

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

Corso di Laurea Magistrale in Informatica

RICONOSCIMENTO DI GESTI:
estensione a più punti del corpo e ad attori multipli

Tesi di Laurea in Sistemi e Applicazioni Multimediali

Relatore:

Chiar.mo Prof. Marco Roccetti

Presentata da:

Cristian Bertuccioli

Co-relatore:

Dott. Gustavo Marfia

Sessione I

Anno Accademico 2011/2012

*... Ai miei familiari,
che mi hanno sostenuto durante
il percorso universitario ...*

Indice

1	Introduzione	1
2	Il riconoscimento delle azioni	5
2.1	Tecnologie: una prima classificazione	5
2.1.1	Dispositivi contact-based	5
2.1.2	Dispositivi vision-based	7
2.2	Aree di applicazione	9
2.2.1	Linguaggio dei segni	10
2.2.2	Realtà virtuale	11
2.2.3	Assistenza medico-sanitaria	11
2.2.4	Videosorveglianza	11
2.2.5	Musei ed installazioni pubbliche	12
2.3	Riconoscimento del corpo: casi di studio	12
2.3.1	Skin detection	13
2.3.2	Template matching	14
2.3.3	Definizione della sagoma umana	15
2.4	Il museo come contesto	17
2.4.1	Lavori correlati	17
2.4.2	Limiti ed esigenze	22
2.4.3	Interagire all'interno del Museo	23
2.5	La soluzione adottata	25

2.5.1	Metodo dei Massimi e Minimi	26
3	Descrizione del sistema: una panoramica	31
3.1	Caratteristiche e configurazione	32
3.2	Passi principali	32
3.3	Riconoscimento del corpo	34
3.4	Estrazione dei punti del corpo	35
3.4.1	Ricerca Massimi Locali	36
3.4.2	Filtraggio dei Gap	37
3.5	Rilevazione dei movimenti	38
4	Descrizione del sistema: dettagli implementativi	41
4.1	Scelte progettuali	42
4.1.1	OpenCV	43
4.2	Segmentazione	45
4.2.1	Filtri di smoothing	45
4.2.2	Sottrazione dello sfondo	50
4.2.3	Filtro morfologico: Apertura	51
4.2.4	Identificazione dei blobs	55
4.3	Conversione in griglia	56
4.4	Estrazione dei punti	57
4.4.1	Classi di punti	57
4.4.2	Popolazione delle liste di riferimento	59
4.4.3	Estrazione delle proporzioni	59
4.4.4	Rilevazione dei piedi	60
4.4.5	Rilevamento top-down	62
4.4.6	Ottenimento dei punti rimanenti	64
4.5	Riconoscimento movimenti	67
4.5.1	Filtro di Kalman discreto	68
4.5.2	Zone sensibili	71

INDICE

4.5.3 One dollar	71
5 Risultati dei test ed analisi	75
5.1 Prestazioni	76
5.2 Limiti e possibili errori	78
Conclusioni	81
Ringraziamenti	85
Bibliografia	87

Elenco delle figure

2.1	Dispositivi contact-based	8
2.2	Dispositivi vision-based	10
2.3	Skin detection	14
2.4	Algoritmo convex hull	16
2.5	Hyper	22
2.6	Servizi intangibili	26
2.7	Applicazione del metodo massimi e minimi	27
2.8	Analisi picchi Tortellino X-Perience	28
3.1	Scenario ipotetico	33
3.2	Trasformazione in griglia	35
3.3	Estrazione dei punti dal blob	36
3.4	Classificazione delle proporzioni del corpo	38
3.5	Linee che definiscono le proporzioni	39
3.6	Esempio del riconoscitore di gesti	40
4.1	Fasi principali del sistema	42
4.2	Architettura della libreria OpenCV	44
4.3	Diagramma di flusso per rilevare i punti	46
4.4	Tipi di rumore	47
4.5	Esempio di convoluzione	47
4.6	Filtri di smoothing	48

ELENCO DELLE FIGURE

4.7	Filtro mediano: esempio sulle matrici	49
4.8	Filtro morfologico di erosione	53
4.9	Filtro morfologico di dilatazione	54
4.10	Filtro morfologico di apertura	54
4.11	Esempio di errore nella rilevazione di un punto approssimato .	58
4.12	Ricerca dei massimi relativi	63
4.13	Esempio di riconoscimento	67
4.14	Equazione ricorsiva del filtro di Kalman	69
4.15	Applicazione filtro di Kalman	69
4.16	Riconoscimento tramite zone sensibili	72
4.17	\$1 Dollar	73
5.1	Esempio di errore	78
5.2	Il frameset analizzato	80

Elenco delle tabelle

5.1	Performance dell'algoritmo in tutto il frameset	76
5.2	Performance dell'algoritmo quando viene effettuata un'azione .	77

Capitolo 1

Introduzione

Negli ultimi anni si è assistito ad una radicale rivoluzione nell'ambito dei dispositivi di interazione uomo-macchina. Da dispositivi tradizionali come il mouse o la tastiera si è passati allo sviluppo di nuovi sistemi capaci di riconoscere i movimenti compiuti dall'utente (interfacce basate sulla visione o sull'uso di accelerometri) o rilevare il contatto (interfacce di tipo touch). Questi sistemi sono nati con lo scopo di fornire maggiore naturalezza alla comunicazione uomo-macchina. Le nuove interfacce sono molto più espressive di quelle tradizionali poiché sfruttano le capacità di comunicazione naturali degli utenti, su tutto il linguaggio gestuale.

Essere in grado di riconoscere gli esseri umani, in termini delle azioni che stanno svolgendo o delle posture che stanno assumendo, apre le porte a una serie vastissima di interessanti applicazioni.

Ad oggi sistemi di riconoscimento delle parti del corpo umano e dei gesti sono ampiamente utilizzati in diversi ambiti, come l'interpretazione del linguaggio dei segni [8], in robotica per l'assistenza sociale [1], per indicare direzioni attraverso il puntamento [2], nel riconoscimento di gesti facciali [1], interfacce naturali per computer (valida alternativa a mouse e tastiera), ampliare e rendere unica l'esperienza dei videogiochi (ad esempio Microsoft

Kinect[©] e Nintendo Wii[©]), nell’*affective computing*¹.

Mostre pubbliche e musei non fanno eccezione, assumendo un ruolo centrale nel coadiuvare una tecnologia prettamente volta all’intrattenimento con la cultura (e l’istruzione).

In questo scenario, un sistema HCI deve cercare di coinvolgere un pubblico molto eterogeneo, composto, anche, da chi non ha a che fare ogni giorno con interfacce di questo tipo (o semplicemente con un computer), ma curioso e desideroso di beneficiare del sistema. Inoltre, si deve tenere conto che un ambiente museale presenta dei requisiti e alcune caratteristiche distintive che non possono essere ignorati.

La tecnologia immersa in un contesto tale deve rispettare determinati vincoli, come:

- non può essere invasiva;
- deve essere coinvolgente, senza mettere in secondo piano gli artefatti;
- deve essere flessibile;
- richiedere il minor uso (o meglio, la totale assenza) di dispositivi hardware.

In questa tesi, considerando le premesse sopracitate, si presenta una sistema che può essere utilizzato efficacemente in un contesto museale, o in un ambiente che richieda soluzioni non invasive. Il metodo proposto, utilizzando solo una webcam e nessun altro dispositivo personalizzato o specifico, permette di implementare i servizi di: (a) rilevamento e (b) monitoraggio dei visitatori, (c) riconoscimento delle azioni.

Questo documento è organizzato come segue. Nel capitolo 2, viene presentato il problema del riconoscimento delle azioni, analizzato lo stato dell’arte

¹Branca specifica dell’intelligenza artificiale che si propone di realizzare calcolatori in grado di riconoscere ed esprimere emozioni.

fornendo diversi esempi di applicazione; saranno analizzati delle soluzioni per valutare quanto possano essere inglobate nel contesto museale, specificandone i limiti e le esigenze.

Nel capitolo 3, viene fornita una panoramica generale sul metodo proposto, definendone gli steps principali e proponendo la definizione di *gap*.

Nel capitolo 4, è descritta, in modo più specifico e tecnico, la struttura del sistema, analizzando più a fondo gli steps, gli algoritmi utilizzati per l'*image processing*, e l'estensione proposta al metodo dei *massimi e dei minimi*. Nel capitolo 5, infine, si valuta l'efficienza e l'efficacia del sistema.

Capitolo 2

Il riconoscimento delle azioni

Riconoscere la postura di un essere umano significa poterne specificare la posizione e l'orientamento delle varie parti del corpo.

Nel corso degli ultimi decenni, la ricerca si è concentrata su questo tipo di problema.

