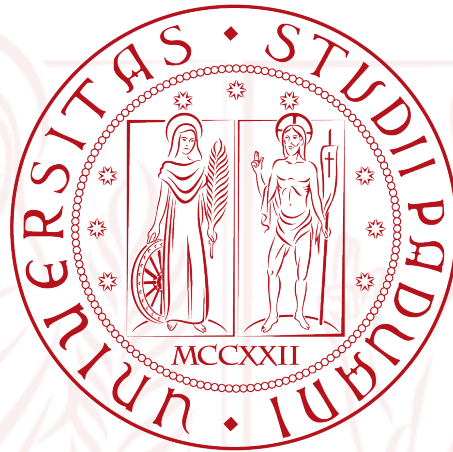


Università degli Studi di Padova



Corso di Laurea Magistrale in Ingegneria
Informatica

**Ambienti multimodali per
l'apprendimento: progetto e
implementazione di un algoritmo di
tracciamento per l'interazione
gesto-suono**

14 Aprile 2014

Relatore: Antonio Rodà

Studente : Simone Rinaldo

Anno Accademico 2013/2014

Sommario

Negli ultimi decenni, con la creazione di nuove tecnologie nell'ambito informatico, il tema dell'interattività si è spostato da semplice studio accademico a realtà di ogni giorno. L'interattività infatti è presente in molte delle nostre azioni quotidiane. In ambito scolastico, oltre all'utilizzo di computer e tablet, numerosi sistemi interattivi sono stati progettati e introdotti nelle scuole come LIM e registri elettronici.

Questa tesi si inserisce all'interno di un progetto dell'università di Padova, la Stanza Logo-Motoria, un sistema interattivo con il quale è possibile creare un ambiente di realtà aumentata (augmented reality), tramite l'uso di tecniche di tracciamento video, con l'utilizzo di immagini e suoni come feedback.

La Stanza Logo-Motoria è un ambiente interattivo multimodale per l'apprendimento. Il progetto si avvale della collaborazione di ricercatori nell'ambito psicologico ed educativo, allo scopo di realizzare uno strumento di supporto ai tradizionali metodi didattici. Con l'ausilio della tecnologia e seguendo l'approccio enattivo, è possibile aiutare le scuole a fornire modalità di insegnamento efficaci, che tengano conto delle diverse capacità degli allievi anche nei casi di specifiche difficoltà di apprendimento.

Il sistema hardware della stanza è molto semplice, infatti è costituito da una webcam installata al soffitto e un computer, questo per abbassare i costi e la complessità del progetto in modo tale da renderlo effettivamente integrabile nelle ordinarie attività didattiche.

Allo stato attuale la Stanza Logo-Motoria è in grado di individuare la posizione dell'utente nello spazio e in base a questa informazione eseguire diverse

applicazioni. L'obiettivo della tesi è quella di ampliare le informazioni ricavate dal sistema aggiungendo oltre alla posizione dell'utente il movimento delle mani, in modo da creare gli strumenti per migliorare l'esperienza delle varie applicazioni che possono girare nella Stanza.

Il testo della tesi si suddivide principalmente in quattro sezioni. Nel primo capitolo vengono introdotti i concetti di interattività, multimedialità ed enattività nel contesto dell'apprendimento, seguendo con la descrizione della Stanza Logo-Motoria confrontandola con l'attuale stato dell'arte di sistemi analoghi. Nel secondo capitolo viene descritto effettivamente il progetto di tesi e come è stato risolto il problema del tracking delle mani. Nel terzo capitolo viene valutato il sistema testandolo con diverse persone e comportamenti, cercandone i limiti e robustezza. Infine l'ultimo capitolo è lasciato alle conclusioni dove vengono presentati possibili sviluppi futuri, portando idee su come poter migliorare il sistema.

Indice

Sommario	i
1 Sistemi Interattivi multimodali per l'apprendimento	1
1.1 Apprendimento interattivo	1
1.2 Enattività	3
1.3 Sistemi multimodali	4
1.4 Progetti esistenti	5
1.4.1 VoicePainter	5
1.4.2 The Ghost in the Cave	6
1.4.3 KidsRoom	7
1.5 Prototipo della Stanza Logo-Motoria	8
1.5.1 Resonant Memory	12
1.5.2 La Fiaba Magica	13
2 La Stanza Logo-Motoria	15
2.1 Stato attuale del Progetto	15
2.2 Tecnologie Utilizzate	16
2.2.1 OpenCV	16
2.2.2 Openframework	17
2.3 Tracking delle mani	18
2.3.1 Istogrammi	19
2.3.2 Calcolo del centro	20
2.3.3 Algoritmo per il tracking delle mani	22
2.4 Comunicazione Dati	26

2.4.1	Open Sound Control	27
2.4.2	Dati Trasmessi	28
2.5	PseudoCodice	28
3	Valutazioni	31
3.1	Solidità e robustezza.	31
3.2	Test con diverse corporature	32
3.3	Precisione in base alla posizione	34
3.4	Comportamento a braccia piegate	37
4	Conclusioni	43
	Bibliografia	45

Capitolo 1

Sistemi Interattivi multimodali per l'apprendimento

1.1 Apprendimento interattivo

Negli ultimi decenni, con la creazione di nuove tecnologie nell'ambito informatico, il tema dell'interattività si è spostato da semplice studio accademico a realtà di ogni giorno. L'interattività infatti è presente in molte delle nostre azioni quotidiane. La semplice navigazione web, con l'avvento del Web 2.0 e grazie a tecnologie come Flash, JavaScript, HTML5, è diventata una vera esperienza interattiva dove le pagine web, in base alle nostre azioni, si modificano in tempo reale dandoci una nuova esperienza web. Le tecnologie touchscreen utilizzate su tablet e smartphone sono veicoli di interattività che rendono più intuitivo e veloce la comunicazione uomo-macchina. Kinect e controller Wiimote hanno dato la possibilità di creare nuovi paradigmi dell'intrattenimento videoludico con l'esaltazione dell'interattività come vettore del divertimento. In ambito scolastico, oltre all'utilizzo di computer e tablet, numerosi sistemi interattivi sono stati progettati e introdotti nelle scuole. Ne presento alcuni:

Registro elettronico

Il registro elettronico è un software che permette di gestire la vita di classe, in particolare la registrazione di assenze, voti, argomento delle materie nelle varie ore, tutto in modo elettronico, senza alcun impiego di materiale cartaceo. Questo porta la possibilità di essere visionato in tempo reale da parte di studenti e genitori, migliorando quindi la comunicazione scuola famiglia e rendendo tutto più trasparente.

L.I.M.

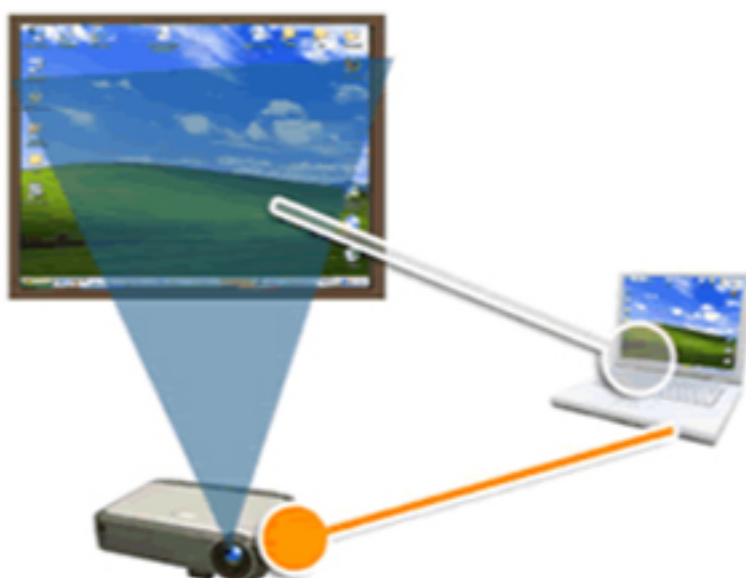


Figura 1.1: Schema di una L.I.M.

La LIM, acronimo di Lavagna Interattiva Multimediale, è un dispositivo elettronico grande come le tradizionali lavagne sulla cui superficie è possibile scrivere, gestire immagini, riprodurre video e navigare nel web. Il sistema è costituito da uno schermo, un proiettore e un computer collegati assieme come possiamo vedere in Figura [1.1]. La differenza principale tra un tradizionale proiettore è l'interattività che questa offre, infatti oltre a visualizzare

sullo schermo le azioni svolte a computer è possibile registrare, tramite touchscreen o pennarello elettronico, le cose scritte o disegnate sulla lavagna. La LIM permette di far entrare il multimediale in classe e di coinvolgere direttamente i ragazzi, pure mantenendo il classico paradigma della lezione frontale. Il vantaggio è quello di poter creare un nuovo stile di apprendimento, in linea con il linguaggio a cui le nuove generazioni sono ormai abituate, completamente sotto il controllo e la guida dell'insegnante.

Banchi Smart



Figura 1.2: SmartTable

Se con la LIM sostituiamo la lavagna, con i “Smart Table” andiamo a sostituire i tradizionali banchi di legno. Questo supporto permette all'alunno di lavorare in un ambiente digitale, con il vantaggio che gli insegnanti potranno caricare le attività direttamente sul banco dello studente. A causa del costo elevato questi dispositivi non sono molto diffusi ed esistono in piccole realtà in cui si è riuscito a trovare uno sponsor esterno.

