggio attenzione chirurghi
	BMSCI	Controllo di neuroprotesi
Domotica	EEG-BCI	Controllo di oggetti IoT
Robotica militare	SSVEP-EEG-BCI	Monitoraggio attenzione sentinelle
	EEG-BCI	Controllo di droni (stormi)
Robotica d'intrattenimento	EEG-BCI	Mindflex della Mattel

Tabella 2.1. Riepilogo dei principali progetti di HMI mediante BCI

È stata ulteriormente considerata con particolare interesse l'interfaccia neurale utilizzata per la creazione del sistema in oggetto. A seguire, un elenco dei principali utilizzi del dispositivo NextMind (SSVEP-EEG-rBCI):

- Controllo di un braccio robotico mediante l'HoloLens 2 (AR-HMD) per operazioni di selezione e posizionamento (pick and place).
- Creazione di un'interfaccia duale rBCI-pBCI per manovre aeronautiche.
- Realizzazione di un sistema di casa e sedia a rotelle intelligenti.
- Riabilitazione di pazienti tramite giochi a scopo formativo.
- Realizzazione di una performance artistica in Giappone.

Come precedentemente indicato, l'obiettivo principale del progetto sviluppato risulta essere quello di studiare un sistema che possa permettere ad utenti con limitazioni motorie o difficoltà di linguaggio di interagire con oggetti circostanti mediante l'ausilio di un cobot. L'aspetto innovativo del sistema si può individuare nell'interfaccia utente in AR proiettata poiché in letteratura non sono presenti progetti di HCI mediante BCI con UI simili.

Capitolo 3

Tecnologie

A seguire saranno introdotte le tecnologie hardware e software utilizzate per lo studio preliminare e per lo sviluppo del sistema in oggetto.

3.1 Tecnologie Hardware

3.1.1 NextMind

Brain Computer Interface

Le Brain Computer Interface (BCI), oppure Brain Machine Interface (BMI), sono delle interfacce che permettono la comunicazione tra cervello e macchine. Il funzionamento di questi dispositivi, di cui si può osservare un esempio in figura 3.1 [35], si può riassumere in cinque fasi: acquisizione del segnale, elaborazione del segnale, estrazione delle caratteristiche, classificazione delle caratteristiche e controllo dell'interfaccia [36].

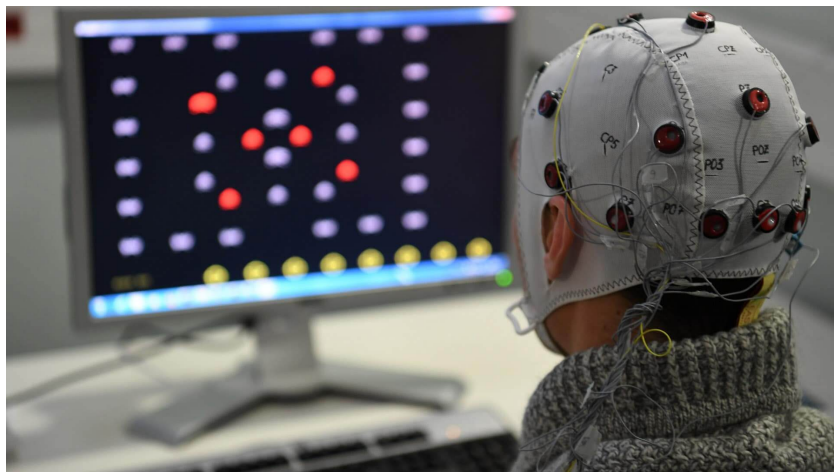


Figura 3.1. Esempio di BCI non invasiva

Esistono principalmente tre tipi di BCI:

- BCI invasive: gli elettrodi sono posizionati nella corteccia cerebrale misurando l'attività neuronale. La qualità del segnale è la più elevata (SNR alto) ma necessitano di un'operazione neurochirurgica per essere impiantate dalla quale possono derivare complicazioni post-operatorie come la formazione di tessuto cicatriziale o infezioni.
- BCI semi-invasive: gli elettrodi sono posizionati sulla superficie esposta del cervello (ECoG). Vantaggi dell'elettrocorticografia sono una maggior risoluzione spaziale, una maggior ampiezza, una buona resistenza del segnale ai rumori ed un rischio clinico minore oltre ad una miglior robustezza dell'impianto nel lungo periodo.
- BCI non invasive: i sensori sono posizionati sullo scalpo per misurare i potenziali elettrici prodotti dal cervello (EEG) o il campo magnetico (MEG). Ci sono diverse tecniche non invasive usate per la ricezione di segnali dal cervello:
 - Elettroencefalografia (EEG): misura l'attività elettrica del cervello dalla superficie dello scalpo. È la tecnica più usata per quanto riguarda le BCI non invasive e si può basare principalmente su tre diversi tipi di segnali: P300 (potenziali positivi che si manifestano circa 300ms dopo lo stimolo, fanno parte degli ERP), VEP (potenziali elettrici che si presentano solamente in presenza di stimoli esterni e sempre nelle stesse tempistiche e modalità, i più utilizzati sono gli SSVEP generati in risposta a treni di stimoli ripetuti alla stessa frequenza) e SMR (onde cerebrali generate quando le aree sensomotorie corrispondenti sono inattive, le MI-BCI sono un esempio).
 - Magnetoencefalografia (MEG): misura il campo magnetico generato dalle correnti nel cervello.
 - Tomografia ad emissione di positroni (PET): è una tecnica di imaging nucleare usata per osservare i neurotrasmettitori.
 - Risonanza magnetica funzionale (fMRI): misura le attività cerebrali identificando cambiamenti associati al flusso sanguigno.
 - Spettroscopia nel vicino infrarosso (NIRS): misura i cambiamenti del flusso sanguigno tramite risposte emodinamiche associate al comportamento neuronale. Presenta miglior risoluzione spaziale dell'EEG [37] e miglior risoluzione temporale della fMRI.

Un'ulteriore classificazione può essere individuata nel modo in cui viene esercitato il controllo sulla macchina: si possono identificare BCI attive, reattive e passive [38]. Le BCI attive e reattive permettono all'utente un'interazione basata sul proprio controllo volontario, quelle attive (aBCI) attraverso la stimolazione mentale di processi motori mentre quelle reattive (rBCI) attraverso la risposta cerebrale a stimoli visivi. Le BCI passive (pBCI), invece, misurano l'attività cerebrale permettendo di analizzare risposte psicofisiche come lo stress durante determinati compiti.

Dispositivo NextMind [39]

Dispositivo non invasivo di tipo reattivo basato sugli Steady State Visually Evoked Potentials (SSVEP), utilizza la tecnologia EEG integrando nove sensori da posizionare a contatto con lo scalpo sulla parte occipitale della testa necessari a rilevare l'attività cerebrale generata dalla stimolazione della corteccia visiva.



Figura 3.2. Sensore NextMind basato su tecnologia EEG-SSVEP

La rBCI in figura 3.2 permette l'interazione uomo-macchina (HMI) attraverso la semplice attenzione visuale verso un particolare componente chiamato NeuroTag. Come si può osservare in figura 3.3, un NeuroTag è un componente composto da uno sfondo a tinta unita, una texture intermittente e un feedback per comunicare all'utente lo stato dello stesso.

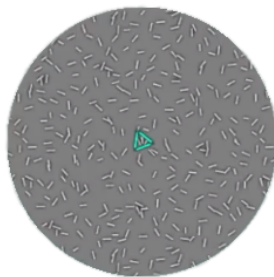


Figura 3.3. Esempio di NeuroTag

L'interfaccia neurale si collega tramite Bluetooth Low Energy (BLE) all'elaboratore nel quale deve essere in esecuzione l'applicazione che permette di visualizzare la UI contenente i NeuroTag. Attraverso algoritmi di machine learning, l'attività cerebrale viene decodificata ed associata ai relativi comandi che la macchina deve eseguire.

3.1.2 Stäubli TX2-60

Braccio robotico industriale e modulare caratterizzato da 6 gradi di libertà (DOF), permette una capacità di carico di 4.5Kg ed una portata massima di 920mm. Prodotto dall'azienda svizzera Stäubli International AG [40] con lo scopo di offrire un'elevata produttività, il robot, mostrato in figura 3.4, propone un'ottima combinazione di velocità, rigidità, dimensione e precisione.



Figura 3.4. Robot Stäubli TX2-60

Mediante un controller è possibile comandare il robot manualmente agendo sui singoli giunti o pilotando l'estremità (end effector) nello spazio lineare e rotazionale, alternativamente è anche possibile controllare il robot mediante messaggi ROS¹.

Di ultima generazione, il robot può essere utilizzato sia con scopo di produttività che con scopo di collaborazione (cobot), scopo per il quale introduce delle funzionalità per la sicurezza degli operatori:

- Safe limited speed: il robot diminuisce la velocità ($\leq 250\text{mm/s}$) quando l'operatore è in prossimità, riguadagnandola quando l'operatore si allontana.
- Safe stop: il robot si ferma durante l'interazione con l'operatore.
- Safe zones: è possibile definire delle zone sicure in aree predefinite.
- Safe tool: permette di supervisionare la dimensione e velocità di utensili e gripper.
- Safe touch: il robot reagisce in 10ms al contatto con l'operatore.

¹Sistema operativo per lo sviluppo e controllo di robot

3.1.3 HoloLens 2

Visore per realtà aumentata (HMD-AR) prodotto e sviluppato da Microsoft [41], permette di far visualizzare e manipolare all'utente degli ologrammi grazie a diverse tecnologie prima tra tutte la presenza di lenti olografiche trasparenti utilizzate come schermi. L'HMD, riportato in figura 3.5, implementa diversi sensori tra cui 2 telecamere ad infrarossi per il tracciamento degli occhi, 4 videocamere a luce visibile per il tracciamento della testa, un sensore di profondità e per il calcolo dell'orientamento un accelerometro, un giroscopio ed un magnetometro.

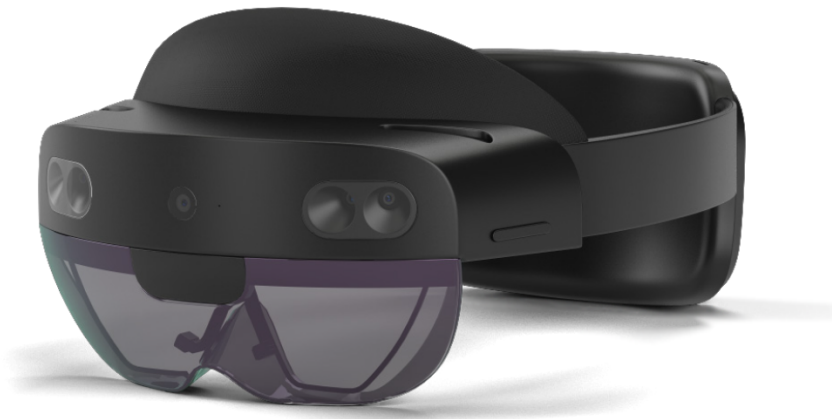


Figura 3.5. Microsoft HoloLens 2

L'interazione è possibile prevalentemente in 3 modi: attraverso il tracciamento delle mani (utilizzo di gesture²), il tracciamento degli occhi (oculometria) e, infine, attraverso comandi vocali.

Per disporre nell'ambiente reale degli oggetti virtuali, il dispositivo utilizza il sensore di profondità a tempo di volo (TOF) con l'obiettivo di mappare spazialmente l'area circostante creando una mesh³ ambientale in tempo reale.

Oltre ad un processore ed una HPU, l'HoloLens 2 integra un coprocessore dedicato esclusivamente alla gestione degli algoritmi di intelligenza artificiale utilizzati per processare i dati dei sensori in tempo reale ed estrarne le informazioni tridimensionali sull'ambiente circostante.

²Movimenti del corpo per il controllo di dispositivi tecnologici

³Reticolo poligonale composto da vertici, spigoli e facce

3.1.4 Oculus Rift

Visore per realtà virtuale (HMD-VR) sviluppato da Oculus VR, azienda acquistata da Meta [42] nel 2014, permette all'utente la totale immersione in un ambiente virtuale stereoscopico. Integra uno schermo OLED con una risoluzione di 2160x1200 pixel (1800x1200 pixel per occhio), una frequenza di aggiornamento di 90Hz e un angolo di visione di 110°. Presenta, inoltre, dei sensori per il calcolo dell'orientamento: un accelerometro, un magnetometro e un giroscopio.



Figura 3.6. Oculus Rift e controller Oculus Touch

Il sistema di figura 3.6, composto dal visore e dai due controller Oculus Touch (anche chiamati Half Moon), viene tracciato in maniera estremamente precisa grazie a due torrette all'infrarosso esterne che devono essere posizionate davanti all'area di movimento e permettono di ricostruire in tempo reale l'esatta posizione dell'utente. Il sistema di controllo Oculus Touch permette una tracciabilità a 360° delle mani e, grazie alla presenza di sensori tattili, è capace di registrare anche i semplici movimenti delle dita.

Oculus, inoltre, propone un'interfaccia di sistema (Oculus Home) che consente un accesso facilitato alle funzioni del visore oltre a monitorare lo stato del dispositivo e le sue connessioni. Si tratta di un Sistema Operativo che crea una sala virtuale nella quale l'utente può interagire con i vari menù e selezionare le diverse applicazioni presenti in libreria.

3.1.5 Gripper elettromagnetico

Il robot Stäubli TX2-60 utilizzato nel progetto, essendo modulare, non presenta un gripper⁴ predefinito. È stato, quindi, costruito utilizzando i seguenti componenti: una MCU Arduino Micro, un modulo Bluetooth, un elettromagnete, un relè e due batterie MN21 da 12V utilizzate per l'alimentazione. La figura 3.7 rappresenta schematicamente il circuito elettronico del gripper, il codice Arduino invece è consultabile in appendice A.

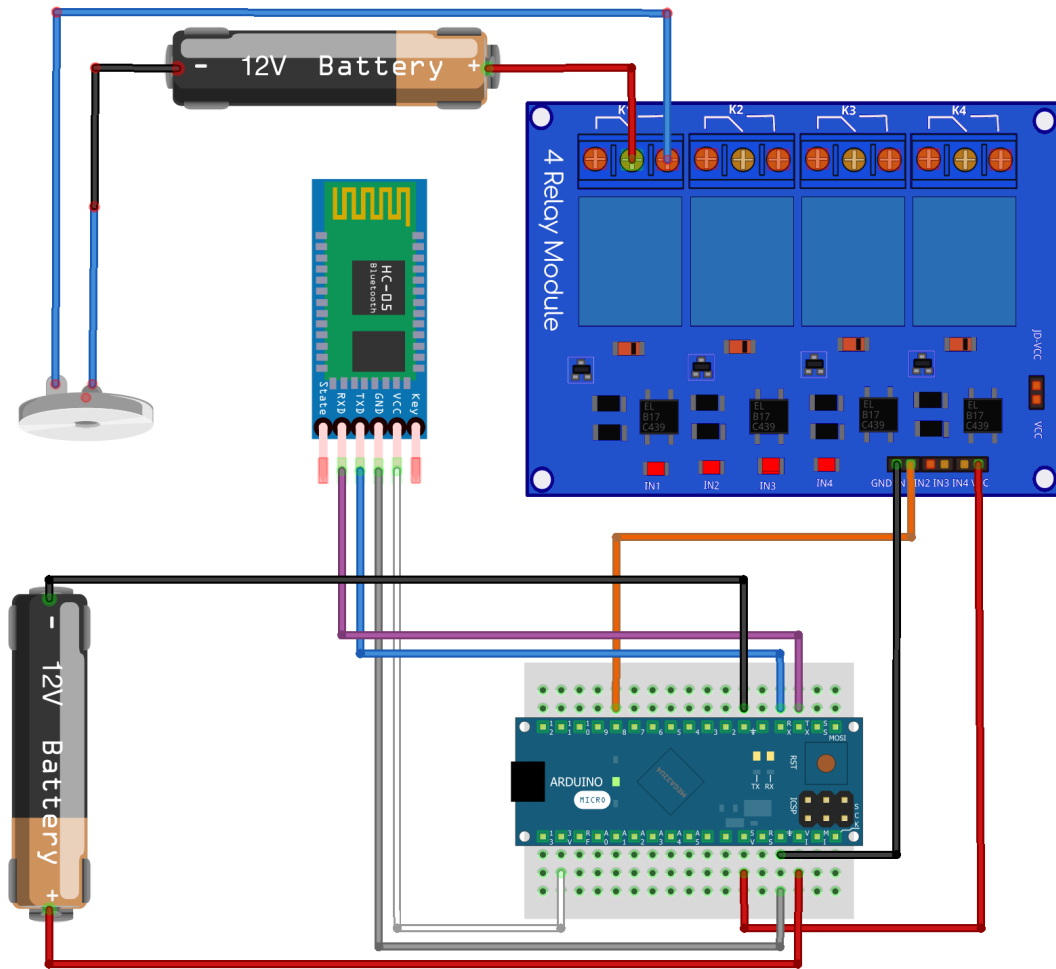


Figura 3.7. Circuito elettronico del gripper elettromagnetico

⁴Pinza o elettromagnete all'estremità del robot, utilizzato per afferrare oggetti o utensili

Arduino Micro

La scheda Arduino Micro, basata sul microcontrollore ATmega32U4 e mostrata in figura 3.8, è dotata di 20 pin di input/output digitali di cui 7 possono essere utilizzati come segnali PWM e 12 come input analogici. La tensione in ingresso consigliata è di 7-12V ed integra un quarzo che fornisce una frequenza di clock⁵ di 16MHz. La scheda complessivamente risulta essere molto piccola (48x18mm) e leggera, ideale per progetti in cui la dimensione è di rilevanza non trascurabile.

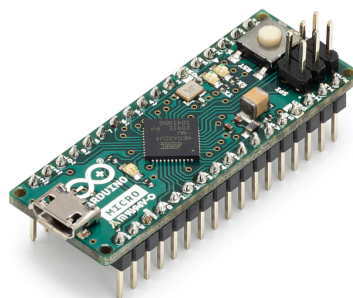


Figura 3.8. MCU Arduino Micro

Modulo Bluetooth

Il modulo Bluetooth HC-05 è un modulo che permette di convertire una porta seriale UART in una porta Bluetooth. Consente, quindi, ad un microprocessore (MCU) di comunicare con un dispositivo dotato di Bluetooth trasmettendo e ricevendo dati in tempo reale. Questo modulo, riportato in figura 3.9, è basato sul chip BlueCore4-Ext e può agire sia da Master che da Slave (a seconda delle esigenze) in modo tale da permettere la creazione di una rete di dispositivi utilizzando lo stesso tipo di modulo.

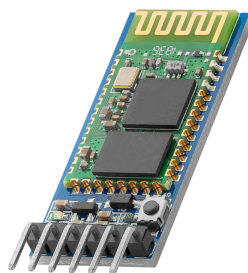


Figura 3.9. Modulo Bluetooth HC-05

⁵Segnale periodico utilizzato per sincronizzare il funzionamento di dispositivi elettronici digitali

Elettromagnete

Un elettromagnete è un componente composto da un nucleo in materiale ferromagnetico sul quale è avvolto un solenoide ed ha come scopo quello di generare un campo magnetico da una corrente elettrica. Il dispositivo mostrato in figura 3.10 si comporta, quindi, come una calamita controllata da un interruttore: quando l'interruttore è aperto non circola corrente e l'elettromagnete non genera un campo magnetico, quando invece l'interruttore è chiuso la corrente transita con conseguente generazione di un campo magnetico.



Figura 3.10. Elettromagnete 12V

Relè

Un relè è un interruttore elettromeccanico che mediante un impulso elettrico devia il flusso di corrente ed è composto da tre parti: una bobina, una ruota dentata ed un contatto elettrico. La ruota dentata è collegata al contatto ed ogni volta che viene erogata corrente al relè, viene mandato un impulso alla bobina che permette di ruotare di un piccolo movimento la ruota dentata la quale aprirà o chiuderà il contatto. In figura 3.11 è riportato il relè a 4 canali utilizzato per il progetto.

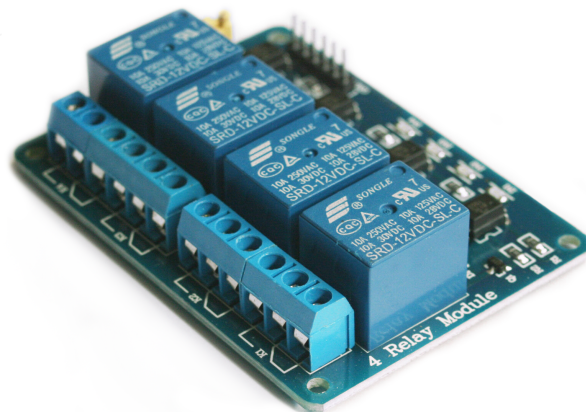


Figura 3.11. Relè a 4 canali

3.1.6 Proiettore laser

Il proiettore di figura 3.12, impiegato nel progetto al fine di visualizzare l'interfaccia utente (UI), utilizza un laser semiconduttore a lunga durata come sorgente di luce. Un grosso vantaggio nell'utilizzo di un proiettore laser piuttosto che un tradizionale proiettore con lampada risulta essere, oltre ad un consumo ridotto e ad una maggior velocità di accensione, una luminosità più intensa della proiezione.



Figura 3.12. Proiettore laser LG HF80LA

Il proiettore HF80LA prodotto da LG [43] utilizza la tecnologia Digital Light Processing (DLP), ovvero la creazione dell'immagine avviene mediante la modulazione della luce attraverso una schiera di milioni di specchi di dimensione micrometrica. Il risultato è un'immagine che risulta essere molto nitida e luminosa (fino a 2000Lumens) anche in ambienti con intensa luce solare o elettrica.

3.2 Tecnologie Software

3.2.1 Unity

Unity [44], la cui UI è mostrata in figura 3.13, è un motore grafico multiplatforma per lo sviluppo di applicazioni e videogiochi sviluppato da Unity Technologies e rilasciato nel 2005. Ha riscosso un notevole successo poiché, oltre ad essere un software gratuito, permette di sviluppare per molteplici piattaforme a partire dallo stesso codice sorgente.

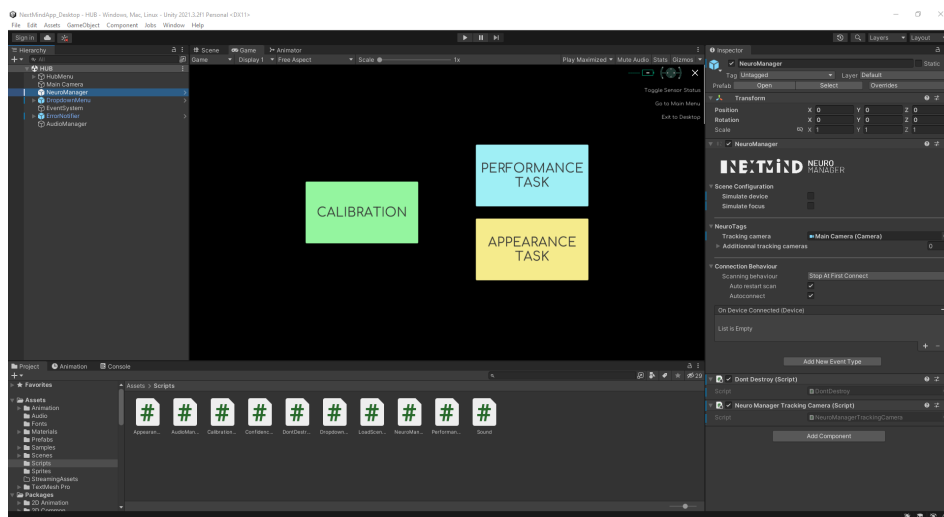


Figura 3.13. Interfaccia utente Unity

Il linguaggio di programmazione utilizzato è C# e le piattaforme supportate sono classificabili in 5 tipologie: piattaforme desktop (Windows, Mac, Linux, UWP), piattaforme mobile (iOS, Android, tvOS), console (PlayStation, Xbox, Nintendo Switch, Stadia), piattaforme VR/MR (Oculus, PlayStation VR, Windows Mixed Reality e altre) e piattaforme web (WebGL).

Nonostante in principio il programma supportasse fondamentalmente lo sviluppo di videogiochi, attualmente è largamente utilizzato anche per altri fini come simulazioni, controllo di robot oppure per la creazione di digital twin⁶ di ambienti ed oggetti. Inoltre, per l'agevolazione dello sviluppo di applicazioni, la piattaforma ha implementato l'Asset Store: un catalogo di script, modelli, SDK e template importabili direttamente nei progetti e adattabili a seconda dell'esigenza specifica.

Durante lo studio preliminare del progetto sono stati utilizzati il Mixed Reality Toolkit (MRTK) (per lo sviluppo AR sul Microsoft HoloLens 2) e l'Oculus XR Plugin (per lo sviluppo VR sull'Oculus Rift). Inoltre, per integrare il sensore NextMind è stato utilizzato il Software Development Kit (SDK) dedicato messo a disposizione dalla casa produttrice.

⁶Replica virtuale informatica di una risorsa fisica

3.2.2 Visual Studio

Microsoft Visual Studio [45] è un ambiente di sviluppo integrato (IDE) sviluppato da Microsoft e rilasciato nel 1997. Supporta la creazione di progetti per diverse piattaforme e permette l'utilizzo di molteplici linguaggi di programmazione tra cui C#, Visual Basic .Net, C++, Java e JavaScript.

