

Politecnico di Milano

Polo regionale di Como

Corso di Laurea in Ingegneria Informatica

Anno accademico 2003/2004



BROWSING GESTURES: interfacciamento multimodale di un guanto virtuale per la navigazione di informazioni

Tesi di Laurea in Ingegneria Informatica di:
Faverio Jacopo Matr. 640109
Sangalli Daniele Matr. 635466

Relatore: prof. Thimoty Alessandro Maria Barbieri

Indice Generale:

Interfacciamento uomo macchina.....	8
1 .1 Interazione con le macchine	8
1.1.1 Cenni storici.....	8
1.1.2 Interazione uomo macchina.....	10
1.1.3 Le Interfacce.....	10
1.2 L'evoluzione delle interfacce.....	12
1.2.1 Human Computer Interaction	12
1.2.2 Le interfacce multimodali	14
1.3 Interazione tra macchina e linguaggio naturale.....	16
1.3.1 MMI EMMA W3C	16
1.3.2 Browser Helper Object, Interazione col il Browser di Internet Explorer.	19
1.4 Le periferiche d'interazione	20
1.4.1 Analisi / Come comunichiamo con i computer? Disabili e informatica - dispositivi di input.....	20
1.4.2 Dispositivi glove per l'interazione	25
Aspetti dell' implementazione.....	30
2.1 Perché usare il gesturing	30
2.2 La periferica adottata.....	31
Progettazione	33
3.1 Requisiti dell'architettura per il browsing gestures.....	33
3.1.1 Funzionamento generale.....	34
3.2 Progettazione dell'architettura generale.....	39

3.2.1 Scelte tecnologiche e architetture.....	40
3.3 Implementazione quanto p5	45
3.4 Rete neurale per il riconoscimento del gesture-path....	49
3.4.1 Introduzione e scelta delle Reti Neurali	49
3.5 Riconoscimento dei gesti	51
3.5.1 Sistema di gesturing	51
3.5.2 Algoritmo di smussamento e conversione dei dati in ingresso	54
3.5.3 Riconoscimento di forme con le reti neurali	56
3.6 La tecnologia dei Browser Helper Object.....	58
3.6.1 Introduzione	58
3.6.2 Implementazione.....	61
3.7 Ambiente e scopo del filtro	64
3.8 Implementazione Filtro.....	68
Testing e valutazione risultati.....	70
4.1 Analisi dei risultati	70
4.2 Caratteristiche generali del lavoro	71
4.3 Training	72
4.4 Sessione sperimentale	72
4.5 Test per il riconoscimento dei gesture	73
4.5.1 Prove effettuate dai trainer	73
4.5.2 Prove effettuate dagli altri utenti.....	74
Conclusioni e Sviluppi Futuri.....	79
5.1 Conclusioni	79
5.2 Sviluppi futuri	80

Indice delle Tabelle/Figure:

Fig 1. MMI FrameWork.....	18
Fig 2. Sostegno ergonomico per avambraccio	22
Fig 3. Tastiera ridotta	23
Fig 4. Schema tracciamento oculare.....	24
Fig 5. DataGlove.....	26
Fig 6/7. Prototipo di force-feedback DataGlove (DataGlove, 2004)	27
Fig 8. Guanto p5 della Essential Reality	29
Fig 9. Menu di configurazione delle azioni.....	37
Fig 10. Estratto del registro di sistema contenente la configurazione di un singolo utente	38
Fig 11. Pulsante per impostazione di default	39
Fig 12. Tastiera del guanto P5 e le relative funzioni associate.....	41
Fig 13. Schema dell'architettura utilizzata	42
Fig 14. Modello dati EMMA	43
Fig 15. Instance data EMMA.....	43
Fig 16. Fase di Learning della Rete Neurale	44
Fig 17. Riconoscimento dei tracciati.....	45
Fig 18. I sei gradi di libertà del guanto	46
Fig 19. Le reti neurali: il cervello umano	50
Fig 20. Schema generale del gesture recognition.....	52
Fig 21. Algoritmo di smussamento del path. Ricerca dei picchi	55
Fig 22. Rete neurale multilayer	57
Fig 23. Visualizzazione del CLSID del BHO tramite il Registry Editor	59

Fig 24. Come il browser inizializza il BHO. Il "BHO site" è l'interfaccia usata per inizializzare una connessione.....	60
Fig 25. Sequence Diagram di un applicazione senza filtro..	69
Fig 26. Sequenze Diagram con applicazione del filtro	69
Fig 27. Portatile sul quale sono state effettuate le prove ...	71
Fig 28. Tabella parametri valutativi	73
Fig 29. Momento durante i test	74
Fig 30. Legenda usata durante la fase di prova	75
Fig 31. Tabella riassuntiva dei dati raccolti durante le prove	76
Fig 32. Diagramma riassuntivo di valutazione del parametro R	77
Fig 33. Diagramma riassuntivo di valutazione del parametro U	78
Fig 34. Diagramma riassuntivo di valutazione del parametro N	78
Fig 35. Glove P5. Sistema con connessioni fisiche tra ricevitore e periferica	81
Fig 36. Superglove. Sistema con ricevitore ad ultrasuoni	81

Ringraziamenti

Giunti al termine del percorso universitario, vorremmo ringraziare le persone che ci sono state vicine in questi anni. Un sentito ringraziamento va al Prof. Barbieri, per il supporto tecnico ed operativo e per la capacità di stimolarci ed accompagnarci con entusiasmo durante lo sviluppo della tesi.

Ai nostri genitori che hanno condiviso con noi le ansie e le gioie in tutti questi anni di studio.

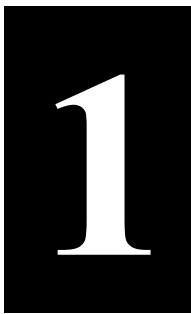
Un caloroso grazie a tutti gli amici con i quali abbiamo trascorso questi anni di università e a tutte le persone che ci sono state vicine.

Daniele, Jacopo

Sommario

Il continuo sviluppo e la crescente diffusione di dispositivi informatici ha sollecitato lo studio e la realizzazione di interfacce che tengano in primo piano le caratteristiche dell'utente come persona. E' sempre più avvertita l'esigenza di mettere a punto nuovi sistemi facili da apprendere e utilizzare, e che tengano conto delle abitudini cognitive e dei modelli mentali degli utenti ai quali il dispositivo è destinato, in vista di una sempre più efficace interazione tra utente e sistema informatico. L'area di studio specifica, oggetto del presente lavoro, è quella delle interfacce multimodali, il cui obiettivo è consentire una migliore comunicazione tra l'uomo e il computer. Tra tutte le soluzioni già realizzate e studiate si è scelto di approfondire, in questa sede, l'interfacciamento di un guanto virtuale per la ricerca, in rete, di informazioni attraverso un Web Browser. L'obiettivo era di individuare i minimi requisiti possibili per consentire l'utilizzo di tale tecnologia anche su applicazioni web già esistenti senza inibire l'utilizzo di dispositivi di input già supportati, quali mouse e tastiera. Ci siamo proposti, dunque, di realizzare un sistema estremamente semplice, che risponda a queste esigenze. Gli unici requisiti richiesti, lato client, sono la disponibilità di un normale Browser Internet Explorer (versione 4.0 o successive) nel quale il plugin debba essere installato usando un'architettura client/server e, lato server, un web server che supporti il paradigma dei filtri di elaborazione delle richieste da parte del client. Al termine della fase di implementazione dell'architettura abbiamo sperimentato il suo potenziale applicativo saggiandolo su un insieme di persone con differenti livelli di dimestichezza con i personal computer, per potere verificare la effettiva facilità di utilizzo.

Dall'analisi dei risultati ottenuti si conclude che l'obiettivo di realizzare un sistema di navigazione nella Rete, il più possibile intuitivo e di facile uso con il minimo costo di implementazione in applicazioni già esistenti può dirsi conseguito, con successo. In conclusione, il risultato della sperimentazione ha confermato l'utilità di una periferica, come il guanto virtuale, che estende il naturale uso della mano, anche nella costruzione di modalità di utilizzo della Rete fruibili attraverso applicazioni web già presenti.



Interfacciamento uomo macchina

1 .1 Interazione con le macchine

1.1.1 Cenni storici

Definizione e filosofia delle interfacce

I computer “ragionano” attraverso il linguaggio binario fatto di 0 e 1, linguaggio incomprensibile per quasi tutte le persone che ogni giorno riescono comunque ad utilizzare abilmente il proprio PC per le attività più disparate. Questo perché, per poter

consentire un uso di massa del computer senza dover essere programmatori, sono stati introdotti dei sistemi che consentono di *mediare* e *umanizzare* il dialogo tra utente e calcolatore: le **interfacce**.

Chiaramente non si parla di interfacce esclusivamente in ambito informatico. Il termine deriva dal latino: *inter facies, fra le facce*. E' un qualcosa che si inserisce fra entità, sistemi, cose o persone diverse tra loro mettendoli in grado di comunicare.

Qualsiasi strumento che ci permette di interagire con il mondo circostante nel modo più adatto alle nostre caratteristiche fisiche e sensoriali, può essere considerato un'interfaccia: ha un ruolo di intermediazione tra ambienti che utilizzano norme e codici differenti.

Restringendo il campo all'ambito informatico, l'**interfaccia uomo-computer** è tutto ciò che sta tra l'utente e l'elaboratore centrale e può essere di tipo *hardware* o *software*. Nel primo caso si tratta dei dispositivi fisici come la tastiera, la stampante o il mouse. Nel secondo invece ci si riferisce al modo in cui un programma si presenta e permette di essere utilizzato, ad esempio tramite la sua presentazione grafica, i menù, le icone ed i pulsanti. Nonostante entrambi svolgano un ruolo di mediazione, nell'hardware questa mediazione è prettamente di tipo fisico, nei software invece è ad alto contenuto simbolico.

1.1.2 Interazione uomo macchina

Cenni storici

Nel 1960 J. C. R. Licklider, psicologo sperimentale, pubblicò un saggio intitolato *“La simbiosi uomo-computer”*, che ha avuto moltissima influenza sugli studi successivi nel campo dell'informatica e della psicologia. Nel documento affermava che: *«Tra non molti anni la mente umana e i calcolatori saranno interconnessi molto strettamente, e questa alleanza uomo-macchina sarà in grado di pensare così come nessun essere umano ha mai fatto finora, elaborando dati con prestazioni che sono ancora irraggiungibili per le macchine con cui effettuiamo attualmente il trattamento delle informazioni.»*

Nei primi anni '60 gli studi sull'interfaccia utente seguirono due percorsi differenti, uno aveva come obiettivo il raggiungimento dell'*interattività* e l'altro la *ricchezza sensoriale*. Era emersa l'esigenza di estendere l'utilizzo del computer ad un utenza sempre più vasta, ma l'alto costo delle macchine era un grosso impedimento. L'interattività entrava così in gioco, facendo in modo che più utenti potessero condividere uno stesso computer.

1.1.3 Le Interfacce

Gli studi nel campo delle interfacce

Con l'apparizione della prima interfaccia grafica, è stata riservata un'attenzione sempre maggiore alla realizzazione di interfacce che tenessero in primo piano le caratteristiche

dell'utente come persona. Dagli anni '80 in poi, l'utente è stato man mano messo al centro del processo di progettazione, giungendo a quello che oggi è definito l'*User Centered Design*: il fruitore viene coinvolto sin dai primi passi del processo di sviluppo, in modo da realizzare sistemi facili da apprendere e utilizzare. Questo approccio tiene conto delle abitudini cognitive e dei modelli mentali degli utenti ai quali il prodotto è destinato.

Quasi tutte le GUI (Graphic User Interface) per personal computer, oggi sono basate sul principio della "WIMP" (windows, icons, mouse, pointer), in base al quale una serie di finestre rendono possibile l'utilizzo contemporaneo di diverse parti dello schermo ed il mouse consente una modalità di interazione non più solo alla tastiera.

Le case produttrici hanno inizialmente cercato di imporre propri standard nel settore delle interfacce, ma dato che, in tal modo, si correva il rischio di creare inutile confusione, le modifiche si sono successivamente concentrate solo su dettagli secondari, come le dimensioni, i colori, i caratteri o le forme degli angoli delle finestre, senza intervenire troppo sulle convenzioni di base.

Dalle ricerche sui possibili sviluppi delle GUI si sono sviluppati altri due filoni, uno relativo all'**interazione in linguaggio naturale** (Natural Language Processing o NLP), l'altro alla **manipolazione diretta** (Direct Manipulation, DM).

Gli studi sul NLP puntano alla creazione di un'interfaccia che metta in grado l'utente di comunicare col computer attraverso il linguaggio naturale, come si fa tra esseri umani. Non si tratta di

una sfida facile da vincere, considerando due caratteristiche fondamentali del linguaggio naturale: l'*ambiguità* e la *ridondanza*. Sappiamo che una parola, a seconda delle circostanze in cui viene utilizzata, può avere significati differenti. Per ora si è riusciti a realizzare interfacce in grado di comprendere un numero limitato di istruzioni (un esempio sono i telefoni cellulari, che identificano il nominativo da chiamare grazie al riconoscimento vocale del nome).

1.2 L'evoluzione delle interfacce

1.2.1 Human Computer Interaction

Sempre più discipline sono coinvolte nello studio dell'interazione uomo-macchina

La *Human Computer Interaction* (HCI), nata da poco più di dieci anni, si occupa dei problemi connessi alla progettazione di interfacce uomo-macchina, cercando di offrire utili strategie e suggerimenti per rendere possibile un'efficace interazione fra l'utente ed il computer. I ricercatori si sono resi conto presto che i problemi relativi all'organizzazione e alla gestione del lavoro, la salute, i fattori neuro fisiologici e i fattori ambientali possono influenzare l'interazione uomo-computer. Ad occuparsi di questi problemi sono deputate discipline quali l'ergonomia cognitiva, la psicologia, le scienze cognitive, la semiotica, la fisiologia, l'informatica e l'industrial design. L'HCI costituisce un ambito

interdisciplinare di ricerca che si pone all'intersezione di queste discipline. Le ricerche inizialmente presero il nome di *man-machine-interaction* e successivamente vennero indicate col termine, attualmente corrente, di *human-computer interaction*. I computer sono diventati strumenti utilizzati non solo da esperti, ma da una vasta gamma di utenti per scopi lavorativi, ludici ed educativi. Questo fenomeno ha indotto i progettisti a studiare sistemi computerizzati che si adattino ai diversi tipi di bisogni che un utente può avere. Perché un computer diventi un prodotto di consumo dovrebbe essere ben progettato. Ciò non significa giungere fino ad una progettazione personalizzata per il singolo utente, ma andare incontro ai bisogni e alle capacità di classi di fruitori.

Gli obiettivi della Human Computer Interaction sono quelli di progettare sistemi computerizzati utili, sicuri, usabili e funzionali. Nonostante non esista una teoria generale e unificata della HCI, c'è comunque un principio fondamentale che la riguarda: le persone utilizzano il computer per svolgere compiti (il termine deve essere interpretato in modo generico e non strettamente connesso al lavoro). Quindi nel processo di progettazione sono coinvolti principalmente tre "elementi": l'uomo, il computer e l'attività da svolgere. Il sistema predisposto deve sostenere l'utente nello svolgimento dei suoi compiti e questo porta a galla un quarto elemento: l'usabilità. Se il sistema forza l'utente e lo mette in difficoltà nello svolgimento del suo compito significa che non ha un buon grado di usabilità.