2.1 Tecnologie: una prima classificazione

In questa sezione si presenta una prima macro classificazione sulle tecnologie per il riconoscimento dei gesti. Esistono due tipi principali di tecnologie: (1) contact-based e (2) vision-based. Di seguito si esporrà e si valuteranno le due tipologie.

2.1.1 Dispositivi contact-based

Dispositivi basati sul contatto sono vari: accelerometri, schermi multi-touch, guanti sensorizzati sono i principali esempi che utilizzano queste tecnologie. Alcuni dispositivi, come Nintendo Wii-mote[©] comprendono anche un mix tra accelerometri e sensori ottici.

Si possono classificare questi dispositivi nelle seguenti categorie:

- **Meccanici:** per esempio, Immersion[©] propone il “CyberGlove II [©]” un guanto wireless sensorizzato per il riconoscimento dei gesti delle mani. Animazoo[©] propone una tuta chiamata “IGS-190[©] per catturare i movimenti del corpo. Inoltre questi dispositivi possono essere utilizzati in associazione con altre tecnologie, utilizzati, ad esempio, per modellare le “traiettorie” ottenute dal riconoscimento dei gesti con cybergloves attraverso tracciatori magnetici [3]
- **Inerziali:** questi dispositivi misurano la variazione del campo magnetico terrestre, al fine di rilevare il movimento. Rientrano in questa categoria gli accelerometri (per esempio Wii-mote[©]) e giroscopi (ad esempio IGS-190[©]). Schlömer et al. propongono di riconoscere gesti con un controller Wii in modo indipendente dal sistema di destinazione utilizzando Modelli di Markov nascosti (HMM) [4], permettendo all’utente di impostare ed imparare gesti personalizzati per navigare tra contenuti multimediali.
- **Tattili:** come gli schermi multi-touch che sono diventati sempre più utilizzati ed integrati nella vita di tutti i giorni (ad esempio tablet, smartphone). Recentemente la Disney Research¹ ha sviluppato *Touché*[©], una nuova tecnologia di riconoscimento che propone una nuova tecnica di rilevamento capacitivo della frequenza, che non solo può rilevare un evento di tatto, ma allo stesso tempo riconosce configurazioni complesse delle mani e del corpo umano durante l’interazione[5].
- **Magnetici:** questi dispositivi misurano la variazione di un campo magnetico artificiale per rilevare dei movimenti in esso.
- **Ultrasuoni:** in questa categoria rientrano i tracciatori di movimento come: emettitori di ultrasuoni, dischi sonori (collegati ad una persona),

¹È una rete di laboratori di ricerca supportata dalla Walt Disney Company[©], ha lo scopo di perseguire l’innovazione scientifica e tecnologica della società.

che riflettono gli ultrasuoni, e sensori che si basano sul tempo di ritorno del suono. Questi dispositivi non sono precisi, ma possono essere una buona soluzione in ambienti dove i sistemi precedenti potrebbero riscontrare problemi (ad esempio ambienti con poca luce o soggetti a perturbazioni magnetiche).

La figura 2.1 illustra qualche esempio dei dispositivi sopracitati.

2.1.2 Dispositivi vision-based

I sistemi di riconoscimento gesti vision-based contano su una o più telecamere al fine di analizzare e interpretare il moto dalle sequenze video catturate.

Analogamente ai dispositivi contact-based, anche questi possono essere vari.

Per esempio, si possono distinguere i seguenti sensori:

- **Termocamere ad infrarossi:** tipicamente utilizzate per la visione notturna, queste telecamere forniscono in genere un'immagine disturbata della sagoma umana ed in ogni caso con una scarsa risoluzione. La tecnologia dei proiettori ad infrarossi sta alla base del sensore di profondità proposto da Microsoft[©] con il dispositivo Kinect[©], che attraverso una telecamera sensibile alla stessa banda del proiettore, permette di ottenere una mappa di profondità che colloca il giocatore in uno spazio tridimensionale (ottenendo i dati da una sola dimensione)[6].
- **Telecamere monoculari:** le telecamere più comuni dato il loro prezzo più conveniente rispetto altri dispositivi. Per riconoscere attori e gesti in ogni fotogramma catturato, occorre fare uso di tecniche ed algoritmi di analisi delle immagini.



Figura 2.1: Esempi di dispositivi contact-based. (a) Cyber Glove II[©], (b) IGS-190[©] (c) Disney Touché[©]

- **Telecamere stereo:** la stereovisione fornisce informazioni direttamente nello spazio tridimensionale, grazie ad un processo di triangolazione delle immagini. Ottenendo quindi immagini con informazioni sulla pro-

fondità (dette Disparity Maps), le applicazioni possono essere molteplici, ad esempio Keller et al. propongono un sistema, che se posto in un autovettura, è in grado di rilevare pedoni in strada in modo autonomo [7].

- **Telecamere PTZ²:** è una telecamera che è in grado di orientarsi e utilizzare lo zoom autonomamente, ovvero quando nota dei cambiamenti dei pixel tra un fotogramma e l'altro.
- **Marcatori:** alcuni sistemi richiedono di posizionare dei marcatori in vari punti del corpo al fine di individuarne i movimenti. Con questo sistema si è in grado di ottenere movimenti complessi, a patto di posizionare più marcatori.

La figura 2.2 mostra degli esempi sui sistemi vison-based citati in precedenza.

2.2 Aree di applicazione

Ad oggi sistemi di riconoscimento delle parti del corpo umano e dei gesti hanno occupato diversi domini di applicazione. Dall'interpretazione del linguaggio dei segni, agli ambienti virtuali che dispongono di interfacce uomo-macchina intelligenti, la quantità di domini aumenta continuamente e le soluzioni proposte sono sempre più efficienti, fino ad arrivare ad installazioni pubbliche e musei.

Proprio per la particolarità di quest'ultima area di interesse, il museo è stato scelto come il contesto per l'applicazione del metodo.

In questa sezione si prendono in rassegna alcuni dei settori di applicazione di sistemi per il riconoscimento di gesti da sequenze video.

²Acronimo di pan-tilt-zoom

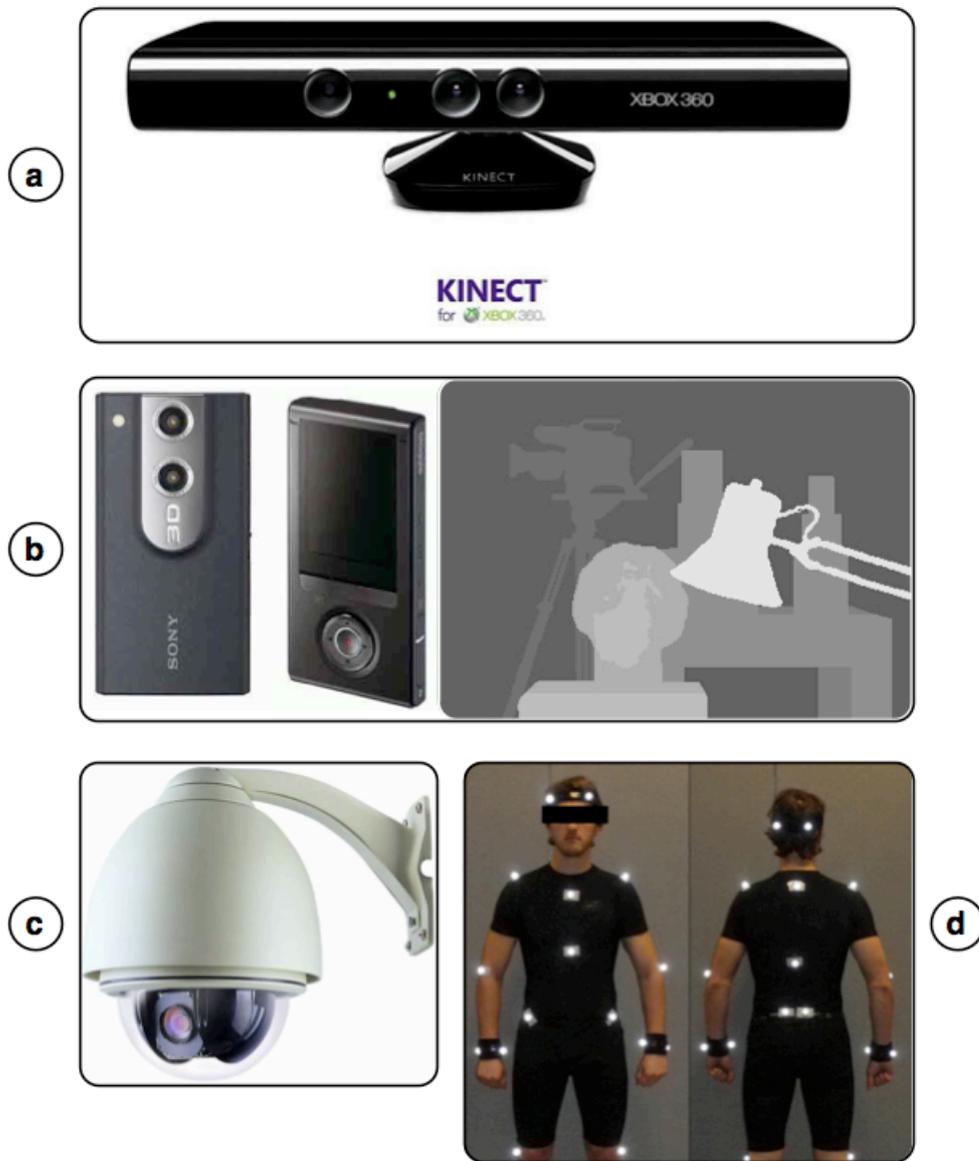


Figura 2.2: Esempi di dispositivi vision-based. (a) Microsoft Kinect[®], (b) telecamera stereo, a destra una disparity map, (c) telecamera PTZ, (d) marcatori

2.2.1 Linguaggio dei segni

Mentre nel riconoscimento vocale l'obiettivo è quello di trascrivere il discorso in testo, lo scopo del riconoscimento del linguaggio dei segni è quello

di trascrivere gesti definiti in testo.