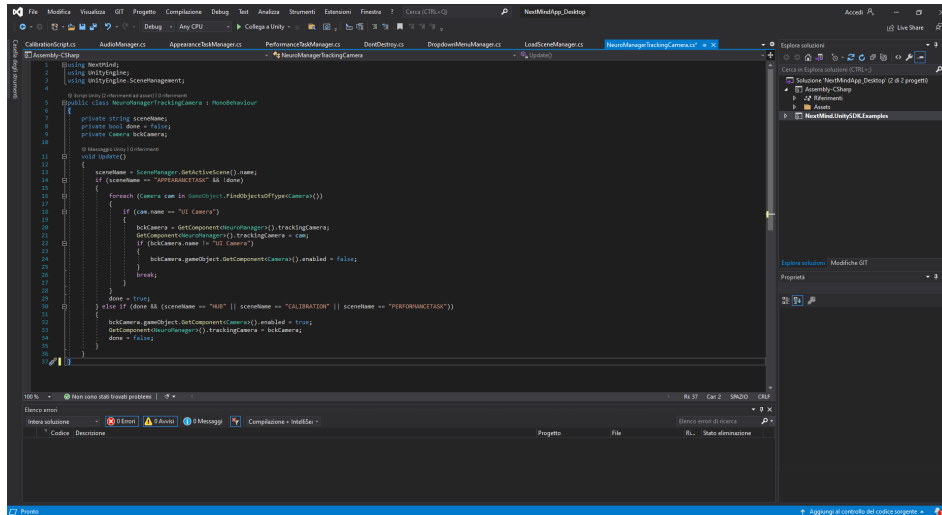


Figura 3.14. Interfaccia utente Visual Studio

L'IDE, oltre a fornire diversi strumenti per l'analisi delle prestazioni, integra la tecnologia IntelliSense che permette l'individuazione di errori sintattici e logici senza la necessità di compilazione del codice. Altre funzionalità che hanno permesso a Visual Studio di diventare uno degli ambienti di sviluppo più utilizzati al mondo sono, tra le tante, la presenza delle cosiddette azioni veloci per correggere rapidamente gli errori con soluzioni consigliate, la possibilità di rinominare variabili e funzioni applicando le modifiche in tutto il codice, la possibilità di visualizzare la definizione di tipi e funzioni e, infine, di cercare opzioni e proprietà tramite una barra di ricerca integrata.

Nel progetto, Visual Studio è stato utilizzato su Windows principalmente per la scrittura del codice responsabile del funzionamento delle applicazioni. È stato inoltre usato per compilare ed eseguire in fase di sviluppo l'applicazione per lo studio preliminare del NextMind su Microsoft HoloLens 2.

3.2.3 Visual Studio Code

Visual Studio Code [46] è un editor di codice sorgente sviluppato da Microsoft e rilasciato nel 2015. La differenza con il parente Visual Studio è che, essendo Visual Studio un IDE, integra la possibilità di creare programmi eseguibili mentre Visual Studio Code è un semplice editor di testo con diverse funzionalità di evidenziazione della sintassi. La UI, invece, risulta essere simile a quella di Visual Studio (figura 3.14).

Uno dei più grandi vantaggi di Visual Studio Code rispetto a molti altri editor di codice, oltre ad offrire anche una versione online, è l'implementazione di funzionalità comuni a Visual Studio come la tecnologia IntelliSense e la possibilità di correggere eventuali errori con soluzioni consigliate dal programma. È inoltre possibile programmare nella maggior parte dei linguaggi come la famiglia C, HTML, Python, PHP, Java, Ruby e altri ancora.

Nel progetto, Visual Studio Code è stato utilizzato su Linux per redigere lo script Python di color detection (rilevamento di colori per riconoscimento di oggetti).

3.2.4 MoveIt

MoveIt [47] è una piattaforma gratuita di manipolazione robotica che permette di sviluppare complesse applicazioni tramite l'utilizzo di ROS.

I principali vantaggi dell'utilizzo della piattaforma sono i seguenti:

- Pianificazione dei movimenti: la possibilità di pianificare e previsualizzare le traiettorie che il robot deve compiere è di notevole importanza per sviluppare un'applicazione che implementa la HCI.
- Manipolazione simulata: è possibile analizzare l'ambiente e interagire con gli oggetti virtuali attraverso una simulazione di presa.
- Cinematica inversa (IK): la piattaforma permette di calcolare mediante un solutore IK la soluzione (se esistente) della posizione dei giunti del robot data una posizione e orientamento nello spazio, essenziale per il calcolo delle traiettorie di movimento e per ottenere soluzioni efficienti.
- Controllo delle collisioni: utilizzando primitive geometriche, modelli o nuvole di punti, è possibile evitare gli ostacoli nell'ambiente durante i movimenti del robot.

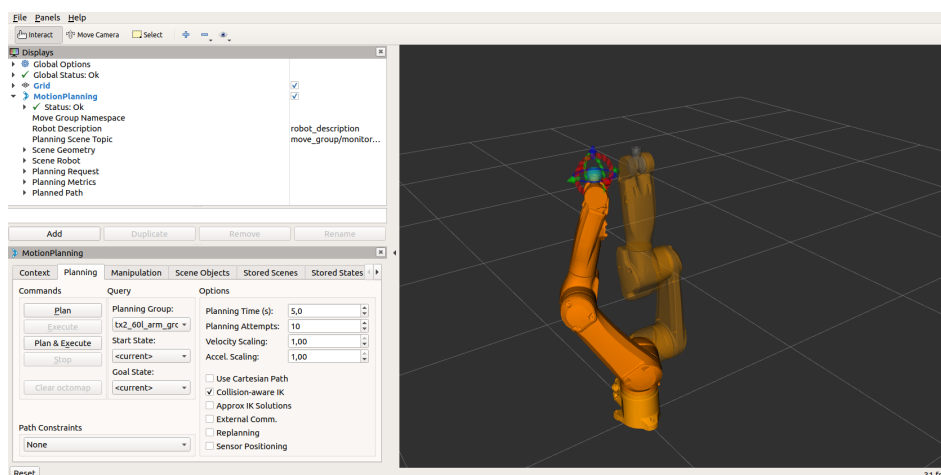


Figura 3.15. Interfaccia utente MoveIt

Nell'interfaccia utente, consultabile in figura 3.15, è possibile individuare una sezione dedicata ai settaggi del progetto e un'altra in cui è possibile previsualizzare e monitorare graficamente i movimenti del robot tramite l'utilizzo di modelli tridimensionali.

Grazie alle sue numerose funzionalità e algoritmi di pianificazione, sono numerose le aziende che utilizzano MoveIt tra le quali Google, Microsoft, NASA e Samsung.

La piattaforma è stata utilizzata nel progetto su Linux come nodo di un'infrastruttura ROS per calcolare i movimenti del robot attraverso il solutore IK integrato.

3.2.5 Arduino IDE

Applicazione multiplatforma in Java, permette di scrivere in maniera semplice del codice sorgente e di caricarlo nella memoria Flash delle schede Arduino o dei componenti compatibili. L'IDE [48] è fornito di una libreria software C/C++ chiamata Wiring ed offre la possibilità di scaricare ed includere altre librerie presenti online.



```

Blink $
This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
32 Arduino/Genuino Uno on COM1

```

Figura 3.16. Interfaccia utente Arduino IDE

L'ambiente di sviluppo integrato di Arduino è stato pensato principalmente per iniziare alla programmazione i neofiti, tanto da richiedere agli utenti esclusivamente la definizione di due sole funzioni (individuabili in figura 3.16) per la creazione di un file eseguibile:

- Setup: funzione invocata una sola volta all'inizio dello script, viene utilizzata per inizializzare i settaggi iniziali.
- Loop: funzione che viene invocata ripetutamente, contiene il corpo principale del codice.

Nel progetto, Arduino IDE è stato utilizzato su Windows per scrivere il codice sorgente (appendice A) del gripper elettromagnetico.

Capitolo 4

Studio preliminare

Antecedentemente allo sviluppo del sistema, è stato svolto uno studio preliminare al Politecnico di Torino con lo scopo di definire delle regole per lo sviluppo di un'interfaccia utente (UI) per un'applicazione che integri l'utilizzo della BCI NextMind [49]. Un altro importante proposito del progetto è stato quello di valutare se le prestazioni e l'usabilità dell'interfaccia neurale variassero in funzione del dispositivo di visualizzazione: per lo studio sono stati impiegati un monitor, un HMD-AR (Microsoft HoloLens 2) ed un HMD-VR (Oculus Rift).

4.1 Architettura dell'applicazione

L'applicazione, realizzata mediante l'utilizzo del motore grafico Unity integrato al SDK messo a disposizione da NextMind, permette di connettersi con il dispositivo per mezzo della tecnologia BLE ed è stata sviluppata per le tre piattaforme considerate. Permette una valutazione della UI tramite due specifici test: uno oggettivo in cui viene analizzata la prestazione dell'utente considerate diverse configurazioni di NeuroTag ed uno soggettivo in cui l'utente ha il compito di definire delle preferenze visuali sulla topologia dei NeuroTag. È costituita da quattro scene: Hub, Calibration, Performance task e Appearance task.

4.1.1 Hub

L'hub¹ è la scena iniziale contenente il menù di navigazione e i componenti essenziali per il funzionamento dell'applicazione. Come si può osservare in figura 4.1, presenta 3 pulsanti, i quali permettono all'utente di avviare le relative scene, ed un menù persistente a discesa. Tramite quest'ultimo, è possibile nascondere le informazioni sul sensore connesso all'applicazione (nome, percentuale di batteria e qualità dei contatti), caricare l'hub (utile se l'utente in un'altra scena volesse avviarne una diversa) ed uscire dall'applicazione.

¹In informatica, dispositivo che funge da nodo di smistamento dati

Il menù a discesa è uno dei componenti che vengono conservati tra le scene ma non risulta essere l'unico, gli altri componenti persistenti utilizzati dall'applicazione per il corretto funzionamento (ma che non partecipano alla UI visuale) sono i seguenti:

- NeuroManager: componente necessario per il corretto funzionamento del NextMind, gestisce i parametri di connessione dell'interfaccia neurale e permette di simulare il dispositivo tramite l'utilizzo del cursore.
- AudioManager: gestisce i feedback audio associati ai NeuroTag.
- Main Camera: è la camera che il NeuroManager utilizza per determinare quali NeuroTag sono visibili all'utente e, quindi, attivarli.
- EventSystem: gestisce gli eventi del sistema come le interazioni utente.
- ErrorNotifier: permette di notificare eventuali errori di connessione con il NextMind.

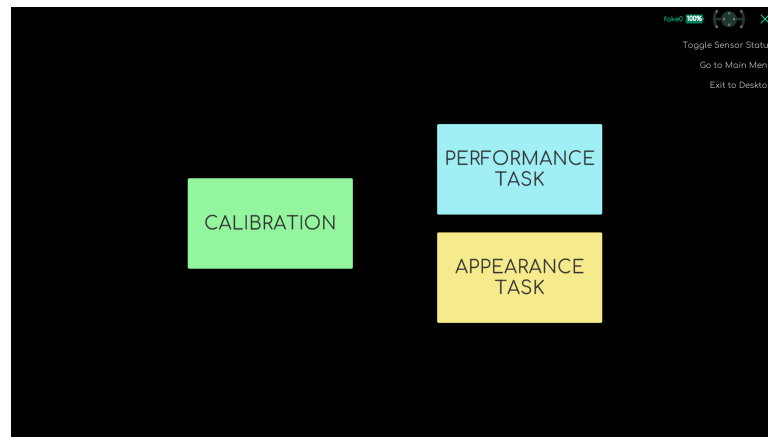


Figura 4.1. Interfaccia utente Hub

4.1.2 Calibration

La scena dedicata alla calibrazione del NextMind è composta da un vettore di pannelli che l'utente può scorrere tramite due pulsanti di navigazione. Attraverso questi pannelli è possibile inizializzare la connessione del dispositivo, guardare dei video guida su come indossarlo, controllare la qualità dei contatti ed, infine, avviare il processo di calibrazione.

Come mostrato in figura 4.2, per assicurarsi una buona qualità del segnale ricevuto dall'interfaccia neurale, l'applicazione mostra facendo uso dei colori quanto i sensori del dispositivo sono correttamente a contatto con lo scalpo dell'utente: il grigio o il rosso indicano una mancanza di contatto, il giallo una mediocre qualità mentre il verde attesta un'ottima qualità del segnale ricevuto. Inoltre, è anche assegnato un punteggio da 1 a 5 per riassumere la condizione del segnale trasmesso, dove 1 equivale ad un segnale molto debole e 5 ad un segnale ideale.

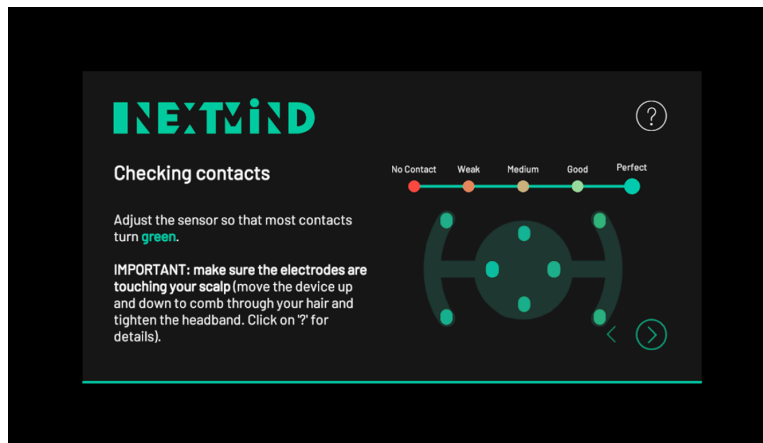


Figura 4.2. Pannello di calibrazione per il controllo della qualità dei contatti

Una volta ottenuto un punteggio accettabile nella fase precedente, è possibile iniziare la procedura di calibrazione. Questa consiste nel mantenere la concentrazione guardando un NeuroTag, mostrato in figura 4.3, il quale viene attivato ogniqualvolta il dispositivo dovesse rilevare il segnale elettrico prodotto dal cervello dell'utente. Il NeuroTag presenta una texture intermittente con frequenza variabile che permette al NextMind di adattarsi all'attività cerebrale dell'utente e, quindi, di migliorare le prestazioni generali dell'esperienza. Come ultimo evento, l'applicazione quantifica assegnando una valutazione da 1 a 5 (1 insufficiente, 5 ottimo) il prodotto finale della calibrazione.

Questo processo risulta essere di fondamentale importanza per la corretta interpretazione da parte della BCI del segnale ricevuto ed è necessario che vada a buon fine dal momento che, in caso contrario, l'interfaccia neurale risulterebbe notevolmente imprecisa.

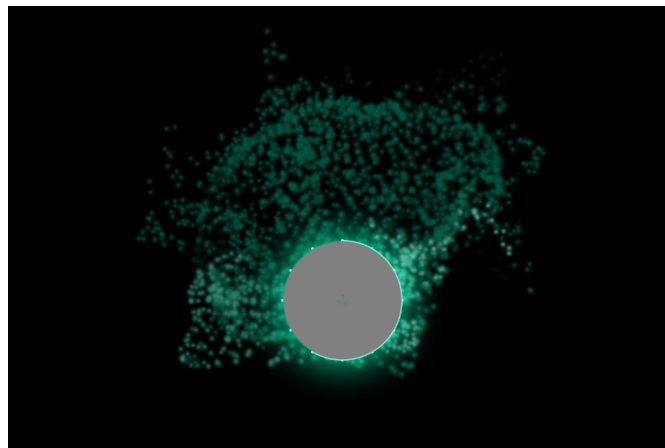


Figura 4.3. NeuroTag utilizzato per la calibrazione del NextMind

4.1.3 Performance task

È la scena dedicata all'analisi prestazionale oggettiva del NextMind in funzione di diverse configurazioni di NeuroTag. Nella schermata di settaggio del test, consultabile in figura 4.4, è possibile definire i seguenti parametri:

- **Confidenza:** percentuale che determina la sicurezza dell'algoritmo di IA per l'attivazione dei NeuroTag, i passi possibili sono 25%, 50%, 75% e 100%. Un valore basso equivale ad una maggior rapidità di selezione ma anche ad una presenza di falsi positivi più elevata mentre un valore alto equivale ad un sistema più robusto ma una maggior lentezza in termini di HCI.
- **Quantità:** permette di definire il numero di NeuroTag presenti durante il test, da un minimo di 2 ad un massimo di 10 (numero massimo di NeuroTag che il NextMind permette di attivare contemporaneamente).
- **Dimensione:** definisce la scala dei NeuroTag tramite un numero intero da 1 (NeuroTag piccoli) a 5 (NeuroTag grandi).
- **Distanza:** determina la lontananza spaziale tra i NeuroTag, sono possibili valori da 1 (NeuroTag attaccati) a 10 (lontananza massima ottenibile dalla configurazione).

Oltre ai parametri sopraelencati, è possibile abilitare o disabilitare cliccando sui relativi pulsanti circolari due ulteriori funzionalità: Maintained Trigger e Repetitions Code. Il primo permette all'utente di selezionare uno stesso NeuroTag più volte consecutivamente senza la necessità di distogliere lo sguardo (il NeuroTag è nuovamente innescato dopo 3 secondi di attivazione continua) mentre il secondo consente ripetizioni nel codice generato per il test. Queste funzionalità sono state introdotte per permettere una maggiore flessibilità del sistema ma non sono state utilizzate durante gli esperimenti.

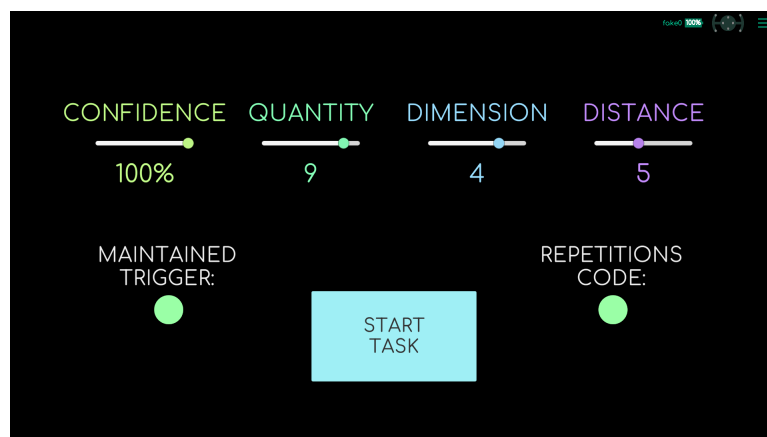


Figura 4.4. Schermata di configurazione della performance task

Una volta definita una configurazione è possibile iniziare il test. Come mostrato in figura 4.5, il test consiste nell'immissione di un codice numerico di 10 cifre mediante la

BCI: non appena il NeuroTag corretto viene attivato, questo si colora temporaneamente di verde emettendo un suono di corretta selezione e la relativa cifra nel codice diventa permanentemente verde, se invece viene selezionato un NeuroTag diverso da quello corretto (a causa di un errore da parte dell'utente o di un falso positivo) il NeuroTag si colora temporaneamente di rosso ed emette un suono di errata selezione. La presenza dei feedback visuali e sonori è di particolare importanza per permettere all'utente di percepire se ha correttamente selezionato il NeuroTag valido o se ha commesso un errore.

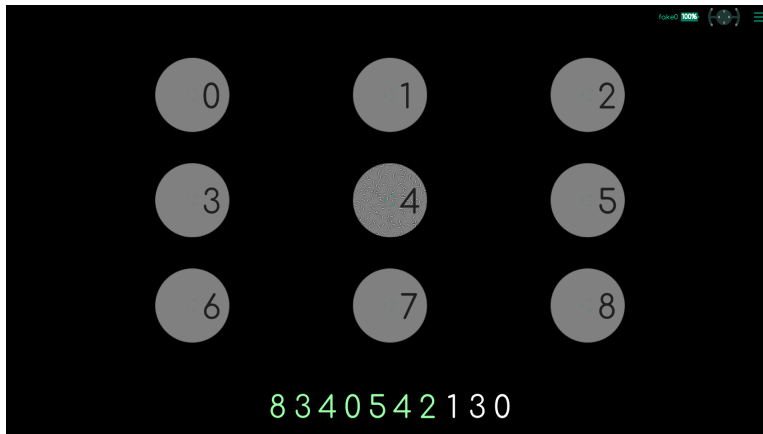


Figura 4.5. Performance task

Non appena immessa l'ultima cifra del codice numerico dell'esperimento, il sistema segnala gli errori complessivi del test e il tempo che l'utente ha impiegato per immettere tutte le cifre. Come mostrato in figura 4.6, vengono inoltre visualizzati i parametri inseriti per l'esperimento corrente e, tramite i pulsanti relativi, è possibile tornare alla schermata di configurazione dell'esperimento oppure al menù principale.



Figura 4.6. Schermata risultati della performance task

4.1.4 Appearance task

È la scena progettata con lo scopo di analizzare in maniera soggettiva l'utilizzo del NextMind: risulta infatti possibile modificare l'estetica dei NeuroTag per individuare i componenti più efficaci per quanto riguarda l'interazione e l'esperienza dell'utente (UX). Visualizzata in figura 4.7, la schermata per la personalizzazione dell'aspetto dei NeuroTag, oltre alla possibilità di scelta della forma, permette la personalizzazione dei seguenti componenti:

- Feedback: definisce il riscontro visuale all'attivazione dei NeuroTag.
- Materiale: modifica l'opacità dei NeuroTag.
- Texture: è l'immagine ad intermittenza che permette alla BCI tramite la tecnologia EEG-SSVEP di attivare i NeuroTag.

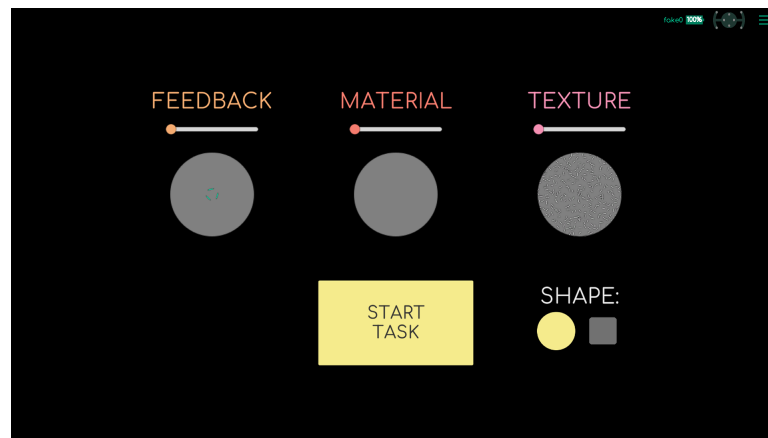


Figura 4.7. Schermata di configurazione dell'appearance task

In figura 4.8 sono identificabili le possibili personalizzazioni per ogni componente. Le tipologie di feedback si distinguono in: tre linee convergenti in un triangolo, un bagliore verde che si accende all'attivazione dei NeuroTag ed una barra di caricamento che permette di visualizzare il livello di confidenza dell'algoritmo di IA. Per il materiale, il valore del canale alfa² può essere 0.33 (molto trasparente), 0.66 (mediamente visibile) oppure 1 (completamente opaco). Sono infine state create due texture prendendo come ispirazione quella predefinita (linee corte e caotiche): una composta da piccoli cerchi ed una composta da righe lunghe ed incidenti.

²Descrive il valore di trasparenza dei pixel di un'immagine digitale

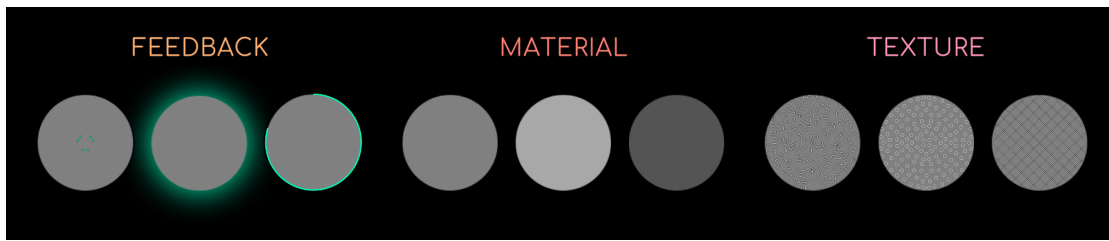


Figura 4.8. Possibili personalizzazioni dei NeuroTag

Durante l'esperienza, raffigurata in figura 4.9, l'utente può interagire con una scena contenente quattro cubi tramite dieci diversi NeuroTag. I quattro NeuroTag superiori si comportano come interruttori e permettono di accendere e spegnere i cubi luminosi attraverso un'animazione, i quattro NeuroTag inferiori, mantenendo la concentrazione in modo continuo, permettono di cambiare il colore dei relativi cubi mentre i due NeuroTag laterali permettono, sempre mantenendo la concentrazione, di girare i cubi accesi in senso orario (NeuroTag di sinistra) o antiorario (NeuroTag di destra). È inoltre presente un riscontro audio diverso per i quattro NeuroTag superiori (suono di attivazione) e per i restanti sei NeuroTag (suono di continua attivazione riprodotto in ciclo) così da comunicare all'utente, anche attraverso il senso dell'udito, informazioni riguardanti l'attivazione dei NeuroTag. Sono, infine, presenti due pulsanti appositamente mimetizzati negli angoli inferiori dello schermo (per non contaminare l'esperienza) che l'utente può utilizzare per tornare alla personalizzazione dei NeuroTag oppure per tornare al menù principale.