1.2.2 Le interfacce multimodali

C'è chi vede nel futuro del computer non più il ruolo di un semplice strumento, ma piuttosto una sorta di assistente in grado di svolgere una serie infinita di attività attraverso l'uso di particolari dispositivi

Ma l'evoluzione che si vuole raggiungere non riguarda esclusivamente la capacità di un computer di svolgere determinate operazioni, ma anche il tipo di interazione che coinvolge l'utente e il sistema. A tale scopo è sorta una nuova area di studio della Human Computer Interaction: quella sulle **interfacce multimodali**, il cui obiettivo è di consentire una migliore comunicazione tra l'uomo e il computer attraverso una sorta di antropomorfizzazione di quest'ultimo.

Si parla di **multimodalità** quando un qualsiasi tipo di interazione coinvolge più di un canale percettivo (o input di comunicazione). L'esempio più vistoso è senz'altro la comunicazione umana. Durante una conversazione vengono stimolati più sensi contemporaneamente: la vista, l'udito, il tatto ed anche l'olfatto. (Di passaggio ricordiamo che occorre distinguere *multimodalità* e *multimedialità* . La prima è usata specificatamente per riferirsi alle modalità di input dell'interazione, mentre la multimedialità sottolinea la presenza contemporanea di più meccanismi di output: audio, video etc.).

Rimanendo nell'ambito della Human Computer Interaction, una definizione di multimodalità è la seguente: *la comunicazione con sistemi di computer attraverso le modalità di input percettivi*

comunemente utilizzate dall'uomo per interagire con il mondo. Quindi ci riferiamo a una comunicazione non più ristretta all'uso della tastiera o del mouse, ma che coinvolge anche la parola, i gesti o la scrittura manuale (in questo caso non si intende la videoscrittura, ovvero mediata da una tastiera, ma piuttosto la tradizionale scrittura a mano).

La comunicazione monomediale è generalmente imposta agli utenti dalle restrizioni tecnologiche. Si pensi al telefono ad esempio: non tutti i generi di informazioni sono facili da trasmettere e assimilare quando la parola è l'unica modalità a disposizione. Ad un livello tecnologico più sofisticato anche l'interfaccia grafica di Windows, tanto comune e impiegata dalla maggior parte degli utilizzatori di PC, subisce le restrizioni imposte dal fatto che solitamente lo schermo, la tastiera e il mouse siano gli unici dispositivi disponibili.

L'utilizzo del computer necessita la conoscenza e la capacità di utilizzare una serie di strumenti sia hardware che software. Gli utenti più esperti spesso si stupiscono davanti ai neofiti che si trovano impacciati nel manovrare in modo coordinato uno strumento tanto comune come il mouse, o che non hanno idea di come si raggiungano e ottengano le informazioni sulla disponibilità di un volo senza dover telefonare ad un'agenzia di viaggi.

Chi utilizza un computer da tempo quasi non si rende più conto che lo fa attraverso un preciso standard di comunicazione con la macchina, standard che è tipicamente costituito sia da comandi inseriti in un linguaggio formale profondamente strutturato, sia

da azioni tipo "seleziona e rilascia" gestite dal mouse. Attualmente queste modalità sono molto evolute e continuano a perfezionarsi, ma certo non raggiungono i livelli di complessità del linguaggio naturale.

Se si riuscisse davvero a superare i limiti attualmente imposti dalla tecnologia, molti servizi diverrebbero molto più semplici da fruire e di conseguenza molto più attraenti per un pubblico più ampio. Ad ottenere vantaggi da un miglioramento della qualità dell'interazione fra uomo e macchina, non sarebbero soltanto le persone poco esperte che riuscirebbero ad utilizzare il computer con più facilità. Si pensi alle persone disabili, che pur essendo in grado di apprendere gli standard attuali, hanno problemi di tipo fisico che impediscono loro di interagire correttamente con il computer. L'utilizzo del linguaggio naturale potrebbe ridurre questi ostacoli.

1.3 Interazione tra macchina e linguaggio naturale

1.3.1 MMI EMMA W3C

Lo standard attuale per il multimodale

Per interagire con il linguaggio naturale, il consorzio W3C ha definito uno standard per descrivere e immagazzinare le informazioni multimodali provenienti dal mondo esterno: l'"Extensible MultiModal Annotation markup language" (EMMA).

La lingua estendibile di annotazione di MultiModal è una disposizione di scambio di dati per i sistemi di amministrazione di interazione in cui EMMA rappresenta l'input dell'utente. Il discorso ed i dispositivi di riconoscimento di calligrafia, i motori di linguaggio naturale, gli interpreti di mezzo e i componenti multimodal di integrazione generano il markup di EMMA.

Il W3C definisce in oltre un framework per connettere i dati EMMA a sistemi di interazione multimodale. Il framework proposto identifica la maggiorparte dei componenti per ogni sistema multimodale.

Ogni componente rappresenta un set di funzioni correlate. Il framework identifica i tag usati per descrivere le informazioni richieste dai componenti e il flusso di informazioni tra i vari dati provenienti dai diversi input.

L'MMI (Multimodal Interaction Framework) di W3C descrive i modi di input e gli output più usati che possono essere estesi per periferiche future.

Il framework MMI non è un'architettura bensì un livello di astrazione al di sotto di essa.

Un architettura definisce come i componenti sono allocati in periferiche hardware e il sistema di comunicazione che permette a più periferiche di comunicare tra di loro. Il Framework MMI è pensato come base di sviluppo in termini di markup, scripting e styling.

Qui sotto viene esplicitata l'interazione tra i componenti chiave del framework MMI di W3C:

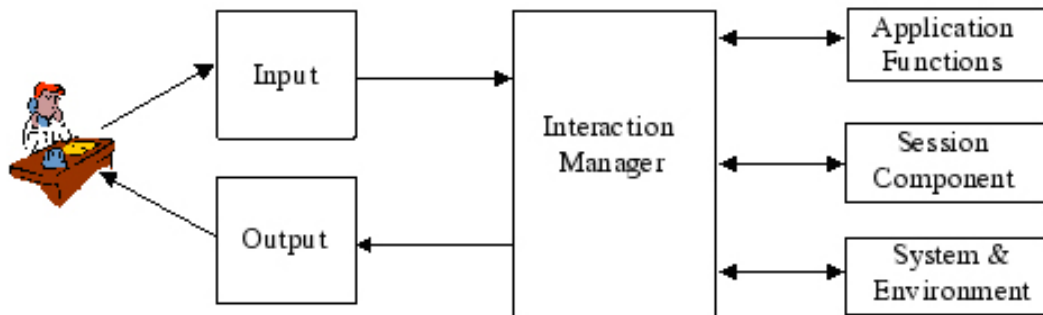


Fig 1. MMI FrameWork

Human user — Uno User che immette informazioni di input nel sistema e si mette in attesa del output del sistema. Parleremo di User in termini di Human User, sebbene esso possa essere sostituito da una macchina per applicazioni di regression testing per verificare la stabilità del sistema in caso di cambio di input.

Input — Un implementazione multimodale userà svariate tipologie di input come il parlato, la scrittura a mano, la scrittura per mezzo di una tastiera e altre.

Output — Un implementazione multimodale userà svariate tipologie di output .Per esempio potremmo avere il parlato, un testo, file audio e grafici ecc..

Interaction manager — E' un componente logico che coordina le informazioni e controlla il flusso d'esecuzione delle modalità

d'interfacciamento dei componenti di input e output. L'interaction manager mantiene lo stato d'interazione e il contesto dell'applicativo, risponde agli input per mezzo d'interfacce e cambia lo stato del sistema. L'interaction manager controlla questi cambiamenti e coordina gli input e gli output tra le interfacce. In determinate architetture l'interaction manager può essere implementato come singolo componente, in altre come la somma di piccoli componenti.

Session component — Il Session component fornisce un interfacciamento all'interaction manager come supporto al controllo dello stato, e per sessioni persistenti o temporanee di applicazioni multimodali.

System and Environment component — Questo componente abilita l'interaction manager per scovare cambiamenti e per rispondere ai devices, secondo le preferenze dell'utente e le condizioni ambientali.

1.3.2 Browser Helper Object, Interazione col il Browser di Internet Explorer.

Estendere le capacità del browser Internet Explorer

Per interagire attivamente coi dati provenienti da una navigazione nella rete si hanno due possibilità.

La prima è sviluppare un browser apposito con funzioni avanzate, il che richiederebbe anni di sviluppo e si andrebbe

fuori dagli standard imposti, invece la seconda possibilità verte sullo sviluppo di un BHO.

BHO: dall'inglese Browser Helper Object, assistente del browser. Un BHO e' un piccolo programma che estende le funzionalità di Microsoft Internet Explorer (IE). Esempi di BHO sono le barre aggiuntive in IE, ma anche funzioni non visibili.

Il browser come tutte le applicazioni basate su Win32 si carica in un certo spazio di memoria. Col BHO si possono sviluppare dei plugin appositi per il browser in COM(Components Object Model) che si caricheranno in memoria all' avvio. Questi componenti si allocano nella stessa contesto di memoria riservata al browser possono compiere qualsiasi azione nella finestra corrente. Per esempio può ricevere i tipici eventi che si manifestano nel browser, come vai Avanti, Stop, Aggiorna, ecc.. e accedere ai sotto menu e toolbar installate. Praticamente lavorano come una spia che vive in simbiosi con l'istanza del browser.

1.4 Le periferiche d'interazione

1.4.1 Analisi / Come comunichiamo con i computer? Disabili e informatica - dispositivi di input

I computer offrono nuove possibilità alle persone disabili, in vista di una migliore integrazione sociale e professionale: nello studio, nel lavoro e anche nella vita relazionale

Un settore che ha ottenuto ottimi risultati nel campo delle interfacce è stato quello rivolto allo sviluppo di **ausili per persone disabili**. Inizialmente le persone con ridotta capacità motoria o con problemi visivi, incontravano grosse difficoltà nell'utilizzo del computer. Da diversi anni però il problema dell'accessibilità dei disabili alle tecnologie informatiche ha interessato numerosi ricercatori e produttori, sia di hardware che di software. Così oggi il mercato offre una vasta gamma di soluzioni ai problemi dovuti a quasi ogni tipo di handicap.

Dispositivi di input

Nel caso dei disabili motori, il principale problema lo si riscontra nell'immissione dei dati attraverso i dispositivi standard, che sono il mouse e la tastiera.

Ci sono una serie di accessori che rendono le tastiere più accessibili alle persone disabili, consentendo di manovrarle non solo con le mani, ma anche con i piedi o con particolari puntatori fissati su caschetti o da tenere in bocca. Inoltre l'utilizzo di specifici sostegni ergonomici per i polsi o per gli avambracci, limitano lo sforzo nel raggiungimento di tasti e favoriscono il movimento su un piano orizzontale.



Fig 2. Sostegno ergonomico per avambraccio

Esistono anche delle tastiere alternative a quelle standard. Le principali categorie sono:

tastiere espanse: sono di dimensioni molto più grandi rispetto a quelle comuni.

tastiere a membrana: sono dispositivi di input programmabili sulla quale si dispongono membrane (mappature di tastiera) semplificate e personalizzabili, premendo i simboli rappresentati si attivano di volta in volta le funzioni ad essi associati;

tastiere ridotte: al contrario delle espanse, si tratta di dispositivi con tasti minuscoli e ravvicinati. Sono adatte a persone che non hanno un ampio raggio d'azione con gli arti superiori, ma mantengono una buona motricità fine delle dita (è necessario compiere piccoli movimenti per attivare i comandi).



Fig 3. Tastiera ridotta

Ulteriore soluzione sono gli **emulatori di tastiera**. Sullo schermo del computer appare l'immagine di una tastiera, che può essere attivata con il mouse o con dispositivi di puntamento alternativi.

Il dispositivo di puntamento più diffuso è senz'altro il mouse. Anche in questo caso per coloro che hanno difficoltà ad utilizzarlo, esistono in commercio numerosissime interfacce con caratteristiche particolari.

Esistono degli **emulatori del mouse** come quelli azionabili con la bocca sono dispositivi di puntamento in grado di simulare il funzionamento di un mouse a due bottoni. Sono controllabili tramite movimenti della bocca o del mento. L'equivalente del "click" del mouse tradizionale, è possibile tramite i sensori sensibili al soffio o al morso. I sistemi più sofisticati consentono di spostare il puntatore sullo schermo con il movimento oculare, questo grazie ad un led posizionato sotto il monitor e collegato ad una videocamera che percepisce i movimenti oculari e li trasferisce al computer.

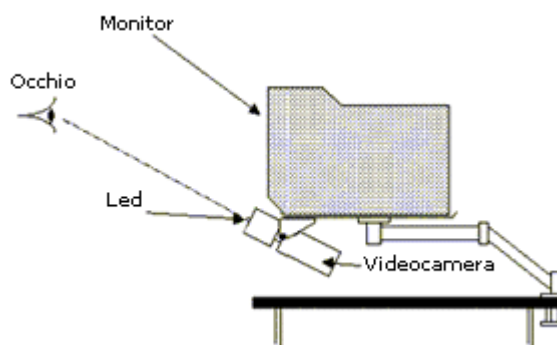


Fig 4. Schema tracciamento oculare

Sistema di tracciamento oculare

Ulteriore possibilità per i disabili motori che abbiano però conservato una buona padronanza del linguaggio, sono i software di riconoscimento vocale. Consentono di pronunciare le parole e visualizzarle poi su schermo invece che doverle digitare su tastiera e anche di poter effettuare le operazioni svolte solitamente con l'ausilio del mouse. Il livello raggiunto dalla tecnologia di riconoscimento, permette di avere vocabolari molto estesi ed un alto livello di precisione di riconoscimento.

Per i non vedenti invece sono a disposizione le **tastiere Braille**: sui tasti della tastiera sono stampati in rilievo le lettere in caratteri Braille.

1.4.2 Dispositivi glove per l'interazione

Periferiche atte per l'interazione gestuale

Diversamente dai tradizionali metodi di interazione che usano strumenti pilotati dalla mano (quali il mouse), l'utilizzo diretto delle gestualità della mano senza nessun trasduttore intermedio fa assumere alla comunicazione uomo-macchina un aspetto pressoché naturale, ed inoltre sfruttando appieno le capacità gestuali dell'utente può ridurre il tempo necessario per l'apprendimento.

Alcuni studi effettuati circa l'utilizzo delle gestualità nella comunicazione fra le persone hanno rilevato che il 70% - 80% di ciò che viene detto durante un dialogo può essere espresso esclusivamente mediante gestualità che coinvolgano tutte le parti del corpo. Da ciò l'importanza attribuita all'utilizzo di gesti della mano nella interazione uomo-macchina. I sistemi che sfruttano questa nuova modalità di interazione possono fornire un prezioso strumento per le persone dotate di capacità motorie limitate, permettendo di utilizzare solo alcuni limitati movimenti per interagire con la macchina.