Studi per questo tipo di obiettivo sono molteplici nel campo delle tecnologie assistive, un esempio significativo è fornito da Fang et al. 2007 che attraverso modelli di transizione del movimento riescono ad implementare un vocabolario con più di cinquemila segni con un'ottima accuratezza [8].

2.2.2 Realtà virtuale

Sistemi di realtà virtuale consentono ad un utente di interagire con un ambiente simulato dal computer (sia esso un esempio del mondo reale o un'istanza di un mondo immaginario). Questi sistemi comprendono immersive gaming, simulatori di volo e controllo remoto.

In questo caso, il riconoscimento dei gesti viene utilizzato come mezzo di comunicazione con il mondo virtuale.

2.2.3 Assistenza medico-sanitaria

Un utile (e curioso) esempio di come queste tecnologie possono esse applicate in campo medico è dato da Keskin et al., che propongono un sistema multimodale di riconoscimento dei gesti, dove un avatar (rappresentato da un forma di una testa di un medico tridimensionale) parlante, comunica in modo autonomo con i pazienti dell'ospedale costretti a letto, rispondendo a un totale di nove comandi gestuali e fornendo un feedback immediato al paziente (ed eventualmente al personale sanitario) [9].

Altri sistemi, invece, utilizzando una rete di sensori, monitorano i pazienti nel sonno riconoscendo posture e movimenti cardio-respiratori a rischio [10].

2.2.4 Videosorveglianza

Il principale scopo dei sistemi di videosorveglianza è la sicurezza. Per quanto riguarda il riconoscimento dei gesti, l'obiettivo principale è quello di

rilevare azioni violente e/o furtive in certe zone a rischio.

Vi sono applicazioni che vanno dal riconoscimento di azioni malevole (ad esempio scippi) in ambienti circoscritti, come in ascensore, utilizzando tecniche di optical flow [11], fino a cercare di predire intenti violenti in zone affollate tramite modelli di predizione [12].

2.2.5 Musei ed installazioni pubbliche

Si può osservare come, anche in un contesto museale, la tecnologia può essere utilizzata ugualmente per creare un legame invisibile ed immediato tra il visitatore e le opere al suo interno.

L'intento dei sistemi sviluppati per queste aree di interesse è quello di cercare di immergere (parzialmente o totalmente) il visitatore, come fosse un gioco, nello stesso tempo rendendo l'esperienza istruttiva.

2.3 Riconoscimento del corpo: casi di studio

Nella sezione seguente si vedranno degli approcci di studi che sono stati utili nella definizione del metodo presentato.

Si tratta di approcci prevalentemente vision-based, che adottano tecniche ed algoritmi più o meno avanzati sulle immagini prese in input (ovvero ogni fotogramma) da un flusso video di una normale telecamera monoculare.

Questi casi non riguardano il riconoscimento di azioni in senso stretto, piuttosto il riconoscimento della figura umana (o parti di essa), in modo da ottenere (e successivamente tracciare) i diversi punti del corpo, importanti per la definizione dei gesti.

2.3.1 Skin detection

Il processo di segmentazione del colore della pelle permette di discriminare tra i pixel che appartengono o che non appartengono a zone di pelle “scoperte” di una persona.

Gli algoritmi che si basano su un approccio cromatico necessitano di una prima fase di training per poter creare un modello probabilistico per la distribuzione del colore della pelle.

La fase di training è un’analisi, eseguita a priori, degli istogrammi di un ampio set di immagini campione, che contengono informazioni su come, stocasticamente, l’intensità del colore si distribuisce su una zona riconosciuta come appartenente alla pelle.

Di solito, questi modelli presentano dei limiti derivanti soprattutto dal database utilizzato: se troppo eterogeneo presenterà una probabilità alta di riconoscere anche zone di non interesse, se troppo omogeneo (od un set ristretto di immagini), l’immagine maschera estratta per la pelle può presentare dei buchi (figura 2.3).

Per ovviare a questi problemi si utilizzano modelli che impiegano classificatori gaussiani semplici (SGM³) o misti(GMM⁴) [37] [38].

Questi algoritmi vengono utilizzati spesso per immagini statiche, piuttosto che per sequenze video, in quanto spesso, i diversi pixel dei fotogrammi di una videocamera sono soggetti a cambiamenti di intensità, che, nonostante si definisca un modello robusto, potrebbero portare alla rilevazione di falsi positivi.

³Single Gaussian Model

⁴Gaussian Mixture Model

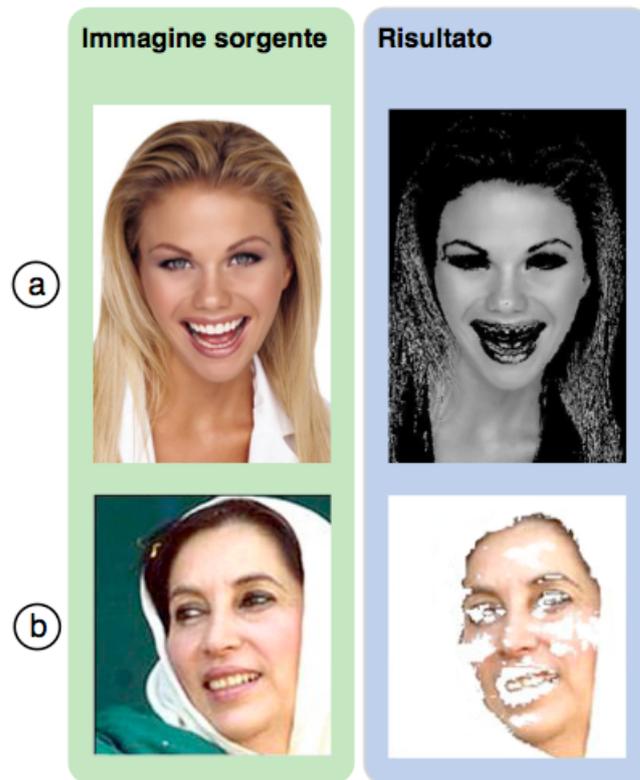


Figura 2.3: Errori della skin detection.

(a) esempio con una soglia bassa: la maschera della pelle contiene anche regioni non interessate (i capelli), (b) esempio con una soglia alta: la maschera estratta per la pelle può presentare dei buchi.

2.3.2 Template matching

È una tecnica di elaborazione delle immagini per la ricerca di piccole parti di un'immagine che corrispondono ad un'immagine modello. Algoritmi di Template matching possono essere suddivisi in due approcci: *feature-based* e il *model-based*.

Nell'approccio **feature-based** vengono definiti dei template pattern di

feature⁵ che descrivono il prototipo del particolare da ricercare, come bordi o angoli, che fungeranno come primo valore di riferimento per trovare la posizione del migliore match (tipicamente un'area) che presenti dei pattern di feature simili [39].

L'approccio **model-based**, o globale, utilizza l'intero template, generalmente utilizzando confronti basati sulla somma (o differenza) tra template e immagine sorgente (ad esempio SAD, SSD, cross-correlation), e ne determina la posizione migliore, testando il match tra i due, valutato in tutto o in un campione ristretto dell'immagine.

Questi algoritmi, se correntemente istruiti, ovvero se si genera un buon modello del template, sono robusti, ma non sempre precisi, inoltre in un sistema real-time, abbassano le performance (dove per performance si intende sia gli fps, che il ritardo dovuto alla computazione) e potrebbero rendere il sistema non confortevole da utilizzare. Inoltre, può “confondersi” se c'è uno spostamento tra due frame consecutivi di grandi dimensioni (effetti *motion blur*).