1.2 Enattività

Tutte le tecnologie presentate nel paragrafo precedente permettono di conferire un grado di interattività alle lezioni, ma un aspetto sostanziale che tali tecnologie non possono fornire è l'utilizzo del movimento all'interno di



Figura 1.3: SmartTable in uso da dei bambini.

un ampio spazio fisico [7]. Di particolare interesse a questo proposito è il concetto di apprendimento enattivo. L'enazione è un termine introdotto dallo psicologo cognitivista Jerome Bruner per indicare uno dei modi possibili di organizzare la conoscenza, insieme alla modalità iconica e simbolica [8]. Le modalità iconica e simbolica si riferiscono rispettivamente ad un tipo di interazione con la realtà che avviene attraverso immagini (come diagrammi o illustrazioni) o attraverso simboli che richiedono di essere tradotti e interpretati (parole, formule matematiche...). L'apprendimento enattivo deriva invece dall'azione, la conoscenza viene costruita mediante l'interazione con il mondo attraverso attività motorie e che coinvolgono l'intera sfera sensoriale. Riconosciuta come la modalità attraverso cui viene costruita la conoscenza nelle prime fasi dello sviluppo cognitivo [9], l'enazione è la maniera più diretta, naturale e intuitiva di apprendimento. Basti pensare a un neonato, incapace di parlare o leggere, riesce comunque ad apprendere facendo. Non a caso i bambini amano manipolare oggetti e materiali, così come gironzolare e guardare, che sono tutte forme enattive di apprendimento.

1.3 Sistemi multimodali

Si parla di multimodalità quando un qualsiasi tipo di iterazione coinvolge più di un canale percettivo. Prendiamo ad esempio la televisione, essa uti-

lizza due canali quello uditivo e quello visivo per comunicare informazione. È facile capire il vantaggio di utilizzare sistemi multimodali: la banda di informazione trasmessa è più ampia di un sistema monomodale. Oltre a un semplice vantaggio di banda è da riconoscere una maggiore robustezza agli errori in quanto se uno dei canali ha dei problemi gli altri possono compensare. Sempre restando nell'esempio della televisione un telegiornale può essere ascoltato da non vedente, oppure guardato da un non udente. Interessante è valutare la capacità di apprendimento attraverso questi sistemi. Amthor arriva a quantificare l'apprendimento dichiarando che le persone memorizzano circa il 20% di quanto ascoltano, il 40% di quello che vedono e ascoltano e il 75% di quello che vedono, ascoltano e fanno [10]. Poichè la struttura cognitiva umana è formata da canali indipendenti con capacità limitata, la modalità di insegnamento che utilizza un solo canale percettivo può incorrere nel rischio di sovraccaricare la capacità cognitiva degli studenti, rispetto invece alla presentazione multimodale dei contenuti (sincronizzazione di molteplici canali sensoriali). Per questi motivi i sistemi multimodali sono indicati nell'utilizzo educativo.

1.4 Progetti esistenti

Presentiamo ora una serie di progetti che hanno cercato di abbracciare l'idea dell'apprendimento interattivo, enattivo attraverso l'utilizzo di sistemi multimodali.

1.4.1 VoicePainter

Il VoicePainter[3] è un sistema che permette di dipingere, su una tela virtuale, attraverso l'uso della propria voce e del movimento del corpo. Tale sistema trasforma lo spettatore della performance multimodale in un performer-attore, proponendo così una nuova metafora artistica.

Il sistema è costituito da un array di otto microfoni i cui segnali sono processati in tempo reale per la localizzazione ed il tracking della posizione

dell'utente. Il rendering grafico è retro-proiettato sullo schermo al fine di ottenere una perfetta corrispondenza tra la posizione della sorgente vocale e la posizione del rendering grafico da questa prodotto. In questo modo il feedback visivo è temporalmente e spazialmente correlato con l'evento sonoro. Dal segnale processato vengono inoltre estratti una serie di parametri indicativi di alcune caratteristiche notevoli del segnale vocale, che risultano mappati in altrettante caratteristiche del tratto grafico.

Oltre alla applicazione in ambiti artistici e di intrattenimento, il sistema apre possibili stadi di ricerca nella riabilitazione vocale.

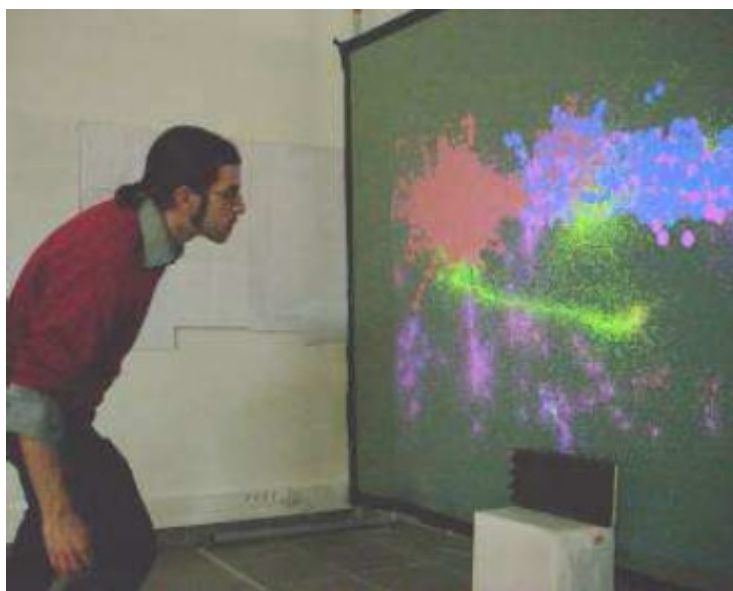


Figura 1.4: VoicePainter in azione.

1.4.2 The Ghost in the Cave

Ghost in the Cave[4] è un gioco interattivo collaborativo, nel quale due squadre di giocatori usano la loro voce e i movimenti del proprio corpo come input al sistema. il progetto propone due sfide alle squadre:

- Capire come utilizzare la voce e i propri movimenti per prendere il controllo dell'avatar e condurlo alla ricerca delle tre caverne.

- Far cambiare l'umore dell'avatar in base al fantasma trovato all'interno di ogni caverna.



Figura 1.5: The Ghost in the Cave

1.4.3 KidsRoom

The KidsRoom[6] è un spazio interattivo completamente automatizzato per bambini. Utilizzando immagini, luci e suoni una camera da letto viene trasformata in un mondo fantastico in cui poter giocare. Oggetti della stanza diventano personaggi di una avventura che partecipano attivamente alla storia guidando le scelte e le azioni del bambino.

The KidsRoom è costituita da quattro videocamere, sei computer, quattro casse, un microfono e due video-proiettori puntati su due diverse pareti della stanza. La posizione e le azioni dei bambini vengono registrate e usate come input per la narrazione. Tecniche di computer grafica vengono utilizzate per modificare il contenuto della stanza in base alle reazioni dei bambini.



Figura 1.6: Esterno dell'ambiente interattivo multimediale KidsRoom

1.5 Prototipo della Stanza Logo-Motoria

Tutti i sistemi multimodali interattivi presentati nel capitolo precedente sono accomunati da un costo elevato e dall'utilizzo di tecnologie complesse, che richiedono per la loro gestione personale informatico e tecnico specializzato. Spesso costituiti da apparecchiature ingombranti e difficili da installare, il loro utilizzo è stato sempre ridotto dentro confini limitati di tempo (nel corso di una visita o un progetto speciale) e di spazio (presso il laboratorio/istituzione dove l'ambiente era installato). La certezza che l'adozione di tali tecnologie sarebbe davvero efficace se potesse contare su una maggiore diffusione e un utilizzo continuativo nelle scuole, affiancato alle ordinarie attività didattiche, ha portato alla progettazione e sviluppo del sistema Stanza Logo-Motoria.

Tali premesse sono alla base delle seguenti linee guida adottate per la sua progettazione.

- Basso costo della tecnologia utilizzata (computer, sensori, interfacce...).
Necessario affinché il sistema possa essere adottabile in scuole di ogni



Figura 1.7: La Stanza Logo-Motoria installata in un'aula della scuola "E. Frinta" di Gorizia

ordine e grado, a prescindere da differenze economiche.

- Usabilità del sistema. Con l'obiettivo di ottenere la maggiore diffusione possibile del sistema Stanza Logo-Motoria è indispensabile che gli insegnanti siano in grado di gestire/utilizzare il sistema senza difficoltà, pur non possedendo competenze avanzate di informatica.
- Customizzabilità del sistema. Necessario affinché gli insegnanti siano in grado di adattare e produrre autonomamente i contenuti delle attività didattiche da svolgere nella Stanza Logo-Motoria.
- Robustezza del sistema. In maniera da poter funzionare in un ampio spettro di condizioni, indipendentemente dall'ambiente in cui viene installato. Questo requisito è necessario per ridurre la necessità da parte della scuola di richiedere l'intervento di tecnici per la manutenzione del sistema e dover per questo investire ulteriori risorse economiche.

Sulla base di queste linee guida il Centro di Sonologia Computazionale del Dip. di Ingegneria di Padova e Serena Zanolla, insegnante di sostegno e dottoranda di ricerca dell'università di Udine, hanno realizzato una versione della Stanza Logo-Motoria, che attualmente è installata presso la Scuola Primaria "E. Frinta" di Gorizia (Istituto Comprensivo Gorizia). Questa versione utilizza una webcam come dispositivo di acquisizione e l'ambiente di programmazione grafico EyesWeb XMI per l'elaborazione dei dati e per la gestione delle attività interattive. La webcam posizionata al centro del soffitto riprende il movimento degli utenti su una superficie quadrata. Degli algoritmi di motion-capture elaborano le immagini e restituiscono il baricentro delle persone che si muovono nello spazio.

Il progetto è stato pensato in modo tale che possa essere ampliato con l'aggiunta di nuove funzionalità. Per questo motivo è stato strutturato gerarchicamente in tre livelli:

- **Piattaforma.** Ovvero i moduli hardware (videocamera o altri dispositivi di acquisizione, unità di elaborazione, diffusori, proiettore...) e software (programmi finalizzati all'interpretazione dei dati in input derivanti dai dispositivi di acquisizione, alla rielaborazione e alla trasmissione delle informazioni sul movimento, alla gestione dei dispositivi di riproduzione audio e video e alla costituzione di un host per le applicazioni) che costituiscono l'infrastruttura necessaria al funzionamento del sistema.
- **Applicazioni.** Pacchetti software, compatibili con la piattaforma, che implementano le differenti attività interattive che possono essere svolte con la Stanza Logo-Motoria.
- **Contenuti.** Unità di apprendimento multimediali suddivise per ambito disciplinare e pensate per ogni diversa applicazione della Stanza Logo-Motoria[1].