Purtroppo questo particolare tipo di guanti in genere è poco confortevole e molto fastidioso. Essi limitano i movimenti perché spesso sono fisicamente connessi con il computer, e in genere non sono in grado di fornire la posizione relativa della mano nello spazio, a meno che non siano dotati di sofisticati (e costosi) sensori per il rilevamento della posizione.

Tuttavia, il dispositivo che meglio si presta al riconoscimento della gestualità umana (gesture recognition) è il guanto (vedi Figura 12). Con esso è possibile definire una ragguardevole combinazione di comandi utili allo svolgimento di numerose attività di interazione con ambienti ed oggetti virtuali.



Fig 5. DataGlove

Già a partire dalla fine degli anni '80 vennero descritti da Sturman (1989) e Zimmerman (1987) taluni guanti che, una volta indossati, potevano rilevare posizione della mano e flessione delle dita. Sebbene questa periferica si sia evoluta nel corso degli anni, essa è ancora oggi oggetto di ricerca, utilizza tipicamente tracciatori di tipo elettromagnetico a sei DOF e sensori per il rilevamento della flessione delle dita.

Nel corso degli anni, si sono sviluppati altri tipi di guanti, più sofisticati, in grado di simulare la forza resistente (force feedback) esercitata dagli oggetti virtuali sulle articolazioni della mano durante il contatto; essi sono però costosi e difficili da calibrare, particolarmente macchinosi e invasivi .

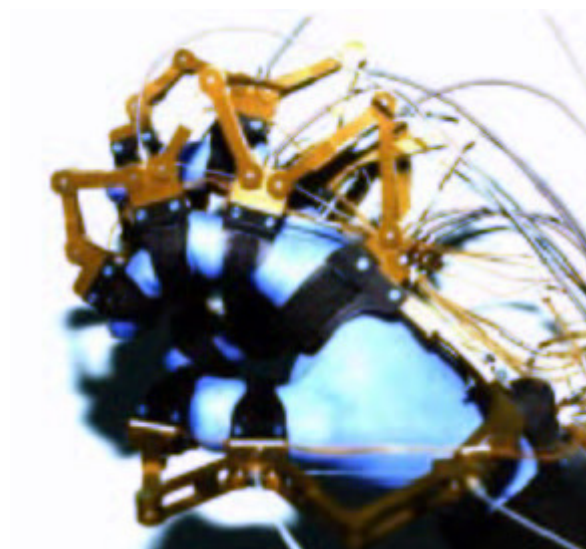
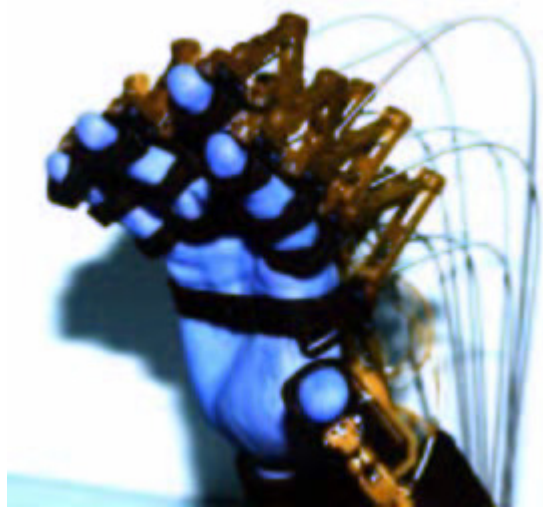


Fig 6/7. Prototipo di force-feedback DataGlove (DataGlove, 2004)

Recenti studi in questo campo hanno permesso di sviluppare un particolare tipo di guanto detto "Pinch Glove" in grado di intercettare il contatto tra le punte delle dita, permettendo in questo modo di evitare l'utilizzo dei costosi sensori atti al rilevamento della flessione delle dita, dando all'utente la possibilità di afferrare oggetti virtuali ed un modo alternativo di utilizzo di una serie di comandi e regole euristiche. L'utilizzo di questo particolare tipo di guanto è facilitato dal fatto che la periferica non necessita, per un corretto funzionamento, di alcuna calibrazione.

La periferica da noi scelta è il P5 della "essential reality". Il P5 è una periferica a 6DOF (Degree of Freedom) che sfrutta la tecnologia di tracking ottico ad infrarossi per il rilevamento della posizione e dell'orientamento nello spazio.



Fig 8. Guanto p5 della Essential Reality

Il dispositivo si compone di un ricevitore a raggi infrarossi e di un guanto, ad esso collegato tramite un cavo, provvisto di una serie di marker (identificatori di posizione) posti sugli angoli della struttura che costituisce il polso del guanto, e di cinque sensori posti sulle dita mediante i quali è possibile rilevare l'angolo di flessione delle articolazioni .



Aspetti dell' implementazione

2.1 Perché usare il gesturing

La potenza del linguaggio gestuale

Mentre i computer diventano sempre più potenti, i modi di interazione, come la tastiera, il mouse limitano tuttora l'interazione uomo-computer a un più basso livello di manipolazione.

Numerosi studi ed esperimenti riportati in letteratura (Bolt et al., 1992) hanno confermato la superiorità dell'utilizzo della mano, in particolare di entrambe le mani, nel descrivere determinate

azioni gestuali se confrontata a metodi di interazione indiretti che prevedono l'utilizzo di strumenti interattivi.

La mano offre una maggior possibilità di espressione gestuale ed anche un più elevato livello di interazione, dal momento che, indossato il guanto, l'utente può interagire direttamente con il proprio corpo nello spazio tridimensionale.

L'utilizzo della penna invece, presuppone un tipo di interazione caratterizzata da un maggiore livello di astrazione mentale perchè il processo di gesture recognition (riconoscimento del gesto) avviene tramite penna in maniera indiretta.

2.2 La periferica adottata

Le caratteristiche del glove P5

Il dispositivo scelto, il P5 Glove prodotto da Essential Reality (Essential Reality, 2004) non supporta force feedback e presenta limitate caratteristiche se confrontato con altri dispositivi più costosi. Esso è caratterizzato da un modesto livello di precisione nella rilevazione dei dati e per di più sono parzialmente limitate alcune funzionalità motorie dell'utente. Quest'ultimo, deve ricordarsi di non uscire dal campo d'azione delimitato dal ricevitore a raggi infrarossi, per evitare discontinuità nel flusso dei dati d'ingresso.

Tuttavia questa periferica, dalle modeste funzionalità, rappresenta un ottimo compromesso tra qualità di interazione e costo d'acquisto, peraltro inferiore ai cento euro.

L'implementazione di una struttura software capace di garantire una valida interfaccia tra la periferica P5 e il browser, rappresenta il punto di partenza per lo sviluppo di un set di comandi gestuali utili a semplificare l'interazione con il web browser di Internet Explorer.

Nei paragrafi seguenti tale realizzazione sarà discussa in dettaglio rivolgendo particolare attenzione alla struttura dati e ai diversi aspetti di ingegnerizzazione affrontati.



Progettazione

3.1 Requisiti dell'architettura per il browsing gestures

In questo paragrafo vengono descritte le funzionalità che sono state previste in fase di analisi e specifica dei requisiti, nell'ambito del progetto di browsing gestures.

3.1.1 Funzionamento generale

Secondo i nostri requisiti, il plugin di gesturing attraverso il guanto virtuale doveva proporsi come uno strumento di facile utilizzo ed installazione, sia dal punto di vista del Browser Internet, sia delle applicazioni web. Il plugin avrebbe reso possibile la multimodalità anche su web application esistenti, apportando il minor numero di variazioni. A questo scopo avrebbe dovuto utilizzare componenti il più diffusi possibile per interfacciare il guanto P5 con il Browser Internet Explorer e poter dialogare con l'applicazione web attraverso un Framework MMI.

In un flusso di lavoro di creazione di un'applicazione Web multimodale, avrebbe dovuto essere possibile lo svolgimento delle seguenti fasi:

- Individuazione dell'insieme delle azioni.
 - In questa fase si individuano un insieme di azioni necessarie al fine di far cooperare il Web Browser con il Multimodal Interaction Framework. Le azioni sono selezionate da un vocabolario predefinito, presente all'interno del plugin, scegliendo quali di esse debbano reagire all'interazione dell'utente.
- Mappatura dei cammini navigazionali in una base di dati.
 - In seguito alla definizione dell'insieme delle azioni, si inseriscono i dati relativi ai cammini navigazionali; tali dati sono necessari per definire in che modo

l'applicazione web debba reagire all'interazione con l'utente e di conseguenza rendere possibile la navigazione tramite il guanto virtuale.

- Associazione tra azioni e cammini navigazionali
 - L'ultima fase consta nell'associazione tra l'insieme delle azioni e i cammini navigazionali, ottenendo come risultante una mappa degli effetti delle azioni su un'applicazione web. Quest'ultima operazione è necessaria in quanto fornisce all'applicazione web, soggetta ad interazione da parte dell'utente tramite guanto virtuale, le informazioni necessarie a dirigere quest'ultimo verso una determinata risorsa (ad es. una pagina html, jsp, o una servlet) in funzione della combinazione tra azione svolta e pagina visibile nel suo Web Browser in quel determinato momento.

Accedendo ad un normale URL all'interno della rete, l'utente Internet deve essere in grado d'interagire con l'applicazione web attraverso la medesima pagina Internet, indistintamente con l'ausilio del guanto virtuale e/o altre periferiche di input (ad es. tastiera, mouse).

La navigazione con l'ausilio del guanto si svolge secondo le modalità previste dal programmatore dell'applicazione web, il quale avrà creato una mappa efficiente e completa delle interazioni a disposizione dell'utente Internet.

La navigazione deve avvenire all'interno di un Browser Internet Explorer, versione 4.0 o successive, nel quale è stato precedentemente installato l'apposito plugin per il supporto del

quanto virtuale P5. Una volta installato il plugin, all'utente devono risultare trasparenti tutte le fasi di caricamento e trasmissione dei dati provenienti dal guanto verso l'applicazione web, i quali devono avvenire in modo automatico.

In ogni momento l'utente deve essere in grado di disabilitare l'utilizzo del plugin e conseguentemente del guanto virtuale. Ogni istanza del Web Browser IE, inoltre, deve permettere l'utilizzo del guanto in modo indipendente dalle altre; a titolo di esempio, supponendo di avere creato due istanze del Web Browser nel medesimo momento, il plugin può essere contemporaneamente abilitato in un'istanza e disabilitato nell'altra.

L'utente può configurare il vocabolario delle azioni che può compiere con l'utilizzo del guanto virtuale, associando ad ogni azione una delle due tipologie di reazioni disponibili:

- interazione su Web Browser Internet Explorer

Viene eseguito un comando sull'istanza di Internet Explorer correntemente attiva. Tale comando può essere, ad esempio, la massimizzazione della finestra. I comandi disponibili sono raccolti in un elenco e possono essere selezionati durante la fase di associazione Azione->Comando.

- interazione su Applicazione Web

Viene eseguito, in modo totalmente trasparente all'utente, l'invio di informazioni concernenti l'azione

appena svolta con l'utilizzo del guanto virtuale. L'invio di tali dati, verso il web server, è attuato utilizzando il metodo POST e rappresentando i dati secondo le specifiche XML per EMMA (Extensible MultiModal Annotation markup language).

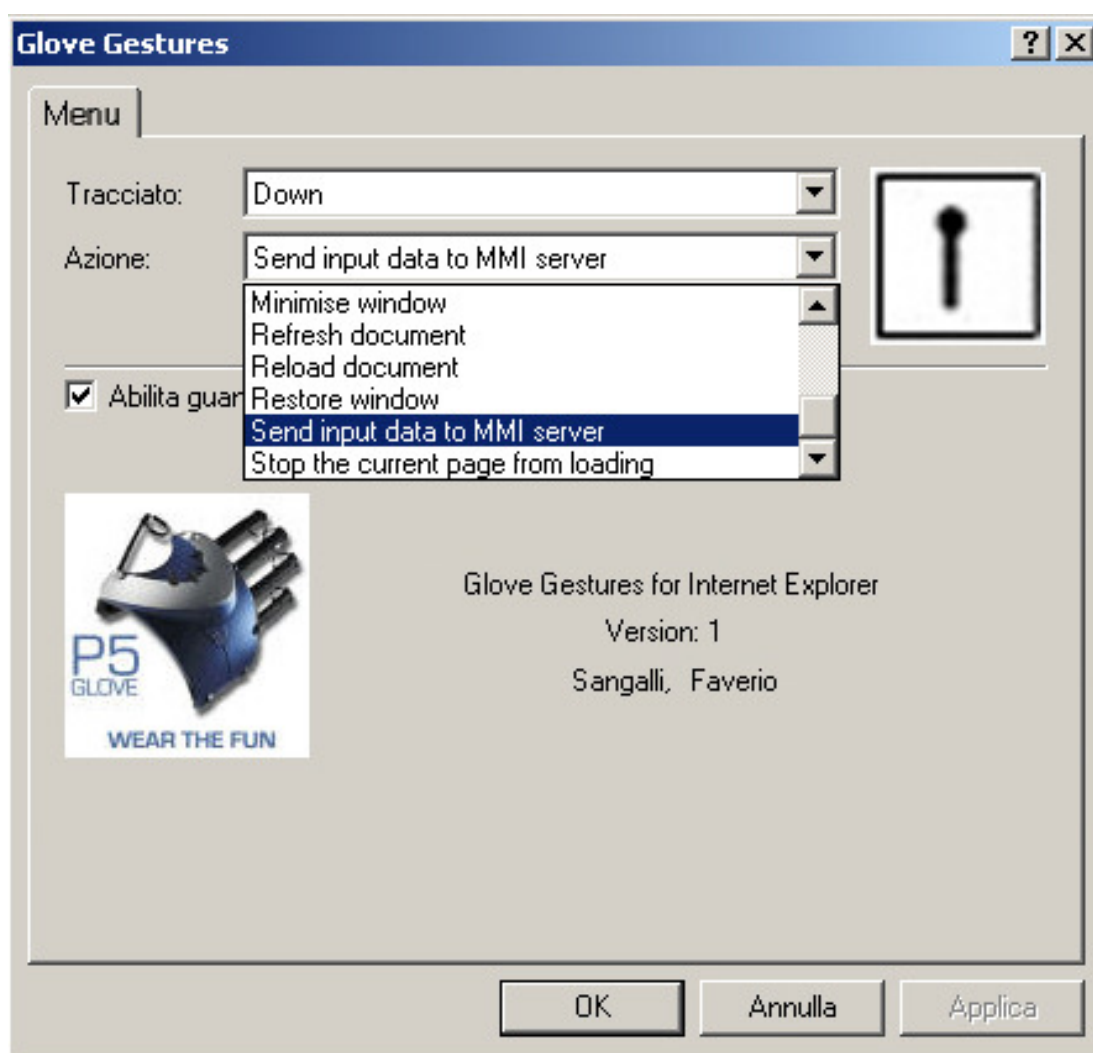


Fig 9. Menu di configurazione delle azioni

La configurazione del vocabolario viene salvata all'interno del registro di sistema di Windows.