2.3.3 Definizione della sagoma umana

Questo metodo, spesso affiancato alle tecniche citate nei paragrafi precedenti, a partire da una sagoma e considerando i vincoli fisici (e la topologia) del corpo umano, permette di ottenere la posizione approssimata delle diverse parti del corpo.

⁵Con feature si intendono punti o regioni salienti di un'immagine, il cui intorno (tipicamente ristretto rispetto alle dimensioni dell'immagine stessa) è identificabile come corrispondente in modo il più possibile ripetibile, indipendentemente dalle differenze presenti fra le immagini.

Un lavoro pionieristico basato su questa tecnica, è fornito da Haritaoglu et al. 1998, che propone un sistema in tempo reale per valutare sia la postura che le diverse parti del corpo umano. Per ottenere questo tipo di risultato costruisce un modello che combina una prima stima della posa confrontando la funzione ottenuta con i diversi valori dell'istogramma⁶ dell'immagine con delle funzioni stimate, successivamente applicava un algoritmo per l'analisi della convessità (*convex hull algorithm*) ottenendo una mappatura dalle parti del corpo [40]. La figura 2.4 mostra un esempio pratico di come funziona questo metodo.

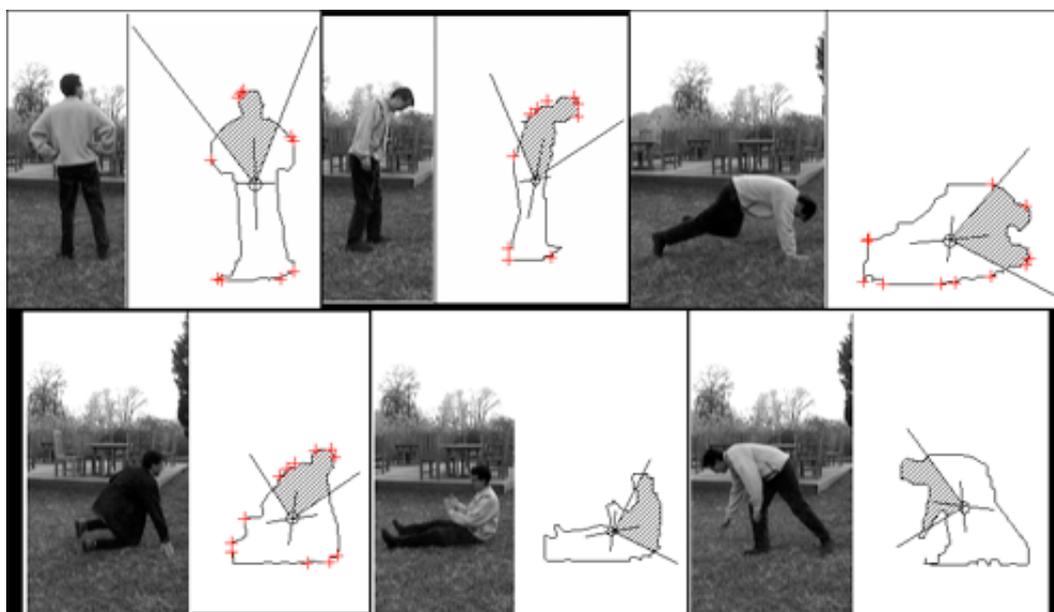


Figura 2.4: *Ghost*: un sistema che, applicando l'algoritmo di convex-hull, riesce ad identificare segmenti della forma umana e un'approssimazione della postura.

Altri approcci proposti successivamente, si basano non solo sulla sagoma, ma vi creano all'interno una sorta di modello di scheletro della figura umana.

⁶L'istogramma in un'immagine digitale, è la distribuzione dell'intensità dei valori, per ogni canale, di un'immagine digitale. In pratica traccia il numero di pixel per ogni valore tonale [56].

In base a questo modello si può ottenere postura e punti rilevanti del corpo (estremità e testa) [41, 42].

Inoltre, prendendo come punto di riferimento il punto mediano del blob, si possono ottenere sia la distanza da questo punto ad una parte del corpo rilevata (r), sia l'orientamento di quest'ultima rispetto l'asse verticale (θ). In base a questi due parametri, catturati durante un'azione, e plottati in coordinate polari, è possibile definire dei pattern per determinate azioni [43].

2.4 Il museo come contesto

Una parte consistente della ricerca, negli ultimi anni, è stata indirizzata all'applicazione di nuove ed avanzate tecnologie nel contesto dei musei e mostre pubbliche.

In questa sezione si elencheranno, e saranno descritti brevemente, diversi lavori che hanno preso in considerazione il museo come contesto applicativo, si analizzeranno i vincoli dell'ambiente museale e le scelte adottate per lo sviluppo del metodo proposto.

2.4.1 Lavori correlati

Il lavoro di ricerca in questo ambito non si è orientato solo verso lo sviluppo di applicazioni multimediali statiche (dove ad esempio si presentano ricostruzioni 3D accurate di una civiltà antica), ma ha compreso anche la realizzazione di sistemi di interazione avanzati, dove, ad esempio, viene posto davanti al visitatore la possibilità di partecipare a spettacoli digitali, così come la possibilità di influenzare o addirittura decidere che cosa visualizzare, o cambiare il corso di una presentazione.

In seguito si elencheranno e si descriveranno alcuni dei sistemi che sono stati sperimentati e che implementano dei servizi sempre più spesso necessari

in un ambiente museale, come: *rilevamento dei visitatori, monitoraggio e riconoscimento delle azioni*.

- **Rilevamento dei visitatori**

Un servizio che rileva la presenza di un visitatore in una determinata zona (oppure occupa una superficie), lascia spazio ad un'ampia gamma di applicazioni che, potenzialmente, potrebbero beneficiarne.

Comprendono sistemi che vanno da guide turistiche digitali a didascalie digitali (che possono “apparire” accanto ad un'opera), ma possono includere anche menù digitali e in genere applicazioni finalizzate a fornire esperienze coinvolgenti e spiegazioni riguardo alle opere che si trovano in prossimità di uno o più visitatori.

Altrettanto ampio è l'insieme delle tecnologie che sono state impiegate nel fornire servizi di questo tipo.

Gli autori di Bay et al. 2006, per esempio, rilevano la posizione di un visitatore in un modo piuttosto non convenzionale, utilizzando tecniche di pattern recognition.

In breve, i visitatori che volevano fruire di questo tipo di tecnologia sono stati equipaggiati con un tablet e gli si è consigliato di fare delle foto agli artefatti, di loro interesse, su cui volevano ottenere maggiori informazioni.

L'applicazione in esecuzione sul tablet, sfruttando tecniche di pattern recognition e algoritmi di matching, riconoscevano l'artefatto dall'immagine, tra tutti quelli esposti nel museo; inoltre individuavano così la posizione e l'orientamento del visitatore [13].

Il progetto *PEACH* (Stock et al. 2007) sfrutta un approccio simile, in cui la presenza dei visitatori viene rilevata attraverso dispositivi muniti di identificazione a radio frequenza (detti tag RFID) ed infrarossi. In entrambe queste due soluzioni occorreva utilizzare diversi dispositivi

hardware specifici per far sì che il sistema funzionasse.

Ad esempio, utilizzando la tecnologia ad infrarossi, venivano utilizzati dei segnalatori (IR beacon) per stabilire la distanza di un dispositivo mobile da una data posizione, e questo creava dei noti inconvenienti, come: doveva essere disponibile una linea diretta tra il mittente ed il ricevitore, e la presenza di ostacoli tra il segnale impediva la comunicazione.

Utilizzando invece dei tag RFID, transponder e ricetrasmittitori potevano comunicare all'interno di una data distanza, di solito entro qualche metro, senza la necessità di una linea diretta, anche se, in stanze affollate, le prestazioni venivano abbattute dall'attenuazione del segnale dovuto alle onde elettromagnetiche emanate dal corpo dei visitatori [14].

- **Monitoraggio dei visitatori** Il problema del monitoraggio di un visitatore in ogni momento può essere visto come un'estensione del rilevamento. Diversamente da quest'ultimo, si pone come obiettivo quello di essere in grado di seguire un visitatore ovunque vada e tracciare le sue azioni, sia all'interno che all'esterno del museo.

La localizzazione all'esterno non è un grosso problema se consideriamo che esistono tecnologie come GPS che forniscono una valida ed efficiente soluzione e che forniscono stime sulla posizione con una buona precisione.

Tuttavia, l'utilizzo di questa tecnologia richiede al museo di fornire ad ogni visitatore un dispositivo portatili equipaggiato con GPS.

Per quanto riguarda la localizzazione all'interno di un museo, generalmente, viene utilizzato l'RSSI⁷ di un segnale WiFi ed in genere

⁷*received signal strength indicator*, è una misura della potenza di un segnale radio ricevuto.

adottando due approcci:

- utilizzando la relazione tra potenza del segnale(RSSI) e distanza
- costruendo una mappa degli RSSI per ogni stanza, dove i valori della potenza del segnale vengono associate alle zone in cui vengono misurati.