Figura 4.9. Appearance task

4.2 Test

L'applicazione, sviluppata per essere particolarmente flessibile, permette la modifica di numerosi parametri generando una molteplicità ingente di combinazioni. Per questa ragione, constatato che il testare tutte le combinazioni sarebbe stato molto impegnativo per gli utenti ed estremamente dispendioso in termini di tempo, sono state svolte delle valutazioni precedenti ai test da esperti del settore con lo scopo di definire delle configurazioni più rilevanti ed eventualmente escludere dei parametri marginali.

4.2.1 Valutazione degli esperti

Sono stati coinvolti 3 ricercatori in ambito UI, AR e VR del Politecnico di Torino. Dopo aver calibrato con successo il dispositivo, è stato richiesto loro di sperimentare più parametri possibili su tutti e tre i dispositivi di visualizzazione in esame.

Per quanto riguarda l'appearance task, tutti e tre hanno concordato su come la trasparenza influenzi negativamente l'interazione con i NeuroTag poiché ne riduce il contrasto (fondamentale per le interfacce neurali basate sui SSVEP) ed aumenti il tasso di falsi negativi³. La texture predefinita (linee corte e caotiche) e il feedback predefinito (tre linee convergenti in un triangolo) sono stati considerati i più efficaci per tutte le interfacce, motivo per cui è stato deciso di utilizzarli per la calibrazione (in aggiunta al bagliore) e per la performance task (senza il bagliore) durante i test di usabilità. Anche il terzo feedback (barra di caricamento) è stato apprezzato poiché fornisce un preciso riscontro visuale di quello che la BCI decodifica, è stato fatto notare però che potrebbe risultare distraente all'utente durante compiti in cui la concentrazione risulta essere un fattore fondamentale.

Relativamente alla performance task, invece, è stato notato come il decremento della percentuale di confidenza dell'algoritmo di IA infici negativamente sul tasso di falsi positivi⁴ e, di conseguenza, è stato deciso di mantenere il livello di confidenza al 100% durante i test. Sono inoltre state proposte cinque configurazioni, descritte in tabella 4.1, tali da sintetizzare al meglio le molteplici configurazioni possibili.

Alla fine, ognuno dei tre ricercatori ha preferito un dispositivo di visualizzazione diverso, circostanza che enfatizza la necessità di un'ulteriore valutazione dei tre dispositivi.

	<i>Config. 1</i>	<i>Config. 2</i>	<i>Config. 3</i>	<i>Config. 4</i>	<i>Config. 5</i>
<i>Confidence</i>	100%	100%	100%	100%	100%
<i>Quantity</i>	9	6	6	9	9
<i>Dimension</i>	5	3	1	2	5
<i>Distance</i>	10	6	10	4	1

Tabella 4.1. Configurazioni testate nella performance task

³Si verificano quando il sistema non risponde all'intento dell'utente di attivare un NeuroTag

⁴Si verificano quando il sistema innesca un NeuroTag non selezionato dall'utente

4.2.2 Test di usabilità

Successivamente alle valutazioni degli esperti, è stata testata l'usabilità della BCI integrata ai dispositivi di visualizzazione prendendo in considerazione i commenti espressi dai ricercatori. Il gruppo adibito alla sperimentazione consta di dodici studenti dai venti ai trent'anni, nove della facoltà di ingegneria del Politecnico di Torino e tre della TU Wien. Il gruppo risulta essere inoltre eterogeneo in termini di esperienza con i dispositivi AR e VR e composto sia da partecipanti di sesso maschile che di sesso femminile.

Eccetto gli ultimi tre, ogni tester ha eseguito sia la performance task che l'appearance task per ognuno dei tre dispositivi di visualizzazione mostrati in figura 4.10 e, dopo l'utilizzo di ogni singolo dispositivo, ha provveduto a compilare il questionario SUS⁵. Sono stati attuati alcuni accorgimenti come lo spegnimento della luce durante l'utilizzo del monitor e l'orientamento dell'utente verso un muro bianco durante l'utilizzo del Microsoft HoloLens 2 con l'obiettivo di migliorare le prestazioni del NextMind. Inoltre, l'ordine dei dispositivi testati è stato variato per ogni utente con lo scopo di evitare qualsiasi tipo di imparità dovuta alla dimestichezza dell'utente con la BCI che, nella maggioranza dei casi, aumenta con l'utilizzo. L'interfaccia neurale è stata calibrata prima delle prove su ogni dispositivo assicurandosi che la qualità della calibrazione fosse sempre accettabile per non contaminare i risultati. Alla conclusione di ogni sessione di test è stato infine chiesto agli utenti un riscontro su tutta l'esperienza e se avessero preferito qualche interfaccia in particolare o meno.



Figura 4.10. Tester durante l'utilizzo dei tre dispositivi di visualizzazione

Appearance task

Per l'appearance task sono stati dati cinque minuti ad ogni utente (su ogni dispositivo) per eseguire una serie prestabilita di azioni, permettendogli di modificare forma e componenti dei NeuroTag (feedback, materiale e texture) con lo scopo di stabilire la configurazione migliore in termini di usabilità. Le preferenze degli utenti di ogni singolo parametro sono state appuntate alla fine dei cinque minuti.

⁵Questionario composto da 10 domande utilizzato per valutare la soddisfazione dell'utente nell'utilizzo del sistema

Performance task

Per la performance task è stato richiesto agli utenti di immettere cinque codici di dieci cifre casuali mediante l'utilizzo del NextMind. Ad ogni codice, la configurazione dei NeuroTag in termini di numero, dimensione e distanza è stata variata secondo le configurazioni in tabella 4.1. Al termine di ognuna delle cinque prove sono stati registrati il numero di falsi positivi e il tempo impiegato per il completamento della prova.

4.3 Risultati

Inerentemente all'appearance task, tutti i feedback sono stati apprezzati con una leggera preferenza per quello predefinito (tre linee convergenti in un triangolo), il materiale più efficace è risultato essere quello completamente opaco mentre la texture che ha riscosso il maggior successo è la predefinita (linee corte e caotiche) con qualche eccezione in presenza di difficoltà di utilizzo da parte di alcuni utenti sui dispositivi AR e VR.

Per quanto riguarda la performance task, la tabella 4.2 mostra gli errori medi (E) e i tempi medi (T) della prova per ogni configurazione ($C1-C5$) su tutti i dispositivi di visualizzazione in esame (MO : monitor, AR : HoloLens 2, VR : Oculus Rift).

	$C1$		$C2$		$C3$		$C4$		$C5$	
	E	T	E	T	E	T	E	T	E	T
MO	0,83	50,92	0,42	46,42	0,64	56,73	1,00	40,00	0,83	38,50
AR	0,20	98,20	0,50	68,60	0,44	108,00	1,11	70,78	0,50	57,10
VR	0,00	64,14	0,57	71,00	0,33	81,33	0,83	57,17	1,14	40,14

Tabella 4.2. Errori e tempi medi di ogni configurazione

In media, il sistema visualizzato su monitor presenta il minor tempo medio per tutte le configurazioni mentre il sistema visualizzato in VR tramite l'Oculus Rift presenta un numero medio di errori minore in tre delle cinque configurazioni. La configurazione 2, mostrata in figura 4.11 a sinistra e caratterizzata da tutti parametri intermedi (quantità, dimensione e distanza dei NeuroTag), si è dimostrata la più solida agli errori (considerando che non tutti i tester sono riusciti a completare le prove con tutti e tre i dispositivi come spiegato in seguito) mentre la configurazione 5, mostrata in figura 4.11 a destra e caratterizzata da un'elevata quantità di NeuroTag con dimensione massima e posizionati alla distanza minima, si è attestata essere la più veloce con cui interagire. Dal momento che il tempo medio della seconda configurazione non è di molto maggiore rispetto a quello della quinta configurazione, è possibile affermare che la configurazione 2 sia la migliore per minimizzare gli errori senza incidere eccessivamente sulla rapidità di selezione.

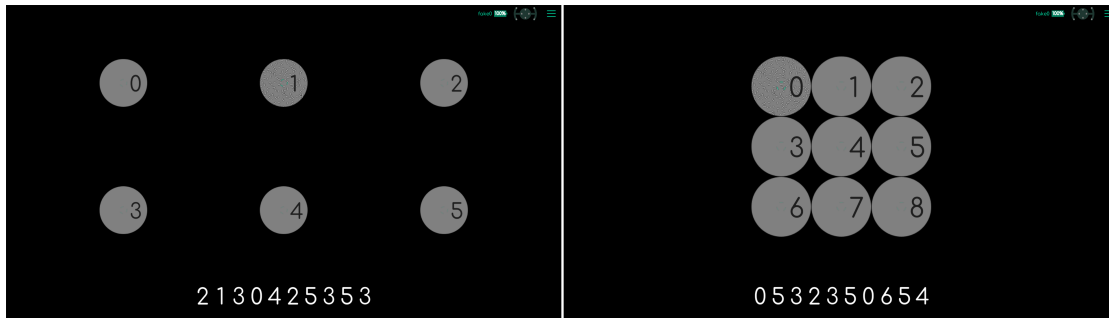


Figura 4.11. Configurazioni 2 (sinistra) e 5 (destra) della performance task

Relativamente alle valutazioni soggettive dei tre dispositivi di visualizzazione, sono presentati i risultati del questionario SUS per ogni utente ($U1-U12$) nella tabella 4.3 a seguire. È inoltre riportato il valore medio di soddisfazione per ogni dispositivo nella colonna *AVG*: si noti come il monitor sia stato il sistema preferito con una valutazione tra buono ed eccellente mentre il sistema AR è risultato essere il peggiore. Per quanto riguarda tre dei primi nove utenti, i test contraddistinti da caselle bianche sono, come precedentemente citato, i pochi che non sono stati portati a termine per problemi di calibrazione causati dall'indossabilità non universale dei sistemi NextMind-HoloLens2 e NextMind-OculusRift. Gli ultimi tre utenti, invece, non hanno provato il sistema in VR poiché, oltre ad essere gli unici a testare il sistema alla TU Wien di Vienna, non si è notata una differenza ingente tra monitor e Oculus Rift durante i nove test precedenti.

	$U1$	$U2$	$U3$	$U4$	$U5$	$U6$	$U7$	$U8$	$U9$	$U10$	$U11$	$U12$	<i>AVG</i>
<i>MO</i>	100	95	77,5	45	90	77,5	97,5	90	62,5	81,3	87,5	56,3	80
<i>AR</i>	37,5	85	37,5	47,5	35		95	77,5		50	59,4	34,4	55,9
<i>VR</i>	72,5	77,5	57,5		62,5	77,5	60	97,5					72,1

Tabella 4.3. Risultati del questionario SUS per ogni utente

Appurato che la differenza tra il monitor e il sistema AR è notevole, è stato eseguito in seguito un two-sample t-test⁶ per verificare se il monitor è statisticamente migliore del sistema AR. Il valore p ⁷ risultate per una distribuzione a due code è 0.012872 e, di conseguenza, è possibile escludere l'ipotesi nulla con un intervallo di confidenza del 95%.

Per concludere, l'esperienza complessiva è stata valutata positivamente dagli utenti nella sezione dei commenti finali ed il sistema è risultato essere solido.

⁶Test statistico utilizzato per verificare se tra le medie di due gruppi di dati sono presenti differenze statistiche significative

⁷Livello di significatività osservato

Capitolo 5

Sistema BCI-Robot

Il sistema progettato e sviluppato all'Università tecnica di Vienna permette, tramite l'utilizzo di una BCI, il controllo di un braccio robotico per operazioni di selezione e posizionamento (pick and place). Consiste nel rilevamento di oggetti posizionati in uno spazio di lavoro raggiungibile dal cobot intorno ai quali vengono visualizzati, tramite proiezione laser, i bounding box¹ associati a dei NeuroTag per permettere la selezione tramite il NextMind. Una volta selezionato un NeuroTag, il braccio meccanico provvede ad effettuare un'operazione di raccolta dell'oggetto associato e di spostamento dello stesso in un'area di deposito. In figura 5.1 è possibile visionare il robot e la UI proiettata.

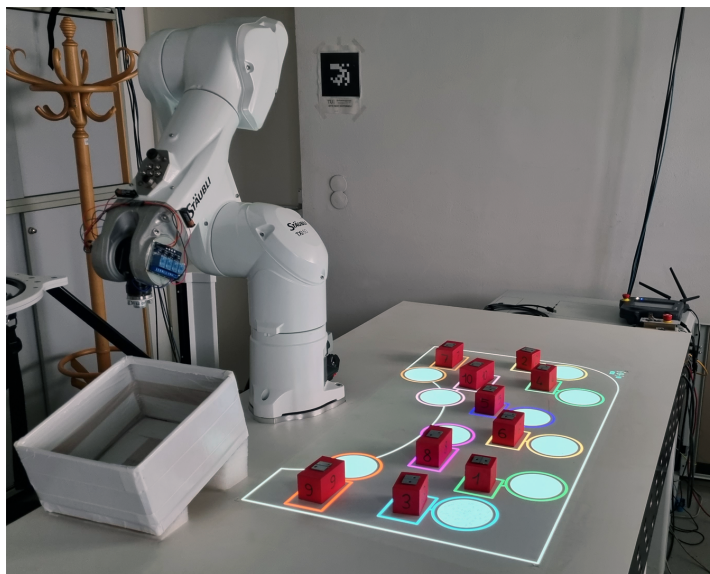


Figura 5.1. Robot, area di deposito e interfaccia utente del sistema

¹Rettangoli con area minima che permettono la delimitazione degli oggetti

Successivamente alla fase di calibrazione del NextMind e della videocamera per il rilevamento degli oggetti, è possibile eseguire due diversi algoritmi di posizionamento dei NeuroTag: il primo prova ad adattarne la disposizione in funzione dell'ubicazione degli oggetti mentre il secondo si limita a collocarli in un'area predefinita della UI. Il sistema provvede, inoltre, ad aggiornare l'interfaccia utente proiettata rimuovendo i bounding box ed i NeuroTag associati agli oggetti rimossi dallo spazio di lavoro.

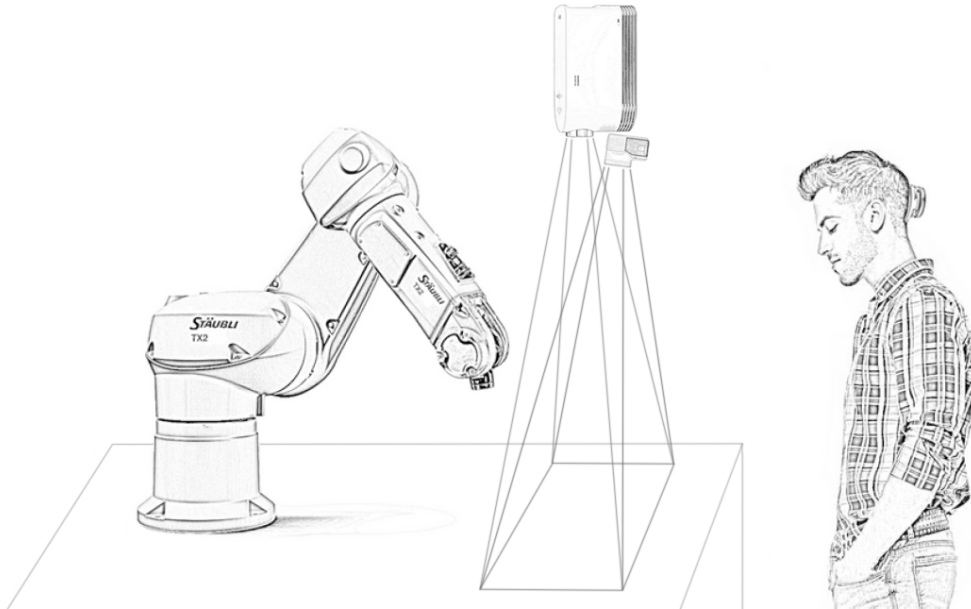


Figura 5.2. Schema del sistema BCI-Robot

Come si può evincere da figura 5.2, la videocamera è stata integrata al proiettore per ridurre la distanza tra la fonte di proiezione dell'interfaccia utente e il dispositivo per il rilevamento degli oggetti. Sebbene sia corretta da uno script Python, il cercare di limitare il più possibile un'eventuale distorsione prospettica risulta essenziale per una calibrazione accurata del sistema. Il proiettore laser, connesso tramite HDMI ad un computer Windows, riproduce un'applicazione Unity per la visualizzazione della UI mentre la videocamera, cablata tramite USB ad un computer Linux, trasmette un flusso dati continuo per permetterne l'elaborazione in tempo reale.

5.1 Infrastruttura del sistema

Il sistema è stato progettato realizzando un'infrastruttura per permettere la trasmissione dei dati necessari alla creazione dell'interfaccia utente e al movimento del braccio meccanico. In figura 5.3 è rappresentato il diagramma dell'architettura del sistema, a seguire la descrizione testuale in riferimento alla figura.

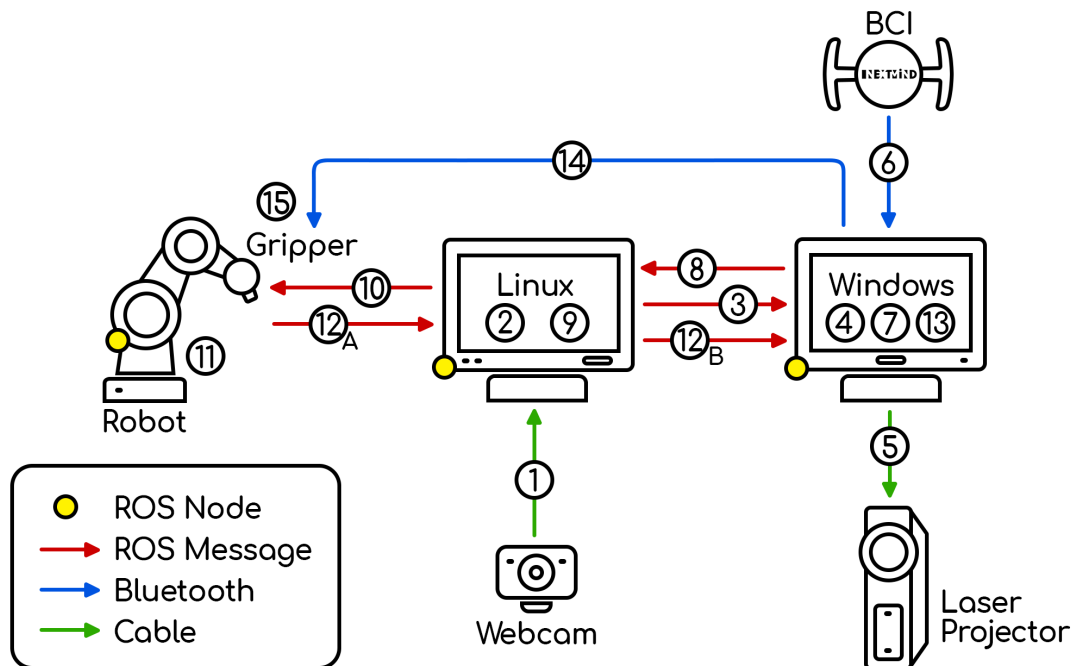


Figura 5.3. Architettura hardware del sistema

- 1) Il flusso di immagini (640x480px a 30fps) viene trasmesso tramite connessione USB dalla videocamera Logitech QuickCam Pro all'elaboratore Linux.
- 2) Successivamente alla calibrazione della videocamera mediante uno script Python, le immagini vengono elaborate estrapolando i contorni ed i bounding box degli oggetti.
- 3) Alla pressione di un tasto viene pubblicato un messaggio ROS composto da un vettore di interi: per ogni bounding box, quattro numeri che rappresentano le ascisse e le ordinate dei vertici superiore-sinistro e inferiore-destro.
- 4) Il vettore di interi ricevuto viene elaborato e, in funzione della quantità, dimensione e posizione degli oggetti, viene eseguito un algoritmo di posizionamento dei NeuroTag.
- 5) L'interfaccia utente contenente i bounding box ed i NeuroTag viene proiettata sugli oggetti nell'area di lavoro.

- 6) Non appena l'utente seleziona un NeuroTag attraverso l'utilizzo del NextMind, viene inviato tramite Bluetooth Low Energy (BLE) il segnale all'applicazione Unity.
- 7) Il sistema elabora il segnale ed individua il bounding box relativo al NeuroTag selezionato, quindi definisce la posizione del centro del bounding box.
- 8) Viene pubblicato un messaggio ROS contenente le coordinate tridimensionali (essendo l'altezza degli oggetti nota) e l'orientamento che l'end effector del robot deve raggiungere per prelevare l'oggetto selezionato.
- 9) Tramite il solutore di cinematica inversa della piattaforma MoveIt, la posizione e l'orientamento presenti nel messaggio ROS ricevuto dal calcolatore Linux vengono elaborati definendo le singole posizioni dei giunti.
- 10) Vengono pubblicati in un messaggio ROS i valori dei giunti definiti dal solutore IK.
- 11) Il robot riceve il messaggio ed inizia il movimento interpolando i valori della posizione corrente con quelli della posizione da raggiungere.
- 12) Durante il movimento, il robot pubblica periodicamente sulla rete ROS messaggi descrittivi della sua posizione corrente che sono ricevuti sia dall'elaboratore Linux, aggiornando in tempo reale la previsualizzazione del movimento sulla piattaforma MoveIt, che dall'elaboratore Windows, sincronizzando l'istanza del robot presente nella simulazione fisica dell'applicazione Unity.
- 13) L'applicazione monitora il raggiungimento dell'end effector del robot alle zone di interesse (l'oggetto da prelevare e l'area di deposito) tramite l'utilizzo delle collisioni simulate nella scena virtuale.
- 14) In caso si verificasse una collisione, l'applicazione comunica un valore al modulo Bluetooth del gripper: in caso di collisione rilevata con il collider² di un oggetto trasmette il valore 1, in caso di collisione rilevata con il collider dell'area di deposito trasmette il valore 0.
- 15) Il gripper del robot, secondo il codice in appendice A, chiude il circuito attivando l'elettromagnete se riceve il valore 1 mentre riapre il circuito disattivando l'elettromagnete se riceve il valore 0.

Il sistema permette di rieseguire il processo molteplici volte: al passo 3, ogniqualvolta viene premuto il tasto per pubblicare i bounding box, il sistema rielabora i dati ricevuti e riesegue l'algoritmo aggiornando l'interfaccia utente proiettata e le associazioni tra i bounding box ed i NeuroTag.

²Componente Unity utilizzato per rilevare le collisioni durante le simulazioni fisiche

In figura 5.4 è consultabile il diagramma di flusso del sistema.

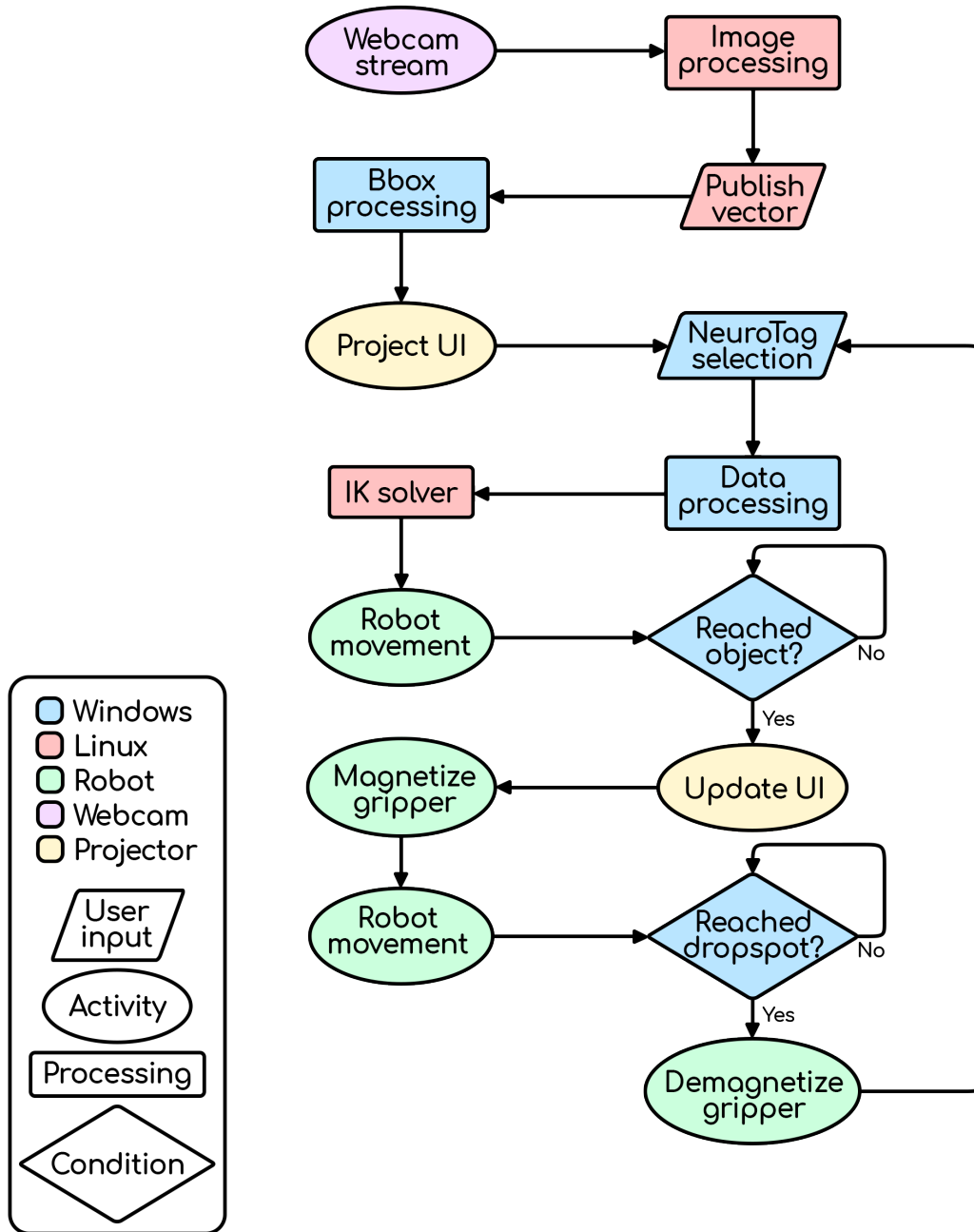


Figura 5.4. Diagramma di flusso del sistema

Infrastruttura ROS

Sono presenti tre nodi³ ROS (`robot_node`, `linux_node` e `windows_node`) che utilizzano i topic⁴ per scambiarsi messaggi⁵. Ogni nodo ROS può operare come ROS Publisher (pubblicando i messaggi nel topic) oppure come ROS Subscriber (iscrivendosi al topic e ricevendo i messaggi pubblicati dai ROS Publisher).