Il registro di sistema è una sorta di archivio nel quale vengono, di volta in volta, memorizzate ed aggiornate le informazioni del software installato dall'utente, delle periferiche hardware, nonché le "personalizzazioni" che l'utente del PC compie sul proprio sistema.

Le impostazioni di un determinato utente, che sta utilizzando Windows, vengono salvate in modo da permettere ad altri di utilizzare la stessa macchina, ma configurazioni di vocabolario differenti.

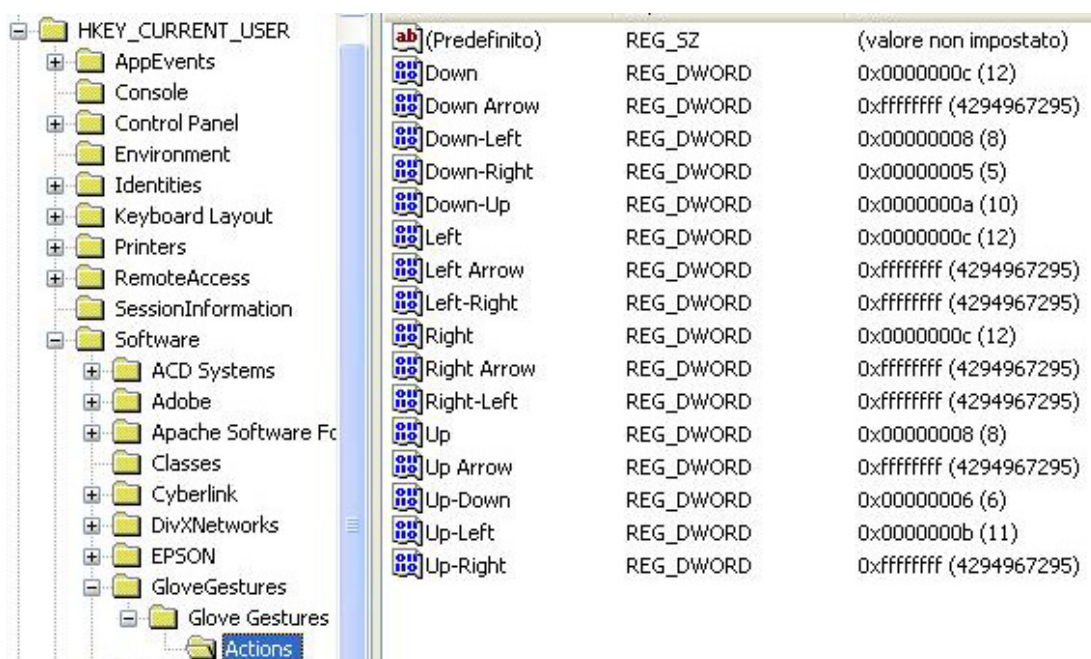


Fig 10. Estratto del registro di sistema contenente la configurazione di un singolo utente

Deve tuttavia sussistere la possibilità di riportare il vocabolario ad uno stato iniziale, dichiarato all'interno del plugin, di *default*.

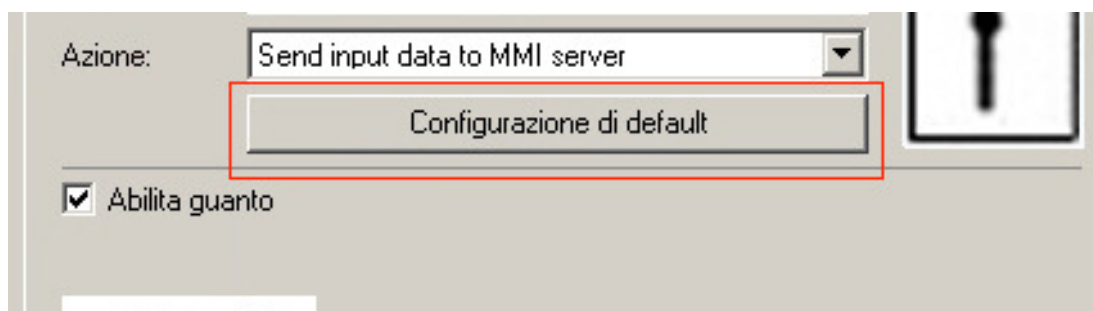


Fig 11. Pulsante per impostazione di default

Nel rispetto del paradigma di interazione multimodale l'utente può navigare attraverso le pagine del sito Web mediante l'utilizzo del guanto virtuale, può eseguire l'inserimento dei dati nelle varie *web form*, grazie all'utilizzo della tastiera, ed inviare tali dati interagendo con il mouse sugli appositi pulsanti.

3.2 Progettazione dell'architettura generale

Il seguente paragrafo dettaglia le scelte progettuali intraprese per la realizzazione dei requisiti specificati. Si è scelto in particolare di mantenere il progetto il più semplice e lineare possibile, scegliendo le soluzioni più efficaci, sfruttando al meglio le tecnologie disponibili, a formare una soluzione progettuale e implementativa ottimale.

3.2.1 Scelte tecnologiche e architetture

L'obiettivo principale era di ottenere un'architettura il più possibile snella e di semplice implementazione, per rendere facile l'installazione e per arrivare rapidamente alla fase di sviluppo.

Per l'implementazione del framework MMI si è scelto di utilizzare il Web Server TomCat, sia in funzione della sua facilità di sviluppo sia per la sua compatibilità con il mondo Java. Dopo una lunga analisi e un'attenta ricerca si è scelto di utilizzare i filtri. Tale funzionalità ha reso possibile l'implementazione di un preprocessore delle richieste entranti nell'applicazione Web, senza dover apportare modifiche all'applicazione già esistente sulla quale verrà installato. Questa funzionalità è presente anche sul Web server di IIS di Microsoft e prende il nome di ISAPI.

Per ottenere una perfetta integrazione con il Web browser Internet Explorer si è scelto di realizzare il Plugin utilizzando le specifiche dei Browser Helper Object, in seguito descritte più nel dettaglio, ed il linguaggio VC++. Grazie all'utilizzo di tale tecnologia è stato possibile interagire, in modo quasi totale con l'istanza di Internet Explorer, ed utilizzare le sue interfacce per pilotare il browser, e quindi rendere possibili azioni come il POST verso il Web Server.

Il passo successivo è stato dettato dalla necessità di eseguire ciclicamente una lettura dei dati provenienti dal guanto. Si è dunque scelto di agganciarsi al thread di ogni istanza di Internet Explorer dal quale il Plugin verrà attivato. Questa soluzione,

scelta dopo una lunga analisi, presenta l'interessante vantaggio di poter attivare o disattivare il plugin all'interno del browser in modo indipendente per ogni istanza.

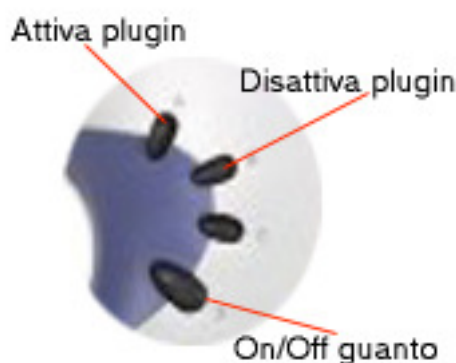


Fig 12. Tastiera del guanto P5 e le relative funzioni associate

Il grande successo di Internet è senza dubbio determinato dal fatto che il protocollo TCP/IP è in grado di essere utilizzato con tecnologie trasmissive e piattaforme software anche molto diverse tra loro, fungendo da "collante universale" per l'intera rete. Basandoci su questi presupposti abbiamo deciso di sviluppare il plugin lato client in modo da utilizzare un semplice comando di POST per l'invio dei dati al Server, inoltre, il filtro Server è stato sviluppato con strumenti e tecnologia Java, in modo da evidenziare ancor più il fattore di compatibilità e la facilità con la quale sia possibile implementare tale applicativo su piattaforme non Microsoft.

L'architettura generale prevede dunque la realizzazione di un plugin lato Client e di un filtro lato Server.

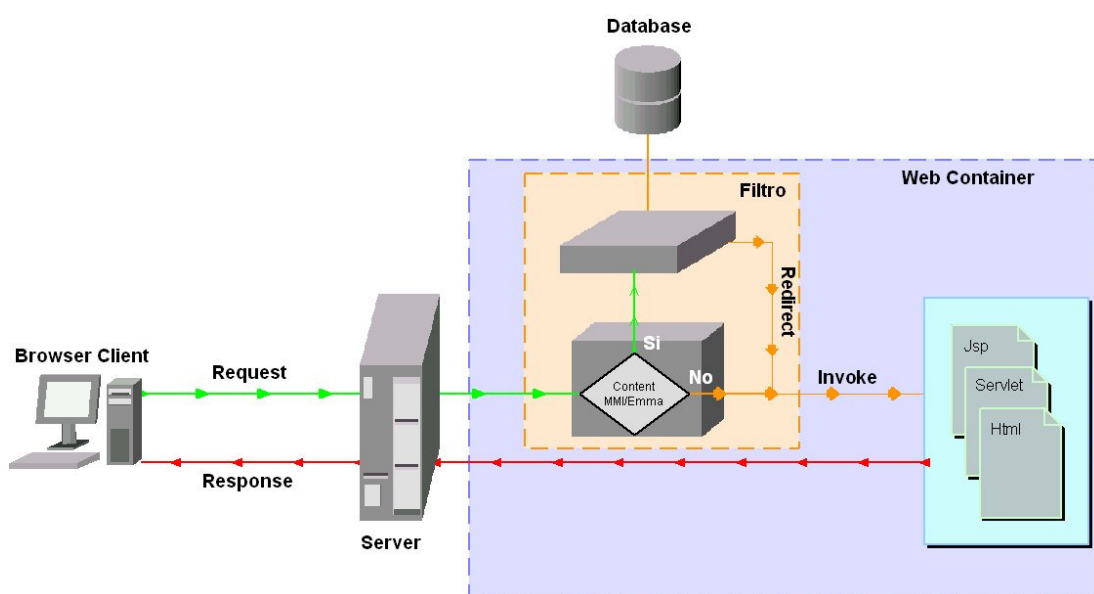


Fig 13. Schema dell'architettura utilizzata

L'altro passo importante e fondamentale per l'implementazione del progetto è stato dettato dalla scelta degli standard. Inizialmente si era pensato di creare uno standard proprietario, ma dopo un'attenta valutazione si è ritenuto opportuno utilizzare le proposte di standard del consorzio W3C per l'implementazione del Framework multimodale MMI e Emma per l'interpretazione della semantica del dispositivo di input.

Una tipica descrizione di EMMA consiste di due tipi di informazione utili nel descrivere i dati raccolti dai dispositivi.

- Data model - uno schema per descrivere i nomi e le strutture dei dati.

```
<emma:interpretation>
  <command_name>
    <parameter>value</parameter>
  </command_name>
</emma:interpretation>
```

Fig 14. Modello dati EMMA

- Instance data – le informazioni raccolte dai vari dispositivi di input. Nel nostro caso vengono rappresentati i dati del quanto.

```
<emma:interpretation>
  <command>
    <glove_path>
      <action>right</action>
    </glove_path>
  </command>
</emma:interpretation>
```

Fig 15. Instance data EMMA

Un ultimo, ma non meno importante aspetto dell'implementazione è stato la scelta di inserire una rete

neurale precedentemente addestrata attraverso un software di terze parti.

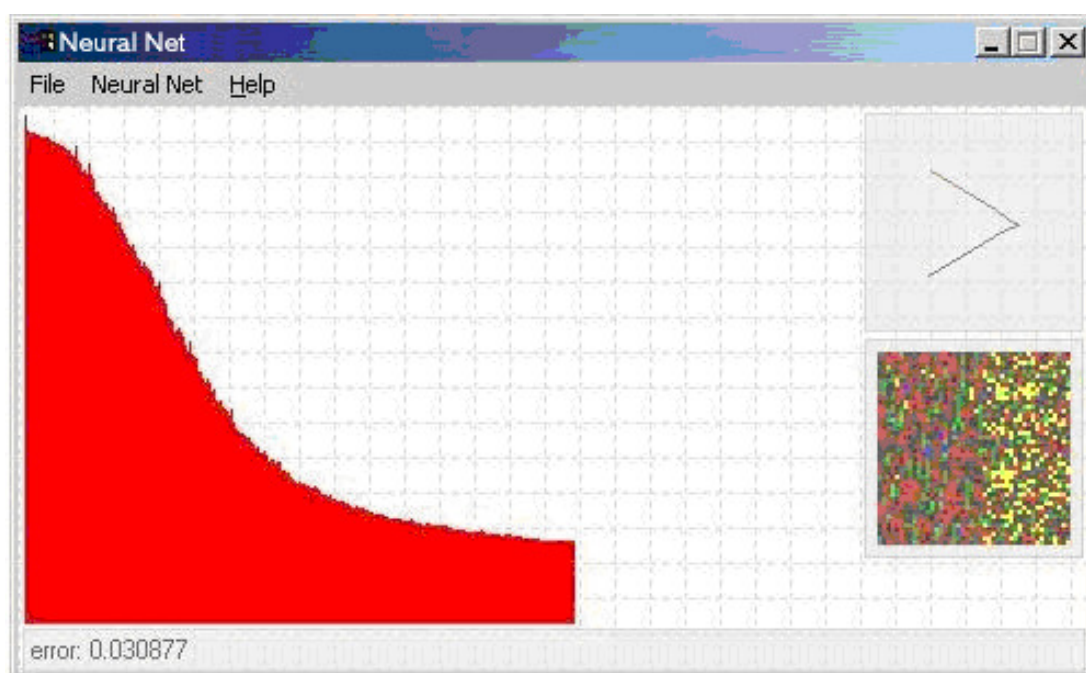


Fig 16. Fase di Learning della Rete Neurale

Tale software, dopo aver eseguito l'addestramento, effettua il salvataggio dei pesi della rete in un file. Grazie a tale file e all'utilizzo della documentazione fornita con il software, si è ottenuta un'implementazione in linguaggio VC++ di una rete neurale multistrato in grado di ottenere, data in ingresso una serie di punti, un unico elemento rappresentante l'azione svolta, tramite il guanto virtuale, selezionando tale elemento da un insieme di tracciati.