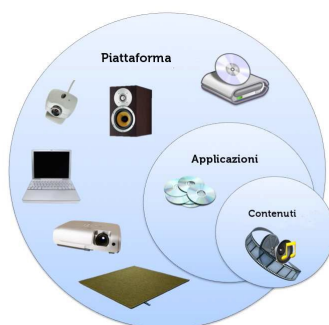


Figura 1.9: I tre livelli del sistema Stanza Logo-Motoria.

1.5.1 Resonant Memory

La prima applicazione realizzata per la Stanza Logo-Motoria è *Resonant Memory*. La stanza viene logicamente suddivisa in nove regioni, una centrale e otto periferiche, alle quali è associato un suono. L'applicazione inoltre è divisa in due fasi diverse tra loro. Nella prima fase che possiamo definire di "esplorazione" l'utente è invitato a posizionarsi su ognuna delle otto regioni periferiche per ascoltare e memorizzare il suono, associato alla posizione. Una volta aver attivato almeno una volta tutte le regioni esterne, si passa alla seconda fase in cui viene abilitata la regione centrale. L'utente infatti posizionandosi al centro della stanza farà iniziare la riproduzione di una storia sotto forma di narrazione audio. Durante la narrazione la storia farà riferimento ad oggetti (luoghi, animali, concetti...), a cui è possibile associare il suono di una ben determinata regione scoperta nella prima fase. L'allievo è quindi incoraggiato a muoversi nella stanza reagendo alla storia, facendogli attivare i vari effetti sonori delle otto regioni periferiche, creando quindi una vera e propria colonna sonora al racconto. Il materiale didattico presentato attraverso questa attività interattiva è più facilmente memorizzabile in quanto vengono stimolati contemporaneamente varie regioni del cervello come l'ascolto, la memorizzazione e il movimento.

1.5.2 La Fiaba Magica

Una seconda applicazione realizzata per la Stanza Logo-Motoria è *La Fiaba Magica*[2]. La Fiaba Magica è un'interfaccia interattiva multimodale nella quale l'utente ha la possibilità di essere il "regista" della storia di *Capucetto Rosso*, decidendo come e quando far continuare la storia. Questa volta la stanza è divisa in tre regioni, il bambino entrando nella prima, quella più distante dal proiettore/monitor, fa partire la riproduzione audio della prima porzione di storia che coincide con la visualizzazione a schermo dei due protagonisti della storia. Una volta ascoltata la porzione di storia, il bambino ha la possibilità di interagire con gli avatar proposti allo schermo semplicemente muovendo il braccio corrispondente al personaggio da animare. Una volta animati almeno una volta entrambi i personaggi, il bambino può accedere alla zona successiva il cui comportamento, fino all'ultima parte, è analogo a quello appena illustrato.

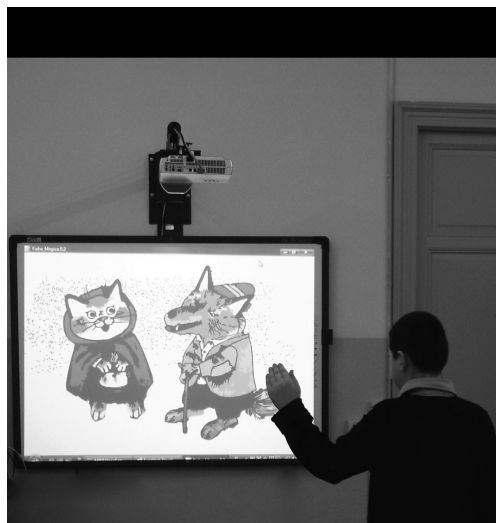


Figura 1.10: FiabaMagica in azione.

Capitolo 2

La Stanza Logo-Motoria

Nel capitolo precedente abbiamo riportato il progetto originale della Stanza Logo-Motoria sviluppato con la piattaforma EyesWeb XMI. Per quanto il progetto sia già funzionante allo stato attuale non è facilmente utilizzabile e configurabile dall'utenza a cui effettivamente il sistema è pensato. EyesWeb XMI ha una interfaccia di programmazione a blocchi ma, per quanto sia veloce e facile da utilizzare nel momento della creazione del prototipo, non è possibile presupporre che un qualsiasi insegnante sia in grado di utilizzarlo. Per questi motivi si è pensato di scrivere un'applicazione più user-friendly avente tutte le funzionalità del progetto iniziale, assicurandosi la possibilità di poter essere effettivamente liberi di personalizzare qualunque aspetto dell'applicazione in quanto scritta da noi.

2.1 Stato attuale del Progetto

Attualmente il nostro software è in grado di eseguire l'applicazione *Resonant Memory*.

Dal punto di vista algoritmico, riguardante la computer vision, è stato risolto il problema della sottrazione dello sfondo, di conseguenza siamo in grado di ricavare dal flusso video della webcam una silhouette della persona nella stanza e quindi individuarne la posizione. Questo è sufficiente per far fun-

zionare la logica che sta sotto alla *Resonant Memory*.

Il progetto prevede inoltre una GUI già sviluppata e funzionale con la possibilità di caricare nuovi contenuti tramite un file XML[17].

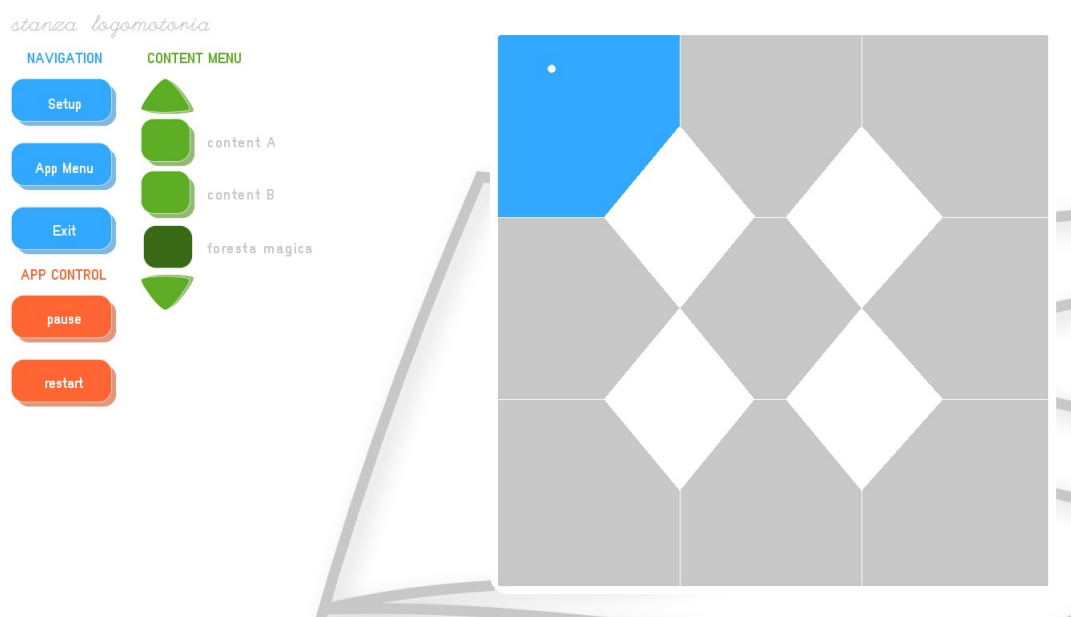


Figura 2.1: L'attuale GUI della Stanza Logo-Motoria

2.2 Tecnologie Utilizzate

Il progetto è scritto in C++, utilizzando OpenFramework 0.8.0 e OpenCV come libreria di computer vision. Per tenere il progetto il più possibile open e multiplatforma come IDE abbiamo usato CodeBlocks molto famoso per la programmazione C.

2.2.1 OpenCV

OpenCV[12] è una libreria open source dedicata alla computer vision il cui sviluppo iniziale risale al 1999, è scritta in C/C++ e supporta i principali sistemi operativi (Linux, Windows e Mac OS X). Implementata con l'idea di essere computazionalmente efficiente essa mira principalmente alle

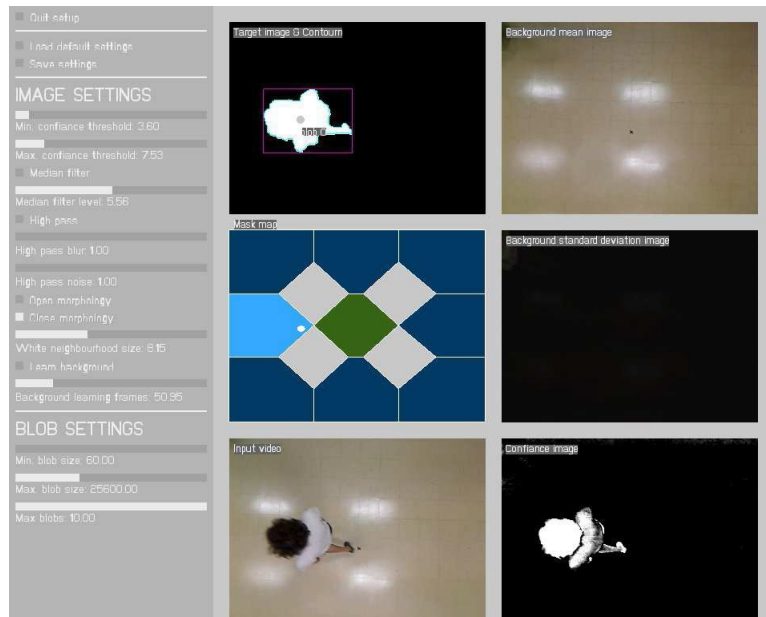


Figura 2.2: Vista Setup della Stanza Logo-Motoria

applicazioni realtime perciò è scritta in linguaggio C ottimizzato e sfrutta hardware dotato di più core. Uno degli obiettivi della libreria è quello di fornire una struttura per la computer vision semplice che possa aiutare gli sviluppatori a scrivere applicativi relativamente complicati in modo veloce. OpenCV contiene moltissime funzioni che spaziano da quelle per uso medicale alla robotica, inoltre è presente una completa libreria per l'apprendimento automatico (Machine Learning Library - MLL).