Stock et al., già citati nel paragrafo precedente, impiegano la tecnologia WiFi in maniera più grossolana rispetto agli approcci descritti, impiegandola per rilevare se un visitatore si trova o meno in una stanza, mentre utilizzano localizzazione più accurata all'interno di una stanza tramite dispositivi as infrarossi o RFID, come descritto nel paragrafo precedente.

Anche tale tipo di approccio, tuttavia, richiede l'uso di dispositivi WiFi [14].

Un'ulteriore alternativa si basa sull'uso di accelerometri [14, 15].

Sulla base dell'equazione cinetica che prende in considerazione posizione, velocità e accelerazione, il sistema ottiene la posizione di un dispositivo dai valori di accelerazione.

Con questa metodologia, però, non si ottengono informazioni precise per tracciare dispositivi collocati in scenari interni, ad esempio i sensori accelerometrici utilizzati dai tradizionali dispositivi portatili (come smartphone) non possono essere utilizzati per calcolare stime affidabili sulla posizione [16].

• Riconoscimento delle azioni

Negli ultimi anni sono state proposte una varietà di modi diversi per far interagire , attraverso interazioni più precise, i visitatori con le opere

che si trovano all'interno di un museo, compreso il riconoscimento delle azioni [17].

Un altro lavoro interessante, basandosi sull'utilizzo di interfacce tattili, permette ai visitatori di compiere azioni come "toccare" sculture virtuali, ottenendo una sorta di percezione del gesto attraverso stimoli tattili fornita da dispositivi force-feedback [18][19].

In particolare, l'autore di Bergamasco 1999, ha sfruttato tali sistemi per dare la possibilità ai visitatori di toccare vere e proprie sculture, contatti altrimenti vietati [21].

Un approccio diverso è invece sfruttato in tutti quei sistemi basati su sensori accelerometrici personalizzati, dove, utilizzando algoritmi che utilizzano modelli di markov nascosti, si è in grado riconoscere una serie di gesti base che un visitatore può effettuare con le mani [20].

Recentemente, un designer francese, utilizzando diverse depth camera di Microsoft Kinect[®] posizionate nelle stanze, ha messo a punto *Hyper*, un sistema (un po' fuori dagli schemi) che permette di immergere l'utente in una versione alternativa della realtà vista attraverso un casco. Invece di avere un punto di vista statico, l'utente è in grado di navigare attraverso l'ambiente 3D e può creare nuovi comportamenti nel mondo iperreale, pur avendo modo di interagire fisicamente con l'ambiente reale (figura 2.5).

Ai visitatori viene fornito un casco con occhiali video ad alta definizione ed un guanto Arduino con sensori di forza che controllano l'ambiente 3D [22].

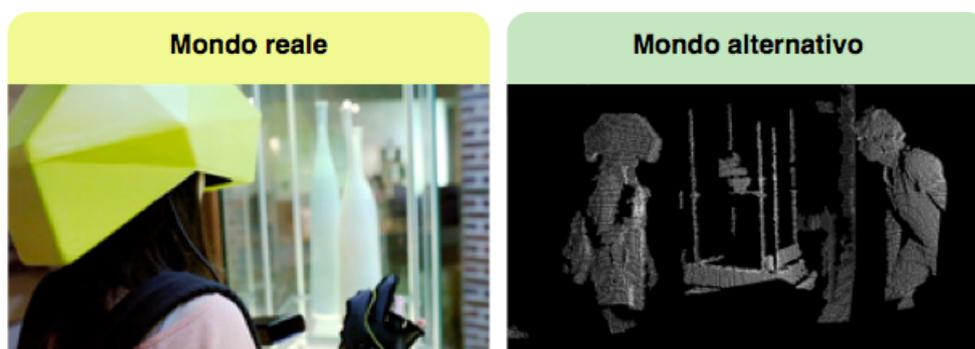


Figura 2.5: Hyper: attraverso un caschetto ed un guanto il visitatore viene immerso in un ambiente 3D dove è in grado di avere un punto di vista dinamico.

2.4.2 Limiti ed esigenze

Alla luce dei lavori precedentemente descritti, si può notare come tutte le soluzioni discusse richiedono ai visitatori l'utilizzo di un qualche tipo di dispositivo hardware.

Questo aspetto è fondamentale in diversi ambienti espositivi pubblici, in quanto spesso non è possibile per le istituzioni artistiche spendere budget importanti per dispositivi hardware, che diventano obsoleti in poco tempo. In questo modo si crea una barriera in ingresso, ovvero si evita la larga diffusione di sistemi multimediali, limitando il loro utilizzo solo alle istituzioni che possono vantare di budget sostanziosi.

Per questo motivo è importante esplorare nuove metodologie che portano a soluzioni più praticabili, dove il visitatore è libero di muoversi senza alcun dispositivo, mentre i sensori necessari per riconoscerlo (e riconoscerne i gesti) sono posti all'interno del museo.

Diversi sistemi hanno messo in pratica questa idea, evitando l'utilizzo, da parte dei visitatori, di qualsiasi dispositivo hardware [23, 24, 25, 26, 27, 28, 29, 30, 31].

Una delle prime opere in questo nel campo, risale agli anni '80, quando si

iniziavano ad utilizzare tecniche di computer vision per riconoscere il corpo di un giocatore come dispositivo di input per un videogioco [32].

Più recentemente, non mancano gli esempi di realtà aumentata, Snibbe e Raffle propongono un metodo dove l'interazione tra telecamere e video proiettori permettono di coinvolgere più utenti contemporaneamente. Nello specifico, i visitatori venivano proiettati in un ambiente virtuale rappresentante la foresta amazzonica ed ogni movimento incauto riconosciuto dalle telecamere poteva scatenare l'attacco di un giaguaro virtuale [33].

Un approccio simile è stato utilizzato in un progetto chiamato *Shadow Garden*, dove venivano catturate le ombre prodotte dai visitatori su uno schermo per proiezioni ed utilizzate per interagire con l'ambiente proiettato [34].

2.4.3 Interagire all'interno del Museo

Sebbene tra tutti gli approcci sopracitati, si possa preferire l'utilizzo di interfacce naturali ed intuitive, che non prevedono quindi l'utilizzo di dispositivi hardware, nella maggior parte, mancano di un approccio sistematico al problema di come sistemi digitali possono meglio inserirsi in contesti culturali.

Solo con tale consapevolezza, è possibile identificare in modo efficace i servizi che sono necessari in una determinata mostra. Infatti, le soluzioni che si incontrano, sono spesso idee derivate da altre applicazioni (ad esempio, da un videogioco) e non adattati alle condizioni specifiche del contesto di applicazione.

I musei, infatti, sono luoghi che differiscono, per molti aspetti, da altri luoghi pubblici. Sono luoghi unici dove i visitatori incontrano rare opportunità di apprendimento e la possibilità di emozionarsi davanti alla bellezza delle opere esposte.

Quindi, l'utilizzo di qualsiasi sistema digitale all'interno di essi, comporta certamente di capire come può essere integrato senza rovinare il fascino dell'ambiente, contribuendo a aumentare la percezione e la partecipazione dei visitatori.

In particolare si dovrebbe cercare di ottenere congiuntamente:

- un set di interazioni che si desiderano in questi ambienti,
- una tecnologia che possa implementare le interazioni il più possibile invisibile e non invasiva (flessibile e facile da spostare).

Ne consegue che lo scopo primario di qualsiasi tecnologia che viene impiegata all'interno di un museo, è quello di sostenere silenziosamente e in modo trasparente il messaggio e la conoscenza che una particolare mostra trasmette ai suoi visitatori, in modo flessibile, adattandosi alle particolari condizioni o vincoli (ad esempio, impostazione fisica della mostra) che vi si possono trovare.

Occorre considerare, però, che tali tipi di interazioni, per essere apprezzate ed utilizzate, dovrebbero essere sia *intuitive*, che *collocate in modo accurato*.

Un'analisi approfondita dello stato dell'arte rivela che anche i visitatori condividono questo approccio, accettando, per la maggior parte, interfacce naturali per l'interazione uomo-macchina costruite su quelle tecnologie che riducono l'utilizzo di qualsiasi dispositivo hardware [35, 36, 33, 26].

Quindi si dovrebbe pensare all'implementazione di sistemi che fanno uso solo di una (o più) webcam, in questo modo è possibile riclassificare le tre istanze (di cui si è parlato nella sezione precedente), come segue:

- Rilevamento intangibile

Le interazioni di questo tipo possono essere catturate in modo molto semplice, ad esempio se un visitatore (ovvero il suo blob) occupa un

determinato spazio (ad esempio la posizione su una parte dell'immagine, un'*area sensibile*), viene generata un certo tipo di risposta.