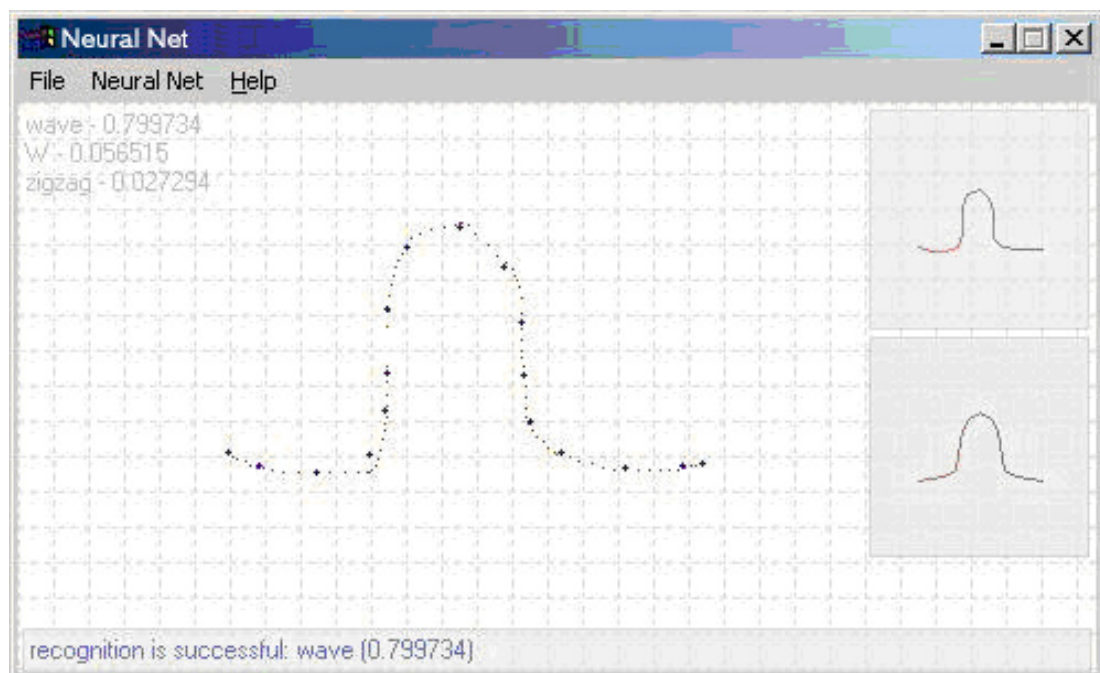


Fig 17. Riconoscimento dei tracciati

3.3 Implementazione guanto p5

Implementazione delle librerie del glove P5

La periferica a *6DOF* (*Degree of Freedom*) P5 sfrutta la tecnologia di *tracking* ottico ad infrarossi per il rilevamento della posizione e dell'orientamento nello spazio.

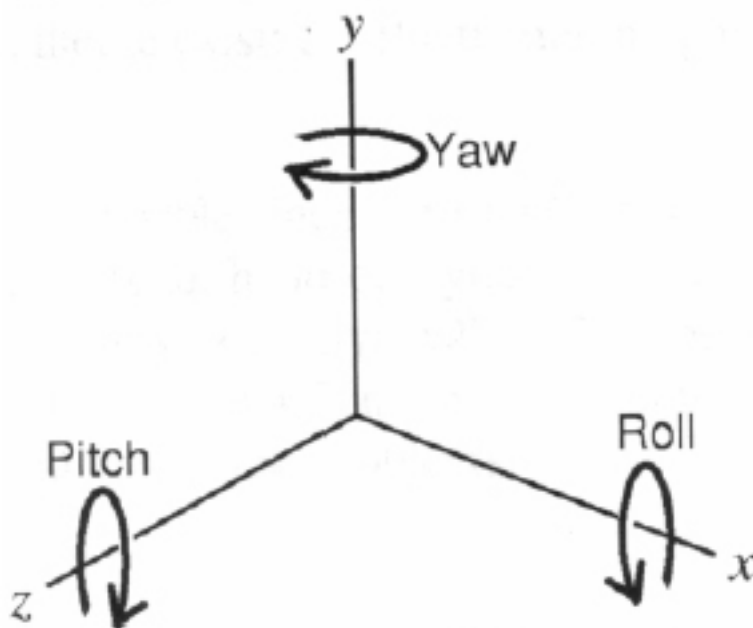


Fig 18. I sei gradi di libertà del guanto

Il dispositivo si compone di un ricevitore a raggi infrarossi e di un guanto, collegati tramite un cavo, provvisto di una serie di *marker* (identificatori di posizione) posti sugli angoli della struttura che costituisce il polso del guanto, e di cinque sensori posti sulle dita mediante i quali è possibile rilevare l'angolo di flessione delle articolazioni. Un pulsante di accensione e altri tre pulsanti ausiliari completano la configurazione di questo economico, ma interessante dispositivo di *tracking*. I *driver* ed un *SDK* (*Software Development Kit*), forniti dalla casa produttrice, permettono l'estrazione delle informazioni relative a posizione, orientamento e flessione delle dita, nonché le informazioni relative allo stato dei pulsanti. Di seguito sono esposti i passi

operativi seguiti per l'implementazione dell'applicazione sviluppata. Il punto di partenza per l'interfacciamento della periferica P5 con il web-browser è stato l'identificazione della struttura dati in uscita dal *driver* fornito in dotazione. Per fare ciò l'applicazione sviluppata ha utilizzato le funzioni *P5Motion*, disponibili con l'*SDK*, necessarie per estrarre le informazioni relative alla posizione (X, Y, Z) e la flessione delle cinque dita (definite tramite le variabili: *P5_THUMB*, *P5INDEX*, *P5_MIDDLE*, *P5_RING*, *P5_PINKY*), l'inizializzazione del dispositivo e per il filtraggio dei dati in uscita. Dopo aver incluso al codice sorgente *P5Motion.h* e *P5Motion.cpp* si è proceduto ad inizializzare l'oggetto *P5Motion*, che gestisce le variabili relative allo spostamento nello spazio, controllando che l'istanza P5 fosse un valido puntatore alla libreria *CPDLL* fornita con l'*SDK*. L'ottimizzazione dei valori relativi all'*SDK* ha permesso la calibrazione dei sensori presenti sul dispositivo necessari al rilevamento della flessione delle dita, e di stabilire la soglia di flessione mediante la modifica della variabile *BEND_THRESHOLD*.

Tale variabile è stata impostata in modo da impedire attivazioni accidentali del sistema per lievi movimenti delle dita, specificandone un valore pari a 30, corrispondente a circa metà flessione del dito (il piegamento delle dita effettuato dall'utente genera valori compresi tra 0 e 63).

Fatto ciò si è proceduto all'implementazione di un thread per l'estrazione ciclica dei dati processati dal dispositivo P5, utilizzando la funzione *Process()* contenuta nella classe *P5Motion*.

Il suddetto thread viene generato e agganciato ad ogni nuova istanza del browser, ed inserisce i dati ricevuti dalla periferica in un *buffer*.

Infine si è proceduto alla taratura del meccanismo di filtraggio dei dati in entrata per mezzo della variabile *P5MOTION_XYZFILTERSIZE*, che ha permesso di impostare le dimensioni del filtro per gli assi *X*, *Y*, *Z* e la variabile *P5MOTION_YPRFILTERSIZE*. Il meccanismo di filtro è stato necessario per evitare fenomeni di tremolio, causa di disturbo nell'acquisizione dei dati dovuti ad involontari spostamenti della mano dell'utente, compensando i dati forniti dal dispositivo di *tracking* ottico.

Tale fenomeno è causato dal fatto che il dispositivo P5 è basato su una tecnologia di *tracking* ad infrarosso a basso costo che non riesce a fornire la precisione offerta dai più costosi dispositivi magnetici, inerziali o da ben più sofisticati dispositivi ottici. Dato che lo scopo del lavoro di tesi non era quello di ottenere un'elevata precisione nel rilevamento dei movimenti, la periferica P5 ha rappresentato un ottimo compromesso tra costo e qualità delle funzionalità offerte.

Configurati adeguatamente i valori del filtro, in modo tale da eliminare i dati provenienti dalla lettura di piccoli spostamenti, è stato possibile ottenere un'acquisizione stabile e fluida del flusso di dati provenienti dalla periferica.

Una volta inizializzata la periferica ed impostati i parametri necessari, si è proceduto all'estrazione dei dati mediante

l'utilizzo delle variabili $fFilterX$, $fFilterY$. Queste variabili vengono estratte ad ogni nuovo *frame*, e contengono i valori filtrati processati dalla periferica relativi alle posizioni sugli assi X , Y . Tali valori (assoluti) forniscono l'entità dello spostamento rispetto ad un riferimento costante posto all'origine.

L'analisi sul formato delle informazioni in uscita dal dispositivo, ottenuta grazie all'utilizzo dei metodi messi a disposizione dall'*SDK*, è stata fondamentale per la formulazione dei passaggi necessari ad una corretta elaborazione dei dati.

3.4 Rete neurale per il riconoscimento del gesture-path

Perché usare le reti neurali nel riconoscimento dei percorsi?

3.4.1 Introduzione e scelta delle Reti Neurali

Supervised learning e Unsupervised learning

Le reti neurali artificiali sono sistemi di elaborazione ispirati al sistema nervoso degli esseri viventi, in grado di svolgere in modo efficiente compiti che risulterebbero, invece, difficilissimi o impossibili per i sistemi tradizionali.

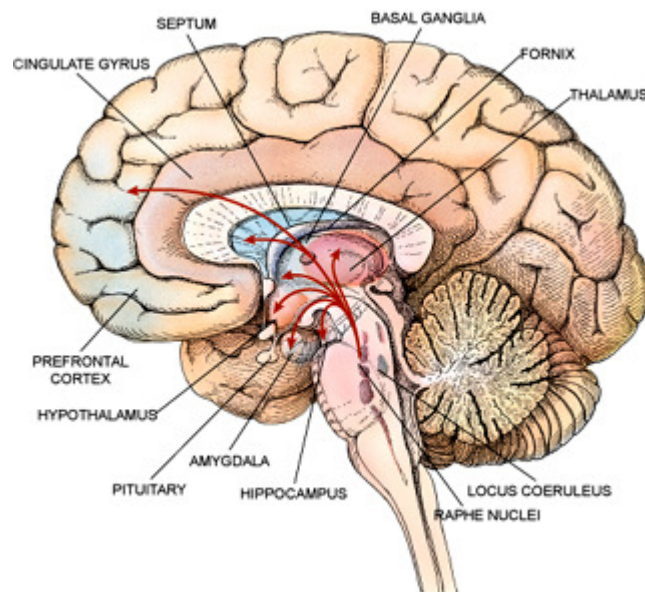


Fig 19. Le reti neurali: il cervello umano

La rete neurale da noi scelta è una multilayer con un algoritmo d'addestramento di tipo back-propagation. Sebbene l'algoritmo di back-propagation sia molto oneroso dal punto di vista computazionale, l'avvento di calcolatori elettronici sempre più potenti e meno costosi, ha permesso di impiegare tale approccio in una vasta gamma di problemi applicativi, raggiungendo spesso risultati significativi. Ciò apre nuove prospettive che spingono gran parte della comunità scientifica a studiare nuovi e più efficienti metodi di addestramento ed analizzare le reali possibilità offerte dalle reti neurali.

Prima di intraprendere l'analisi della rete multilayer, introduciamo alcuni termini che ritorneranno nelle prossime sezioni. Anzitutto, esistono due tipi di apprendimento nelle reti neurali (e in genere nei dispositivi artificiali) :

Addestramento supervisionato (Supervised learning), nel quale vengono forniti al modello connessionistico esempi contenenti un pattern di ingresso x e il corrispondente pattern di uscita y . Da queste coppie ingresso-uscita $(x; y)$, che formano il cosiddetto *training set*, devono essere ottenuti i valori per i parametri caratteristici della rete neurale (tipicamente i *pesi* in essa contenuti).

Addestramento non supervisionato (Unsupervised learning), nel quale gli esempi del training set contengono esclusivamente pattern di ingresso x . Lo scopo del processo di apprendimento è in tal caso quello di individuare eventuali correlazioni tra questi esempi, suddividendoli così "naturalmente" in classi distinte.

3.5 Riconoscimento dei gesti

3.5.1 Sistema di gesturing

Le fasi del gesture recognition

Il riconoscitore di gesti è il cuore delle interfacce gestuali, in questo capitolo verrà progettato un riconoscitore di comandi gestuali in cui verrà adottata una metodologia che si basa sull'elaborazione, mediante una rete neurale multilayer precedentemente addestrata, di N punti acquisiti dal guanto P5.

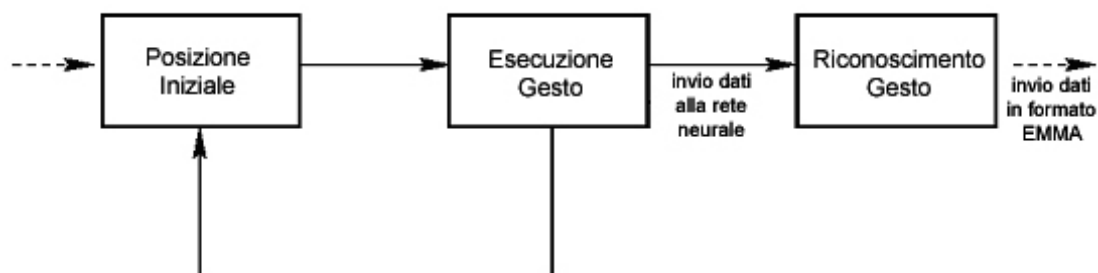


Fig 20. Schema generale del gesture recognition

Di seguito viene proposto un modello di interazione derivato da un lavoro proposto da Thomas Baudel. Questo modello descrive una tipologia d'interazione che permette all'utente di interagire con un sistema automatico attraverso gesti e nello stesso tempo lo lascia libero di usare i gesti per comunicare con il mondo circostante. Il modello si basa sul semplice concetto di individuare l'intenzionalità dell'utente di indirizzare un comando gestuale al sistema.

Ogni comando o sequenza di comandi gestuali è descritto da una posizione iniziale, il quale fa intuire al sistema che si ha l'intenzione di impartire un comando, seguito da un movimento, ovvero un gesto con cui si identifica un preciso comando nell'insieme di quelli definiti. La procedura da seguire per interagire con una interfaccia gestuale di questo tipo, secondo la modalità proposta, è la seguente:

1. riconoscimento da parte del sistema dell'intenzionalità di eseguire un comando. Questo lascia all'utente la libertà di eseguire qualsiasi gesto e muoversi come vuole intorno al

mondo circostante. Inoltre ogni sequenza di comandi deve essere preceduta dalla chiusura a pugno del guanto.

2. Impartizione comando. Il gesto viene eseguito tenendo il guanto chiuso e muovendosi nello spazio circostante, dopodiché si deve lasciare la presa.

3. termine della sessione. Dopo che la rete neurale avrà riconosciuto il gesto precedentemente eseguito, il programma genererà un xml in formato EMMA che sarà poi postato alla pagina corrente.

L'operazione di gesture recognition sta diventando sempre più popolare tra i web browser, primo tra tutti quello di Opera. E' comunque usata da poche applicazioni, ma ha grandi potenzialità nel campo di interfacce intuitive e funzionali.

Il riconoscimento di tali gesti è dovuto alla rete neurale che prende in input le coordinate della periferica P5 e, dopo aver riconosciuto il gesto trasmesso, rilascia in uscita il comando relativo al path ricevuto in ingresso. Nel prossimo paragrafo verrà descritto più in dettaglio il meccanismo adottato per il riconoscimento dei gesti.

3.5.2 Algoritmo di smussamento e conversione dei dati in ingresso

Accumulazione ed elaborazione dei dati di input

I campioni ricevuti, dopo aver tracciato un path con la periferica, saranno analizzati dalla rete neurale e poi convertiti in azioni.

Vi sono vari campioni predefiniti di path, ma per comodità ci limiteremo ad usare i più semplici ed intuitivi per interagire in modo spontaneo col browser. Naturalmente l'uso dei campioni più complessi, quali le lettere, è possibile allo scopo di dirigere una molteplicità di funzioni del browser.