2.2.2 Openframework

OpenFramework è un toolkit open source, scritto in C++, designato per *creative coding*. Pensato per creare applicazioni audio e grafiche, questo framework permette di unire la potenzialità del linguaggio C++ con la semplicità di utilizzo, riunendo in esso tutte le librerie necessarie per questi tipi di applicazioni. Oltre ad essere open source è anche progettato per essere multiplatforma, infatti è possibile usarlo in ambiente Windows, Mac OS X, Linux, iOS e Android.

La creazione di questo progetto nasce dall'esigenza di Zach Lieberman, professore alla Parson School of Design, di condividere con i propri studenti la libreria che utilizzava per i propri lavori. Questa libreria, sviluppata al MIT Media Lab's Aesthetics and Computation Group, aveva però due grossi problemi, oltre a non essere mantenuta (grosso problema in informatica), aveva una licenza non open. Per questi motivi Zach con l'aiuto di un suo ex-studente Theo Watson iniziarono a costruire una libreria simile ma con licenza open source (licenza MIT), per poi renderla pubblica [14].

Oltre alle proprie funzioni, tramite openframework è possibile usare le librerie comunemente conosciute in questi ambiti:

- OpenGL, GLEW, GLUT, libtess2 e cairo per la grafica
- rtAudio, PortAudio, OpenAL e Kiss FFT o FMOD per audio input, output e analisi
- FreeType per i caratteri
- FreeImage per caricamento/salvataggio di immagini
- Quicktime, GStreamer e videoInput per riproduzione video e registrazione
- Poco per le utility
- OpenCV per la visione artificiale
- Assimp per il caricamento di modelli 3D

Quindi ancora una volta semplicità d'uso senza avere limitazioni di potenzialità grazie a queste librerie con licenza analoga a quella del framework [13].

2.3 Tracking delle mani

Arriviamo ora al punto focale della tesi, partendo dalla silhouette precedentemente elaborata, dobbiamo riconoscere la posizione delle mani. In

circolazione ci sono parecchi algoritmi/librerie che riconoscono ed eseguono il tracking delle mani o addirittura che creano l'intero scheletro della figura umana, il problema è che partono tutti da un presupposto che noi non abbiamo: la ripresa frontale.

L'esempio più semplice da portare è quello del kinect, dispositivo dotato di una telecamera e doppio sensore di profondità a infrarossi, che riesce a riconoscere una o più persone davanti al dispositivo tracciandone l'intero scheletro. Nel nostro caso il video viene ripreso dall'alto da una webcam quindi sicuramente questi algoritmi non possono essere utilizzati.

Per questi motivi nasce l'esigenza di scrivere un algoritmo ad hoc per l'applicazione, sfruttando le poche informazioni che abbiamo a disposizione.

2.3.1 Istogrammi

L'istogramma è un tipo di rappresentazione compatta molto utile per l'elaborazione delle immagini. In generale, un istogramma rappresenta graficamente la distribuzione dei dati. È un grafico bidimensionale in cui sull'asse delle ascisse troviamo le classi dei dati, mentre sulle ordinate i valori di frequenza. Un esempio pratico nel caso dell'elaborazione di immagini si ha se prendiamo un'immagine in scala di grigio: il suo istogramma dei valori di grigio avrà sulle X i valori di intensità da 0 a 255, e sulle Y saranno rappresentati le frequenze di questi valori all'interno dell'immagine. Avremo così una raffigurazione intuitiva della distribuzione dei valori di grigio dell'immagine.

L'utilizzo in questo progetto degli istogrammi sarà leggermente diversa, in quanto verranno impiegati per valutare la posizione X/Y dell'utente e delle sue mani partendo da una silhouette in bianco e nero precedentemente elaborata. Dalla silhouette verranno generalmente calcolati due istogrammi, uno delle frequenze dei pixel bianchi per colonna e l'altro per le righe. Valutando questi istogrammi (altezza massima, area) verranno ricavati informazioni su dove, nello spazio X/Y, effettivamente sono presenti le feature cercate[18].

2.3.2 Calcolo del centro

Nell'attuale implementazione della Stanza Logo-Motoria è già presente un metodo che ritorna il baricentro della silhouette. Questo metodo era stato implementato per far funzionare l'applicazione *Resonant Memory*, l'algoritmo semplicemente partendo dal Blob ne calcola il bounding box, cioè il rettangolo che include la silhouette, e di questo ne calcola il centro. Per quanto questo metodo risulti efficace in quel contesto, non è sufficientemente preciso per il nostro utilizzo. Quello di cui abbiamo bisogno è la posizione della testa o comunque del busto visto dall'alto.

Per capire meglio il problema basta guardare la Figura [2.3]. In bianco abbia-

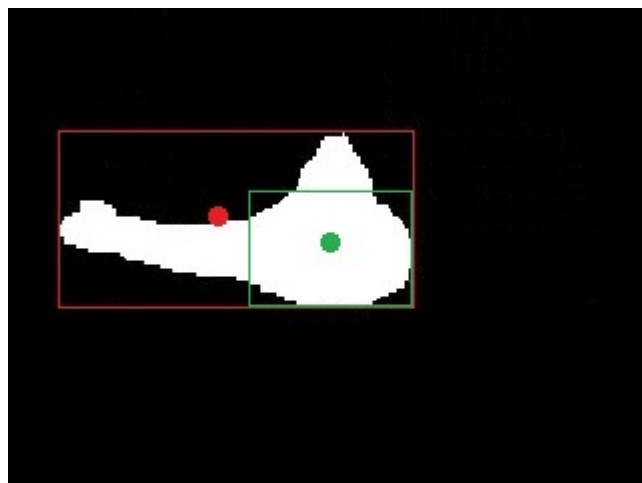


Figura 2.3: Esempio di silhouette. In rosso il bounding box e il centro trovato con esso, in verde il centro desiderato.

mo la silhouette della persona all'interno della stanza, con il braccio sinistro in posizione orizzontale. In rosso possiamo vedere la bounding box e il suo centro. Per quanto il centro di colore rosso possa essere considerato valido per individuare la posizione della persona sulla stanza, non ci dà nessuna informazione utile per individuare le braccia. Quello che noi cerchiamo è il punto verde, il centro del busto della persona. Individuando quel centro possiamo poi presupporre le braccia partire da esso in qualunque direzione. Utilizzando gli istogrammi e rangionando sulle proporzioni del corpo umano

è possibile arrivare facilmente a un metodo per individuare il busto. Per prima cosa creiamo gli istogrammi delle frequenze verticali e orizzontali come mostrato in Figura [2.4]. Grazie a OpenFramework questa operazione è



Figura 2.4: Istogrammi delle frequenze verticali e orizzontali della silhouette molto semplice in quanto, grazie al metodo *getPixels()*, è possibile ottenere un array di pixels ordinati in row major di un'immagine, dell'attuale frame di un video o, in questo caso, dell'immagine catturata dalla webcam. Dai pixel, considerandone il colore, si riempiono i due array che rappresenteranno gli istogrammi delle frequenze. Ora non dobbiamo far altro che riuscire, partendo da questi istogrammi, delimitare un bounding box relativo al busto della persona. Per fare questo calcoliamo l'area totale dell'istogramma e cerchiamo due indici x_1 , x_2 tali che:

- $x_2 > x_1$
- L'area dell'istogramma compreso tra x_1 , x_2 sia almeno il 60% dell'area totale.

- La differenza $x_2 - x_1$ sia minima

Trovati gli estremi del busto trovarne il centro è immediato $(x_1 + x_2)/2$, per una maggiore robustezza e per eliminare certi artefizi della silhouette ho preferito ampliare le dimensioni del busto trovato di un 30%. Anche se questo porta a includere nel busto parte del braccio non risulta in un problema in quanto il nostro fine è quello di individuare le mani.

2.3.3 Algoritmo per il tracking delle mani

Dopo questo preambolo su come individuare il centro, abbiamo tutti i dati e gli strumenti per passare al vero e proprio algoritmo di tracking. La prima operazione da fare è quella di creare dei nuovi istogrammi delle frequenze verticali e orizzontali escludendo dalla silhouette il bounding box precedentemente calcolato del busto. Questo è facilmente ottenibile in quanto basta, nel momento in cui si scorre l'array di pixel, saltare i punti le cui coordinate X/Y (considerando ovviamente l'array come una matrice) soddisfano queste condizioni contemporaneamente:

- $(X > \text{CoordinataCentroX} - \text{DimensioneBustoX}/2)$
- $(X < \text{CoordinataCentroX} + \text{DimensioneBustoX}/2)$
- $(Y > \text{CoordinataCentroY} - \text{DimensioneBustoY}/2)$
- $(Y < \text{CoordinataCentroY} + \text{DimensioneBustoY}/2)$

Per meglio capire come funziona l'algoritmo conviene spiegarlo ad esempi partendo da quello più semplice fino ad arrivare al caso più complesso.

Un braccio obliquo

Esaminiamo la Figura [2.5], in blu possiamo vedere gli istogrammi ricavati con l'algoritmo precedente, ora bisogna valutarli per estrapolare da essi una coordinata X/Y che identifichi la posizione della mano sullo spazio.