I topic definiti nel sistema sono quattro:

- `/bbox_topic` (passo 3): è utilizzato per pubblicare il vettore di interi contenente i due vertici significativi di ogni bounding box. Il nodo `linux_node` agisce da ROS Publisher mentre il nodo `windows_node` agisce da ROS Subscriber.
- `/targetpos_topic` (passo 8): è utilizzato per pubblicare la posizione e l'orientamento da far raggiungere all'end effector del robot. Il nodo `windows_node` agisce da ROS Publisher mentre il nodo `linux_node` agisce da ROS Subscriber.
- `/joints_topic` (passo 10): è utilizzato per pubblicare i valori dei giunti del robot da ottenere. Il nodo `linux_node` agisce da ROS Publisher mentre il nodo `robot_node` agisce da ROS Subscriber.
- `/currentpos_topic` (passo 12): è utilizzato per comunicare la posizione in tempo reale del robot. Il nodo `robot_node` agisce da ROS Publisher mentre i nodi `linux_node` e `windows_node` agiscono da ROS Subscriber.

In figura 5.5 è rappresentato il diagramma della rete ROS progettata per il sistema.

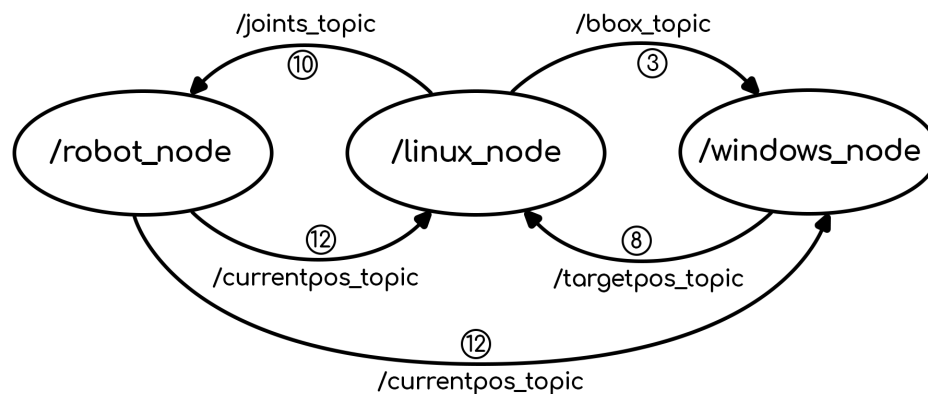


Figura 5.5. Grafo della rete ROS

³Processi che eseguono calcoli, comunicano attraverso i topic

⁴Bus nominali su cui i nodi si scambiano messaggi

⁵Strutture dati

5.2 Rilevamento degli oggetti

Per il rilevamento degli oggetti è stato prodotto uno script Python utilizzando Visual Studio Code su Linux. Inizialmente si è pensato di utilizzare il sistema di intelligenza artificiale YOLO⁶ integrato nella videocamera OAK—D ma, a causa del tempo insufficiente per la creazione del dataset e per l'addestramento della rete neurale, si è deciso di estrapolare la posizione degli oggetti dal flusso di immagini tramite l'utilizzo del colore. Sono stati creati, infatti, degli oggetti rossi in modo tale da permetterne agevolmente l'identificazione. Gli oggetti impiegati, mostrati in figura 5.6, sono blocchi di legno di larghezza ed altezza predefiniti (5cm) e lunghezza variabile (dai 5cm ai 10cm). I dieci parallelepipedi sono stati tagliati, dipinti usando una tempera per legno rossa, numerati ed, infine, sono state incollate al centro della faccia superiore di ogni blocco due piastrine metalliche per permetterne l'attrazione in caso di prelievo da parte del gripper elettromagnetico.



Figura 5.6. Oggetti utilizzati per i test del sistema

5.2.1 Calibrazione videocamera

La prima di due fasi del programma Python è costituita dal processo di calibrazione: consiste nel calibrare prospetticamente e cromaticamente la videocamera. Come mostrato nelle figure 5.7 e 5.8, il programma presenta i seguenti componenti: quattordici parametri modificabili, un flusso di immagini in 4:3 (prodotto in tempo reale dalla videocamera con in sovrapposizione un poligono contornato di rosso) e, accanto, un flusso di immagini in 16:9 (maschera cromatica della porzione di immagine inscritta dal poligono rosso e corretta prospetticamente). I controlli del programma sono consultabili in appendice B.

⁶Algoritmo di intelligenza artificiale per la computer vision in tempo reale

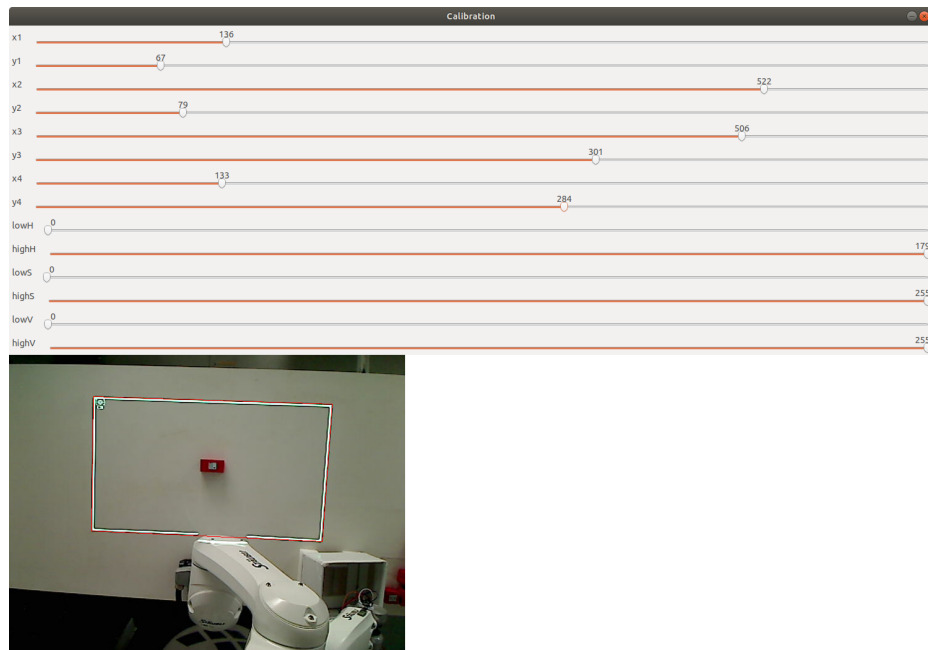


Figura 5.7. Calibrazione prospettica della videocamera

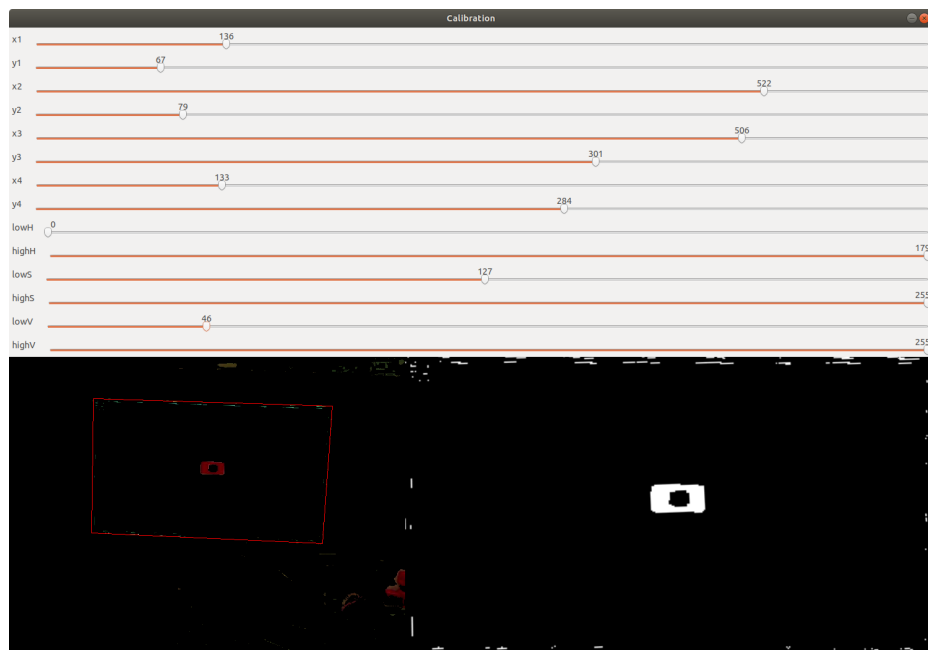


Figura 5.8. Calibrazione cromatica della videocamera

I primi otto parametri permettono di modificare le coordinate dei quattro vertici del poligono rosso e tollerano valori da 0 a 639 per le ascisse e da 0 a 479 per le ordinate (la risoluzione dell'immagine è 640x480px). In questo modo è possibile far combaciare esattamente il poligono con un'immagine proiettata sull'area di lavoro delineante il confine di proiezione per correggere prospetticamente la porzione ritagliata definendo una matrice di trasformazione prospettica.

Gli ultimi sei parametri, invece, permettono di definire i limiti superiori ed inferiori degli intervalli di tonalità (H), saturazione (S) e luminosità (V) da preservare nella maschera sottostante. Il flusso di immagini codificato tramite il modello di colori RGB è, infatti, convertito nel modello di colori HSV prima di essere elaborato per consentire un rendimento migliore. Modificando singolarmente i parametri H, S e V (H tollera valori da 0 a 179, S e V tollerano valori da 0 a 255) è possibile isolare in maniera abbastanza precisa un intervallo di colori, operazione necessaria per estrapolare le informazioni spaziali degli oggetti presenti nell'area di lavoro. La maschera mostrata in figura 5.8 identifica in bianco la presenza di un oggetto mentre in nero l'area vuota inscritta nel poligono rosso.

5.2.2 Identificazione oggetti

Una volta completata la fase di calibrazione e quindi definiti i quattordici parametri, è possibile procedere con la fase di identificazione degli oggetti. Mediante l'utilizzo della libreria OpenCV per la Computer Vision, sono state estrapolate dalle maschere del flusso di immagini le informazioni inerenti ai contorni degli oggetti (verdi in figura 5.9). Sono stati, inoltre, individuati e scartati i contorni caratterizzati da un'area inferiore ad una soglia stabilita empiricamente (per escludere i contorni generati dal rumore digitale della videocamera) e quelli inscritti in contorni di grado gerarchico superiore (per escludere i contorni generati dalle placche metalliche al centro degli oggetti). In questo modo è stato successivamente possibile definire i bounding box degli oggetti (blu in figura 5.9) identificando i minimi ed i massimi delle ascisse e delle ordinate di ogni contorno. Tramite la pressione di un tasto, è infine possibile pubblicare sulla rete ROS (topic `/bbox_topic`) il vettore di interi rappresentativo dei bounding box generati dall'immagine corrente.



Figura 5.9. Interfaccia utente per il rilevamento degli oggetti

5.3 Architettura dell'applicazione

L'applicazione, progettata in Unity sul calcolatore Windows, gestisce il processo di creazione dell'interfaccia utente proiettata sull'area di lavoro. Realizza una fase di calibrazione del NextMind, utilizzata anche per la visualizzazione del confine di proiezione adoperato durante la calibrazione prospettica della videocamera, e due scene integranti i due diversi algoritmi di posizionamento dei NeuroTag. È possibile navigare tra le scene ed interagire con i componenti presenti interamente attraverso l'utilizzo della tastiera dal momento che, essendo la UI proiettata su un tavolo, l'interazione tramite mouse sarebbe risultata disagiata. I controlli dell'applicazione sono consultabili in appendice B.

5.3.1 Calibrazione NextMind

È la scena iniziale dell'applicazione e permette la calibrazione del dispositivo neurale. Visualizza un perimetro verde per delineare il contorno di proiezione durante la fase di calibrazione prospettica della videocamera. Analoga alla scena dello studio preliminare, consente la connessione con il NextMind e, come mostrato in figura 5.10, il controllo della qualità del segnale tramite i sensori dell'interfaccia neurale. Attraverso l'uso dei colori, è possibile identificare i contatti dei singoli elettrodi: il grigio o il rosso indicano una mancanza di contatto, il giallo una mediocre qualità mentre il verde attesta un'ottima qualità del segnale ricevuto. Risulta inoltre possibile definire, mediante l'utilizzo del tastierino numerico, l'altezza dell'utente la quale sarà utilizzata per la simulazione fisica della scena integrante l'algoritmo adattivo di posizionamento dei NeuroTag. L'altezza, se definita, appare nella UI nell'angolo superiore-destro mentre, se non dovesse essere definita durante la fase corrente, verrebbe impostata un'altezza predefinita di 180cm.

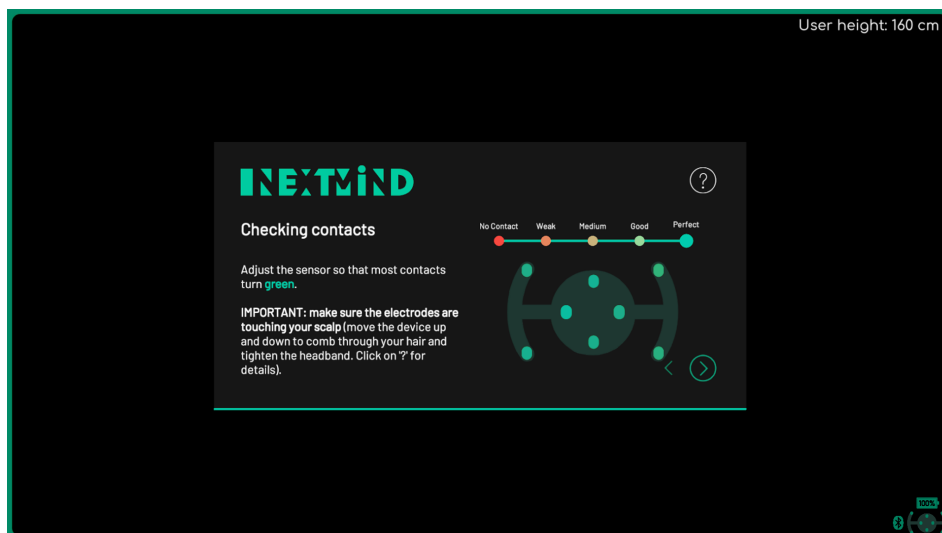


Figura 5.10. Interfaccia utente per il controllo qualità dei contatti

Successivamente al controllo qualità dei contatti, durante il quale viene assegnato un risultato per riassumere la condizione del segnale trasmesso (mancanza di contatti, scarsa, mediocre, buona, perfetta), l'utente procede con l'effettiva calibrazione della BCI: mantenendo la concentrazione sul NeuroTag presente in figura 5.11, questo viene attivato ogniqualvolta il dispositivo dovesse rilevare il segnale elettrico prodotto dal cervello dell'utente. Il NeuroTag è composto, infatti, da una texture intermittente con frequenza variabile che permette al NextMind di adattarsi all'attività cerebrale dell'utente e, quindi, di migliorarne le prestazioni in termini di precisione e velocità.

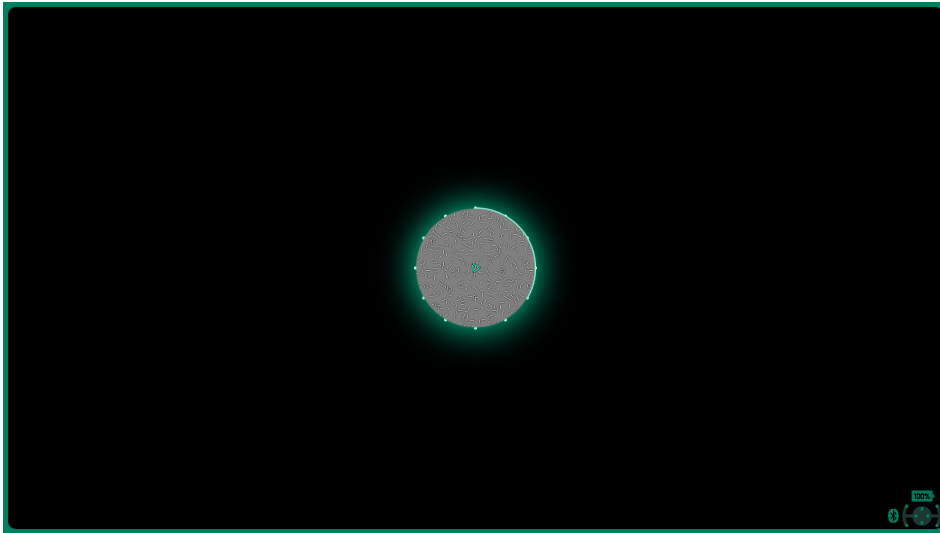


Figura 5.11. NeuroTag utilizzato durante la calibrazione del NextMind

Come mostrato in figura 5.12, l'applicazione quantifica infine, assegnando una valutazione da 1 a 5 (1 insufficiente, 5 ottimo), il prodotto finale della calibrazione. Questo processo risulta essere di fondamentale importanza per la corretta interpretazione da parte della BCI del segnale ricevuto ed è necessario ottenere un buon punteggio dal momento che, in caso contrario, l'interfaccia neurale risulterebbe notevolmente imprecisa.

I componenti persistenti tra le varie scene ed utilizzati dall'applicazione per il corretto funzionamento sono i seguenti:

- NeuroManager: componente necessario per l'opportuno funzionamento del NextMind, gestisce i parametri di connessione dell'interfaccia neurale e permette di simulare il dispositivo tramite l'utilizzo del cursore.
- CVCamera: è la camera simulata corrispondente alla videocamera Logitech Quick-Cam Pro che il NeuroManager utilizza per determinare quali NeuroTag sono visibili e, quindi, attivarli.
- EventSystem: gestisce gli eventi del sistema come le interazioni utente.
- SensorCanvas: presenta delle icone informative su BCI e sistema.

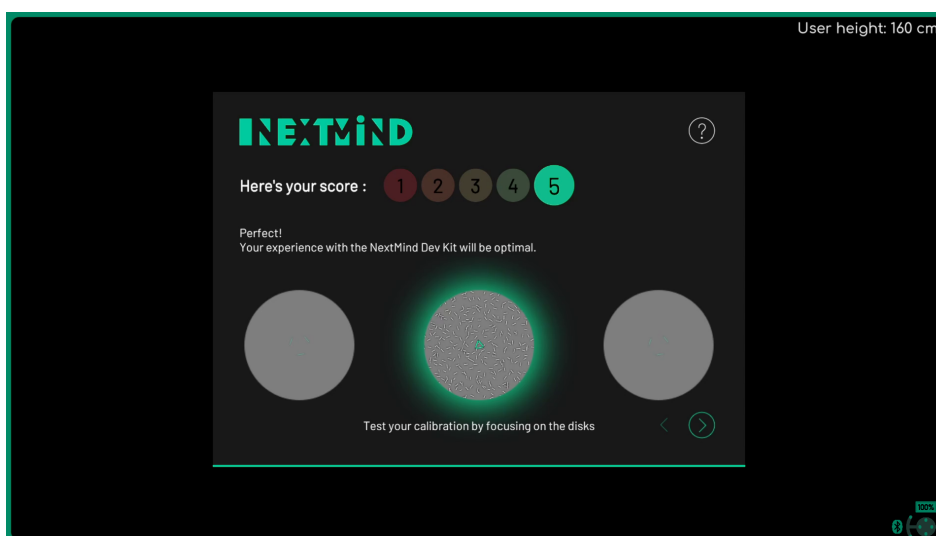


Figura 5.12. Punteggio riassuntivo ottenuto durante la fase di calibrazione

La regione dell'interfaccia utente adibita alla visualizzazione delle icone informative, consultabile in figura 5.13, è composta da quattro elementi:

- Percentuale batteria: è la carica rimanente del dispositivo NextMind connesso.
- Contatto sensori: illustra la qualità del segnale ricevuto tramite il colore dei sensori.
- Bluetooth: si attiva quando l'applicazione si connette tramite il modulo Bluetooth al gripper del robot.
- Lucchetto: tramite il tasto 'L' della tastiera è possibile attivare o disattivare i NeuroTag proiettati con lo scopo di prevenirne l'innesco involontario e il conseguente movimento del robot (durante le fasi di test, i movimenti del robot sono automaticamente sospesi e, quindi, il blocco dei NeuroTag è temporaneamente disabilitato).

Le icone sono posizionate nell'angolo inferiore-destro durante la fase di calibrazione ed esecuzione dell'algoritmo adattivo mentre, necessitando degli angoli inferiori per il posizionamento dei NeuroTag, durante l'esecuzione dell'algoritmo non adattivo viene effettuato un riarrangiamento delle icone nella parte superiore dell'interfaccia utente.



Figura 5.13. Regione della UI adibita alla visualizzazione delle icone informative

5.3.2 Interfacce utente

Dalla scena di calibrazione è possibile eseguire i due algoritmi integrati, dettagliati nella sezione a seguire, attraverso la pressione del tasto '1' per l'algoritmo adattivo e '2' per quello non adattivo. È inoltre possibile tornare alla scena di calibrazione attraverso il tasto 'backspace'.

Le due scene contenenti i due algoritmi risultano essere simili per quanto riguarda i componenti presenti se non per delle differenze in termini di interfaccia. Come si può notare in figura 5.14, è presente una simulazione della scena reale tramite la realizzazione di un modello tridimensionale del robot Stäubli TX2-60 utilizzato nel sistema e l'utilizzo di cubi rappresentativi degli oggetti rilevati dalla videocamera. Sono inoltre presenti due camere: la CVCamera, corrispondente alla videocamera fisica ed utilizzata per la creazione in proiezione ortografica dell'interfaccia utente proiettata (oltre ad essere la camera di tracciamento del NeuroManager), e la UserCamera, corrispondente alla posizione della testa dell'utente (determinata dall'altezza definita in fase di calibrazione) ed utilizzata per la computazione dell'algoritmo di posizionamento adattivo. La maggior parte degli oggetti in scena sono provvisti di collider per garantire una corretta simulazione fisica durante i movimenti del robot (aggiornati in tempo reale mediante messaggi ROS sul topic /currentpos_topic) oppure per permettere l'identificazione del raggiungimento da parte del robot delle zone di interesse (gli oggetti dispongono di un collider usato come innesco ed è inoltre presente un oggetto vuoto con un collider equivalente all'area di deposito degli oggetti). I messaggi ROS sono gestiti dal componente ROSManager che, attraverso la creazione di un ROSBridge definito da un URL, permette l'iscrizione del /windows_node come ROS Publisher e come ROS Subscriber ai topic relativi di figura 5.5.

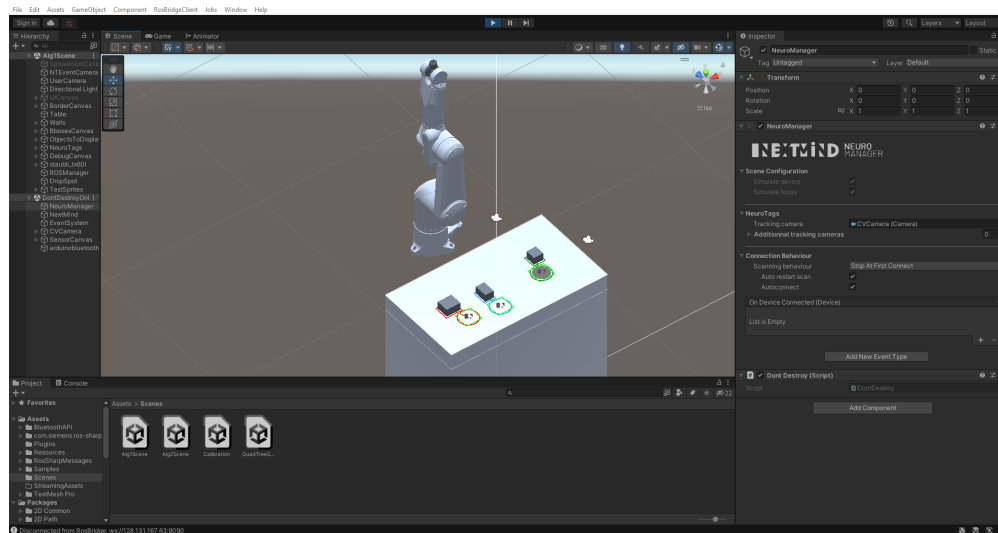


Figura 5.14. Interfaccia utente della scena Unity integrante l'algoritmo adattivo

È stata definita l'area raggiungibile dal braccio meccanico mediante la proiezione di una griglia composta da 1296 celle (48x27) e, conseguentemente, è stata identificata l'area di posizionamento oggetti ammessa per permetterne il prelievo da parte del robot. In figura 5.15 è possibile identificare l'area in questione nell'interfaccia di sinistra (utilizzata per l'algoritmo adattivo che permette l'utilizzo di tutta la UI disponibile). A destra, invece, è possibile distinguere l'interfaccia dell'algoritmo non adattivo che, al contrario del primo, necessita di uno spazio destinato esclusivamente alla proiezione dei NeuroTag (definito dalla zona inferiore dell'immagine).