Un metodo di questo tipo non richiede algoritmi di segmentazione o di rilevazione della sagoma umana.

- Localizzazione intangibile

In questo caso le figure sono segmentate, ottenendo uno o più blob (di ogni figura umana rilevata), successivamente viene calcolato un punto rappresentativo per ognuno di essi, dando la possibilità di tracciarlo costantemente .

- Riconoscimento intangibile delle azioni

Infine, per questo tipo di servizio occorre riconoscere e tracciare più punti il più possibile significativi per ogni figura catturata, dato che un'azione può aver origine da qualsiasi parte del corpo umano e può essere più o meno articolata.

In tal caso, infatti, non solo vengono segmentati i blob del corpo, ma utilizzando un algoritmo efficiente si individuano le parti del corpo rilevanti.

Nella figura 2.6 è mostrato un piccolo schema che mette in risalto le differenze sotto il profilo geometrico [44].

2.5 La soluzione adottata

Partendo dai casi di studio descritti nella sezione (2.3), si sono analizzati pregi e difetti, considerando come vincolo importante il contesto di un ambiente pubblico o una mostra in un museo; vi sono inoltre estrapolate informazioni (e tecniche) utili che sono state poi implementate.

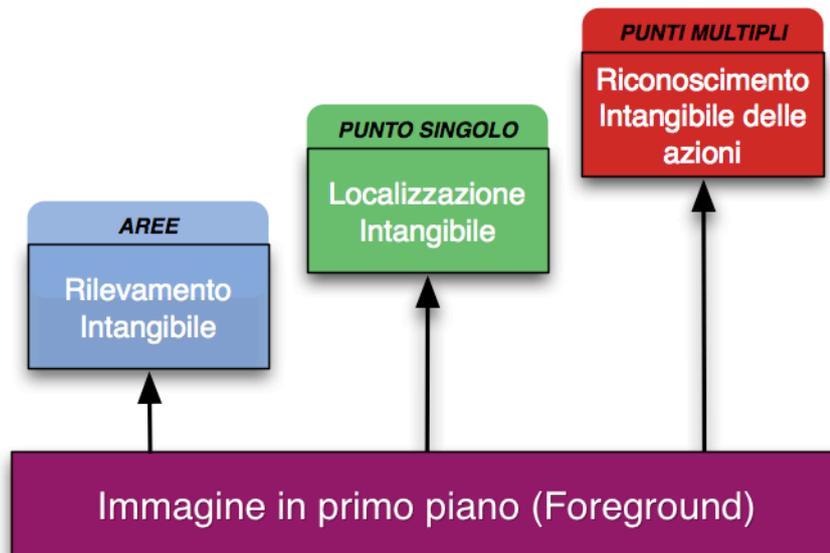


Figura 2.6: le tre tipologie di servizi intangibili discusse

Di seguito verrà presentata la soluzione adottata, come estensione di una metodologia dimostratasi efficace quanto semplice da implementare.

2.5.1 Metodo dei Massimi e Minimi

Per sviluppare il sistema, si è fatto riferimento ad un metodo proposto da Rocchetti et al., applicato con successo più volte, sia in ambiente museale, più precisamente al museo di Palazzo Poggi a Bologna (dove l'utente poteva sfogliare mappe antiche come se realmente sfogliasse l'artefatto), sia in ambiente pubblico, ovvero al Shanghai World Expo nell'ottobre 2010, con il progetto Tortellino X-Perience (dove veniva proposto ai giocatori di seguire e copiare le azioni di una sfoglina nel preparare un tortellino) [45, 35, 46]. La figura 2.7 mostra le installazioni.

Entrambi i casi di studio utilizzano una webcam (posta in posizioni diverse, a seconda, ovviamente, di come il gesto deve essere effettuato), e un



Figura 2.7: In alto Tortellino X-Perience, in basso l'installazione a Palazzo Poggi.

metodo che prevede l'utilizzo di una fase di segmentazione delle mani e l'identificazione dei punti di interesse (ovvero il punto più alto raggiunto dalle dita) analizzando i massimi locali del contorno del blob delle mani trovato.

In particolare, quando un visitatore posiziona le braccia all'interno di un frame, che viene catturato posizionando la webcam in posizione sopraelevata sopra il giocatore, una prima mano può essere immediatamente individuata nel punto massimo, ottenuto in qualsiasi direzione del blob catturato (figura 2.8). Il problema, quindi, si riduce a trovare la posizione della seconda mano.

Questa non può essere identificata con la stessa metodologia di prima, ovvero scegliendo il secondo massimo, che, come si evince nel grafico *b* della

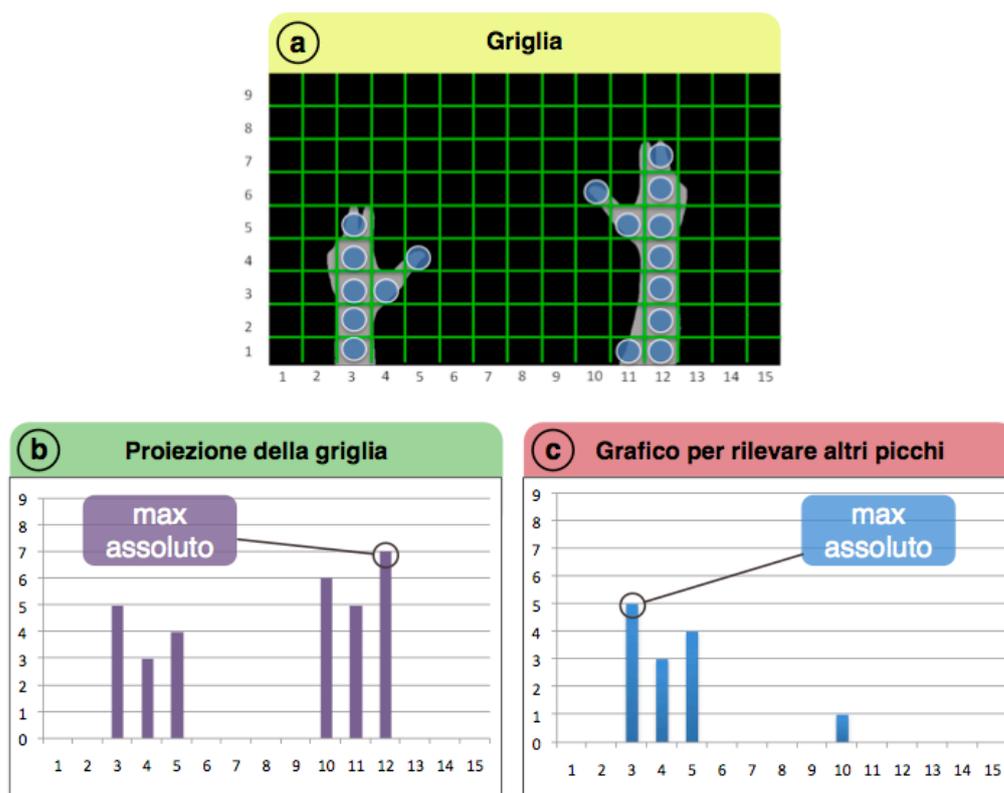


Figura 2.8: Come vengono trovati i due punti che identificano le mani in Tortellino X-Perience.

figura 2.8, potrebbe rappresentare il pollice della stessa mano.

Questo problema può essere facilmente risolto osservando che esisterà sempre un *gap* (ad esempio, un valore minimo) tra due massimi che rappresentano le due mani, a meno che le due mani non si sovrappongano.

Pertanto viene utilizzata una procedura di filtraggio, che si basa sul *gap* che si osserva tipicamente tra i due massimi che identificano i contorni delle due mani; in questo modo è possibile distinguere tra i punti che sono rilevanti da quelli che non lo sono e quindi identificare precisamente le due mani (immagine *c* della figura 2.8).

Una volta che le due mani vengono rilevate, all'interno di ogni fotogram-

La soluzione adottata

ma, si riconoscono le azioni con un metodo di cui si parlerà nei capitoli seguenti.

Utilizzando sempre lo stesso metodo, inoltre, si possono rilevare anche movimenti a grana fine effettuati con le mani [47].

Nel prossimo capitolo verrà descritta questa procedura e come è stata estesa l'idea sopracitata per poter identificare fino a cinque varie parti del corpo (testa, mani e piedi).

Capitolo 3

Descrizione del sistema: una panoramica

Considerando i concetti espressi nel capitolo precedente, si cercherà di esporre un metodo che sia in grado di implementare tutte le operazioni di cui sopra con l'utilizzo di un semplice dispositivo hardware come una (o più) telecamera od una webcam.