Per comodità lavoriamo prima sull'asse X, scorriamo l'array dell'istogramma

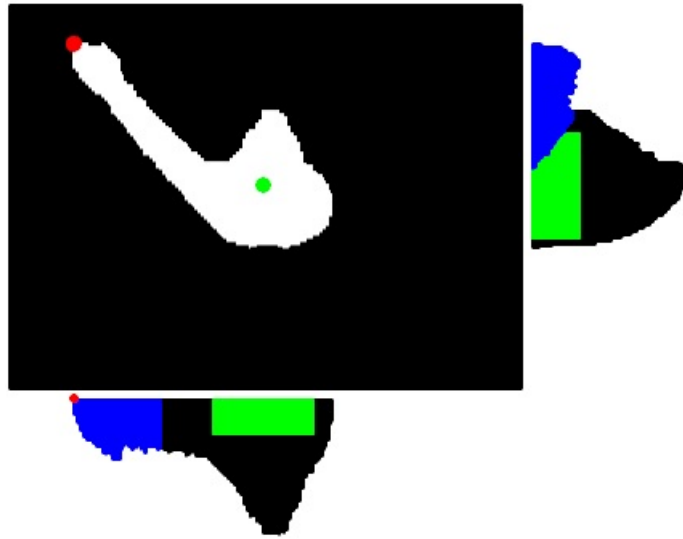


Figura 2.5: Modalità debug dell'applicazione, un solo braccio obliquo.

blu fino a che non troviamo un valore diverso da zero salvandoci questo indice che chiameremo x_1 . Ora proseguiamo fino a che non troviamo un valore uguale a zero, l'indice precedente a questo lo salveremo e lo chiameremo x_2 . Facendo in questo modo avremo in x_1 e x_2 gli estremi dell'istogramma che non sono altro che l'inizio e la fine del braccio. Per capire quale di questi due indici rappresenta la coordinata X della mano, non facciamo altro che servirci della posizione del centro del busto precedentemente calcolata. La coordinata della mano sarà quella più distante dal centro del corpo. Si procede analogamente per la coordinata Y.

Sempre in Figura [2.5] il punto rosso è la posizione della mano trovata in questo modo che come possiamo vedere è corretta.

Un braccio parallelo ad un'asse

Osserviamo la Figura [2.6] in questo caso il braccio è parallelo all'asse delle X, in questo caso dovremmo portare una piccola modifica all'algoritmo. Per quanto riguarda la coordinata X possiamo, senza nessun problema, lavorare in modo analogo al caso precedentemente spiegato. Per la coordinata Y

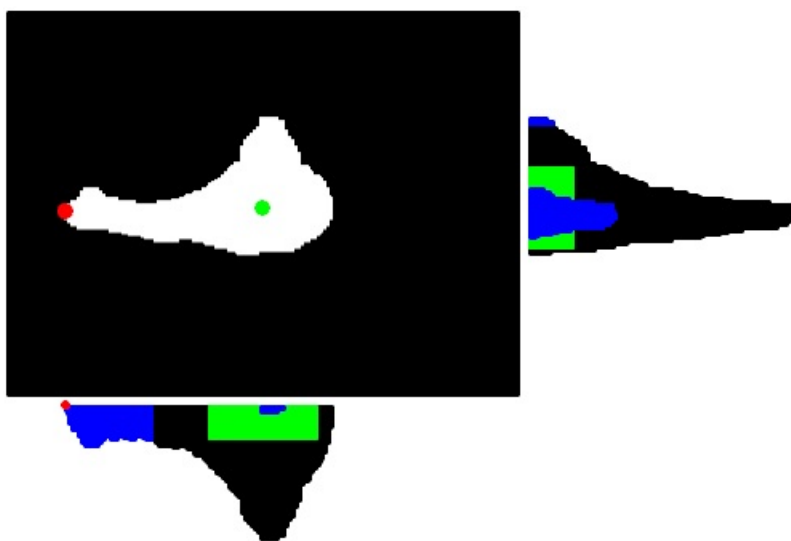


Figura 2.6: Modalità debug dell'applicazione, un solo braccio parallelo a X.

invece dobbiamo modificare l'algoritmo. Come possiamo notare l'istogramma in questo caso presenta una forma diversa, si estende più in altezza che in larghezza. Questo è dovuto al fatto che il braccio si trova in una posizione perpendicolare all'asse di cui cerchiamo la coordinata. La posizione della mano in questo caso non è più data dall'estremo più distante dal centro ma dalla coordinata della massima altezza dell'istogramma. Alla luce di questo possiamo modificare l'algoritmo in modo tale che nel mentre scorre nel cercare x_1 e x_2 memorizzi anche la posizione e il valore del picco massimo. Avendo questi dati possiamo subito riconoscere i due casi:

- Se $(x_2 - x_1) < AltezzaMassima \Rightarrow$ Prendiamo la posizione del picco massimo.
- Se $(x_2 - x_1) > AltezzaMassima \Rightarrow$ Prendiamo tra x_1 e x_2 il più distante dal centro.

Come possiamo vedere anche questa volta con il punto rosso è riportata la posizione trovata dal sistema che anche questa volta risulta precisa.

Due braccia



Figura 2.7: Modalità debug dell'applicazione, due braccia

In Figura [2.7] siamo di fronte al caso in cui l'utente muove contemporaneamente le due braccia. Per quanto utilizzando gli algoritmi precedenti effettivamente troviamo le due coordinate X e le due coordinate Y delle due mani, ci troviamo di fronte ad un problema di ambiguità. Non riusciamo a collegare in modo univoco una coordinata X con la propria coordinata Y. Anche se, ingannati da questo esempio, un metodo per risolvere questa ambiguità potrebbe essere quello di collegare a un istogramma sviluppato in altezza sulle asse delle X con uno sviluppato in larghezza sulle Y e viceversa, non è corretto. Infatti se le braccia sono in posizioni oblique come abbiamo visto nella Figura [2.5] gli istogrammi risultanti sono su tutti e due gli assi sviluppati in larghezza. Siamo costretti quindi per risolvere questo problema a costruire un nuovo istogramma della coordinata opposta per ogni braccio che individuiamo. Per essere più precisi, sempre per comodità partiamo dall'asse X, individuiamo il primo braccio trovando x_1 e x_2 e la sua relativa

coordinata seguendo gli algoritmi visti in precedenza. A questo punto creiamo un nuovo istogramma Y saltando le coordinate che soddisfano oltre alle solite condizioni sulla bounding box del busto anche le seguenti:

- $X < x_1$
- $X > x_2$

In questo modo nel nuovo istogramma saranno presenti le frequenze relative solo al braccio che stiamo cercando di rilevare in questo momento eliminando l'ambiguità'.

Pulizia Istogramma

Prima di dare in pasto all'algoritmo gli istogrammi è opportuno farne una pulizia da eventuale rumore o falsi positivi generati da posizioni strane del corpo. Osserviamo ad esempio la Figura [2.6] in cui sia nell'asse X che nell'asse Y è possibile notare dei pezzi di istogramma generato dalla testa dell'utente sporgente verso l'alto. Se non eliminiamo quelle imperfezioni il sistema riconoscerà quella testa sporgente come un vero e proprio braccio. Possiamo quindi, per risolvere questo problema, a calcolare l'area totale occupata dall'istogramma intero e da tutti i suoi pezzi. A questo punto se uno dei pezzi ha un'area inferiore al 10% dell'area totale lo ignoriamo.

2.4 Comunicazione Dati

Il progetto da me realizzato al momento non è integrato direttamente con il sistema principale, per questo motivo ho predisposto un metodo per comunicare i risultati della mia computazione verso l'esterno. Questo, anche una volta integrato al progetto Stanza-Logomotoria, è comunque una feature da conservare in quanto avere la possibilità di poter collegare la propria applicazione ad altre è un comportamento sicuramente gradito. Per essere utile e utilizzabile la comunicazione deve seguire degli standard riconosciuti, per questo motivo utilizziamo Open Sound Control.

2.4.1 Open Sound Control

Open Sound Control è un protocollo di comunicazioni per computer, sintetizzatori sonori e altri dispositivi multimediali attraverso una rete ethernet. Prendendo i benefici delle moderne tecnologie di rete e del mondo degli strumenti musicali, riesce a ad essere un protocollo accurato, flessibile, multifunzionale, organizzato e documentato. OSC è stato sviluppato al CNMAT da Adrian Freed e Matt Wright con lo scopo di condividere dati relativi a una performance musicale tra strumenti musicali, computer e altre periferiche multimediali. Viene spesso utilizzato come alternativa allo standard MIDI, quando c'è bisogno di un quantitativo di parametri musicali maggiori. I messaggi OSC vengono trasmessi usando per la maggior protocolli ethernet e UDP/IP dando agli utilizzatori una grossa flessibilità sui tipi di data da inviare, in modo da permettere le varie applicazioni di comunicare ad alto livello. Le caratteristiche principali di questo formato sono:

- Uno schema dinamico dei nomi in stile URL
- Dati numerici di tipo simbolici e ad alta risoluzione
- Un linguaggio che permette di specificare più destinatari per un messaggio
- Delle variabili per la gestione delle tempistiche molto precise
- Possibilità di creare agglomerati di messaggi da inviare contemporaneamente.

Ci sono diverse implementazioni di OSC per i vari settori in cui è utilizzato come nei sistemi sonori real-time, sintetizzatori software, linguaggi di programmazione, periferiche hardware e nei strumenti che gestiscono le interazioni nel web.

2.4.2 Dati Trasmessi

Grazie a OpenFrameworks e un suo Add-ons (ofxOsc) il compito di inviare dati è notevolmente semplificato. Per prima cosa, dopo aver incluso l'Add-ons al progetto, dobbiamo dichiarare una classe di tipo *ofxOscSender* nel programma per poi inzializzarla attraverso il metodo *setup()* in cui dobbiamo specificare l'Host e la porta a cui mandare il messaggio. I pacchetti di informazioni che spediremo saranno quelli relativi alle mani quindi come URL utilizzeremo */hand1 /hand2* mentre come dati per la posizione due numeri di tipo Float contenenti la coordinata X e la coordinata Y normalizzate all'unità.

In seguito il pezzo di codice che crea il messaggio:

```
ofxOscMessage m;  
m.setAddress("/hand1");  
m.addFloatArg((float)(handXValue/VIDEO_WIDTH));  
m.addFloatArg((float)(handYValue/VIDEO_HEIGHT));  
sender.sendMessage(m);
```

2.5 PseudoCodice

Combinando assieme quello scritto nei paragrafi precedenti possiamo arrivare a scrivere uno pseudocodice di come l'algoritmo si comporta alla fine tenendo conto delle varie possibilità:

Crea istogramma frequenze X,Y

Pulisci istogrammi

Calcola il centro e la dimensione del busto

Crea dei nuovi istogrammi frequenze X,Y ignorando la parte del busto

Pulisci istogrammi

Calcola il numero di pezzi di istogrammi X,Y

Seleziona l'istogramma con più pezzi

Per ogni pezzo di istogramma

Calcolane inizio, fine, posizione del valore massimo

Crea un nuovo istogramma della coordinata opposta ignorando il busto
e le zone esterne a inizio, fine

Del nuovo istogramma calcolane inizio, fine, posizione del valore massimo

Calcola le posizioni della mano

se $\text{fine} - \text{inizio} < \text{PosMax}$ posizione = PosMax

altrimenti posizione = Il più distante dal centro tra Inizio, Fine

Invia via OSC la posizione della mano.