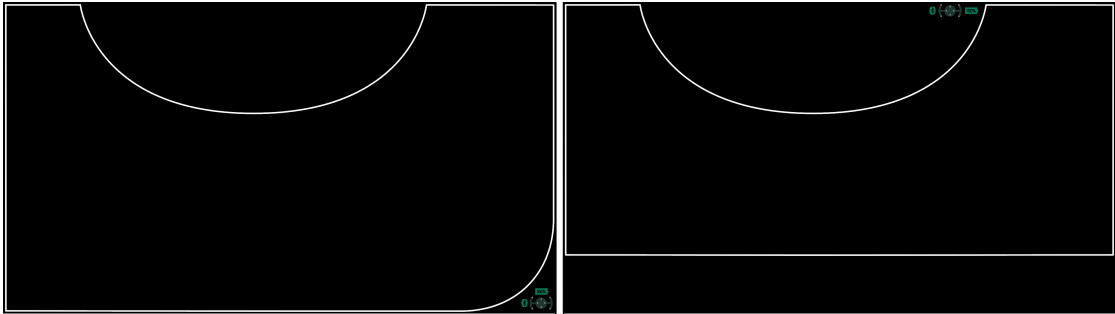


Figura 5.15. Aree di posizionamento algoritmo adattivo (sinistra) e non adattivo (destra)

È infine possibile abilitare l'interfaccia di debug per l'algoritmo adattivo, la quale presenta diverse funzionalità tra le quali la possibilità di visualizzare e manipolare la matrice generata dalla posizione degli oggetti rilevati o simulati, visualizzare la struttura del QuadTree, mostrare i raggi proiettati dal centro degli oggetti al centro dei NeuroTag per individuare eventuali interferenze, passare alla vista dell'utente oppure pubblicare una posa dell'end effector del robot definita durante l'esecuzione.

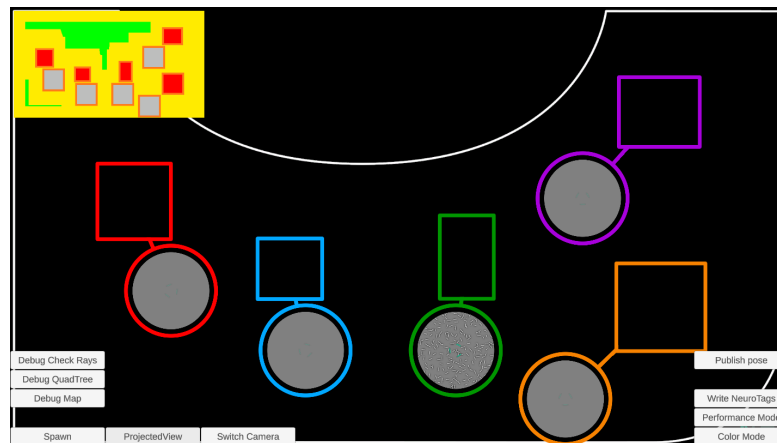


Figura 5.16. Interfaccia utente per il debug dell'algoritmo adattivo

5.4 Algoritmi di posizionamento

Il sistema integra due algoritmi per il posizionamento dei NeuroTag nell'interfaccia utente, uno adattivo ed uno non adattivo. La differenza principale tra i due algoritmi emerge dalle possibili collocazioni dei NeuroTag: mentre il primo ha come obiettivo quello di trovare una disposizione che si adatti all'ubicazione degli oggetti e dell'utente permettendo una UI omogenea, il secondo riserva una porzione dell'area proiettata per l'esclusivo collocamento dei NeuroTag frazionando lo spazio a disposizione. Sebbene l'algoritmo adattivo risulti essere sicuramente di più complessa computazione a confronto di quello non adattivo, permette il miglior sfruttamento dello spazio a disposizione e la maggior vicinanza tra oggetto e NeuroTag associato.

A seguire i due algoritmi descritti in dettaglio.

5.4.1 Algoritmo adattivo

L'obiettivo dell'algoritmo adattivo è quello di analizzare lo spazio disponibile non occupato dagli oggetti e sfruttarlo al meglio per posizionare i NeuroTag associati. L'algoritmo riceve in ingresso il vettore di interi contenente le informazioni relative ai due vertici significativi di ogni bounding box tramite messaggio ROS sul topic `/bbox_topic` e produce le posizioni di tutti i NeuroTag da disporre. Il numero di posizioni in uscita è equivalente al numero di bounding box rilevati dalla videocamera, ovvero $n = d/4$ con d dimensione del vettore ricevuto.

Risulta essere costituito da due processi: la creazione della matrice e il posizionamento dei NeuroTag.

Creazione della matrice

Il primo processo eseguito dall'algoritmo provvede alla creazione di una matrice 500x280px ottenendo una precisione di 2mm/px dal momento che l'area proiettata a disposizione misura esattamente 1000x560mm. Oltre ad aver deciso di discretizzare lo spazio per facilitare l'elaborazione dei dati da parte del calcolatore, la scelta dell'utilizzo di una matrice da 140.000px è stata compiuta per trovare un equilibrio tra risoluzione spaziale e tempo di calcolo necessario alla computazione e scrittura nella matrice.

Lo scopo della matrice è quello di identificare tutte le posizioni nelle quali risulta possibile il posizionamento di un NeuroTag senza sovrapposizione con gli oggetti. La prima fase (figura 5.17.1) consiste nel riempimento della matrice di celle disponibili, ogni cella disponibile equivale ad una possibile posizione per il centro dei NeuroTag considerando il raggio e lo spazio occupato dai contorni della UI di oggetti e NeuroTag proiettati. La seconda fase (figura 5.17.2) provvede ad escludere tutte le celle troppo vicine al bordo della matrice che non permetterebbero al NeuroTag di essere visualizzato nella sua interezza tagliandone una porzione. La terza fase (figura 5.17.3) esclude tutte le celle contenute nei bounding box (rosse in figura), le celle occupate dalla proiezione della UI (arancioni in figura) e le celle troppo vicine agli oggetti per evitarne la sovrapposizione (gialle in

figura). L'ultima fase (figura 5.17.4) consiste nella proiezione fisica attraverso la tecnica del raycasting⁷ dei quattro vertici superiori di ogni bounding box (altezza di 5cm) a partire dalla posizione degli occhi dell'utente (derivante dalla statura definita in fase di calibrazione) per escludere le oclusioni degli oggetti (blu in figura) non permettendo ai NeuroTag di essere posizionati in ubicazioni nascoste agli occhi dell'utente. Successivamente sono anche escluse le celle troppo vicine alle oclusioni che nasconderebbero alcune porzioni dei NeuroTag. In conclusione, è stata ottenuta una lista di celle disponibili per il posizionamento del centro dei NeuroTag (figura 5.17.5).

In figura 5.17 è possibile consultare graficamente le fasi per la creazione della matrice con annessa una legenda illustrativa.

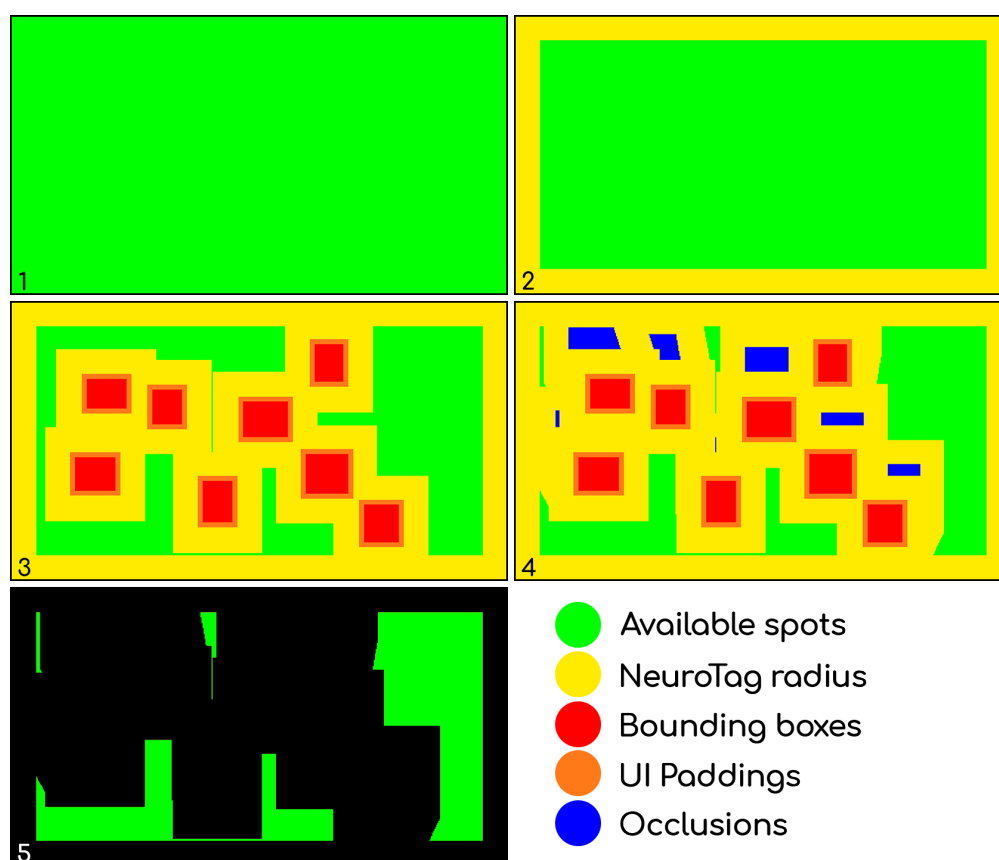


Figura 5.17. Fasi di creazione della matrice dell'algoritmo adattivo

Per il riempimento delle porzioni rettangolari della matrice (fasi 1, 2, 3) sono stati utilizzati dei semplici cicli ottimizzati per evitare di scrivere molteplici volte celle già escluse. Per il riempimento delle oclusioni (fase 4), ottenute come mostrato in figura

⁷Tecnica utilizzata per proiettare raggi da una sorgente verso direzioni definite

5.18 tramite la tecnica del raycasting, è stato invece implementato l'algoritmo scan line [50], ovvero un algoritmo che permette di definire se una cella è interna o esterna ad un poligono attraverso il cambio di polarità ogniqualvolta la linea di scansione interseca uno dei lati del poligono. Ogni cella interna al poligono descritto dalle quattro proiezioni sul tavolo (individuate tramite la collisione dei raggi con il collider del tavolo) viene quindi esclusa dalla matrice.

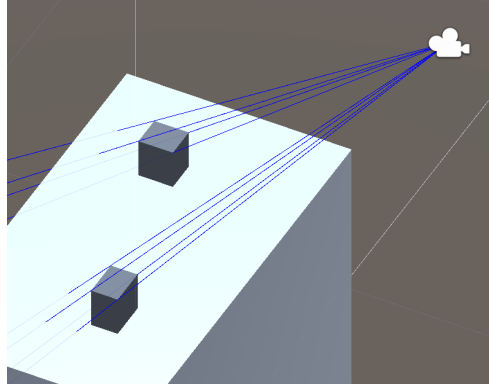


Figura 5.18. Tecnica del raycasting per il calcolo delle occlusioni

Posizionamento dei NeuroTag

Successivamente al processo di creazione della matrice, l'algoritmo procede con il posizionamento dei NeuroTag a partire dalla lista di celle valide appena computata. Come prima fase, i bounding box vengono ordinati in funzione crescente del numero di celle disponibili adiacenti al proprio perimetro (definito dal margine prodotto dal raggio dei NeuroTag). Le occlusioni ed i margini generati da esse non partecipano alla formazione del perimetro essendo questo esclusivamente descritto dall'area composta dai bounding box (celle rosse), UI (celle arancioni) e dal margine rettangolare derivato (celle gialle). In questo modo, come mostrato in figura 5.19, i bounding box con meno celle disponibili adiacenti vengono computati prima essendo, in teoria, i più complessi da risolvere.

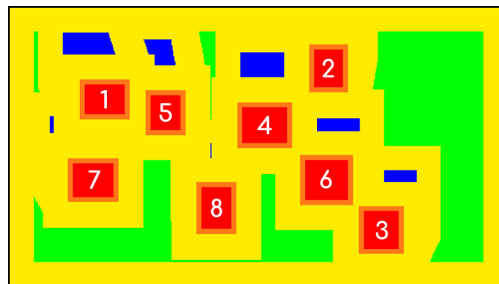


Figura 5.19. Ordine di computazione dei bounding box

Una volta ordinati, il primo bounding box della lista viene computato. A questo punto sono possibili due diverse condizioni: il bounding box presenta almeno una cella disponibile adiacente al suo perimetro oppure non presenta neanche una cella disponibile.

Nel primo caso, viene calcolata l'equazione bidimensionale della retta (parametri m e q) che unisce il centro del bounding box alla posizione dell'utente mediante la formula $(y - y_A)/(y_B - y_A) = (x - x_A)/(x_B - x_A)$ e successivamente viene computata la distanza punto-retta per ogni cella mediante la formula $d(P, r) = |y_P - (mx_P + q)|/\sqrt{1 + m^2}$. In questo modo le celle presenti nel perimetro vengono ordinate in funzione della distanza dalla retta congiungente il bounding box all'utente (figura 5.20) e viene scelta la cella con distanza minore permettendo di posizionare il NeuroTag il più vicino possibile all'utente.

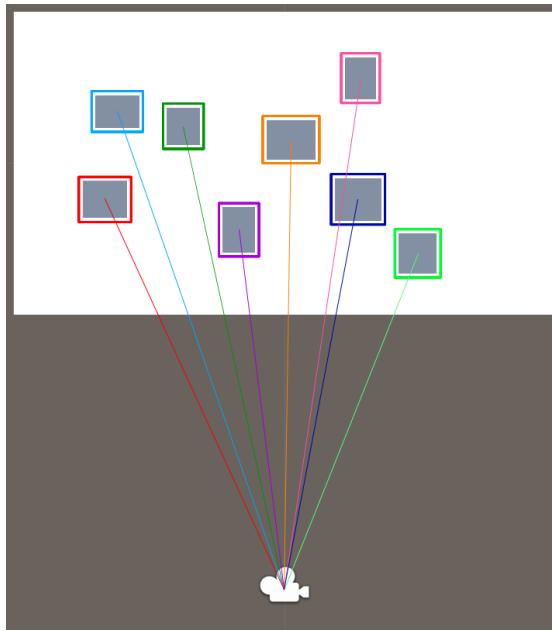


Figura 5.20. Rette congiungenti i centri dei bounding box alla posizione dell'utente

Nel secondo caso, non essendoci celle adiacenti valide, è necessario implementare una modalità di ricerca efficiente per identificare le celle disponibili più vicine al bounding box. È stata quindi implementata una struttura dati ad albero quadramentale per permettere una ricerca funzionale tra tutte le celle disponibili nella matrice. Un albero quadramentale (in inglese QuadTree) è una struttura dati ad albero non bilanciata nella quale tutti i nodi interni hanno esattamente quattro nodi figli. È possibile definire il numero massimo di celle presenti in ogni nodo dell'albero: un numero elevato ne velocizza la generazione (producendo pochi nodi) ma rende più lenta la ricerca mentre, al contrario, un numero basso ne aumenta il tempo di generazione (producendo più nodi) ma permette di velocizzare notevolmente la ricerca. Per il progetto è stato deciso di assentire un massimo di 4 celle per nodo, individuando un compromesso accettabile tra rapidità di generazione dell'albero e dinamismo di ricerca. In figura 5.21 è possibile visualizzare una rappresentazione grafica dell'albero quadramentale generato dalle celle valide della matrice.

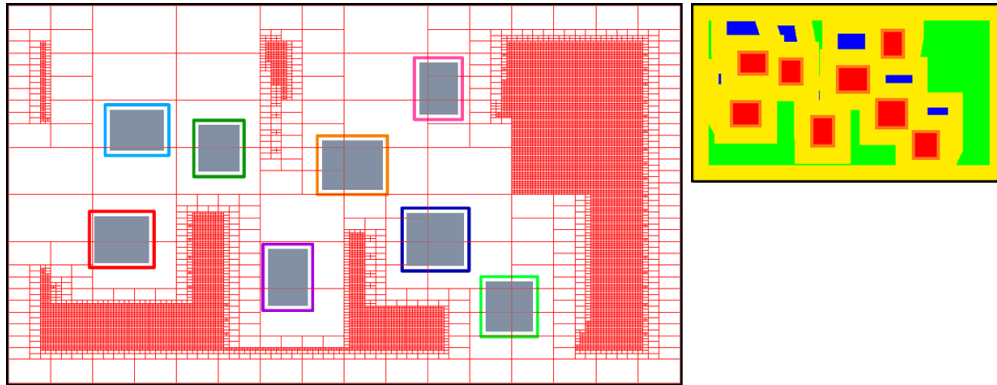


Figura 5.21. Albero quadramentale generato dalle celle valide della matrice

Una volta generata la struttura dati, viene avviata una ricerca attraverso un'area rettangolare incrementale per individuare i nodi più vicini al perimetro del bounding box. Il rettangolo di ricerca, centrato in corrispondenza del bounding box in analisi, inizialmente risulta essere 4px più grande in entrambe le dimensioni del perimetro ma, fino a quando non viene individuata almeno una cella valida, incrementa di 2px alla volta l'altezza e la larghezza dell'area (1px per direzione). Non appena una o più celle valide sono individuate, viene computata l'equazione bidimensionale della retta che unisce il centro del bounding box alla posizione dell'utente (figura 5.20) e calcolata la distanza punto-retta per ogni cella trovata con lo scopo di definire quella con distanza minore, ovvero quella più vicina all'utente.

In entrambi i casi, una volta identificata la cella destinata al posizionamento del centro nel NeuroTag, questo viene scritto nella matrice come mostrato in figura 5.22 per escludere le celle occupate dalla propria proiezione nella UI. Infine, le celle vengono cancellate dalla struttura dati e l'algoritmo riparte ciclicamente dalla prima fase del posizionamento rimuovendo il bounding box appena elaborato e ricalcolando, utilizzando la matrice aggiornata, le celle perimetrali valide dei bounding box rimanenti.

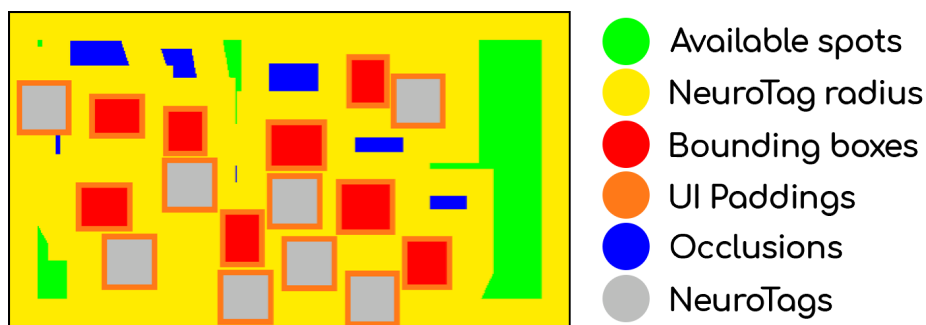


Figura 5.22. Matrice finale completa di NeuroTag

Non appena l'ultimo bounding box viene rimosso dalla lista, l'algoritmo procede a verificare la correttezza della soluzione controllando che tra ogni bounding box e il proprio NeuroTag associato non ci siano interferenze. Per fare ciò, attraverso la tecnica del ray-casting, vengono proiettati dei raggi dai centri dei bounding box ai centri dei NeuroTag associati (figura 5.23) esaminando le prime collisioni rilevate: se tutte risultano essere corrispondenti ai NeuroTag corretti la soluzione può essere definita valida mentre, se anche solo uno dei raggi collide con un altro bounding box o con un NeuroTag non associato al bounding box di origine, la soluzione non viene considerata valida e l'algoritmo riparte dalla creazione della matrice riducendo il diametro dei NeuroTag di 5mm. Dopo 10 tentativi (riduzione da 10cm a 5cm del diametro dei NeuroTag), se l'algoritmo ancora non avesse definito una soluzione valida, i NeuroTag che non presentano un collegamento rettilineo con il bounding box associato vengono proiettati nella zona superiore della UI poiché non utilizzata per il posizionamento degli oggetti e che, quindi, risulta essere vuota.

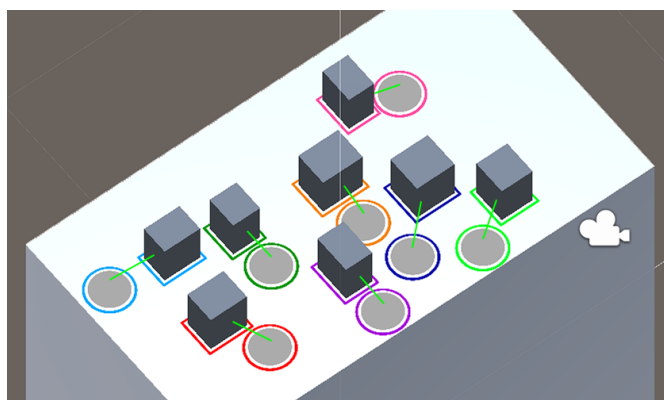


Figura 5.23. Raggi di controllo per la verifica della soluzione

In figura 5.24 è illustrata a sinistra la soluzione dal punto di vista dell'utente mentre a destra le componenti proiettate della soluzione senza l'interfaccia utente circostante.

In appendice C sono consultabili diagramma di flusso e pseudocodice dell'algoritmo.

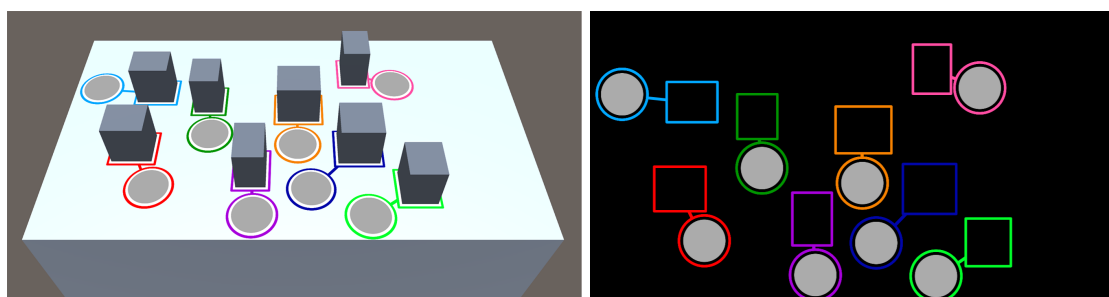


Figura 5.24. Soluzione adattiva osservata dall'utente (sinistra) e proiettata (destra)

5.4.2 Algoritmo non adattivo

L'algoritmo non adattivo risulta essere di più facile progettazione e computazione in confronto di quello adattivo. Presenta, però, il maleficio di riservare un'area dell'interfaccia utente esclusivamente alla proiezione dei Neurotag e che, quindi, non può essere occupata dagli oggetti riducendone lo spazio disponibile per il posizionamento. È stato scelto di dedicare alla disposizione dei NeuroTag la regione della UI più vicina all'utente (lato inferiore dello schermo) così da agevolarne la visualizzazione ed evitare eventuali occlusioni da parte degli oggetti presenti nell'area di lavoro.

L'algoritmo riceve in ingresso il vettore di interi contenente le informazioni relative ai due vertici significativi di ogni bounding box tramite messaggio ROS sul topic `/bbox_topic` e produce le posizioni di tutti i NeuroTag da disporre. Il numero di posizioni in uscita è equivalente al numero di bounding box rilevati dalla videocamera, ovvero $n = d/4$ con d dimensione del vettore ricevuto.

Come primo processo, l'algoritmo ordina i bounding box ricevuti in funzione della posizione del proprio centro da sinistra a destra così da permettere l'associazione tra oggetti e NeuroTag attraverso la corrispondenza cromatica dei contorni proiettati. Successivamente, i NeuroTag vengono associati e collocati in posizioni predefinite in funzione del numero di oggetti rilevati: attraverso la formula matematica a seguire è possibile definire esattamente tutte le posizioni dei NeuroTag da visualizzare a distanza omogenea.

$$ntGap/2 + ntGap * i$$

dove $ntGap = 1/n$, n numero di oggetti rilevati
e $i \in [0, n - 1]$ parametro intero incrementale

Infine, l'interfaccia utente viene proiettata presentando una distanza tra i NeuroTag variabile tra i 2.5cm e i 42.5cm (essendo i NeuroTag di 7.5cm per permettere la possibilità di proiettarne un massimo di dieci per volta considerando il margine introdotto dalla UI).

In figura 5.25 è illustrata a sinistra la soluzione dal punto di vista dell'utente mentre a destra le componenti proiettate della soluzione senza l'interfaccia utente circostante.

In appendice C sono consultabili diagramma di flusso e pseudocodice dell'algoritmo.