Al contrario dei sistemi di riconoscimento OCR (Optical Character Recognition) del testo, le gestures prodotte dal guanto non sono analizzate come pixel di un'immagine ma come una serie di coordinate spaziali.

Il riconoscimento è suddiviso in più parti. La registrazione del path compiuto dal guanto viene immagazzinato in un'array di coordinate e poi passato ad un algoritmo di smussamento e conversione dei dati.

Innanzitutto viene estratto dai dati ricevuti in ingresso, tramite un'apposita funzione, un campione di 32 punti. Questa scelta è stata operata per ottimizzare le prestazioni della rete neurale.

Dopodiché viene applicata una funzione per trovare eventuali picchi tra i punti ricevuti. Questa elaborazione viene fatta per togliere i disturbi provenienti dalla periferica P5.

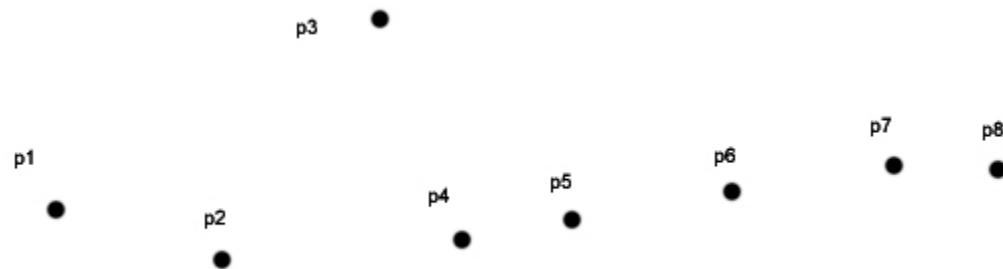


Fig 21. Algoritmo di smussamento del path. Ricerca dei picchi

In seguito si è deciso di trasformare i punti ricevuti in vettori, visto che per il riconoscimento è esclusivamente necessario avere la direzione del path eseguito e non la lunghezza.

$$\cos(\alpha) = \frac{\Delta y}{\sqrt{\Delta x^2 + \Delta y^2}}$$

con $\Delta y = y_2 - y_1$ e $\Delta x = x_2 - x_1$.

Questo procedimento è molto importante poichè ci permette di distinguere un cerchio disegnato in senso orario rispetto ad uno in senso antiorario, anche se entrambi rappresentano geometricamente la stessa figura.

La posizione assoluta del gesture effettuato non ha tuttavia influenza sul risultato. Il grande vantaggio dell'uso degli angoli come dati di input è indipendente dallo scaling. Il riconoscimento del gesture, ad esempio, di un cerchio di 3 cm di diametro ha lo stesso esito di uno di 15 cm di diametro.

Esempio:

Un cerchio disegnato in senso orario produce dei campioni di questo tipo:

280, 259, 226, 209, 193, 147, 99, 78, 57, 18, 355, 330, 302, 270,

Invece un cerchio disegnato in modo antiorario ha i seguenti valori:

98, 120, 154, 193, 228, 259, 273, 319, 340, 5, 35, 47, 75, 90.

La rete neurale riconosce la figura del cerchio anche se viene disegnata dal basso verso l'alto o viceversa.

3.5.3 Riconoscimento di forme con le reti neurali

Il learning della rete neurale

Il riconoscimento di un path è molto difficile da eseguire poiché serve un sistema error tollerant adattabile. Le reti neurali sono molto indicate alla soluzione di questo tipo di problema.

Con le reti neurali si simula la percezione dei dati attraverso stime e tolleranza di errori provenienti da dati inesatti. Non vi sono procedure matematiche atte alla stima di tali valori. Esempi di applicazione di rete neurale sono il riconoscimento del testo, impronte digitali o volti di persona.

La caratteristica più importante di una rete neurale è l'adattabilità. Senza una programmazione specifica, le reti neurali possono solamente imparare dai dati forniti durante una fase iniziale d'addestramento.

Dopo alcune ricerche si è deciso di usare una rete multilayer con uno strato inputlayer costituito da 32 sinapsi, un hiddenlayer di 32 neuroni e un outputlayer di 29 assioni (uno per ogni gesture predefinita).

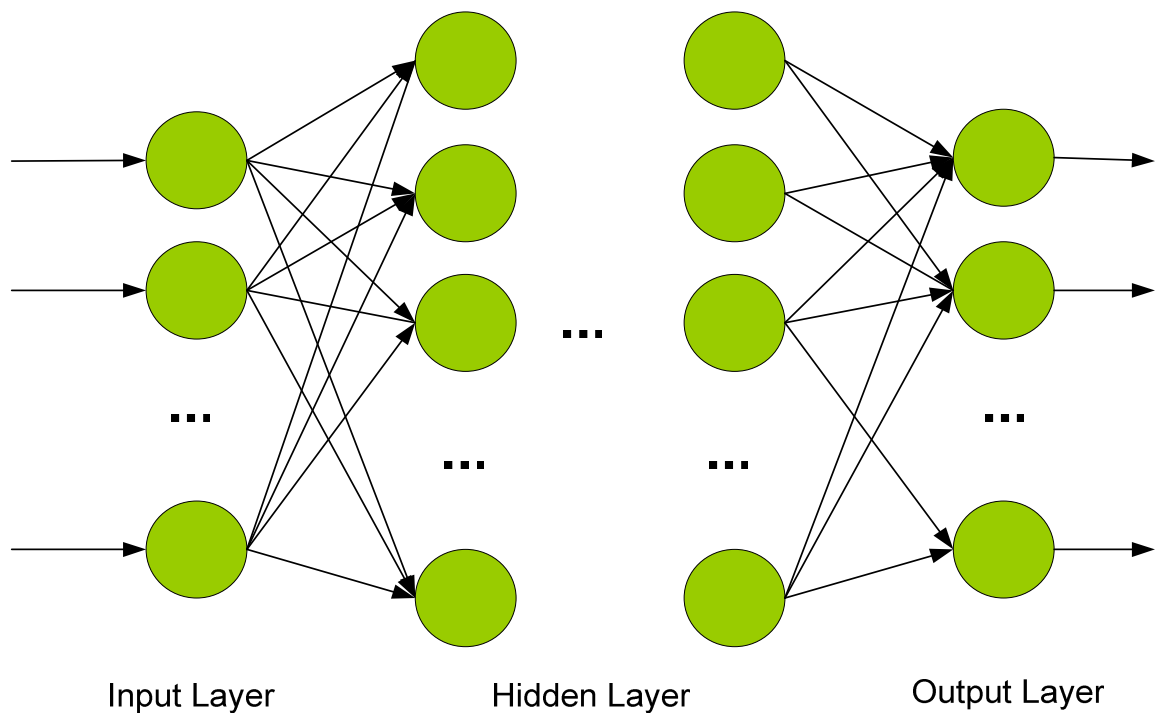


Fig 22. Rete neurale multilayer

La rete mostrata ha più HiddenLayer ed è atta alla risoluzione di problemi molto complicati, noi ne useremo solamente uno.

I collegamenti hanno pesi variabili come valori di soglia di attivazione dei neuroni negli strati collegati. I pesi dei collegamenti hanno come valore numeri da -1,0 a 1,0 e sono settati durante la fase di learning della rete (viene usato l'algoritmo di Backpropagation). I pesi della rete sono stati precalcolati attraverso l'ausilio di un programma di terze parti.

3.6 La tecnologia dei Browser Helper Object

3.6.1 Introduzione

Caratteristiche generali dei Browser Helper Object

Quando viene avviata un'istanza del browser, nel Registro di configurazione viene eseguita automaticamente una ricerca degli identificatori CLSID (CLaSS IDentifier) presenti nella seguente chiave:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser  
Helper Objects
```

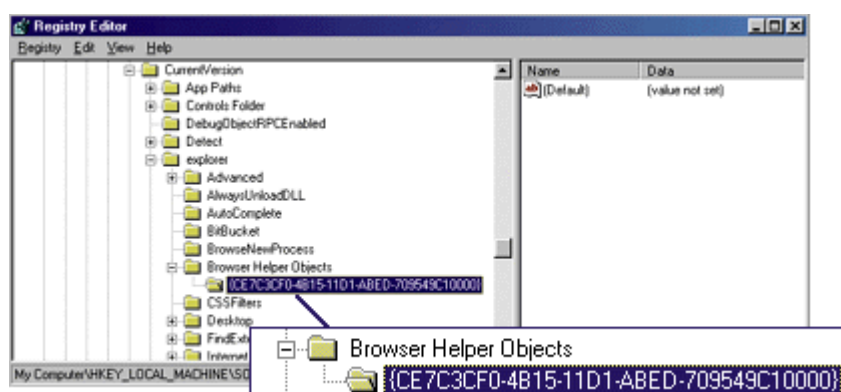


Fig 23. Visualizzazione del CLSID del BHO tramite il Registry Editor

Se questa chiave esiste e presenta dei CLSID, il browser utilizzerà `CoCreateInstance()` per tentare di creare un'istanza di ciascun oggetto elencato come sottochiave in tale chiave. Si noti che il CLSID dell'oggetto creato deve essere elencato come sottochiave e non come valore nominato.

Ad esempio, immaginiamo che nel Registro di configurazione sia presente la seguente voce:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\
  Browser Helper Objects\
    {CE7C3CF0-4B15-11D1-ABED-709549C10000}.
```

Ogni volta che si avvierà Internet Explorer, il programma tenterà di creare un'istanza dell'oggetto a cui l'identificatore CLSID `{CE7C3CF0-4B15-11D1-ABED-709549C10000}` fa riferimento. Queste chiavi di registro possono essere create manualmente

oppure tramite la porzione di codice di registrazione automatica dell'applicazione COM utilizzata.

Quando si crea un Browser Helper Object è consigliabile tenere presente quanto segue:

- L'applicazione creata deve essere un server in-process ovvero una DLL.
- Tale DLL deve implementare `IObjectWithSite`.
- È necessario implementare il metodo `ObjectWithSite::SetSite()`. Attraverso questo metodo l'applicazione riceve un puntatore per `IUnknown` del browser. (in effetti passa un puntatore a `IWebBrowser2` ma l'implementazione di `SetSite()` riceve un puntatore a `IUnknown`.) È possibile utilizzare questo puntatore `IUnknown` per automatizzare il browser oppure per elaborare gli eventi da esso.
- Deve essere registrato come un Browser Helper Object, come descritto in precedenza.

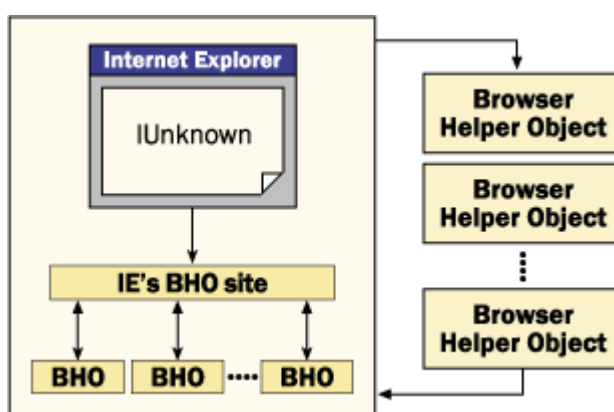


Fig 24. Come il browser inizializza il BHO. Il "BHO site" è l'interfaccia usata per inizializzare una connessione

Per creare un Browser Helper Object si può utilizzare qualsiasi linguaggio che supporti COM e l'implementazione delle interfacce.

Visto che il BHO è un COM in-process server la scelta migliore per costruirne uno è l'ATL(Active Template Library).

L'ATL è già fornito di una sua implementazione dell'interfaccia IObjectWithSite, tra tutte le tipologie di oggetti predefiniti che il wizard ATL COM supporta nativamente vi è l'Internet Explorer Object, che è la struttura di come deve essere un BHO.

Infatti un ATL Internet Explorer Object è un COM in-process server che supporta IUnknown IObjectWithSite e la registrazione automatica nel registro di sistema.

3.6.2 Implementazione

Ricordando che ATL Internet Explorer Object è un semplice oggetto COM server che supporta IUnknown e l'auto-registrazione nel registro di sistema più IObjectWithSite.

Il metodo SetSite() è il punto nel quale il BHO viene inizializzato, tale metodo viene invocato solo una volta per ogni istanza di Internet Explorer creata secondo il suo ciclo di vita. In tale metodo viene inserito il codice per agganciare il ciclo di lettura del guanto all'istanza di Internet Explorer che lo ha invocato in modo da inizializzare la lettura del guanto una sola volta.

La possibilità di agganciarsi al web browser è fornita in principio dal fatto che il metodo `SetSite()` restituisce l'handle della finestra (HWND) dell'istanza del browser in fase di caricamento in modo da poter agganciare il nostro codice.

Un altro utilizzo che successivamente viene fatto dell'HWND è quello di eseguire delle operazioni sulla finestra (massimizzazione, riduzione, ecc.) simulando, tramite l'invio di messaggi, la pressione dei tasti.

Quando si naviga un URL con Internet Explorer è necessario attendere il verificarsi di una serie di eventi sollevati dal browser necessari per garantire che il download del documento richiesto sia stato completamento e inizializzato.

Solamente a questo punto sarà possibile accedere al contenuto attraverso i vari modelli di oggetti esposti, questo significa che è necessario acquisire una serie di puntatori a tali oggetti.

Il primo puntatore indispensabile per l'implementazione è quello a `IWebBrowser2`, l'interfaccia che esegue il render dell'oggetto `WebBrowser`.

Il secondo puntatore necessario è quello di notifica degli eventi. Questo viene registrato come un ascoltatore di eventi provenienti dal browser per permettere la ricezione dell'evento di caricamento di una pagina web.

Per concludere, particolare attenzione si è rivolta verso la fase di automazione di Internet Explorer per l'invio dei dati, inerenti

all'azione svolta dall'utente, utilizzando il metodo POST verso un server HTTP come TomCat.

Parecchi metodi sono utilizzabili per l'invio dei dati verso un HTTP server, quelli di GET e POST sono tra i più comuni. Il metodo di POST è tipicamente utilizzato per inviare i dati di una form a un HTTP server, quando questi eccedono la massima dimensione di trasferimento permessa utilizzando il metodo GET (tipicamente 2 KByte). Tenendo conto della possibilità di futuri sviluppi e relativa crescita nelle dimensioni dei dati trasmessi si è scelto di adottare il metodo POST.

A tal fine Internet Explorer espone il metodo Navigate nell'interfaccia IWebBrowser, il cui prototipo è il seguente:

```
HRESULT Navigate([in] BSTR URL, [in] VARIANT* Flags,  
[in] VARIANT* TargetFrameName, [in] VARIANT* PostData,  
[in] VARIANT* Headers);
```

Allo scopo di eseguire una POST solo i parametri URL, PostData, e Headers sono rilevanti. Per chiamare Navigate e POST form data verso un HTTP server, il parametro di URL deve specificare un indirizzo valido che in fase di analisi si è stabilito essere lo stesso indirizzo della pagina caricata e sulla quale si sta svolgendo l'azione.

Il parametro Headers deve contenere il Content-Type il quale è in formato MMI/EMMA.