Capitolo 3

Valutazioni

In questo capitolo cercheremo di valutare il sistema, mostrando come si comporta in diverse situazioni, cercandone i limiti e proponendo soluzioni per ottimizzare l'utilizzo del sistema. La maggior parte di questi test si baseranno sull'utilizzare persone differenti e cercare posizioni in cui il sistema potrebbe presentare delle difficoltà. A priori abbiamo scartato l'idea di controllare il comportamento dell'algoritmo in base alla velocità di movimento dell'utente in quanto il sistema lavora frame su frame ignorandone quindi l'history.

3.1 Solidità e robustezza.

Grazie alle varie pulizie effettuate sugli istogrammi il sistema è resistente a tutte le imperfezioni della silhouette, ignorando le aree e le frequenze sotto a una certa soglia. Piccole macchie o linee sul video dopo la sottrazione dello sfondo vengono totalmente ignorate dall'algoritmo, iniziano a creare problemi solo imperfezioni della grandezza di una mano e solo se posizionate in certe posizioni. Errori di tale grandezza implicano un problema a monte, come stanza male illuminata o con sfondo troppo variabile per avere un video soddisfacente. In seguito riportiamo il comportamento del sistema utilizzando persone di diverse corporature in posizioni diverse, arrivando infine ad osservare l'algoritmo modificandone le ipotesi di partenza.

3.2 Test con diverse corporature

Abbiamo testato l'algoritmo con diverse persone di corporatura differenti per valutarne la precisione. Quello che abbiamo notato è una indifferenza da parte dell'algoritmo rispetto alle corporature, questo era prevedibile in quanto l'algoritmo tiene conto dell'area della silhouette e quindi lavora in percentuali di area per determinare braccia e busto. Queste percentuali, indipendentemente dalla corporatura, rimangono simili tra persona a persona. Se questo è quello che possiamo dire dell'algoritmo, esiste un problema però tecnico dovuto alla costruzione hardware della stanza. Aumentando l'altezza della persona a cui vogliamo individuare le mani ci troviamo di fronte al problema che la webcam non è in grado di riprendere l'intero corpo, tagliando effettivamente le mani o parte del braccio. In questi casi l'algoritmo offrirà comunque una posizione della mano individuandola a bordo camera. Il problema in questo caso non è risolvibile via software in quanto è il sistema che non riesce a darci le informazioni necessarie per continuare, non possiamo individuare quello che non possiamo vedere.

Il problema, visto l'utilizzo della stanza, non dovrebbe presentarsi in quanto l'utilizzo della Stanza Logo-Motoria è pensato per dei bambini quindi l'altezza dell'utente non sarà tale da infastidire il sistema. Alla luce di questo possiamo consigliare di installare la Stanza-Logomotoria se possibile in una stanza con un soffitto alto in modo da mitigare questo comportamento.

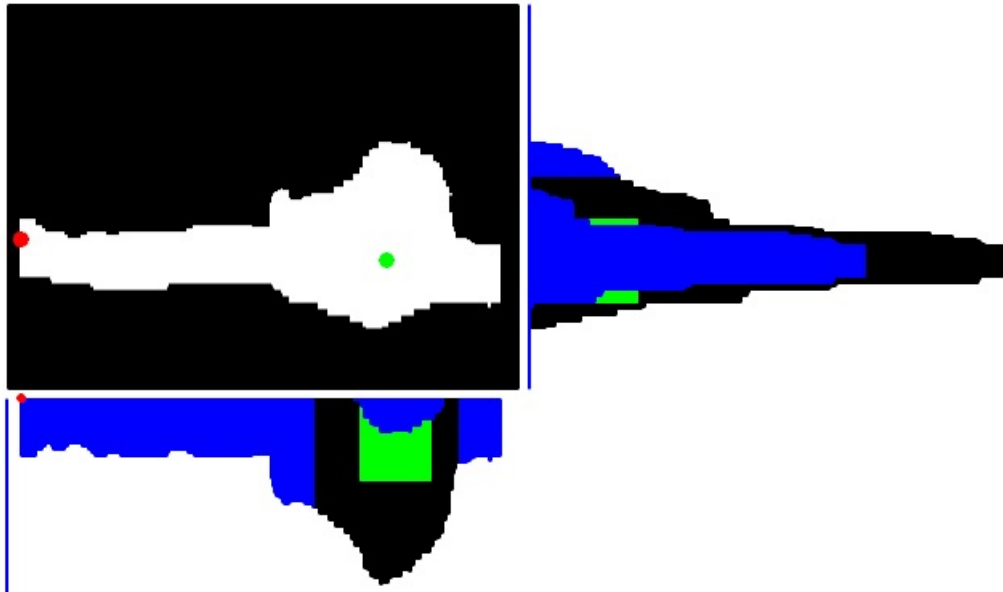


Figura 3.1: Modalità debug dell'applicazione, Utente altezza 1.90m.

In Figura [3.1] vediamo il comportamento dell'algoritmo con una persona dell'altezza di circa 1,90 metri. Il ragazzo ha tutte e due le braccia in posizione orizzontale, la mano sinistra anche se tagliata e non totalmente visibile viene comunque rilevata. Anche se la posizione se vogliamo essere precisi non è quella effettiva per la logica di qualunque applicazione è comunque il comportamento desiderato dal sistema. Per la mano destra invece non è possibile ricavarne una posizione in quanto in questo caso l'intero braccio esce dal campo visivo della camera.

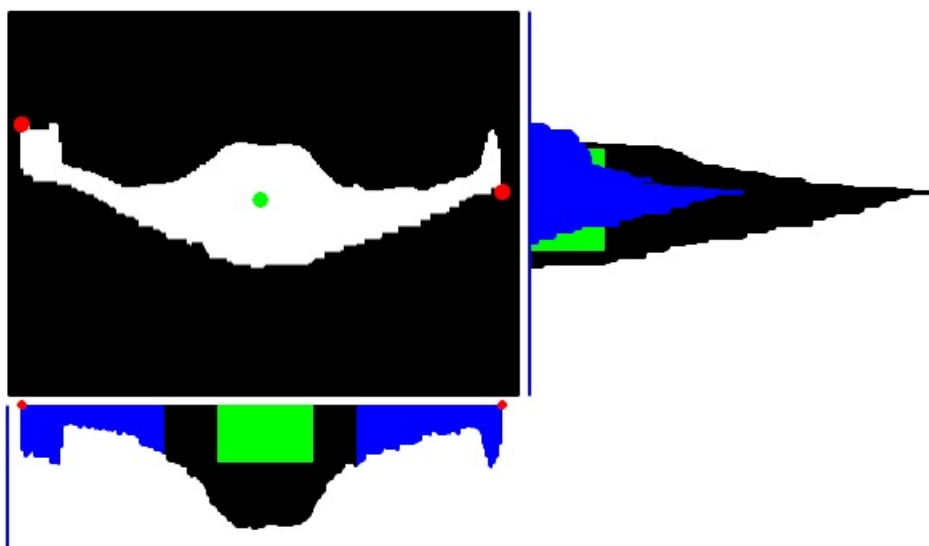


Figura 3.2: Modalità debug dell'applicazione, Utente altezza 1.60m.

In Figura [3.2] l'utente in questione è una donna dall'altezza di 1,60 metri. Come possiamo osservare in questo caso non sussistono problemi nella rilevazione in quanto l'intera silhouette è contenuta nella ripresa.

3.3 Precisione in base alla posizione

Per come è progettato il sistema la posizione dell'utente influisce in modo sostanziale sulla precisione del sistema. L'algoritmo è progettato prendendo come assunzione una ripresa dall'alto perpendicolare al corpo da osservare. Questo è però vero solo quando il corpo si trova esattamente al centro della stanza, in altri punti la ripresa non è più perpendicolare ma è inclinata di un angolo sempre più ampio man mano che ci allontaniamo dal centro. L'inclinazione della camera porta a una deformazione dell'ideale visione del busto, in quanto iniziano ad apparire altri particolari del corpo come le gambe e una distinzione tra busto e testa. Oltre a questo problema, spostandoci dal centro, la probabilità che la mano esca dal campo visivo della telecamera aumenta rendendo insensata la ricerca.



Figura 3.3: Modalità debug dell'applicazione, Utente lontano dal centro.

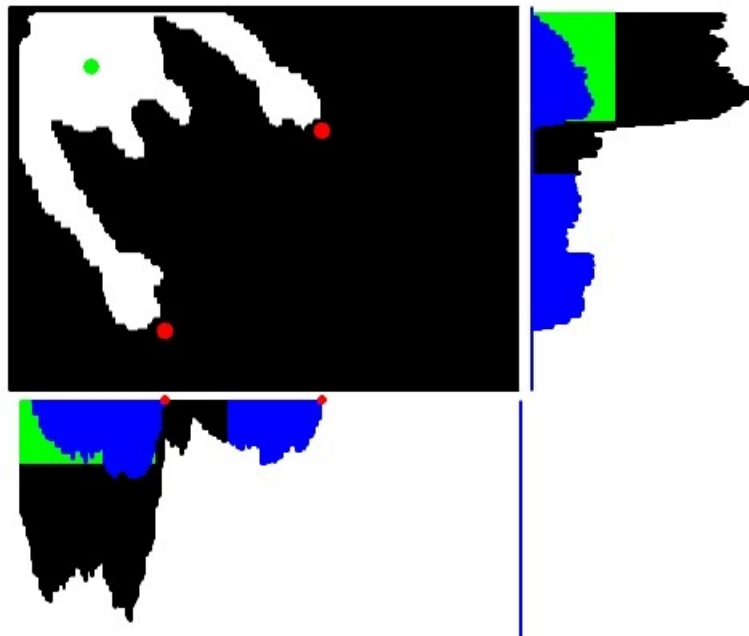


Figura 3.4: Modalità debug dell'applicazione, Utente lontano dal centro.