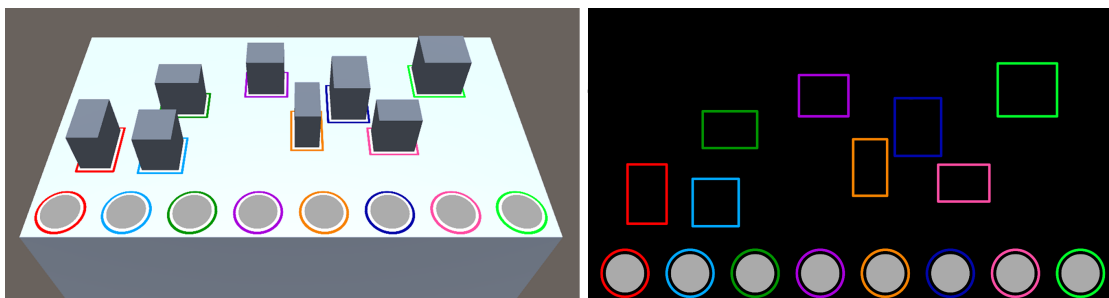


Figura 5.25. Soluzione non adattiva osservata dall'utente (sinistra) e proiettata (destra)

5.5 Colori interfacce utente

I colori utilizzati per la realizzazione delle interfacce utente sono ispirati ai colori impiegati per rappresentare le metropolitane delle più grandi città al mondo [51]. Dal momento che le metropolitane presentano la difficoltà di utilizzare colori percettivamente contrastanti cercando di identificare le singole linee nella maniera più univoca possibile, la creazione della UI del sistema ne condivide la problematicità. Come si può notare nella rappresentazione della ruota dei colori di figura 5.26, i colori scelti sono caratterizzati da una distanza spaziale considerevole con lo scopo di massimizzare il più possibile la differenza percettiva degli stessi e garantire un contrasto cromatico efficace.

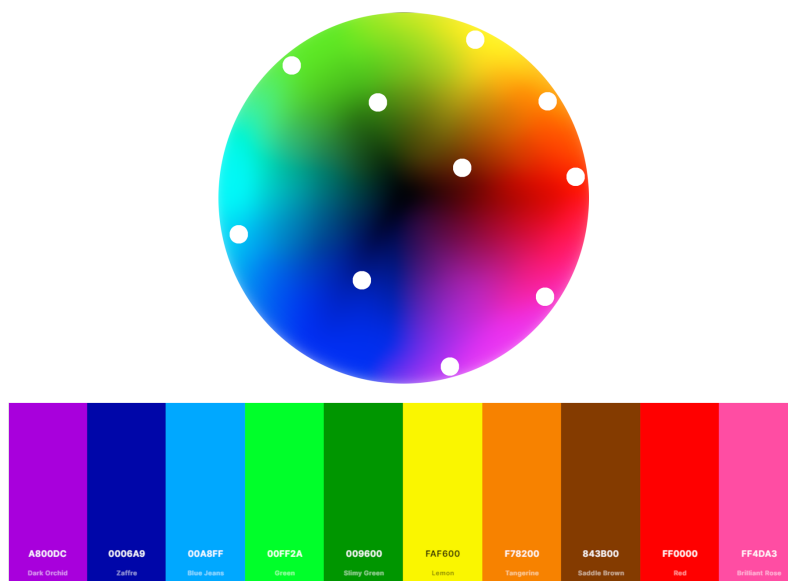


Figura 5.26. Cerchio cromatico e codici dei colori utilizzati per l'interfaccia utente

Successivamente, per evitare che colori simili vengano associati ad oggetti limitrofi, la palette dei colori è stata ordinata cercando di accostare colori complementari o comunque sensibilmente differenti in termini di cromaticanza e luminanza. In figura 5.27 è possibile consultare l'ordine spaziale dei colori dell'interfaccia utente associati agli oggetti ed ai relativi NeuroTag organizzati da sinistra a destra.



Figura 5.27. Palette di colori associata agli oggetti da sinistra a destra

Capitolo 6

Test e risultati

Con lo scopo di analizzare il sistema presentato, sono stati condotti dei test utente atti a valutare la User Experience (UX) e a paragonare gli algoritmi di posizionamento integrati. In questo capitolo sono descritti i test svolti e, successivamente, commentati i risultati ottenuti.

6.1 Test

I test sono stati condotti su un gruppo di studio composto da 22 partecipanti. Il gruppo è caratterizzato dalla presenza di 15 utenti di sesso maschile e 7 di sesso femminile con un'età che varia dai 19 ai 35 anni, presentando un'età media complessiva di 28 anni. Risulta, inoltre, omogeneo in termini di conoscenza del dispositivo NextMind e utilizzo di sistemi basati su AR proiettata, tutti i partecipanti si sono definiti neofiti per quanto concerne l'ambito delle BCI. In figura 6.1 è presente un grafico riassuntivo di età e sesso dei partecipanti (il blu definisce utenti di sesso maschile, il rosa di sesso femminile).

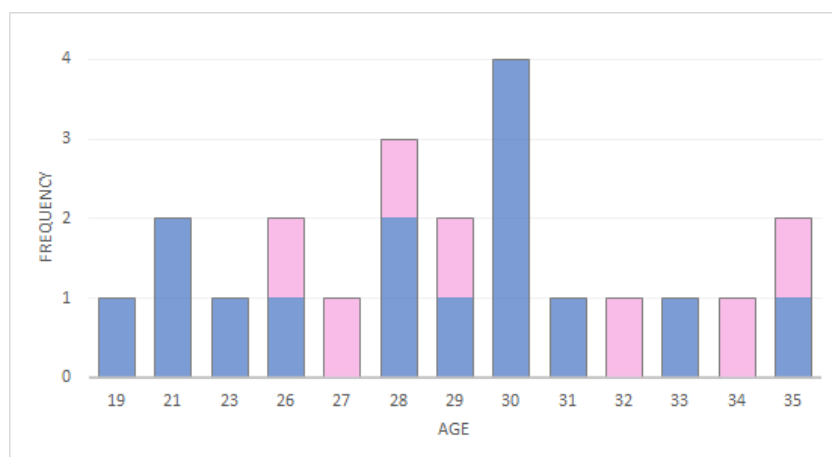


Figura 6.1. Grafico occorrenze età del gruppo di studio

La struttura dei test è costituita da tre fasi principali: la fase di calibrazione, di test e di compilazione dei questionari. Ognuna di queste viene ripetuta in sequenza due volte (una per ogni algoritmo) permettendo così agli utenti di valutare gli algoritmi subito dopo l'utilizzo. Ad ogni utente, l'ordine degli algoritmi provati viene invertito con lo scopo di evitare eventuali dipendenze statistiche dal momento che l'utilizzo del NextMind varia notevolmente in funzione dell'esperienza d'uso. La durata dei test complessiva per ogni partecipante è di circa 1 ora (25 minuti per algoritmo oltre ad una spiegazione iniziale del sistema) che può variare in funzione della rapidità di apprendimento dell'utente.

In figura 6.2 è mostrato il diagramma riassuntivo della struttura dei test utente.

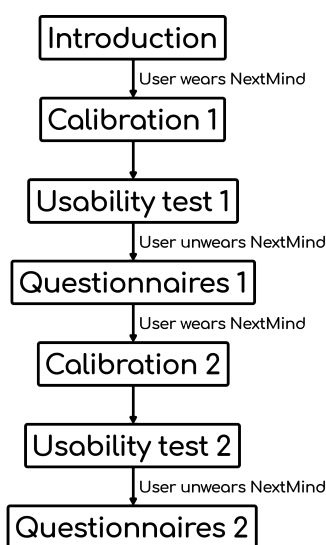


Figura 6.2. Diagramma riassuntivo della struttura dei test utente

Ad ogni utente del gruppo di studio è stato fatto firmare un modulo di consenso informato ed assegnato un identificativo univoco composto da numeri e lettere maiuscole generato dal software NASA Task Load Index (NASA-TLX). Successivamente sono stati illustrati gli obiettivi dello sviluppo del sistema e il suo generale funzionamento.

A seguire, le fasi iterate per entrambi gli algoritmi.

6.1.1 Calibrazione

Precedentemente all'utilizzo del sistema, è necessario effettuare il processo di calibrazione. Questa fase risulta essenziale per monitorare l'attività cerebrale dei partecipanti ed adattare di conseguenza la rete neurale impiegata dal dispositivo NextMind per la decodifica degli impulsi generati dalla stimolazione della corteccia visiva tramite i NeuroTag. Come dettagliato in sezione 5.3.1, la fase di calibrazione consiste nella verifica della qualità dei contatti del NextMind, l'osservazione di un NeuroTag a frequenza variabile ed, infine, nella valutazione complessiva del processo presentando la possibilità di provare tre diversi NeuroTag (figura 5.12) per confermare il corretto funzionamento dell'interfaccia neurale.

In figura 6.3 una partecipante durante la fase di calibrazione dell'interfaccia neurale.

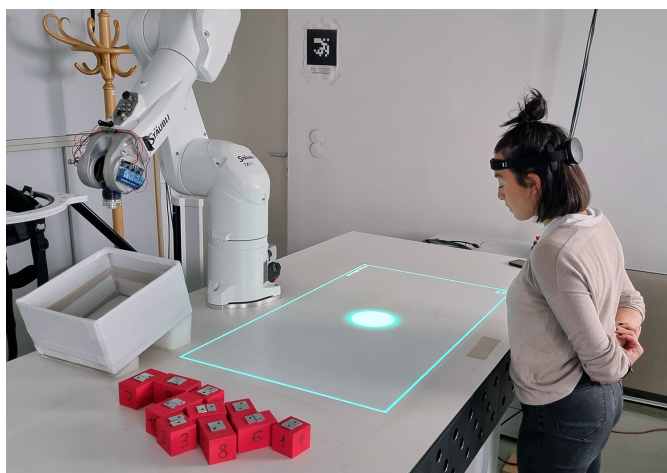


Figura 6.3. Partecipante del test durante il processo di calibrazione

6.1.2 Test di usabilità

Una volta calibrata l'interfaccia neurale, sono stati effettuati i test di usabilità durante i quali il robot non è stato mosso poiché, essendo cronometrati gli intervalli temporali di selezione di ogni NeuroTag, il tempo di movimento del robot (computato dal solutore di cinematica inversa) sarebbe risultato in alterazioni casuali delle misurazioni. Per ogni algoritmo sono stati eseguiti tre test di selezione, introducendo una difficoltà crescente: il primo è consistito nel selezionare tre oggetti in una sequenza predefinita su tre oggetti presenti nell'area di lavoro (3 NeuroTag proiettati), il secondo nel selezionare quattro oggetti su sei presenti nell'area di lavoro (6 NeuroTag proiettati) e l'ultimo nel selezionare cinque oggetti su dieci presenti nell'area di lavoro (10 NeuroTag proiettati). Per ogni utente ed algoritmo sono state impiegate le stesse configurazioni di oggetti e sequenze di selezione per permettere la confrontabilità dei dati (figura 6.4). Una volta attivato un NeuroTag, la UI relativa all'oggetto selezionato scompare permettendo di progredire nelle sequenze di selezione (pensate per far passare lo sguardo su tutta l'interfaccia con lo scopo di individuare eventuali falsi positivi). Come consultabile in appendice B, tramite l'utilizzo del tastierino numerico è possibile proiettare una configurazione (figura 6.4) per facilitare il posizionamento degli oggetti da rilevare oppure avviare le prove. Durante l'esecuzione dei test, per confermarne il corretto funzionamento, viene proiettato un piccolo disco verde nell'angolo inferiore-destro della UI e, se presente, viene ignorato l'eventuale blocco dei NeuroTag attivabile e disattivabile tramite la pressione del tasto 'L'.

Alla conclusione di ogni prova, sono scritti dal sistema in un file di testo i tempi parziali e gli errori compiuti dall'utente per ogni singolo NeuroTag. Successivamente, i file di testo sono stati rinominati con gli identificativi univoci degli utenti generati precedentemente per associare i risultati ai partecipanti.

A seguire, la sintassi impiegata per la scrittura dei dati nel file di testo.

id_test,t1,t2,t3,t4,t5,t_tot,e1,e2,e3,e4,e5,e_tot

Il primo dato identifica il test attraverso una lettera (A, B) ed un numero (1, 2, 3): la lettera determina l'algoritmo utilizzato (A algoritmo adattivo, B algoritmo non adattivo) mentre il numero determina la tipologia del test (1 facile, 2 intermedio, 3 difficile). I valori successivi identificano i tempi parziali calcolati per ogni NeuroTag (con una precisione di tre cifre decimali), il tempo totale della prova, gli errori (o falsi positivi) rilevati per ogni NeuroTag ed il numero di errori complessivi. Sia per i tempi parziali che per gli errori parziali sono presenti 5 valori: per i primi due test, che richiedono la selezione di meno oggetti, è inserito automaticamente il valore di -1 per i termini in eccesso mantenendo così la formattazione e permettendo a posteriori di scartare i valori negativi. Infine, è stato utilizzato il carattere di separazione ',' con lo scopo di semplificare l'inserimento dei dati in un foglio di calcolo tramite la funzione del copia e incolla.

In figura 6.4 sono mostrate le tre configurazioni degli oggetti (proiettabili tramite tastierino numerico) e le relative sequenze di selezione mostrate agli utenti durante i test.

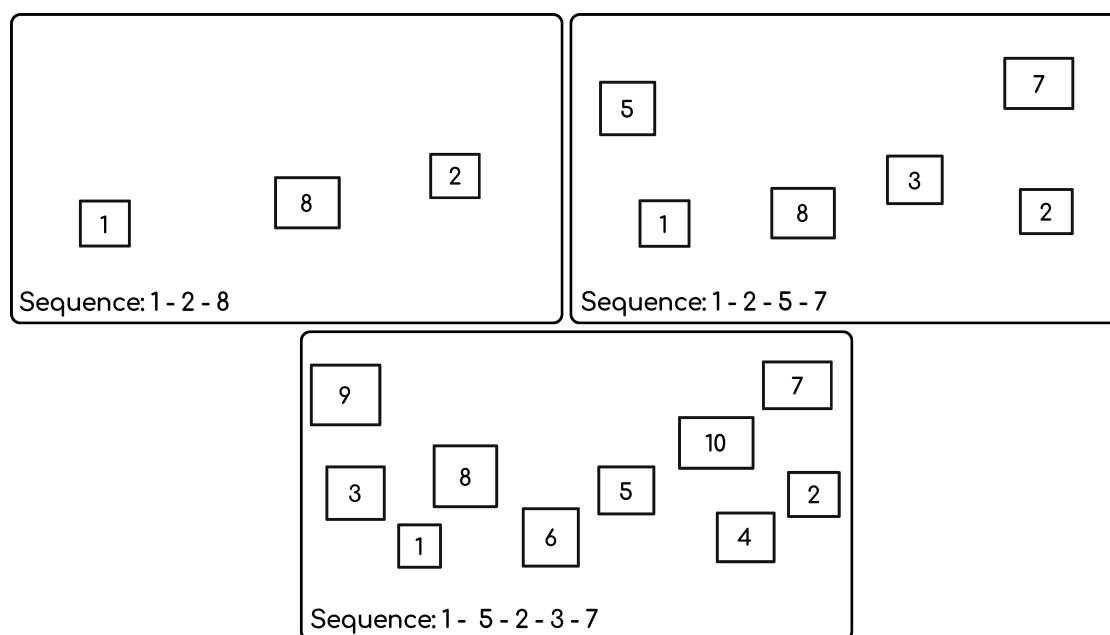


Figura 6.4. Configurazioni e sequenze di selezione dei test di usabilità

6.1.3 Compilazione questionari

Subito dopo aver provato il sistema, è stato chiesto ai partecipanti di compilare alcuni questionari per valutare l'esperienza svolta. Inizialmente sono stati inseriti l'identificativo generato, l'algoritmo di partenza, il risultato della calibrazione (valutazione sensori e valutazione complessiva) ed i dati di età, sesso ed altezza. Successivamente è stato richiesto di rispondere ad alcune domande generali con un numero da 1 a 5 (1 per nulla, 2 leggermente, 3 moderatamente, 4 abbastanza, 5 estremamente) oppure in maniera affermativa o negativa (sì o no) in funzione del quesito. Queste domande sono state formulate in maniera tale da definire precisamente la conoscenza del partecipante in ambito BCI ed AR oltre ad individuare eventuali problemi di vista come miopia, astigmatismo o daltonismo che potrebbero risultare rilevanti per eventuali scarse prestazioni durante i test.

Sono stati, infine, compilati i seguenti questionari (due volte: una per algoritmo) con lo scopo di valutare l'esperienza utente ed il carico di lavoro del sistema.

SUS

Il System Usability Score (SUS) [52] è un questionario atto a valutare soggettivamente l'usabilità di un sistema, è composto da dieci affermazioni alle quali gli utenti devono assegnare un punteggio da 1 a 5 (1 per nulla, 5 estremamente).

A seguire le dieci affermazioni (in inglese, essendo i test svoltisi a Vienna):

- 1) I think I would like to use this system frequently.
- 2) I found the system unnecessarily complex.
- 3) I thought the system was easy to use.
- 4) I think I would need the support of a technical person to be able to use this system.
- 5) I found the various functions in this system were well integrated.
- 6) I thought there was too much inconsistency in this system.
- 7) I would imagine that most people would learn to use this system very quickly.
- 8) I found the system very cumbersome to use.
- 9) I felt very confident using the system.
- 10) I needed to learn a lot of things before I could get going with this system.

NASA-TLX

Il NASA Task Load Index (NASA-TLX) [53] è un questionario atto a valutare il carico di lavoro percepito durante l'utilizzo di un sistema, è composto da due parti: la prima (figura 6.5, sinistra) consiste nell'assegnare una valutazione al carico di lavoro suddiviso in sei classi, la seconda (figura 6.5, destra) consiste nello scegliere per quindici volte quale tra le classi abbia contribuito di più durante l'utilizzo del sistema (la scelta è posta presentando due classi alla volta mutandole ad ogni scelta).

Figura 6.5. Questionario NASA Task Load Index (NASA-TLX)

Le sei classi definite per descrivere il carico di lavoro complessivo sono le seguenti:

- Mental Demand: quanta attività mentale e percettiva è stata richiesta.
- Physical Demand: quanta attività fisica è stata richiesta.
- Temporal Demand: quanta pressione è stata sentita a causa del ritmo con cui si sono verificati i compiti o gli elementi dell'attività.
- Overall Performance: quanta soddisfazione è stata provata durante la prestazione.
- Effort: quanto è stato difficile (mentalmente e fisicamente) conseguire il risultato ottenuto.
- Frustration Level: quanta frustrazione è stata provata.

Valutazione e commento finale

Contrariamente al SUS ed al NASA-TLX, i quali sono stati compilati due volte (una dopo ogni algoritmo), alla conclusione del test è stato chiesto (utilizzando un Single-Item Questionnaire) di valutare soggettivamente l'esperienza per ogni algoritmo in termini di difficoltà mediante una valutazione da 1 a 7 (1 molto difficile, 7 molto facile).

Infine, sono stati lasciati agli utenti alcuni minuti conclusivi per commentare l'esperienza svolta utilizzando una o più frasi e dichiarare eventuali preferenze o considerazioni personali.

6.2 Risultati

I dati registrati durante i test sono stati successivamente analizzati per valutare l'usabilità del sistema ed evidenziare eventuali differenze tra i due algoritmi proposti.

Per ogni insieme di dati, è stato inizialmente utilizzato il test di Shapiro-Wilk per la verifica della normalità, adatto soprattutto per piccoli campioni. Come è evidenziato in figura 6.6, la differenza tra una distribuzione normale ed una distribuzione non normale è la forma della stessa: se presenta una forma a campana è definibile come normale e descrivibile attraverso il valore centrale e la deviazione standard σ (varianza σ^2), contrariamente è definibile come non normale. Se il valore p risultante dal test di Shapiro-Wilk dovesse essere inferiore al livello alfa di 0.05 (ipotesi nulla rifiutata), la distribuzione risulterebbe non normale mentre se dovesse essere superiore (ipotesi nulla non rifiutata) la distribuzione risulterebbe normale. Questa caratterizzazione iniziale degli insiemi di dati registrati risulta essere essenziale poiché permette di definire il processo di analisi dal momento che per distribuzioni normali è necessario utilizzare test parametrici mentre per distribuzioni non normali sono validi esclusivamente i test non parametrici.

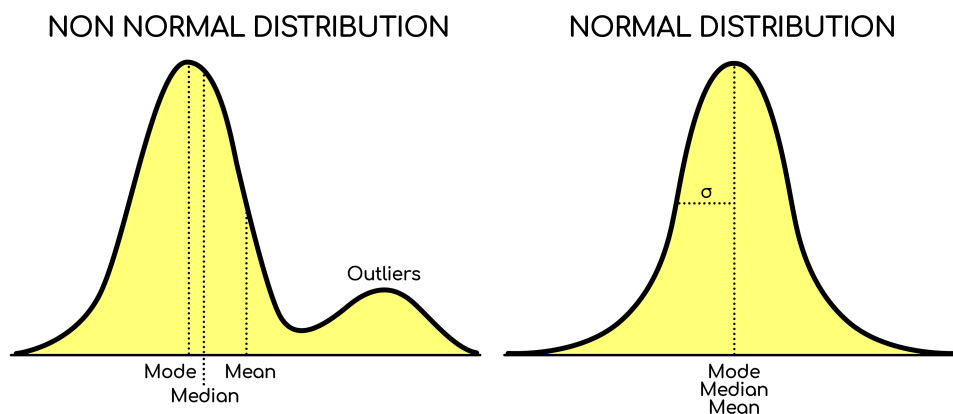


Figura 6.6. Distribuzione non normale (sinistra) e normale (destra)

Successivamente, sono state computate le differenze tra i campioni degli insiemi di dati dei due algoritmi e rappresentate tramite box plot con lo scopo di studiarne la simmetria. Come si evince da figura 6.7, i dati statistici del valore minimo (Min), massimo (Max), mediana (Q_2), primo quartile (Q_1) e terzo quartile (Q_3) sono schematizzati mediante la costruzione di un rettangolo e due linee, una ad indicare l'intervallo complessivo dei valori e l'altra a definirne la mediana. La distribuzione delle differenze risulta essere simmetrica se presenta le seguenti condizioni: la mediana (Q_2) risulta essere il valore centrale tra il primo quartile (Q_1) ed il terzo (Q_3), la differenza tra il primo quartile (Q_1) ed il valore minimo (Min) è la medesima della differenza tra il valore massimo (Max) ed il terzo quartile (Q_3), non sono presenti valori anomali (outlier). Ciascuna coppia di insiemi di dati distribuiti non normalmente è risultata essere asimmetrica, anche a causa del modesto numero di utenti testati.

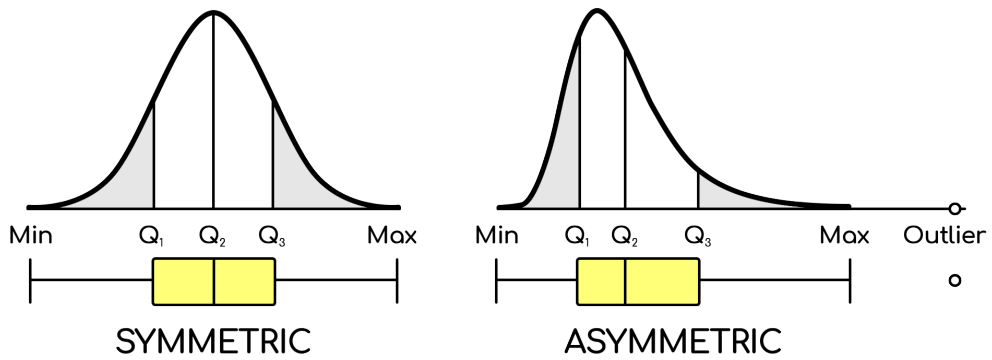


Figura 6.7. Box plot simmetrico (sinistra) e asimmetrico (destra)

Per quanto concerne le distribuzioni non normali e asimmetriche si è proceduto con l'utilizzo del test dei segni, ovvero un test non parametrico che, adoperando i valori mediani, permette di identificare la presenza di una differenza statisticamente significativa tra due insiemi di dati. Per quanto concerne le distribuzioni normali, invece, è stato utilizzato il t-test per campioni dipendenti, ovvero un test parametrico simile al precedente. In entrambi i casi, se il valore p risultante si dimostrasse essere inferiore al livello alfa di 0.05, la differenza si rivelerebbe statisticamente significativa mentre, se si dimostrasse essere superiore, non si avrebbero sufficienti prove per affermare la presenza di una differenza statistica tra i due insiemi di dati analizzati.

6.2.1 SUS

In figura 6.8 sono presenti i punteggi finali di entrambi gli algoritmi suddivisi per utente.

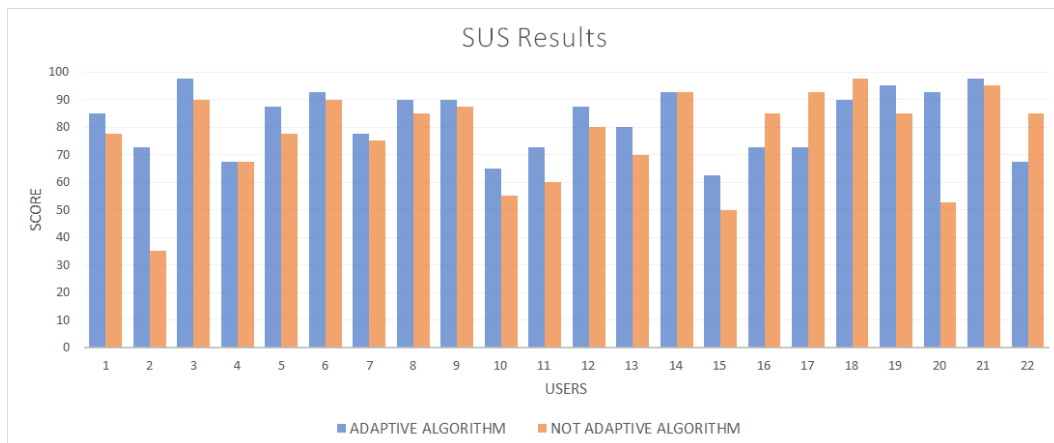


Figura 6.8. Punteggi del System Usability Score (SUS)

Il punteggio finale del System Usability Score è calcolato mediante i seguenti passaggi:

$$\begin{aligned} X &= (\text{Somma punteggi quesiti dispari}) - 5 \\ Y &= 25 - (\text{Somma punteggi quesiti pari}) \\ \text{Punteggio} &= (X + Y) * 2.5 \end{aligned}$$

Questo poiché punteggi elevati nei quesiti dispari (ovvero affermazioni positive) necessitano di un peso positivo e, quindi, devono essere espressi da termini additivi, mentre punteggi elevati nei quesiti pari (ovvero affermazioni negative) necessitano di un peso negativo e, quindi, devono essere espressi da termini sottrattivi.