Questo parametro indica che il POST dovrà essere intercettato dal filtro server ed elaborato secondo gli altri requisiti progettuali.

3.7 Ambiente e scopo del filtro

La necessità di filtrare le informazioni multimodali

Fra le novità introdotte in Servlet API 2.3 c'è un nuovo componente chiamato filtro.

Un filtro è un pre-processore delle richieste che arrivano ad una servlet/jsp e/o un post-processore di risposte generate da una servlet/jsp. E' dunque possibile intercettare, esaminare e modificare gli oggetti request e response legati ad una servlet o jsp.

I filtri sono importanti per un ampio numero di ragioni:

- 1 Essi possiedono la capacità di incapsulare task ricorrenti in unità riusabili. Il codice modulare è più maneggiabile e documentabile, è facile eseguire il debug e se ben strutturato può essere riutilizzato in altre applicazioni web.
- 2 I Filtri possono essere utilizzati per trasformare l'oggetto response proveniente da una servlet o da una pagina JSP. Un'operazione comune per le applicazioni web è quella di

formattare i dati trasmessi al cliente in un formato differente da quello html, senza dover effettuare modifiche ai componenti dell'applicazione web.

Con l'introduzione dei filtri come parte delle specifiche Java Servlet, gli sviluppatori odierni hanno l'opportunità di scrivere componenti di trasformazioni riusabili, portabili attraverso i vari containers.

E' giusto ricordare che un filtro non è una servlet, ma è un componente invocato dal container J2EE quando un client richiede una risorsa per la quale il filtro è stato configurato.

- Intercetta un'invocazione di una richiesta prima che la richiesta venga soddisfatta.
- Esamina la richiesta prima che la richiesta venga soddisfatta.
- Modifica le request headers e request data provvedendo ad una versione rielaborata (customized) dell'oggetto request che riscrive la reale request.
- Modifica la response headers e response data provvedendo a una versione su misura dell'oggetto response che sovrascrive quello reale.
- Intercetta l'invocazione della servlet dopo che la servlet è stata invocata.

Si può configurare il filtro in modo che abbia effetto su una o un gruppo di pagine/servlet; zero o più filtri possono filtrare una o più servlet. Il filtro, pur non essendo una servlet, ha una struttura simile; si tratta di un'interfaccia (`javax.servlet.Filter`) che espone 3 metodi :

1. `public void init(FilterConfig filterConfig)` : Chiamato prima che il filtro entri in servizio e setta l'oggetto di configurazione del filtro.
2. `public void doFilter(final ServletRequest request, final ServletResponse response, FilterChain chain)` : esegue l'attuale lavoro del filtro.
3. `public void destroy()` : Chiamato dopo che il filtro è stato messo fuori servizio.

I metodi seguono lo stesso lifecycle delle servlet. Il comportamento di un filtro è dettato dall'implementazione del metodo `doFilter`.

Il server chiama "`init(FilterConfig)`" quando deve preparare il filtro per il servizio, in seguito chiamerà "`doFilter()`" ogni volta che verrà intercettata una richiesta speciale configurata per l'utilizzo del filtro. L'interfaccia "`FilterConfig`" possiede dei metodi per recuperare il nome dei filtri e i suoi parametri "`init`" e attiva il servlet context. Il server chiama "`destroy()`" per indicare che il

filtro sta per uscire dal servizio. Precedentemente il lifecycle implica un metodo `"setFilterConfig(FilterConfig)"` per l'impostazione della configurazione del filtro. Nel suo metodo `"doFilter()"` ogni filtro riceve la request e response corrente, come pure un `"FilterChain"` contenente i filtri che ancora devono essere processati. Nel metodo `"doFilter()"` un filtro può eseguire con gli oggetti request e response le operazioni che desidera, potrebbe raccogliere dati ed elaborarli chiamando i loro metodi, o modificare gli oggetti dando loro un nuovo comportamento. In seguito il filtro chiama il metodo `"chain.doFilter()"` per trasferire il controllo al filtro successivo. Quando questa chiamata ritorna un filtro può, alla fine dell'esecuzione del proprio metodo `"doFilter()"`, realizzare del lavoro addizionale sull'oggetto response. Può annotare, ad esempio, le informazioni sulla response. Se un filtro desidera fermare l'elaborazione della richiesta e ottenere il pieno controllo della risposta, esso dovrà intenzionalmente non invocare il filtro successivo. Sicuramente l'aspetto più rilevante è che il filtro è assolutamente una struttura non invasiva per qualsiasi applicazione web già esistente; l'aggiunta o la rimozione dello stesso si traduce in una manipolazione del file `web.xml`.

3.8 Implementazione Filtro

Implementazione del filtro per il riconoscimento del formato EMMA/MMI

Il filtro creato si pone come filtro di intercettazione per un modello di controllore frontale su tutte le richieste entranti rivolte alla web application in cui è stato installato. Quando rileva una richiesta, tramite un HTTP POST di dati in formato EMMA eseguita dal client al server, identificata grazie ad un content-type "MMI/EMMA", il filtro cattura l'oggetto request della richiesta entrante e applica dei criteri di analisi appositamente studiati per lo speciale formato di content-type. Il filtro ricevente lo speciale involucro della richiesta ha accesso ai parametri del post attraverso il metodo standard "getParameter()" in modo che possa valutare le azioni da svolgere e ridefinire il comportamento della richiesta.

Il sequence diagram nella figura sottostante illustra il generico funzionamento di un'applicazione web prima dell'inserimento di un filtro di intercettazione.

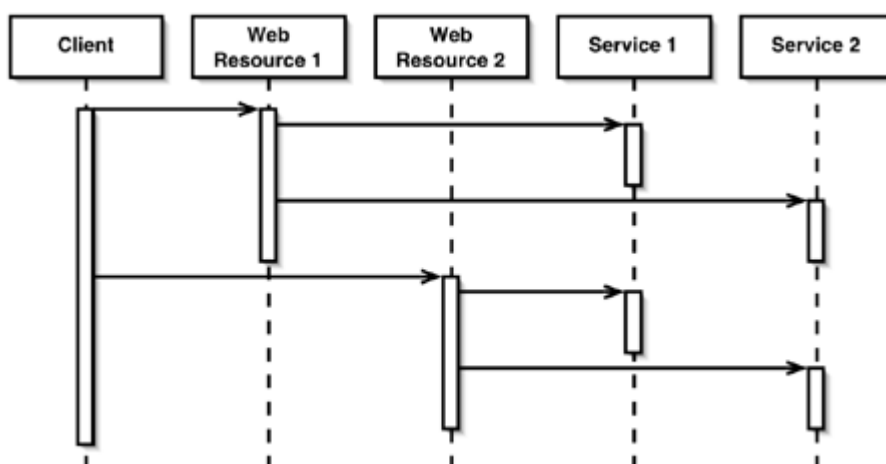


Fig 25. Sequence Diagram di un applicazione senza filtro

Il sequence diagram nella figura sottostante illustra il generico funzionamento di una web application dopo l'inserimento di un filtro di intercettazione.

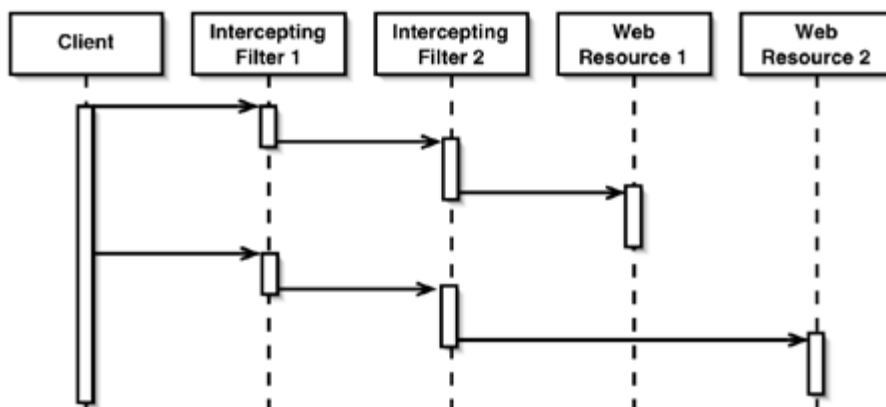


Fig 26. Sequence Diagram con applicazione del filtro



Testing e valutazione risultati

4.1 Analisi dei risultati

Introduzione al testing dell'applicazione

In seguito verranno riportati alcuni risultati sperimentali mediante il prototipo di interfaccia gestuale, illustrato nel capitolo precedente. Vedremo come sono stati soddisfatti i vincoli che ci eravamo proposti nella premessa e mostreremo come con il prototipo realizzato abbiamo ottenuto ottimi risultati.

4.2 Caratteristiche generali del lavoro

L'hardware usato nei test

Vediamo innanzitutto in quali condizioni abbiamo eseguito i test sull'interfaccia gestuale. Per le prove è stata utilizzata la piattaforma hardware illustrata, quindi nessun hardware particolarmente costoso o specializzato.



Fig 27. Portatile sul quale sono state effettuate le prove

Anche se la periferica P5 non possiede grandi qualità per ricezione del segnale, siamo comunque riusciti ad ottenere degli ottimi risultati.

Per la progettazione sono state utilizzate le librerie ATL della Microsoft. L'intero codice sorgente è stato sviluppato con Microsoft Visual C++ versione 8.

4.3 Training

Le fasi del training

Per utilizzare l'interfaccia gestuale è necessario eseguire preventivamente tre fasi.

La prima fase è stata il training della rete neurale attraverso un programma di terzi.

In seguito si è passati alla configurazione delle azioni del plugin dall'apposito pannello di controllo.

La terza fase consiste nella mappatura sul database delle azioni possibili per ogni pagina del sito preso d'esempio.

4.4 Sessione sperimentale

I parametri di valutazione usati nei test

Per dare una valutazione complessiva sul riconoscitore gestuale, sono state eseguite alcune sessioni di prova, nelle quali si è cercato di valutare i seguenti parametri:

- **fattore di riconoscimento**, denotato con **R**, che stima l'efficienza del riconoscitore e viene misurata dal seguente rapporto:

$$R = \frac{\text{pose correttamente riconosciute}}{\text{pose istanziate}}$$

- **semplicità d'uso** del plugin nel settaggio dei parametri, che indichiamo con la lettera **U**.
- Fattore di **spontaneità nella navigazione** attraverso il gesturing, indicato con **N**.

Per la valutazione dei parametri particolari come **U** e **N** abbiamo usato degli aggettivi valutandoli poi con dei numeri simbolici seguendo la scaletta riportata qui sotto:

Ottimo	Buono	Discreto	Sufficiente	Insufficiente	Scarso
5	4	3	2	1	0

Fig 28. Tabella parametri valutativi

4.5 Test per il riconoscimento dei gesture

4.5.1 Prove effettuate dai trainer

I primi risultati

Abbiamo effettuato una sessione di prova dell'interfaccia da noi sviluppatori e una su un gruppo di persone estranee al progetto.

Le prove effettuate da noi hanno dato ottimi risultati prossimi al 100% del riconoscimento. Il sistema navigazionale pensato ci è sembrato intuitivo ed efficiente. Per quanto concerne l'utilizzo della periferica P5 non si è verificato alcun problema.



Fig 29. Momento durante i test

4.5.2 Prove effettuate dagli altri utenti

I test effettuati da utenti estranei all'applicazione

L'utilizzo dell'interfaccia gestuale non è ovviamente una prerogativa dei trainer, ma deve fornire buoni risultati anche su altri utenti che non hanno preso parte allo sviluppo dell'applicazione. Difatti abbiamo sottoposto a una sessione di prova un campione di 20 persone.

Ognuna di queste persone è stata invitata ad eseguire alcune prove. Ogni sessione di prova consiste nel porre l'utente di fronte all'interfaccia, spiegandone sommariamente il funzionamento e chiedendo loro di interagire con il browser

attraverso il gesturing. Le indicazioni circa il vocabolario dei gesti erano illustrate in una piccola legenda cartacea posta vicino alla tastiera.

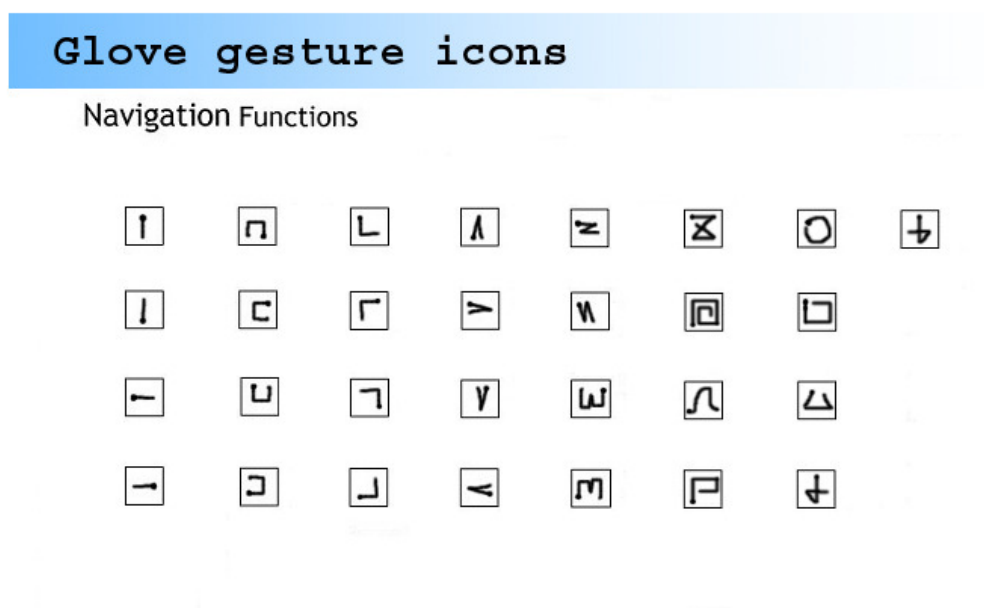


Fig 30. Legenda usata durante la fase di prova

Abbiamo eseguito diverse sessioni, ciascuna caratterizzata da un valore diverso dei parametri precedentemente definiti.

I dati vengono riportati nella tabella sottostante:

Utente	R	U	N
1	47/50	Ottimo	Ottimo
2	43/50	Discreto	Ottimo
3	49/50	Buono	Ottimo
4	48/50	Ottimo	Ottimo
5	44/50	Ottimo	Buono
6	49/50	Ottimo	Buono
7	47/50	Buono	Ottimo
8	47/50	Buono	Buono
9	44/50	Ottimo	Discreto
10	43/50	Discreto	Buono
11	47/50	Buono	Ottimo
12	48/50	Buono	Ottimo
13	42/50	Buono	Buono
14	46/50	Ottimo	Ottimo
15	48/50	Buono	Discreto
16	47/50	Buono	Buono
17	44/50	Buono	Buono
18	43/50	Ottimo	Buono
19	47/50	Ottimo	Buono
20	48/50	Buono	Ottimo
Totale in centesimi	92,1%	78% Buono	80% Buono

Fig 31. Tabella riassuntiva dei dati raccolti durante le prove

I dati mostrano come l'applicazione abbia dato ottimi risultati anche con degli strumenti di costo non elevato. Sicuramente con una periferica migliore avremmo ottenuto il massimo nel riconoscimento dei gesture effettuati.