Nelle Figure [3.3][3.4] vediamo come il sistema riesce a gestire e individuare le mani anche se l'utente si trova in una posizione sfavorevole e cioè lontano dal centro della stanza.

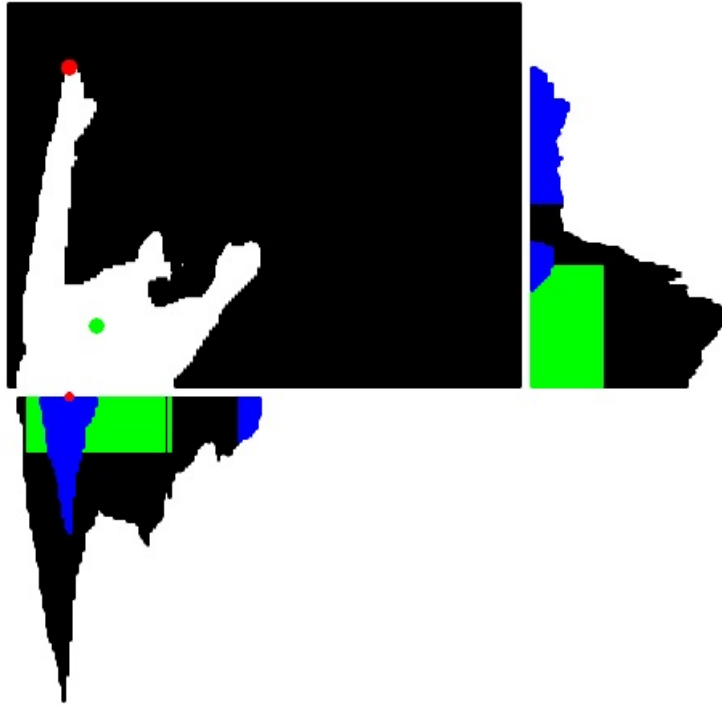


Figura 3.5: Modalità debug dell'applicazione, Utente lontano dal centro errore.

In Figura [3.5] presentiamo un caso in cui l'algoritmo non riesce a rilevare una delle due mani. Il sistema a causa della piccola area associata a quella mano, la ignora considerandola rumore.

3.4 Comportamento a braccia piegate

Anche se l'algoritmo è stato progettato per individuare le mani da un corpo con braccia distese, è interessante osservare come il sistema si comporta quando l'utente tiene le braccia piegate.

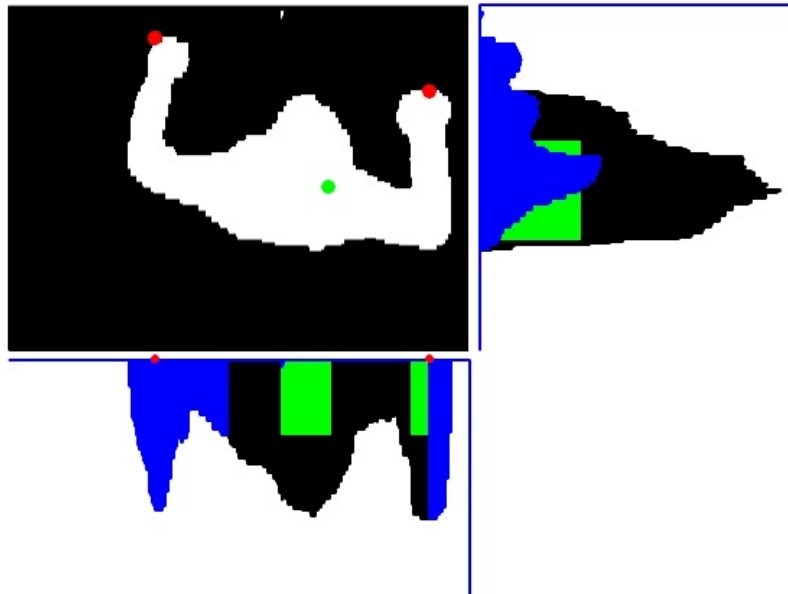


Figura 3.6: Modalità debug dell'applicazione, Braccia Piegate.

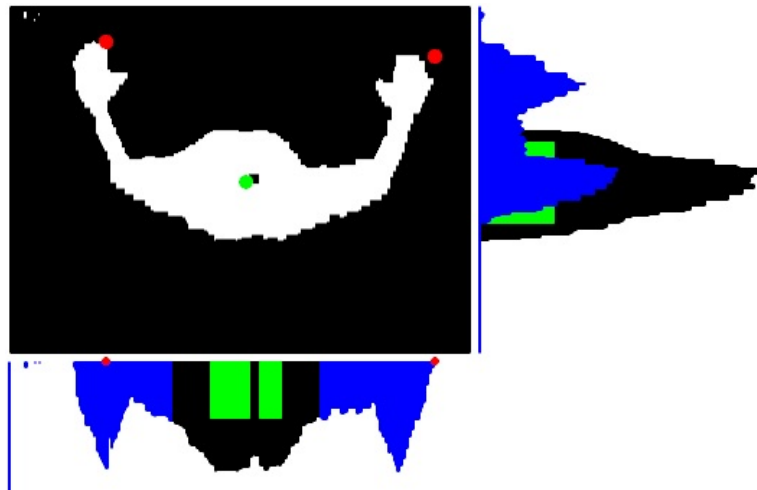


Figura 3.7: Modalità debug dell'applicazione, Braccia Piegate.

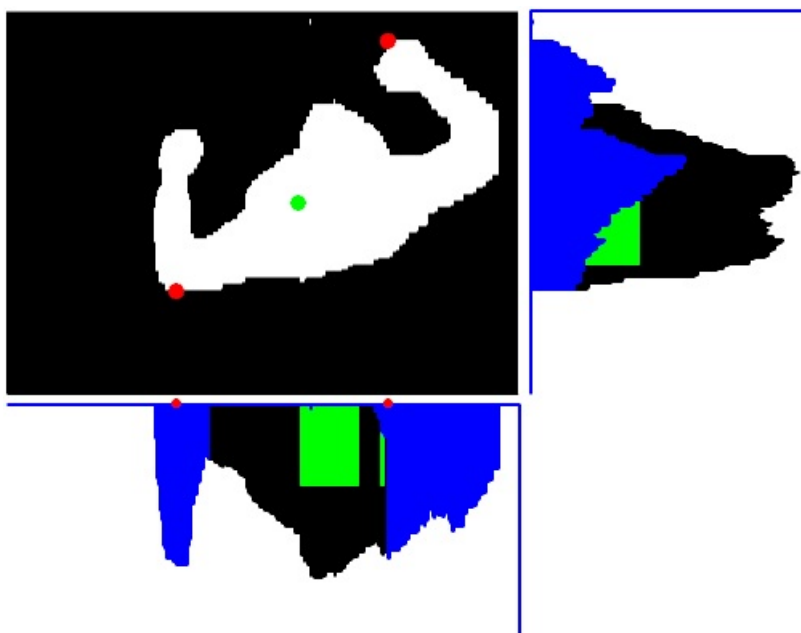


Figura 3.8: Modalità debug dell'applicazione, Braccia Piegate errore rilevamento.

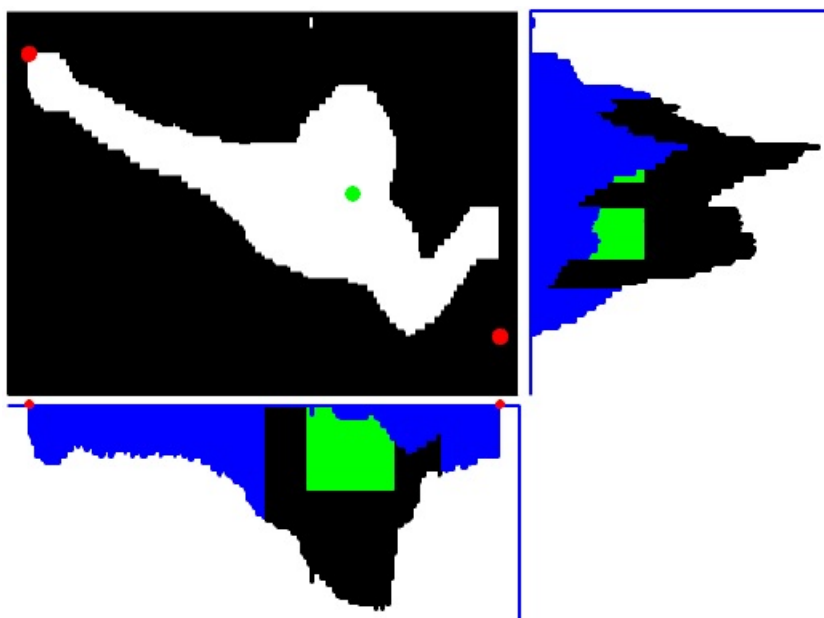


Figura 3.9: Modalità debug dell'applicazione, Braccia Piegate errore rilevamento.

Nelle Figure [3.6][3.7] vediamo la risposta al sistema nel caso in questione. Le immagini derivano da due test differenti, un uomo e una donna di altezza e corporatura differenti. Come possiamo notare in tutti e due i casi l'algoritmo riesce a giudicare correttamente la figura individuando la posizione delle due mani.

Nelle Figure [3.8][3.9] possiamo notare due casi in cui l'algoritmo sbaglia a individuare le mani. Nella prima individua il gomito mentre nella seconda un punto esterno alla silhouette. Anche se a prima vista potrebbero sembrarsi due errori differenti, tutti e due derivano dal fatto che la condizione con cui decidavamo se scegliere l'inizio o la fine del pezzo di istogramma in base alla distanza dal centro non è più valida nel caso di braccia piegate, o almeno non sempre.