In generale, il sistema è caratterizzato da un'ottima usabilità poiché, secondo la scala di interpretazione dei risultati SUS consultabile in figura 6.9, l'algoritmo adattivo ha ottenuto un punteggio medio complessivo di 82.16 su 100 (grado A, categoria eccellente) mentre l'algoritmo non adattivo ha ottenuto un punteggio medio complessivo di 76.59 su 100 (grado B, categoria buona).

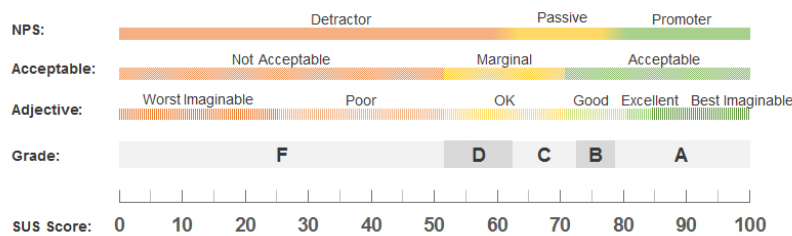


Figura 6.9. Scala di interpretazione dei risultati SUS

Come si evince dalla tabella 6.1, le distribuzioni dei dati analizzati risultano non normali ($p < 0.05$ nel test di Shapiro-Wilk). Di conseguenza, è stato computato il valore mediano per entrambi gli insiemi di dati e, utilizzando il test dei segni, è stato ricavato il valore $p = 0.012$, inferiore al livello alfa di 0.05 risultando pertanto statisticamente significativo.

	<i>Shapiro-Wilk test</i>	<i>Median</i>	<i>Sign test</i>
<i>A_SUS</i>	0,049	34,5	
<i>NA_SUS</i>	0,039	33	0,012

Tabella 6.1. Risultati analisi statistica del System Usability Score (SUS)

Si può, quindi, affermare che l'algoritmo adattivo risulta essere migliore di quello non adattivo in termini di usabilità. In conclusione, il sistema è contraddistinto da un'ottima operabilità poiché, sebbene quello adattivo sia risultato migliore, entrambi gli algoritmi hanno complessivamente ottenuto punteggi molto soddisfacenti.

6.2.2 NASA-TLX

In figura 6.10 sono presenti i punteggi descrittivi del carico di lavoro complessivo di entrambi gli algoritmi suddivisi per utente. Si noti che un carico di lavoro esiguo descrive uno sforzo complessivo contenuto (sia fisico che mentale) e, quindi, risulta essere migliore di un valore elevato che rappresenta un carico di lavoro ingente, caratteristica negativa durante l'uso di un sistema.

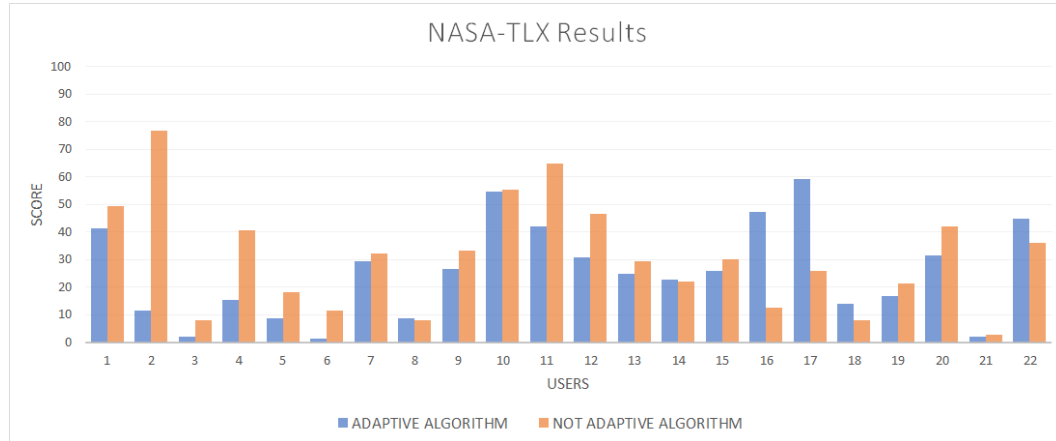


Figura 6.10. Punteggi del NASA-TLX

Il software NASA-TLX provvede a calcolare autonomamente il carico di lavoro complessivo basandosi sulle sei classi descritte precedentemente nel paragrafo relativo della sezione 6.1.3. Secondo gli intervalli di interpretazione dei risultati consultabili in tabella 6.2, l'algoritmo adattivo ha ottenuto un punteggio complessivo di 25.48 su 100 (carico di lavoro medio) mentre l'algoritmo non adattivo ha ottenuto un punteggio complessivo di 30.64 su 100 (carico di lavoro alquanto alto). Questi risultati, sebbene non siano eccessivamente elevati, sono giustificati dallo sforzo cognitivo e fisico che è necessario effettuare durante l'utilizzo dell'interfaccia neurale a cui i partecipanti non sono abituati.

<i>Workload</i>	<i>Value</i>
Low	0-9
Medium	10-29
Somewhat high	30-49
High	50-79
Very high	80-100

Tabella 6.2. Intervalli di interpretazione dei risultati NASA-TLX

Come si evince dalla tabella 6.3, le distribuzioni dei dati analizzati risultano non normali ($p < 0.05$ nel test di Shapiro-Wilk) ad eccezione del carico di lavoro complessivo che risulta essere distribuito normalmente ($p > 0.05$ nel test di Shapiro-Wilk). Ad ogni modo, nonostante sia stato utilizzato il test dei segni per le sei classi distribuite non normalmente (computando i valori mediani degli insiemi) ed il t-test per il carico di lavoro complessivo distribuito normalmente, sono stati ottenuti valori p maggiori al livello alfa di 0.05 non risultando pertanto statisticamente significativi.

	<i>Shapiro-Wilk test</i>	<i>Median</i>	<i>Sign test</i>	<i>Dependent t-test</i>
<i>A_Mental</i>	0,013	110	0,238	
<i>NA_Mental</i>	0,012	100		
<i>A_Physical</i>	0,000	5	0,109	
<i>NA_Physical</i>	0,000	0		
<i>A_Temporal</i>	0,001	20	0,754	
<i>NA_Temporal</i>	0,000	25		
<i>A_Performance</i>	0,019	50	0,454	
<i>NA_Performance</i>	0,055	70		
<i>A_Effort</i>	0,017	100	1,000	
<i>NA_Effort</i>	0,015	85		
<i>A_Frustration</i>	0,000	0	0,227	
<i>NA_Frustration</i>	0,000	0		
<i>A_Overall</i>	0,388			0,230
<i>NA_Overall</i>	0,409			

Tabella 6.3. Risultati analisi statistica del NASA-TLX

Si può affermare, quindi, che il carico di lavoro complessivo durante l'utilizzo del sistema è risultato essere medio-alto, sebbene non si possa attestare una differenza statisticamente significativa nel carico di lavoro durante l'esecuzione dell'algoritmo adattivo e di quello non adattivo.

6.2.3 Single-Item Questionnaire

In figura 6.11 sono presenti le valutazioni finali della complessità delle prove di entrambi gli algoritmi suddivisi per utente. Valori esigui identificano una difficoltà maggiore mentre valori ingenti identificano una difficoltà minore, la scala di valutazione utilizzata è la seguente: 0 estrema complessità della prova, 7 estrema semplicità della prova.

La media complessiva dei punteggi registrati risulta essere di 5.78 su 7 per quanto riguarda l'algoritmo adattivo e di 5.18 per quanto riguarda quello non adattivo. In generale, quindi, le prove svolte dai partecipanti mediante l'utilizzo del sistema non sono state ritenute eccessivamente complesse ma, al contrario, abbastanza semplici.

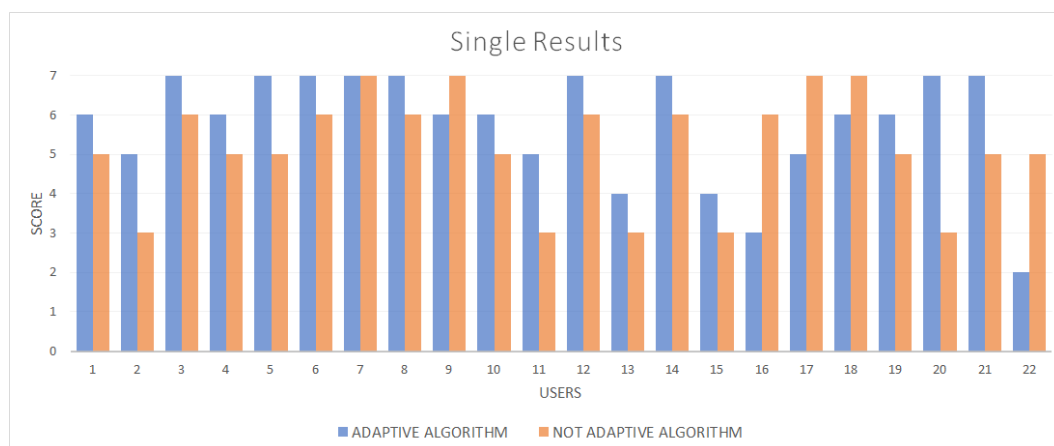


Figura 6.11. Punteggi del Single-Item Questionnaire

Come si evince dalla tabella 6.4, le distribuzioni dei dati analizzati risultano non normali ($p < 0.05$ nel test di Shapiro-Wilk). Di conseguenza, è stato computato il valore mediano per entrambi gli insiemi di dati e, utilizzando il test dei segni, è stato ricavato il valore $p = 0.027$, inferiore al livello alfa di 0.05 risultando pertanto statisticamente significativo.

	<i>Shapiro-Wilk test</i>	<i>Median</i>	<i>Sign test</i>
<i>A_SUS</i>	0,001	6	0,027
<i>NA_SUS</i>	0,005	5	

Tabella 6.4. Risultati analisi statistica del Single-Item Questionnaire

Si può, quindi, affermare che l’algoritmo adattivo risulta essere di più facile utilizzo rispetto a quello non adattivo a parità di prove svolte. Dal momento che i test utente condotti sono stati concepiti come simulazioni di casi d’uso reale, emerge una generale semplicità ed intuitività di utilizzo del sistema.

6.2.4 Prestazioni

Come enunciato in sezione 6.1.2, le prestazioni di ogni utente sono state computate e registrate dal sistema mediante l’utilizzo dei seguenti parametri: i tempi di selezione dei singoli NeuroTag e gli errori (o falsi positivi) rilevati durante la selezione dei singoli NeuroTag. Le distribuzioni di tutti gli insiemi di dati analizzati sono risultate essere non normali e asimmetriche, essendo le prestazioni delle interfacce neurali estremamente dipendenti da caratteristiche individuali (come l’attività cerebrale o la capacità di concentrazione) e, quindi, soggette ad una maggior inconsistenza.

A seguire, sono analizzati entrambi i parametri osservati.

Tempi

In figura 6.12 sono consultabili i tempi medi di selezione dei singoli NeuroTag (comprensivi del tempo di ricerca nella UI) ed i tempi medi di completamento di ogni singola prova per entrambi gli algoritmi.

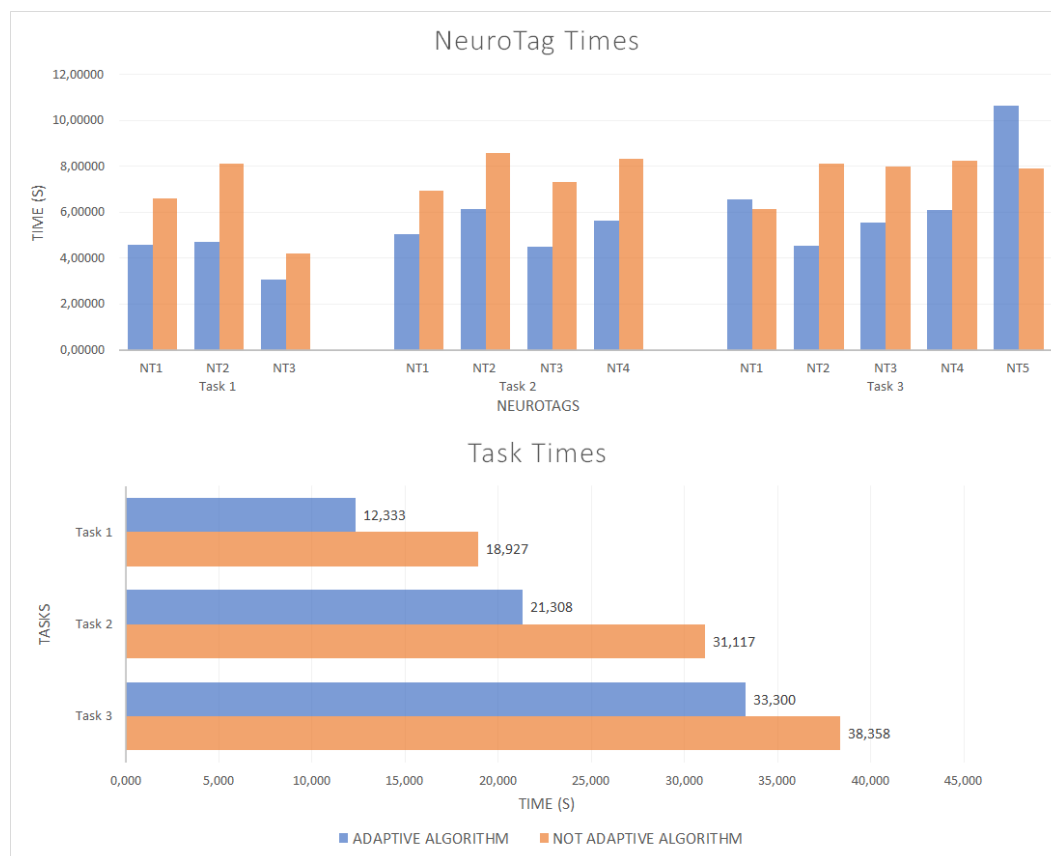


Figura 6.12. Tempi di selezione dei NeuroTag e tempi complessivi delle prove

In generale, l'algoritmo adattivo ha permesso di ottenere tempi notevolmente minori permettendo agli utenti di trovare più facilmente nella UI i NeuroTag associati agli oggetti da selezionare. Questo è stato prevalentemente possibile poiché i NeuroTag sono stati proiettati il più vicino possibile agli oggetti e, quindi, è stato permesso agli utenti di muovere minimamente lo sguardo senza perdere tempo nella ricerca del NeuroTag associato. Sebbene sia evidente la supremazia prestazionale dell'algoritmo adattivo, con l'aumentare dei NeuroTag proiettati si può notare come l'algoritmo non adattivo, offrendo una UI ordinata e lineare, vada a performare più similmente a quello adattivo che invece, con l'aumentare degli oggetti nell'area di lavoro, può risultare maggiormente caotico.

Tra tutti i tempi medi analizzati, è presente un'unica singolarità considerevole: il quinto NeuroTag della terza prova. Questa anomalia è causata dalla posizione del relativo oggetto da selezionare (figura 6.4, oggetto 7 della terza configurazione) essendo questo

molto distante dagli utenti e, quindi, più difficile da selezionare utilizzando l’algoritmo adattivo in confronto a quello non adattivo che permette di disporre tutti i NeuroTag in prossimità della posizione dell’utente.

In tabella 6.5 sono presentati i valori dell’analisi statistica dei tempi di selezionamento. Sebbene i tempi registrati durante l’utilizzo dell’algoritmo adattivo siano abbastanza inferiori rispetto a quello non adattivo, solo il tempo complessivo medio della seconda prova è risultato essere statisticamente significativo, ottenendo un valore p inferiore al livello alfa di 0.05 mediante l’utilizzo del test dei segni.

	<i>Mean (s)</i>	<i>Median (s)</i>	<i>Sign test</i>
<i>A_T1</i>	12,333	11,365	0,134
<i>NA_T1</i>	18,927	15,190	
<i>A_T2</i>	21,308	20,215	0,017
<i>NA_T2</i>	31,117	22,274	
<i>A_T3</i>	33,300	30,660	0,523
<i>NA_T3</i>	38,358	29,965	

Tabella 6.5. Risultati analisi statistica dei tempi di selezionamento

In conclusione, entrambi gli algoritmi sono caratterizzati da un’ottima rapidità di esecuzione, misurando un tempo medio di ricerca e selezione dei singoli NeuroTag di 5.578 secondi per l’algoritmo adattivo e di 6.867 secondi per quello non adattivo.

Errori

In figura 6.13 sono consultabili gli errori complessivi rilevati durante le prove.

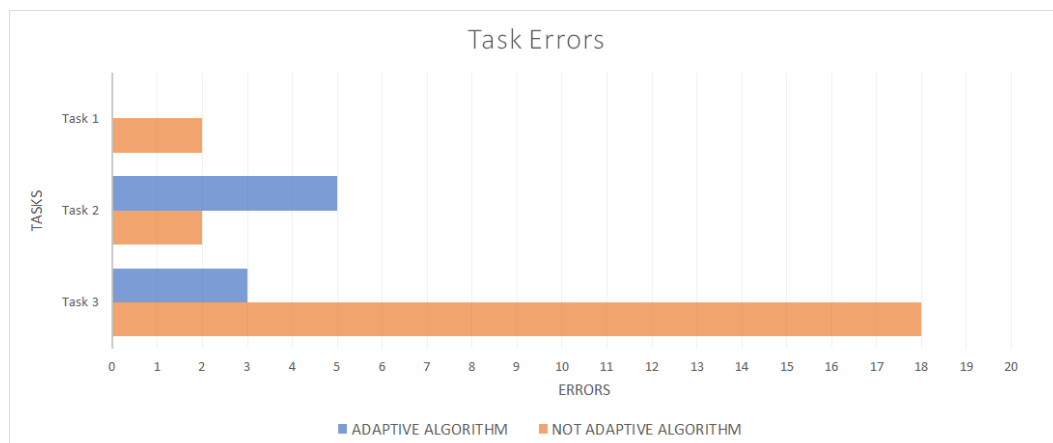


Figura 6.13. Errori complessivi rilevati durante le prove

In generale, il sistema integrato all’algoritmo adattivo è risultato essere molto solido agli errori offrendo una precisione abbastanza consistente. Al contrario, sebbene durante le prime due prove siano stati rilevati complessivamente pochi errori, l’algoritmo non adattivo si è rivelato essere di notevole imprecisione nel corso della terza prova a causa dell’eccessiva vicinanza spaziale dei NeuroTag proiettati, come evidenziato anche precedentemente nel capitolo 4 relativo allo studio preliminare del NextMind. Infatti il NextMind, basandosi sulla tecnologia SSVEP-EEG, permette di innescare i NeuroTag grazie all’attività cerebrale generata dagli stimoli visivi che, se spazialmente troppo vicini, rischiano di sovrapporsi facendo diminuire il livello di confidenza dell’algoritmo di intelligenza artificiale integrato al dispositivo.

In tabella 6.6 sono presenti i valori dell’analisi statistica degli errori complessivi rilevati durante le prove. Tutte le mediane calcolate sono risultate nulle poiché la maggior parte degli utenti è riuscita a non commettere errori producendo una distribuzione estremamente asimmetrica verso destra con molti valori nulli. Infine, come è intuitivo, solo la terza prova è risultata essere statisticamente significativa, ottenendo mediante l’utilizzo del test dei segni un valore $p = 0.021$ inferiore al livello alfa di 0.05.

	<i>Mean</i>	<i>Median</i>	<i>Sign test</i>
<i>A_E1</i>	0	0	0,500
<i>NA_E1</i>	0,09	0	
<i>A_E2</i>	0.23	0	0,687
<i>NA_E2</i>	0.09	0	
<i>A_E3</i>	0.14	0	0,021
<i>NA_E3</i>	0.82	0	

Tabella 6.6. Risultati analisi statistica degli errori

Di conseguenza, è possibile affermare che il sistema integrato all’algoritmo di posizionamento adattivo risulta essere preciso e solido agli errori indipendentemente dal numero di oggetti presenti nell’area di lavoro mentre, al contrario, l’algoritmo di posizionamento non adattivo risulta essere impreciso in presenza di molteplici oggetti nell’area di lavoro a causa della vicinanza spaziale dei NeuroTag proiettati.

Capitolo 7

Conclusioni e sviluppi futuri

Questo capitolo è dedicato alle conclusioni complessive dello studio condotto e, successivamente, alla proposta di eventuali sviluppi futuri per il miglioramento del sistema.

7.1 Conclusioni

Oggi giorno la Human-Computer Interaction (HCI) mediante interfacce neurali, sebbene non sia uno dei sistemi di interazione più popolare, si può definire essere in un periodo di maturazione proponendosi come innovativo in molteplici settori. L'ambito di utilizzo più evoluto è sicuramente la robotica di servizio che, stimolata dal proposito di migliorare lo stile di vita di persone con limitazioni motorie o difficoltà di linguaggio, è oggetto di ricerca da diversi anni. Nonostante le BCI invasive siano in proporzione largamente utilizzate in ambito medico per permettere l'interazione tra uomo e macchina, quelle non invasive, impiegate anche per altri scopi, risultano essere ancora leggermente scomodi da indossare a causa della presenza degli elettrodi. Per quanto precise ed affidabili, infatti, le interfacce neurali non invasive presentano un'usabilità ed un carico di lavoro necessario per l'utilizzo tale da essere tollerabile per un periodo limitato di tempo, rendendole complicate da indossare ed utilizzare per molte ore al giorno in ambienti lavorativi o ludici.

Nel futuro prossimo, l'interazione uomo-robot diventerà progressivamente più importante e quotidiana stabilendo un legame sempre più forte tra umani e macchine. Le interfacce neurali, con ogni probabilità, si evolveranno e saranno concepite nuove tecnologie in grado di decodificare con precisione le intenzioni umane senza la necessità di stimoli esterni come per le BCI basate su SSVEP o l'osservazione di attività biologiche come i cambiamenti del flusso sanguigno. Inoltre, anche a causa dell'avvento del Metaverso¹, è possibile la futura nascita di una neurorealtà, ovvero un sistema neuroreale che si interfaccia direttamente con la biologia dell'utente attraverso una BCI, permettendo di sviluppare una nuova tipologia di immersività.

¹Ipotetica iterazione di Internet come un unico mondo virtuale universale ed immersivo

Dispositivo NextMind

L'interfaccia neurale NextMind si è rivelata essere in generale di ottima precisione ed affidabilità, fornendo un'indossabilità agevole in confronto alla maggior parte delle BCI in commercio. Invece, per quanto riguarda l'integrazione con visori di realtà virtuale e realtà aumentata, non è risultata universalmente indossabile con gli HMD HoloLens 2 (HMD-AR) e Oculus Rift (HMD-VR) poiché eccessivamente stretti per molti utenti.

Utilizzando come dispositivo di visualizzazione della UI il monitor o il visore VR, l'interfaccia neurale ha permesso di ottenere prestazioni soddisfacenti. Inoltre, sebbene integrata al visore AR non si sia dimostrata particolarmente performante, utilizzando l'interfaccia utente in realtà aumentata proiettata si è rivelata essere particolarmente versatile ed efficace grazie alla dimensione ed alla definizione della stessa.

Per quanto riguarda la progettazione dei NeuroTag, è necessario fornire un contrasto elevato tra lo sfondo a tinta unita e la texture ad intermittenza per garantire la massima resa dell'interfaccia neurale. È inoltre importante fornire un riscontro visivo (e se necessario uditivo) all'utente durante il processo di selezione tramite l'animazione di componenti in sovrapposizione o nei pressi del NeuroTag.

Sistema

Il sistema progettato è risultato essere preciso ed intuitivo da utilizzare, presentando come innovazione l'interfaccia utente in realtà aumentata proiettata. Entrambi gli algoritmi progettati si sono rivelati essere efficienti ma, in generale, l'algoritmo adattivo è risultato essere migliore di quello non adattivo in termini di usabilità, facilità d'uso e prestazioni. Complessivamente, però, la scelta migliore dell'algoritmo di posizionamento da usare può dipendere dal caso d'uso poiché entrambi gli algoritmi presentano sia vantaggi che svantaggi.