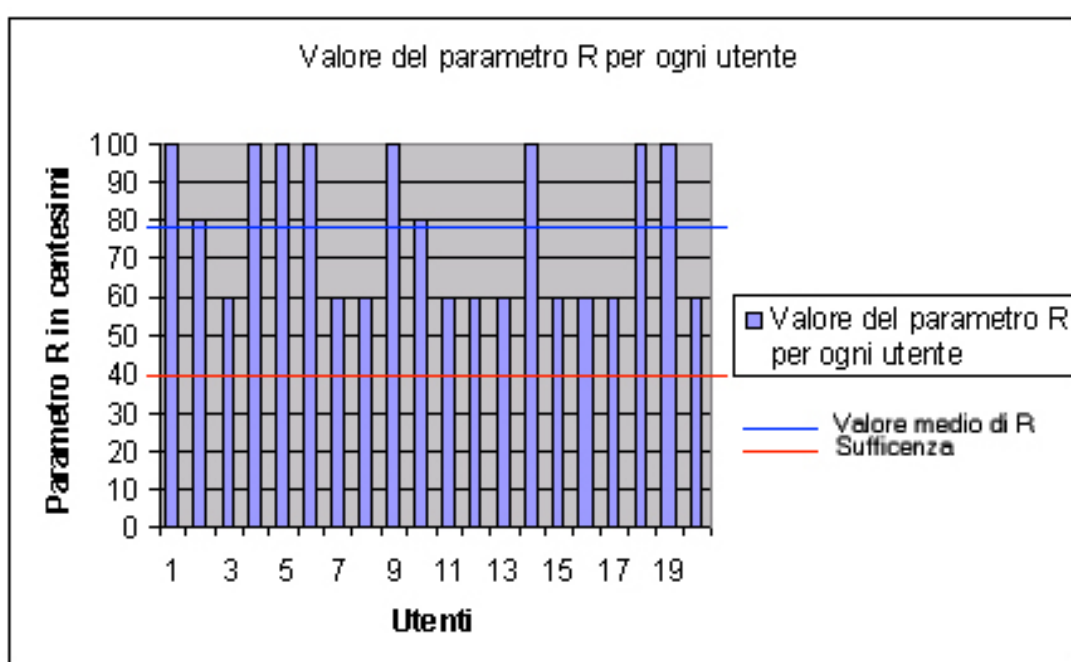


Fig 32. Diagramma riassuntivo di valutazione del parametro R

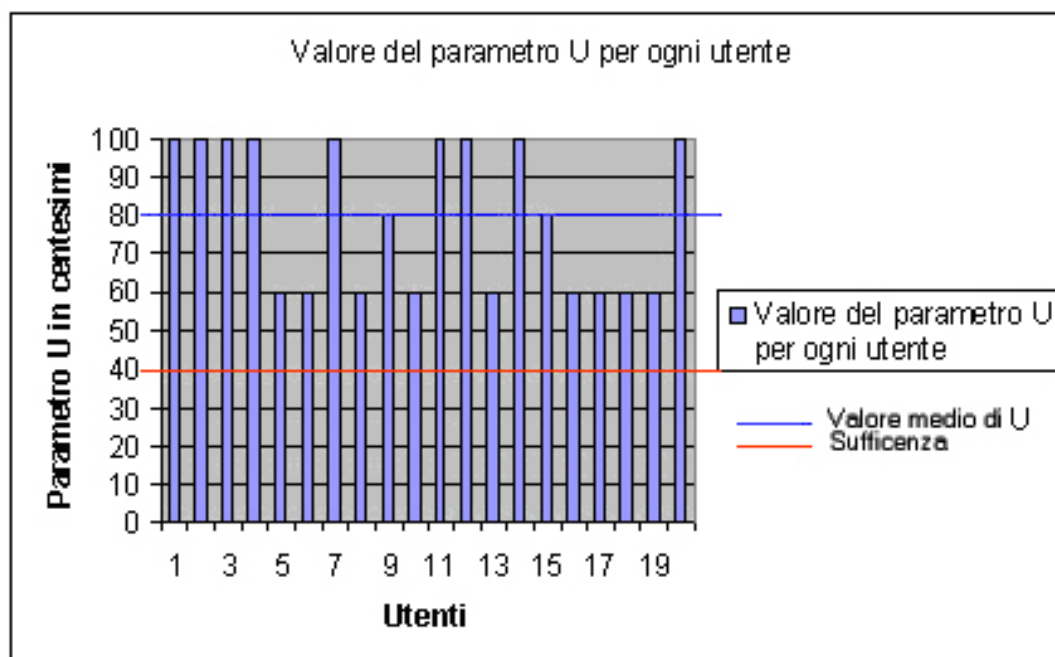


Fig 33. Diagramma riassuntivo di valutazione del parametro U

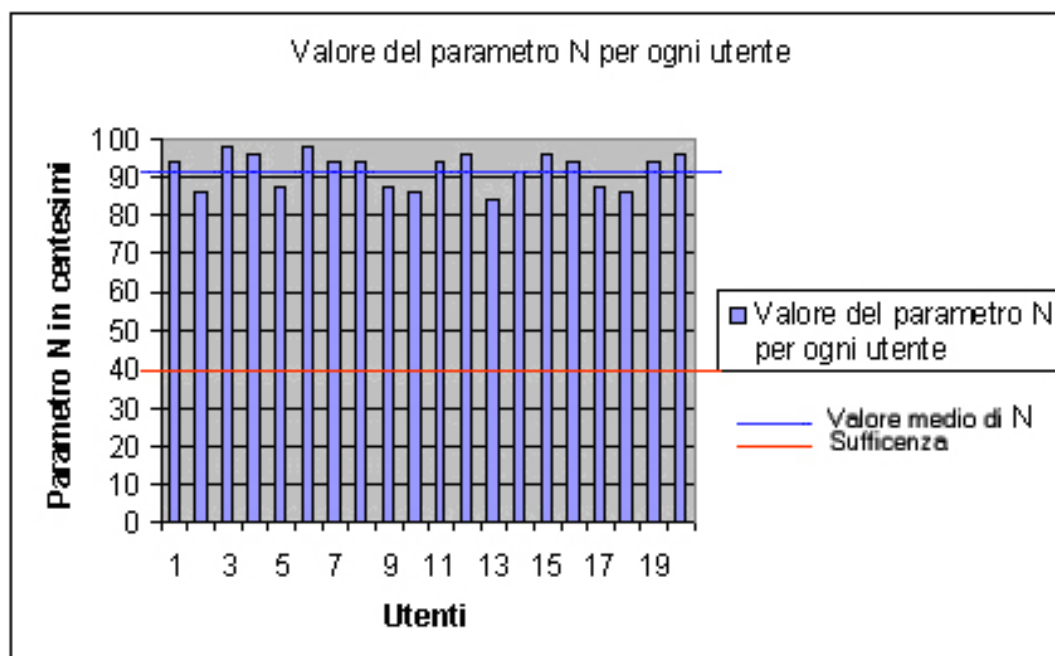


Fig 34. Diagramma riassuntivo di valutazione del parametro N



Conclusioni e Sviluppi Futuri

5.1 Conclusioni

Impressioni e valutazioni finali dell'applicazione

La piattaforma utilizzata è tra le più comuni, l'algoritmo può essere implementato su qualsiasi macchina che disponga di un browser Internet Explorer di Microsoft. Il sistema proposto fa uso di un hardware economico. Quindi il plugin realizzato è proponibile come un'applicazione di largo consumo, anche se ancora non ci sono alcuni fattori culturali e tecnologici per una diffusione di interfacce che lo supportino.

Complessivamente, nel sistema proposto, il fattore di riconoscimento è risultato sorprendentemente alto.

Sono state rispettate tutte le specifiche proposte dai vari standard usati e per quanto riguarda l'impatto sull'utente, è stato positivo. Durante le sessioni di prova del dispositivo con i diversi utenti abbiamo raccolto oltre che i dati sperimentali anche delle opinioni d'uso e dei pareri. Gli utenti che sono stati sottoposti alle sessioni hanno trovato l'interfaccia stimolante e di facile utilizzo, apprezzandone l'aspetto innovativo.

5.2 Sviluppi futuri

Miglioramenti ed integrazioni da apportare

I risultati ottenuti al termine di questo progetto di tesi, ci ha condotto ad un'analisi attenta di quali possano essere gli sviluppi futuri legati all'applicativo sviluppato, tra questi lo sviluppo del plugin per Internet browser differenti (di passaggio ricordiamo Opera, che in associazione con IBM sta progettando un web browser multimodale). Sarebbe interessante collaudare, inoltre, il funzionamento dell'applicativo sviluppato con guanti virtuali più costosi, performanti e innovativi, dotati di ricettori ad ultrasuoni invece che ad infrarossi, così da avere un'area di lavoro più ampia e non essere ostacolati dalle connessioni fisiche tra periferica e ricettore.



Fig 35. Glove P5. Sistema con connessioni fisiche tra ricevitore e periferica



Fig 36. Superglove. Sistema con ricevitore ad ultrasuoni

Particolare attenzione e ulteriore sviluppo andrebbe dato all'interazione con differenti canali multimodali, secondo la proposta di standard EMMA, come ad esempio il textspeech. L'interazione tra diversi tipi di linguaggi human-friendly, come gestualità e voce, porterebbe ad un incremento del fattore **N** (spontaneità nella navigazione di pagine multimodali), come dimostrato da vari trattati sul NLI (Natural Language Interaction).

Per la fase di learning della rete neurale, è stato usato un software di terze parti con ridotte capacità di addestramento; un miglioramento significativo potrebbe essere dato dall'implementazione di un applicazione, specializzata ad incrementare la precisione del riconoscimento del gesture.

La logica dei tracciati navigazionali, con la quale avviene l'interazione tra il server e l'applicazione di gesture recognition, si fonda - come visto in precedenza - sull'utilizzo di una base di dati caricata manualmente rendendo il processo di compilazione oneroso in termini di tempo. Ai fini di ottenere una rapida fase di mappatura si potrebbe realizzare un componente aggiuntivo con la capacità di automatizzare questo processo. Questo componente potrebbe essere disegnato per sfruttare le potenzialità di un CASE (Computer Aided Software Engineering) come Webratio, in grado di sviluppare applicazioni per il Web passando dall'analisi dei requisiti direttamente alla generazione automatica del codice.

Bibliografia

JavaWorld (2004) Filter code with Servlet 2.3 model. Informazioni reperite il 2 Agosto 2004 dal World Wide Web: <http://www.javaworld.com/javaworld/jw-06-2001/jw-0622-filters.html>

Bolt, R. A., Herranz, E. (1992). Two-handed gesture in multi-modal natural dialog. Proceedings of the 5th annual ACM symposium on User interface software and technology.

DataGlove. (2004). L.R.P. Force Feedback Data Glove. Informazioni reperite il 26 Marzo 2004 dal World Wide Web: <http://www.caip.rutgers.edu/~bouzit/lrp/Video/Glove.mpg>

Essential Reality (2004). Pagina ufficiale del dispositivo di tracking P5Glove. Informazioni reperite il 20 Aprile 2004 dal World Wide Web: <http://www.essentialreality.com/index.asp>

Hauptmann, A. (1989). Speech and Gestures for Graphic Image Manipulation Proc. CHI'89, (pp. 241245). Reperito il 10 Febbraio 2004 dal World Wide Web: <http://citeseer.nj.nec.com/hauptmann89speech.html>

Kemble, K. A. (2001). An Introduction to Speech Recognition. Informazioni reperite il 22 Febbraio 2004 dal World Wide Web: <http://www.voicexmlreview.org/Mar2001/features/recognition.htm>

T. J. Darrell and A. P. Pentland. Recognition of space-time gestures using a distributed representation. Technical report, M.I.T. Media Laboratory Vision and Modeling Group, 1995.

J. Davis and M. Shah. Gesture recognition. Technical report, Dep. of Computer Science University of Central Florida, Orlando, 1993.

Starner T. E. Visual recognition of american sign language using hidden markov models. Technical report, Program in Media Arts and Sciences, School of Architecture and Planning, MIT, Cambridge, MA, 1995.

W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In IEEE Intl. Workshop on Automatic Face and Gesture Recognition, Zurich, June 1995.

W. T. Freeman and C. Weissman. Television control by hand gestures. In IEEE Intl. Workshop on Automatic Face and Gesture Recognition, Zurich, June 1995.

T. Heap and F. Samaria. Real-time hand tracking and gesture recognition using smart snakes. Technical report, Olivetti Research Lim. Cambridge UK, June 1995.

C. Hand, I. Sexton, and M. Mullan. A linguistic approach to the recognition of hand gestures. In Designing Future Interaction conference (Ergonomics Society/IEEE). University of Warwick, UK, apr. 1994.

M. K. Hu. Pattern recognition by moment invariants. In Proc.IRE, number 49, Sept. 1961.

Schlenzig J., Hunter E., and R. Jain. Recursive identification of gesture inputs using hidden markov models. In second IEEE workshop on applications of Computer Vision Sarasota, Sarasota, Florida, 1994.

Thorison K., Koonos D., and Bolt R. Multi-modal natural dialogue. In Proceedings ACM CHI, 1992.

Murakami K. and Taguchi H. Gesture recognition using recurrent neural networks. In Proceedings ACM CHI, 1991.

A. KatKere, E. Hunter, D. Kuramura, J. Schlenzig, S. Moezzi, and R. Jain. Robogest: Telepresence using hand gestures. Technical report, Visual Computing Laboratory, University of San Diego, CA, 1994.

A. Khotanzad and J. Lu. Classification of invariant image representations using a neural network. IEEE Trans. on Acoustic, Speech and Signal Processing, 38(6):1028-1038, June 1990.

Shozo Kondo. Automatic recognition of handwriting. Computer Recognition and Human Production of Handwriting, 1989.

R. E. Kahn, M. J. Swain, P. N. Prokopowicz, and R. J. Firby. Real-time gesture recognition with the perseus system. Technical report, Dep. of Computer Science, University of Chicago, IL.

D. McNeill. Hand and mind. Technical Report. University of Chicago Press, Chicago, 1992.

F. Quek. Toward a vision-based hand gesture interface, May 1994.

F. Quek. Non-verbal vision-based interfaces. In Keynote Speech, International Workshop in Human Interface Technology '95, Aizuwakamatsu, Fukushima, Japan, Oct. 1995.

Bolt R. Put that here: Voice and gesture at the graphics interface. In Proceedings ACM SIGGRAPH, 1980.

C. Teh and R. T. Chin. On image analysis by the methods of moments. IEEE Trans. on Pattern Analysis and machine Intellig., 10:496{512, July 1988.

K. Thorisson, D. Koons, and R. Bolt. Multi-modal natural dialogue. In CHI'92 Conference Proceedings. ACM Press, 1992.

Richard Watson. A survey of gesture recognition techniques. Technical report, Dep. of Computer Science, Trinity College, Dublin, 1993.

A. Wilson and A. Bobick. Learning visual behaviour for gesture analysis. In Prooc. of the IEEE Symposium on Computer Vision, Coral Gablef, Florida, Nov. 1995.