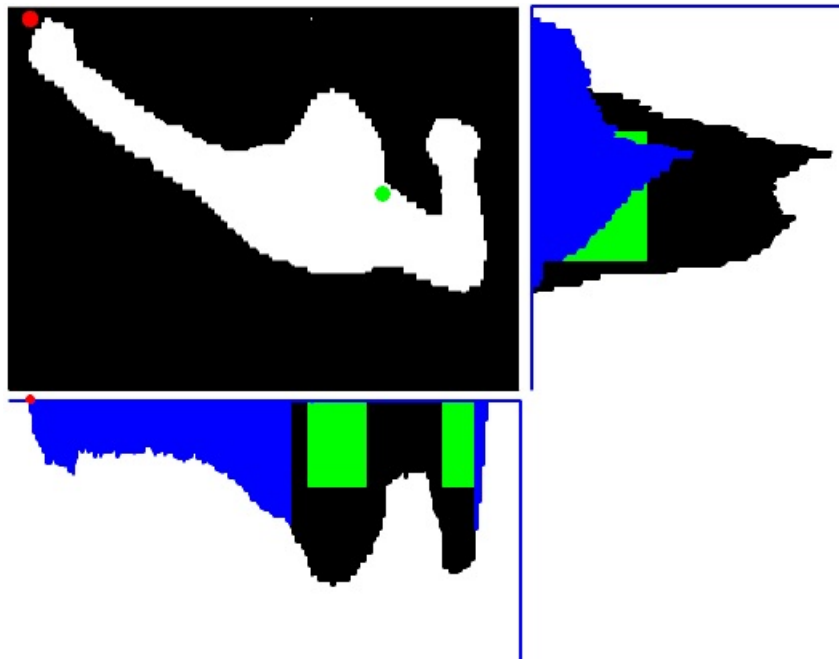


Figura 3.10: Modalità debug dell'applicazione, Braccia Piegate errore rilevamento.

In Figura [3.10] presento un diverso tipo di errore. In questo caso il problema non è causato dall'algoritmo di tracking in sé, ma dalla parte responsabile dell'individuazione del centro e del busto. La speciale posizione del braccio

congiunta con la testa fa credere al sistema che il busto sia la giunzione di busto e braccio destro dell'utente. Questo è un caso limite, infatti aumentando leggermente l'angolo formato dal braccio destro il sistema torna a funzionare correttamente, mentre diminuendolo anche all'occhio umano non è possibile distinguere le varie regioni.

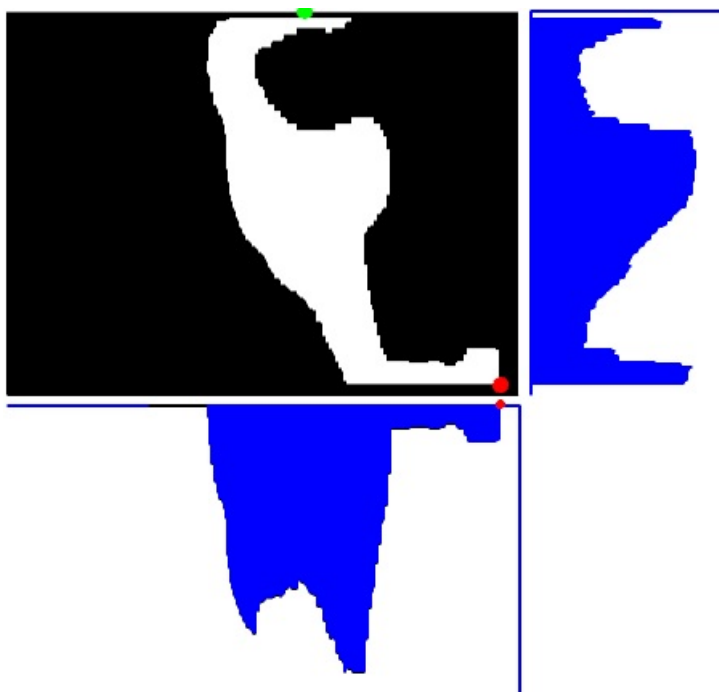


Figura 3.11: Modalità debug dell'applicazione, Braccia Piegate errore rilevamento.

Infine in Figura [3.11] abbiamo un'esagerazione dell'esempio precedente in cui il sistema non riesce a distinguere corpo principale e braccia. Questo è dovuto principalmente da due fattori. Primo la posizione delle braccia piegate che come abbiamo visto precedentemente non è gestita dal sistema. Secondo la non completezza della silhouette che ovviamente danneggia le proporzioni del corpo rendendo difficile il riconoscimento delle varie parti da esso.

Possiamo concludere quindi che se anche l'algoritmo spesso riesce a rilevare correttamente le mani, anche se le braccia sono piegate, non è sicuramente affidabile in quanto il suo funzionamento è aleatorio. Questo controllo sul comportamento del sistema è comunque importante in quanto tra la posizione di riposo delle braccia a quella a braccia distese è facile passare attraverso una fase di braccia piegate. Dagli esempi precedenti abbiamo concluso che il sistema o ignora la mano o la confonde ad esempio con il gomito. In una transazione di pochi secondi prima di arrivare al braccio disteso l'individuare il gomito al posto della mano per la maggior parte delle applicazioni non sarà un problema.

Capitolo 4

Conclusioni

L'algoritmo, come abbiamo visto nel capitolo precedente, si presta a risolvere il problema del tracking delle mani. Per ora abbiamo collegato l'uscita dati OSC a un sintetizzatore sonoro creato in Pure Data[16] che varia la frequenza di un suono in base alla posizione della mano.

In un futuro è auspicabile l'implementazione, ora possibile, dell'applicazione Fiaba Magica. Avendo a disposizione la posizione delle mani è sufficiente implementare la logica e importare i contenuti dal prototipo per avere l'applicazione funzionante. Si apre la possibilità di ampliare l'applicazione *Resonant Memory* con nuove feature come ad esempio una variazione del suono in base al movimento delle mani. Supponiamo ad esempio di dover riprodurre l'effetto sonoro di un temporale, possiamo collegare al movimento delle mani la velocità del vento o addirittura scatenare un tuono se la mano raggiunge una determinata posizione. Tutto questo aumenterebbe il coinvolgimento dell'utente nell'applicazione.

Voglio ricordare che l'algoritmo è progettato per una monoutenza ed è difficilmente adattabile per una multiutenza. Se in un futuro sarà necessario una applicazione che richiede il tracking in un ambiente multiutente bisognerà cambiare approccio al tracking. Partendo dalle silhouette una buona strada è quella di studiare i contorni della figura e sfruttare concavità e convessità delle curve per individuare le feature interessanti. Altra possibilità è quel-

la di partire dal video originale e lavorare anche sui colori per ottenere più informazione possibile da elaborare.

Bibliografia

- [1] S. Zanolla and S. Canazza and A. Rodà and A. Camurri and G. Volpe “Entertaining listening by means of the Stanza Logo-Motoria: an Interactive Multimodal Environment” in *Entertainment Computing*, Elsevier, v. 4, n. 3, pp. 213-220, 2013
- [2] S. Zanolla, F. Romano, F. Scattolin, A. Rodà, S. Canazza, and G. L. Foresti. When sound teaches. In S. Zanolla, F. Avanzini, S. Canazza, and A. de Götzen, editors, *Proceedings of the SMC 2011 - 8th Sound and Music Computing Conference*, pages 64-69, 2011.
- [3] A. de Götzen, R. Marogna, and F. Avanzini. The voice painter. In *Proc. Int. Conf. on Enactive Interfaces*, Pisa, Nov. 2008.
- [4] Rinman, M-L, Friberg, A, Bendiksen, B, Cirotteau, D, Dahl, S, Kjellmo, I, Mazzarino, B, and Camurri, A (2004). Ghost in the Cave - an interactive collaborative game using non-verbal communication. in A. Camurri, G. Volpe (Eds.), *Gesture-based Communication in Human-Computer Interaction*, LNAI 2915, (pp. 549-556), Berlin: Springer Verlag.
- [5] Rinman M-L, Friberg A, Kjellmo I, Camurri, A, Cirotteau D, Dahl S, Mazzarino B, Bendiksen B McCarthy H (2003) EPS - an interactive collaborative game using non-verbal communication. in R. Bresin (Ed.) *Proceedings of SMAC03, Vol II* pp. 561- 563.
- [6] A. F. Bobick, S. S. Intille, J. W. Davis, F. Baird, C. S. Pinhanez, L. W. Campbell, Y. A. Ivanov, A. Sch \tilde{A} $\frac{1}{4}$ tte, and A. Wilson. The kidsroom:

- A perceptually-based interactive and immersive story environment. In PRESENCE, pages 367-391, 1999.
- [7] S. Zanolla, S. Canazza, A. Rodà, and G. De Poli. A learning environment based on movement and sound interaction. In Emerging Software for Interactive Interfaces, Database, Computer Graphics and Animation: Pixels and the New Excellence in Communicability, Cloud Computing and Augmented Reality. Blue Herons, 2012. In press.
- [8] J. Bruner. Toward a theory of instruction. Belknap Press of Harvard University Press, 1966
- [9] J. Bruner. Processes of cognitive growth: Infancy. Clark University Press, Worcester, MA, 1968.
- [10] G. R. Amthor. Multimedia in education: An introduction. Int. Business Mag., 1992.
- [11] G. Bradski and A. Kaehler. Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Incorporated, 2008.
- [12] Sezione about del sito web di OpenCV. <http://opencv.org/about.html>.
- [13] Sezione about del sito web di Openframeworks. www.openframeworks.cc/about/.
- [14] Intervista a Zach Lieberman, fondatore di Openframeworks. www.instituteofplay.org/work/projects/quest-to-learn
- [15] Sito ufficiale del progetto OSC <http://opensoundcontrol.org/>
- [16] Sito ufficiale di Pure Data <http://puredata.info/>
- [17] Leonardo Amico. La Stanza Logo-Motoria: Un ambiente multimodale interattivo per l'insegnamento a bambini in situazione di multi-disabilità. Università di Padova A.A 2011/2012

- [18] Pagina Wiki in inglese riguardante gli istogrammi
<http://en.wikipedia.org/wiki/Histogram>