Si dimostrerà come riesca ad identificare una o più persone e riconoscere fino ad un massimo di 5 parti del corpo per ogni persona rilevata.

In questo capitolo si esporrà il sistema sviluppato in questa tesi, fornendo una nuova metodologia, che implementa un algoritmo per il riconoscimento di persone e delle loro azioni e che soddisfa i requisiti richiesti da un'installazione multimediale in un contesto museale.

In particolare si fornirà un ipotetico scenario, si parlerà delle caratteristiche, di come si è pensato di estendere il *modello dei massimi e dei minimi* discusso nelle sezioni precedenti (cap. 1.5), presentandone l'architettura ad un alto livello di astrazione, infine, di come rilevare il movimento a partire

dai punti calcolati.

3.1 Caratteristiche e configurazione

Le caratteristiche dell’algoritmo proposto che divergono dalle altre soluzioni, sono:

- **Semplicità**, in quanto non utilizza tecniche di image analysis come “skin detection” od “feature detection”.
- **Immediatezza**, perché non utilizza approcci basati su geometria a stella o che fanno uso di approssimazioni poligonali, in aggiunta, anche grazie la prima caratteristica, il costo computazionale viene abbattuto.

Il sistema proposto, utilizzando una webcam posizionata frontalmente rispetto ai visitatori, tiene traccia della posizione di piedi, mani e testa di una o più persone.

Un ipotetico scenario può essere rappresentato dalla in figura 3.1, dove due visitatori si muovono davanti a una mostra od un artefatto, mentre vengono rilevati in tempo reale, sia la loro posizione che i punti corrispondenti alle suddette parti del corpo.

3.2 Passi principali

Si possono individuare i passi principali del sistema proposto, come soluzione dei seguenti problemi:

- Che tecnica adottare per **segmentare la figura umana**.
- Quali **parti del corpo prendere in considerazione**, per far sì che un movimento potesse essere riconosciuto, senza però appesantire la



Figura 3.1: Due visitatori interagiscono con l'opera museale.

computazione, ovvero cercando di ottimizzare il rapporto *precisione/performace* e come ricavarle.

- Una volta ottenuti i suddetti punti, quale metodo sviluppare per la **rilevazione (e classificazione) di particolari movimenti, e quali movimenti prendere in considerazione.**

Supponendo, per facilità, che la figura di ogni giocatore venga analizzata separatamente, ci si può concentrare sul blob di ogni singola persona come se fosse isolata. Nelle sezioni successive saranno analizzati i diversi steps, tenendo conto di questa preconditione.

3.3 Riconoscimento del corpo

Il primo passo da effettuare è il cercare di capire come circoscrivere il blob che identifica un corpo umano.

Questa operazione sta alla base di ogni sistema per il rilevamento di azioni ed è detta *segmentazione*.

La segmentazione è il processo che permette di partizionare un'immagine in regioni significative. Viene utilizzata per ottenere una rappresentazione più compatta, estrarre degli oggetti e permette di partizionare le immagini digitali in insiemi di pixel. Lo scopo della segmentazione è semplificare e/o cambiare la rappresentazione delle immagini in qualcosa che è più significativo e facile da analizzare.

Algoritmi che implementano tale tipo di operazione sono stati studiati a lungo e migliorati [48, 49]. Metodi più semplici sono quelli che adottano tecniche di sottrazione dell'immagine di sfondo: gli oggetti in primo piano vengono rilevati come differenza tra il frame corrente e una rappresentazione (ottenuta con diversi metodi) che rappresenta lo sfondo statico.

Il sistema implementato utilizza un algoritmo di sottrazione dello sfondo veloce ed abbastanza avanzata, che, attraverso permette di rilevare figure in primo piano anche con la presenza di luce e sfondo non omogeneo ed è discretamente robusto in presenza di ombre [50].

Una volta ottenuta l'immagine di foreground, viene trasformata in una *griglia di aree sensibili*, ovvero una sorta di modello degli oggetti in primo piano. La granulosità di questa griglia, ovvero la dimensione in pixel di ogni cella, è determinante per la precisione che si vuole ottenere nel riconoscimento dei punti e dovrebbe essere configurata anche in base alla distanza (massima e minima) dalla quale i visitatori interagiscono con il sistema.

Nella figura 3.2 è mostrato un esempio di come avviene la sottrazione dello sfondo in un fotogramma e come l'immagine segmentata viene trasformata

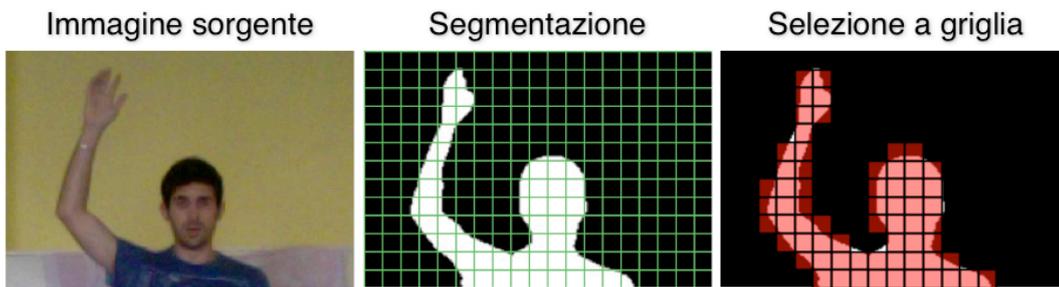


Figura 3.2: Tre steps che illustrano graficamente come si arriva alla selezione a griglia.

in una griglia di aree sensibili.

3.4 Estrazione dei punti del corpo

Si può pensare al processo di definizione dei punti del corpo, importanti per descrivere un'azione, come ad una funzione di trasformazione τ che, sulla base di considerazioni geometriche, prende in input tutti i pixel dell'immagine segmentata e fornisce una stima di tutti i punti ricercati in funzione delle seguenti variabili:

- (a) la posizione prevista di un visitatore (ad esempio seduto , in piedi, ecc);
- (b) quali punti rilevare (ad esempio, cercare solo la testa di un visitatore, o anche le sue mani);
- (c) le caratteristiche dell'ambiente in cui si è collocati (ad esempio, il livello d'illuminazione);
- (d) la posizione della webcam (o delle webcam);
- (e) il principio adottato per il filtraggio

Mentre dal punto (a) al (d) sono chiari, poiché riguardano tutti le proprietà fisiche della zona nonché le proprietà dei punti che sono richiesti, occorre definire il punto (e).

Questo infatti è l'algoritmo, che viene adottato per individuare i punti corrispondenti dalla griglia di aree sensibili che rappresenta un modello degli elementi in primo piano dell'immagine.

Questo principio utilizzata due moduli "a cascata", in modo da essere applicato in situazioni diverse senza stravolgere l'architettura (figura 3.3)

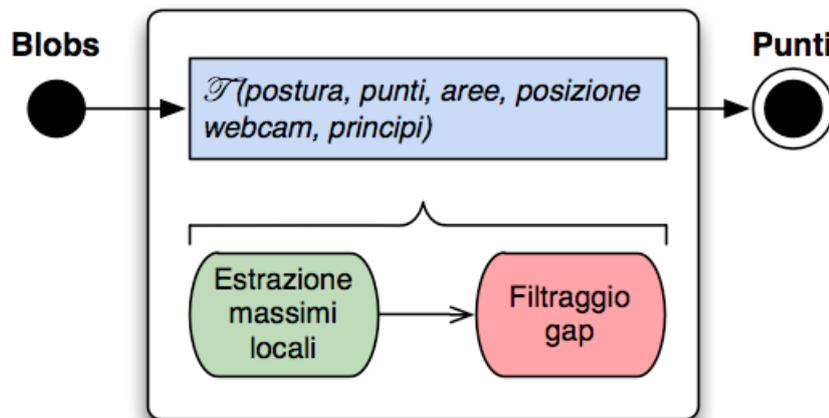


Figura 3.3: Estrazione dei punti dal blob.

3.4.1 Ricerca Massimi Locali

Questo primo modulo è incaricato di rilevare tutti i massimi locali che possono essere ottenuti dalle figure rilevate in foreground. Inizialmente ogni massimo locale è un potenziale punto di interesse, infatti, in questa fase non è detto che un punto rilevato sia effettivamente rappresentativo di una parte del corpo (potrebbe, ad esempio, essere un falso positivo).

3.4.2 Filtraggio dei Gap

Il secondo modulo viene utilizzato per eliminare i falsi positivi, basandosi su considerazioni dell'anatomia umana.

Proporzioni del corpo Anche se ci sono sottili differenze tra gli individui, le proporzioni umane rientrano in un intervallo abbastanza standard. I disegnatori utilizzano come unità base di misura la *testa*, intesa come distanza dalla sommità del cranio fino al mento [51].