Inerentemente all'algoritmo adattivo, il vantaggio maggiore si identifica nella possibilità di utilizzare la totalità dello spazio dell'interfaccia utente proiettata, ottimizzandola in maniera tale da proiettare i NeuroTag il più vicino possibile agli oggetti. Inoltre, l'algoritmo prende in considerazione la posizione virtuale dell'utente, utilizzandola per disporre i NeuroTag nella sua direzione con lo scopo di facilitarne la selezione. Gli svantaggi più rilevanti dell'algoritmo adattivo, invece, sono la complessità computazionale non irrilevante per il calcolo della soluzione (sono necessarie molteplici operazioni fisiche e geometriche) e la possibile assenza di una soluzione valida (nel caso di molti oggetti posizionati vicini tra loro in un angolo dell'area di lavoro). Infine, se l'algoritmo non dovesse trovare una soluzione migliore, è possibile che proietti un NeuroTag nella sezione della UI più lontana dall'utente, rendendone più complessa l'attivazione.

Inerentemente all'algoritmo non adattivo, i vantaggi maggiori si identificano nella semplicità computazionale per il calcolo della soluzione e nell'infallibilità dell'algoritmo nel trovare una soluzione valida dal momento che viene riservata un'area dell'interfaccia utente per l'esclusiva proiezione dei NeuroTag. Per questo motivo, però, l'area di posizionamento degli oggetti risulta essere sensibilmente minore. Un altro vantaggio dell'algoritmo è il proiettare tutti i NeuroTag molto vicini all'utente, facilitandone di conseguenza la selezione. Infine, come commentato in sezione 6.2.4, un grosso svantaggio dell'algoritmo non

adattivo risulta essere l'imprecisione dell'interfaccia neurale in presenza di molti oggetti a causa della vicinanza spaziale dei NeuroTag.

In tabella 7.1 è presente un riepilogo dei vantaggi e degli svantaggi dei due algoritmi di posizionamento progettati.

<i>Algoritmo</i>	<i>Vantaggi</i>	<i>Svantaggi</i>
<i>Adattivo</i>	Ottimizzazione dello spazio Considera la posizione dell'utente NeuroTag vicini agli oggetti	Complesso da computare Presenza di soluzioni invalide NeuroTag distanti dall'utente
<i>Non adattivo</i>	Facile da computare Soluzione sempre valida NeuroTag vicini all'utente	Area di posizionamento minore Impreciso con tanti oggetti NeuroTag distanti dagli oggetti

Tabella 7.1. Riepilogo vantaggi e svantaggi dei due algoritmi progettati

7.2 Sviluppi futuri

A seguire, alcuni possibili sviluppi futuri del sistema progettato.

Rilevamento oggetti

Il sistema attuale permette di identificare la posizione degli oggetti attraverso un codice Python per il rilevamento del colore dal momento che i prototipi degli oggetti utilizzati sono stati dipinti di rosso. Un possibile miglioramento del sistema potrebbe riguardare l'integrazione di un'intelligenza artificiale per identificare e classificare specifici gruppi di oggetti in funzione dell'utilizzo del sistema. Un algoritmo particolarmente veloce per il rilevamento di oggetti è YOLO [54] che, oltre ad essere utile per definire posizione e dimensione degli oggetti, può anche essere impiegato per riconoscerne la tipologia. Questa caratteristica risulterebbe di particolare utilità poiché potrebbe permettere di associare delle informazioni geometriche agli oggetti identificati e, quindi, permettere al braccio meccanico di posizionare il gripper (il quale potrebbe essere sostituito con una pinza a tre dita per permettere anche il prelevamento di oggetti non magnetici) all'altezza e orientamento corretti in funzione dell'oggetto da prelevare.

Inoltre, il flusso di immagini prodotto dalla videocamera Logitech QuickCam Pro è trasmesso alla risoluzione di 640x480px a 30fps, entrambi parametri migliorabili sebbene sia necessario non utilizzare valori troppo elevati dal momento che è comunque necessario un tempo di elaborazione delle immagini che potrebbe rallentare considerevolmente il sistema. Un esempio di videocamera utilizzabile potrebbe essere la OpenCV OAK-D Camera [55] che integra dei modelli di intelligenza artificiale tra i quali YOLO ed, inoltre, è attualmente a disposizione del dipartimento di ricerca all'Università tecnica di Vienna.

Algoritmi di posizionamento

Un miglioramento sostanziale degli algoritmi progettati risulterebbe essere il funzionamento degli stessi in tempo reale. Per permettere ciò, sebbene molte delle operazioni svolte siano già state ottimizzate, è necessario diminuire ulteriormente la complessità computazionale dell'algoritmo adattivo. Uno dei modi migliori sarebbe quello di semplificare la ricerca delle celle valide nella matrice mediante l'utilizzo di una struttura dati più efficiente dell'albero quadramentale utilizzato come, ad esempio, un albero k-dimensionale o un algoritmo ibrido [56]. Tuttavia risulta importante identificare un compromesso tra la velocità di generazione della struttura dati e la rapidità di ricerca poiché, sebbene un'individuazione rapida sia essenziale, l'algoritmo genererebbe diverse strutture dati al secondo.

Tracciamento della posizione utente

L'algoritmo adattivo è già stato progettato e sviluppato per essere dipendente dalla posizione dell'utente nello spazio. Infatti, la posizione tridimensionale simulata dell'utente è utilizzata sia per il calcolo delle occlusioni, come mostrato precedentemente in figura 5.18, sia per il posizionamento dei NeuroTag, come mostrato in figura 5.20. Utilizzando i tracker HTC VIVE [57] attualmente a disposizione del dipartimento di ricerca all'Università tecnica di Vienna, risulterebbe infatti molto semplice l'implementazione del tracciamento dell'utente dal momento che risultano già installati quattro sensori a muro nella stanza contenente il braccio meccanico.

Colori interfaccia utente

Come dettagliato in sezione 5.5, i colori utilizzati per la realizzazione delle interfacce utente sono ispirati ai colori impiegati per rappresentare le metropolitane delle più grandi città al mondo. Una possibile evoluzione cromatica della UI del sistema potrebbe essere definita mediante l'applicazione del teorema dei quattro colori [58] il quale afferma che, data una superficie piana divisa in regioni connesse, sono sufficienti quattro colori per colorare ogni regione facendo in modo che a regioni adiacenti non sia assegnato lo stesso colore. Risulterebbe infatti possibile descrivere univocamente la posizione degli oggetti rilevati mediante un grafo ed identificare quali colori assegnare ad ogni oggetto e NeuroTag associato senza il rischio di proiettare due colori uguali adiacenti. La stessa tecnica è utilizzata per la colorazione delle mappe geografiche di tutto il mondo.

Paragone con tracciamento oculare

Durante questo studio sono stati testati e comparati i due algoritmi di posizionamento proposti. Potrebbe risultare rilevante confrontare l'interfaccia neurale NextMind con un dispositivo per il tracciamento oculare in modo tale da integrare un paragone con una tecnologia di selezione diversa ma che permetta di interagire con il sistema senza l'utilizzo delle mani o della voce. Inoltre, la tecnologia del tracciamento oculare permetterebbe un utilizzo analogo a quello delle BCI basate su stimoli visivi (come il NextMind) da parte di persone con limitazioni motorie o difficoltà di linguaggio.

Sistema integrato ad una sedia a rotelle

Per concludere, uno sviluppo futuro di particolare interesse potrebbe essere l'integrazione del sistema ad una sedia a rotelle installando un braccio meccanico più leggero dello Stäubli TX2-60, il quale risulterebbe eccessivamente pesante per il movimento della sedia, ed un proiettore congiunto ad una videocamera. Tramite un modello di intelligenza artificiale come il sopracitato YOLO sarebbe possibile identificare gli oggetti davanti all'utente sulla sedia a rotelle e, tramite il proiettore, si potrebbero proiettare i NeuroTag (o dei componenti sostitutivi che generino comunque una texture ad intermittenza) direttamente sugli oggetti. Ovviamente, con lo scopo di creare contrasto con la texture proiettata, sarebbe necessario utilizzare degli oggetti opachi oppure installare delle placche opache adibite alla proiezione dei NeuroTag in tempo reale. Così facendo, sarebbe possibile creare una sedia a rotelle intelligente che permetta di prendere degli oggetti non raggiungibili altrimenti da pazienti con disabilità motorie.

Appendice A

Codice Arduino gripper

```
1 int relay = 8;
2 char state = '2';
3
4 void setup() {
5
6   pinMode(relay, OUTPUT);
7   digitalWrite(relay, HIGH);
8   Serial1.begin(9600);
9
10 }
11
12 void loop() {
13
14   if (Serial1.available() > 0) { // Checks incoming data
15     state = Serial1.read(); // Reads data
16   }
17   if (state == '0') {
18     digitalWrite(relay, HIGH); // Turn Relay OFF if 0
19     state = '2';
20   } else if (state == '1') {
21     digitalWrite(relay, LOW); // Turn Relay ON if 1
22     state = '2';
23   } else if (state != '2'){
24     state = '2';
25   }
26
27 }
```


Appendice B

Controlli sistema

UNITY APP

Calibration:

S: start NextMind calibration
Numpad: set user height (del to delete)
Arrows: navigate NextMind calibration
H: hide user interface
1: start algorithm 1
2: start algorithm 2
L: lock tags except for tests
Q: quit

Algorithm 1:

H: hide bboxes and tags (for rescans)
D: debug map
Back: return to calibration
Numpad 1: show test 1 placement
Numpad 2: show test 2 placement
Numpad 3: show test 3 placement
Numpad 4: start test 1
Numpad 5: start test 2
Numpad 6: start test 3
L: lock tags except for tests
R: reset robot parameters manually
Q: quit

Algorithm 2:

H: hide bboxes and tags (for rescans)
Back: return to calibration
Numpad 1: show test 1 placement
Numpad 2: show test 2 placement
Numpad 3: show test 3 placement
Numpad 4: start test 1
Numpad 5: start test 2
Numpad 6: start test 3
L: lock tags except for tests
R: reset robot parameters manually
Q: quit

COLOR DETECTION SCRIPT

Calibration:

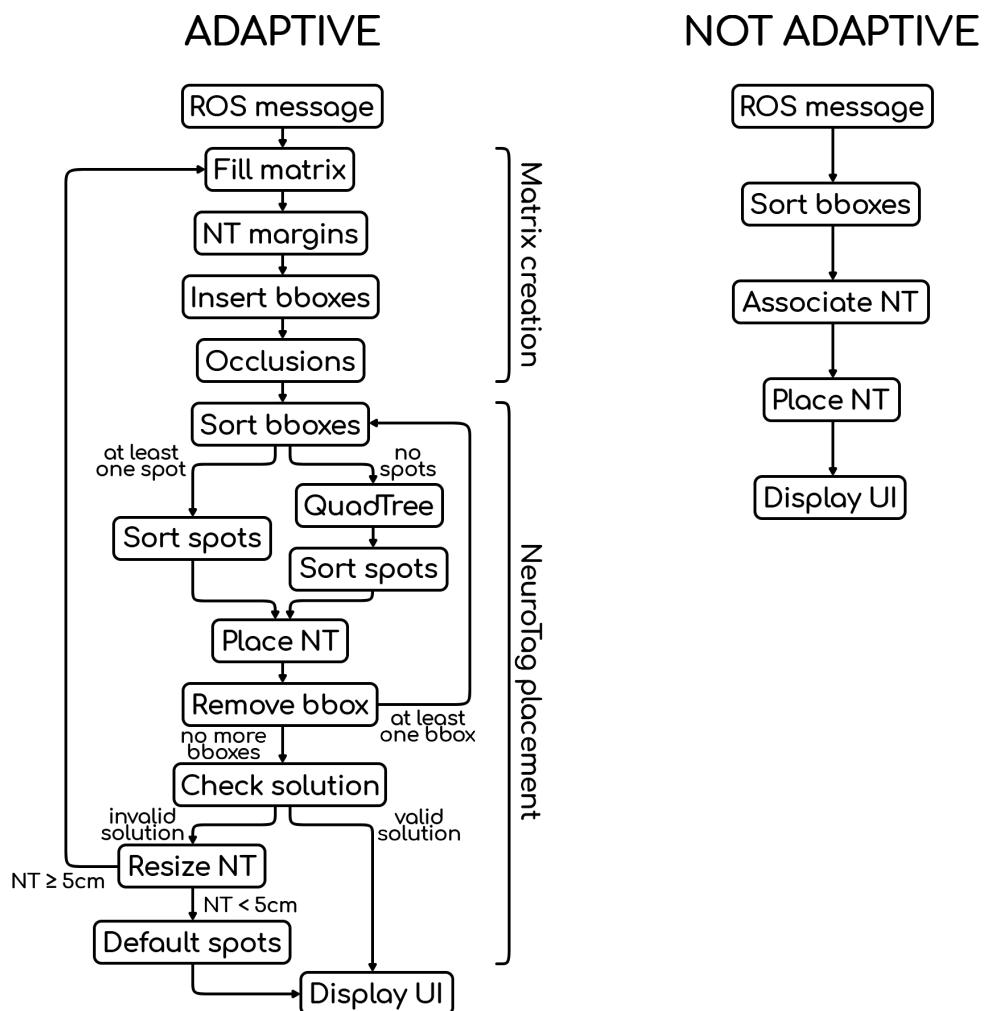
R: reset parameters
C: confirm calibration
Q: quit

Detection:

P: publish bboxes
Q: quit

Appendice C

Algoritmi di posizionamento



Pseudocodice

Algorithm 1 Adaptive

```

receive_vector                                ▷ from ROS message
generate_matrix
while bboxes.count > 0 do
    detect_availableSpots                    ▷ for every bbox
    sort_bboxes                              ▷ by available spots
    if bbox.availableSpots.count > 0 then    ▷ bbox with less available spots
        sort_availableSpots                ▷ by user distance
        place_neuroTag                     ▷ in the nearest spot
    else
        generate_quadtrees
        while availableSpots = 0 do
            increase_researchArea           ▷ of quadtree
            search_availableSpots          ▷ within the research area
            if found_availableSpots then
                sort_availableSpots        ▷ by user distance
                place_neuroTag             ▷ in the nearest spot
            end if
        end while
    end if
    remove_bbox_from_list
end while
check_solution
if invalid_solution then
    invalid_solutions_counter++
    if invalid_solutions_counter < 10 then
        neuroTagRadius -= 0.25            ▷ radius in cm
        goto generate_matrix
    else
        default_spots
    end if
end if
display_ui                                    ▷ projected

```

Algorithm 2 Not adaptive

```

receive_vector                                ▷ from ROS message
sort_bboxes                                  ▷ from left to right
associate_neuroTags                          ▷ with bboxes
place_neuroTags                              ▷ in front of the user
display_ui                                    ▷ projected

```

Bibliografia

- [1] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse et al. “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm.” *Nature* 485”. In: *Donoghue*. Citeseer. 2012.
- [2] L. Schiatti, J. Tessadori, G. Barresi, L. S. Mattos e A. Ajoudani. “Soft brain-machine interfaces for assistive robotics: A novel control approach”. In: *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE. 2017, pp. 863–869.
- [3] J. F. Braun, G. Díez-Valencia, S. K. Ehrlich, P. Lanillos e G. Cheng. “A prototype of a P300 based brain-robot interface to enable multi-modal interaction for patients with limited mobility”. In: *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE. 2019, pp. 78–84.
- [4] F. Burget, L. D. J. Fiederer, D. Kuhner, M. Völker, J. Aldinger, R. T. Schirrmeister, C. Do, J. Boedecker, B. Nebel, T. Ball et al. “Acting thoughts: Towards a mobile robotic service assistant for users with limited communication skills”. In: *2017 European Conference on Mobile Robots (ECMR)*. IEEE. 2017, pp. 1–6.
- [5] T. L. Olsen e B. Tomlin. “Industry 4.0: Opportunities and challenges for operations management”. In: *Manufacturing & Service Operations Management* 22.1 (2020), pp. 113–122.
- [6] Y. Dmytriyev, F. Insero, M. Carnevale e H. Giberti. “Brain–Computer Interface and Hand-Guiding Control in a Human–Robot Collaborative Assembly Task”. In: *Machines* 10.8 (2022), p. 654.
- [7] A. P. Garcia, I. Schjølberg e S. Gale. “EEG control of an industrial robot manipulator”. In: *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*. IEEE. 2013, pp. 39–44.
- [8] X. Chen, B. Zhao, Y. Wang, S. Xu e X. Gao. “Control of a 7-DOF robotic arm system with an SSVEP-based BCI”. In: *International journal of neural systems* 28.08 (2018), p. 1850018.
- [9] C. Wang, B. Xia, J. Li, W. Yang, A. C. Velez, H. Yang et al. “Motor imagery BCI-based robot arm system”. In: *2011 Seventh International Conference on Natural Computation*. Vol. 1. IEEE. 2011, pp. 181–184.
- [10] Y. Zhu, Y. Li, J. Lu e P. Li. “A hybrid BCI based on SSVEP and EOG for robotic arm control”. In: *Frontiers in Neurorobotics* 14 (2020), p. 583641.

-
- [11] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan e M. A. Nicolelis. “Real-time prediction of hand trajectory by ensembles of cortical neurons in primates”. In: *Nature* 408.6810 (2000), pp. 361–365.
- [12] A. Sanna, F. Manuri, J. Fiorenza e F. De Pace. “BARI: An Affordable Brain-Augmented Reality Interface to Support Human–Robot Collaboration in Assembly Tasks”. In: *Information* 13.10 (2022), p. 460.
- [13] J. Marescaux e F. Rubino. “Robotic surgery: Potentials, barriers, and limitations”. In: *European Surgery* 37.5 (2005), pp. 279–283.
- [14] E. Olivieri, G. Barresi e L. S. Mattos. “BCI-based user training in surgical robotics”. In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2015, pp. 4918–4921.
- [15] A. Athanasiou, I. Xygonakis, N. Pandria, P. Kartsidis, G. Arfaras, K. R. Kavazidi, N. Foroglou, A. Astaras e P. D. Bamidis. “Towards rehabilitation robotics: off-the-shelf BCI control of anthropomorphic robotic arms”. In: *BioMed research international* 2017 (2017).
- [16] A. R. Donati, S. Shokur, E. Morya, D. S. Campos, R. C. Muioli, C. M. Gitti, P. B. Augusto, S. Tripodi, C. G. Pires, G. A. Pereira et al. “Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients”. In: *Scientific reports* 6.1 (2016), pp. 1–16.
- [17] M. Alam, W. Rodrigues, B. N. Pham e N. V. Thakor. “Brain-machine interface facilitated neurorehabilitation via spinal stimulation after spinal cord injury: Recent progress and future perspectives”. In: *Brain Research* 1646 (2016), pp. 25–33.
- [18] W. Haohong, G. Shenghao, Z. Yimin, S. Maojia, W. Heng e D. C. Rompapas. “The Mind Commands You: Combining Brain-Computer Interactions with Augmented Reality to Control Internet of Things (IoT) Tools, and Robotic Platforms”. In: *2022 IEEE 5th International Conference on Electronics Technology (ICET)*. IEEE. 2022, pp. 1026–1031.
- [19] M. Rajmohan, S. C. H. Vali, A. Raj e A. Gogoi. “Home automation using brain computer interface (BCI)”. In: *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. IEEE. 2020, pp. 1–7.
- [20] J. Mangion. “Exploitation of EEG-extracted eye movement for a hybrid SSVEP home automation system”. Tesi di laurea mag. University of Malta, 2020.
- [21] R. A. Miranda, W. D. Casebeer, A. M. Hein, J. W. Judy, E. P. Krotkov, T. L. Laabs, J. E. Manzo, K. G. Pankratz, G. A. Pratt, J. C. Sanchez et al. “DARPA-funded efforts in the development of novel brain–computer interface technologies”. In: *Journal of neuroscience methods* 244 (2015), pp. 52–67.
- [22] A. Czech. “Brain-computer interface use to control military weapons and tools”. In: *International Scientific Conference on Brain-Computer Interfaces BCI Opole*. Springer. 2021, pp. 196–204.

-
- [23] S. N. Resalat e F. Afdideh. “Real-time monitoring of military sentinel sleepiness using a novel SSVEP-based BCI system”. In: *2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences*. IEEE. 2012, pp. 740–745.
- [24] J.-H. Jeong, D.-H. Lee, H.-J. Ahn e S.-W. Lee. “Towards brain-computer interfaces for drone swarm control”. In: *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE. 2020, pp. 1–4.
- [25] A. N. Belkacem e A. Lakas. “A Cooperative EEG-based BCI Control System for Robot–Drone Interaction”. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2021, pp. 297–302.
- [26] A. Y. Kaplan, S. L. Shishkin, I. P. Ganin, I. A. Basyul e A. Y. Zhigalov. “Adapting the P300-based brain–computer interface for gaming: a review”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 5.2 (2013), pp. 141–149.
- [27] J. Katona, I. Farkas, T. Ujbanyi, P. Dukan e A. Kovari. “Evaluation of the NeuroSky MindFlex EEG headset brain waves data”. In: *2014 IEEE 12th international symposium on applied machine intelligence and informatics (SAMi)*. IEEE. 2014, pp. 91–94.
- [28] B. J. Hou, J. P. Hansen, C. Uyanik, P. Bækgaard, S. Puthusserypady, J. M. Araujo e S. MacKenzie. “Feasibility of a Device for Gaze Interaction by Visually-Evoked Brain Signals”. In: *2022 Symposium on Eye Tracking Research and Applications*. 2022, pp. 1–7.
- [29] F. Dehais, S. Ladouce, S. Velut, T.-V. Nong, J. T. Tresols, L. Darnet e G. Ferraro. *A Dual Passive & Reactive-Based SSVEP Brain Computer Interface in the Cockpit*. 2021.
- [30] F. Dehais, S. Ladouce, L. Darnet, T.-V. Nong, G. Ferraro, J. T. Tresols, S. Velut e P. Labedan. “Dual Passive Reactive Brain Computer Interface: a Novel Approach to Human-Machine Symbiosis”. In: *bioRxiv* (2021).
- [31] S. Da Col. “Mixed Reality Application for Inspection and Validation in Industrial Environments: Human Performance and Brain-Computer Interface Advantages over Gestures”. Tesi di dott. University of Windsor (Canada), 2021.
- [32] C. Uyanik, M. A. Khan, R. Das, J. P. Hansen e S. Puthusserypady. “Brainy Home: A Virtual Smart Home and Wheelchair Control Application Powered by Brain Computer Interface.” In: *BIODEVICES*. 2022, pp. 134–141.
- [33] O. Geman, I. Chiuchisan e D. Vicoveanu. “Serious Games used in Rehabilitation of Patients with Osteoarthritis”. In: *2021 International Conference on e-Health and Bioengineering (EHB)*. IEEE. 2021, pp. 1–5.
- [34] S. Ono. “the_neuron_001: performance using brain computer interface”. In: *SIG-GRAPH Asia 2021 Real-Time Live!* 2021, pp. 1–1.
- [36] L. F. Nicolas-Alonso e J. Gomez-Gil. “Brain computer interfaces, a review”. In: *sensors* 12.2 (2012), pp. 1211–1279.

- [37] M. Caproni, F. Orihuela-Espina, D. R. James, A. Menciassi, P. Dario, A. W. Darzi e G.-Z. Yang. “An analysis framework for Near InfraRed Spectroscopy based brain-computer interface and prospective application to robotic surgery”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, pp. 2143–2148.
- [38] F. Dehais, S. Ladouce, L. Darnet, T.-V. Nong, G. Ferraro, J. T. Tresols, S. Velut e P. Labedan. “Dual Passive Reactive Brain Computer Interface: a Novel Approach to Human-Machine Symbiosis”. In: *bioRxiv* (2021).
- [49] F. Manuri, A. Sanna, M. Bosco e F. De Pace. “A Comparison of Three Different NeuroTag Visualization Media: Brain Visual Stimuli by Monitor, Augmented and Virtual Reality Devices”. In: *Intervento presentato al convegno 8th International Conference on Human Interaction & Emerging Technologies tenutosi a Nizza (FR)*. 2022.
- [50] K. Lin, Y. Chen, M. Qiu, M. Zeng e W. Huang. “SLGC: a fast point-in-area algorithm based on scan-line algorithm and grid compression”. In: *2016 11th International Conference on Computer Science & Education (ICCSE)*. IEEE. 2016, pp. 352–356.
- [51] S. Griffioen. “Maximizing color differences in metro maps”. Tesi di dott. Faculty of Science e Engineering, 2014.
- [52] J. Brooke et al. “SUS-A quick and dirty usability scale”. In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.
- [54] C. Liu, Y. Tao, J. Liang, K. Li e Y. Chen. “Object detection based on YOLO network”. In: *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE. 2018, pp. 799–803.
- [56] L. Di e E. G. Yu. “Spatial Search”. In: *Urban Informatics* (2021), pp. 683–699.
- [58] R. Thomas. “An update on the four-color theorem”. In: *Notices of the AMS* 45.7 (1998), pp. 848–859.

Sitografia

- [35] Getty Images website. URL: <http://www.gettyimages.co.uk/>.
- [39] NextMind website. URL: <http://www.next-mind.com/>.
- [40] Stäubli website. URL: <http://www.staubli.com/>.
- [41] Microsoft website. URL: <http://www.microsoft.com/hololens/>.
- [42] Meta website. URL: <http://www.meta.com/>.
- [43] LG website. URL: <http://www.lg.com/>.
- [44] Unity website. URL: <http://unity.com/>.
- [45] Visual Studio website. URL: <http://visualstudio.microsoft.com/>.
- [46] Visual Studio Code online. URL: <http://vscode.dev/>.
- [47] MoveIt website. URL: <http://moveit.ros.org/>.
- [48] Arduino website. URL: <http://www.arduino.cc/>.
- [53] NASA website. URL: <http://humansystems.arc.nasa.gov/groups/tlx/>.
- [55] OpenCV website. URL: <http://store.opencv.ai/products/oak-d/>.
- [57] VIVE website. URL: <http://www.vive.com/us/accessory/tracker3/>.