In letteratura sono individuate quattro classi di proporzioni umane:

- **persona media:** in genere alta 7.5 teste (testa compresa);
- **figura ideale:** alta 8 teste, utilizzata per enfatizzare grazia o dare l'impressione di nobiltà;
- **figura fashion:** alta 8.5 teste e largamente usata dall'industria della moda per dare l'illusione di bellezza con linee allungate;
- **figura eroica:** alta 9 teste, un caso estremo di esagerazione, utilizzato nei fumetti.

La figura 3.4 rende meglio l'idea di questa classificazione e di come variano le proporzioni in tutto il corpo utilizzando una classe piuttosto che un'altra. Per il sistema, si sono utilizzate le proporzioni della prima classe, in quanto, utilizzandola, ha fornito ottimi risultati in fase di testing.

Pensando, invece, alle proporzioni come vincoli fisici, si può assumere che ogni coppia di massimi adiacenti, corrispondenti a punti di interesse, sono separati almeno da un valore di minimo che sarà inferiore ad una certa soglia (il cui valore dipende dai punti che vengono valutati di volta in volta).

La figura 3.5 come questi valori vengono utilizzati per definire la linea mediana e trasversale del corpo (in bianco), dove, la linea orizzontale definisce la

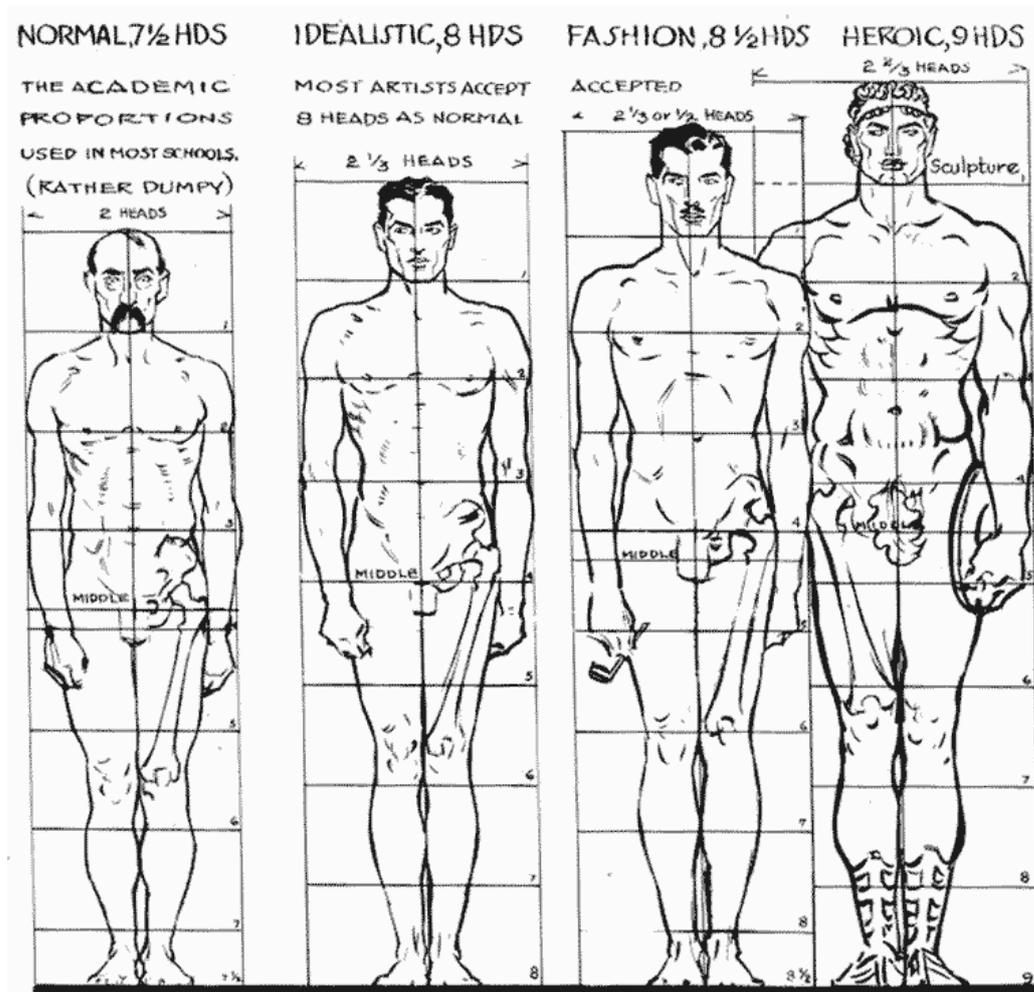


Figura 3.4: Le quattro classi di proporzioni umane.

metà superiore ed inferiore del corpo, mentre la linea spezzata verticale rappresenta un'approssimazione per calcolare la linea longitudinale (in verde), utile per definire la postura.

3.5 Rilevazione dei movimenti

Per rilevare i movimenti sono stati studiati sostanzialmente due approcci: il primo basato su zone sensibili che stimano la “traiettoria” del movimento

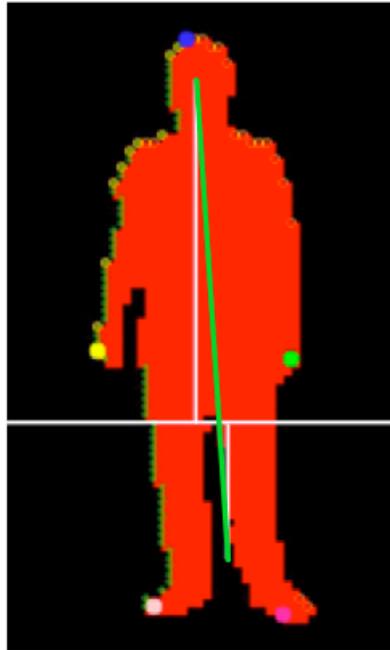


Figura 3.5: Linee che definiscono le proporzioni.

[52], mentre il secondo utilizza un algoritmo che valuta l'insieme di punti ottenuti, normalizzandoli, e restituendo se questo insieme corrisponde, data una certa tolleranza, ad un template del gesto che si vuole riconoscere [53]. In figura 3.6 è presentato un ipotetico caso d'uso in cui vengono utilizzati.

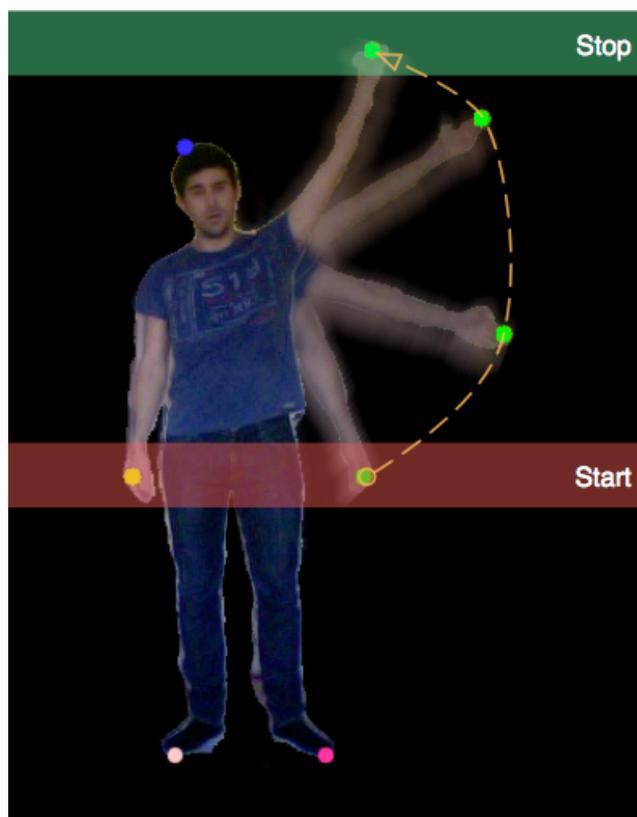


Figura 3.6: Esempio di come vengono rilevati i movimenti.

Capitolo 4

Descrizione del sistema: dettagli implementativi

Dopo aver presentato l'idea generale e valutato la panoramica del sistema, si vedrà quale approccio è stato seguito per l'implementazione del modello presentato.

Il sistema, come mostrato in figura 4.1, si può sostanzialmente suddividere in tre macro-fasi distinte:

- una **fase di preprocessing** effettuata su ogni frame catturato, che permette di segmentare una o più figure umane
- una **fase di conversione** che permette di convertire l'immagine binaria ottenuta in un modello a griglia, che agevola l'operazione successiva e ne alleggerisce la computazione [54].
- una **fase di estrazione dei punti** per ottenere le diverse parti del corpo (mani, piedi e testa)
- una **fase di estrazione del movimento**.

In questo capitolo si mostreranno più in dettaglio i vari steps dell'algoritmo (figura 4.3), affrontando anche i diversi problemi che si sono incontrati

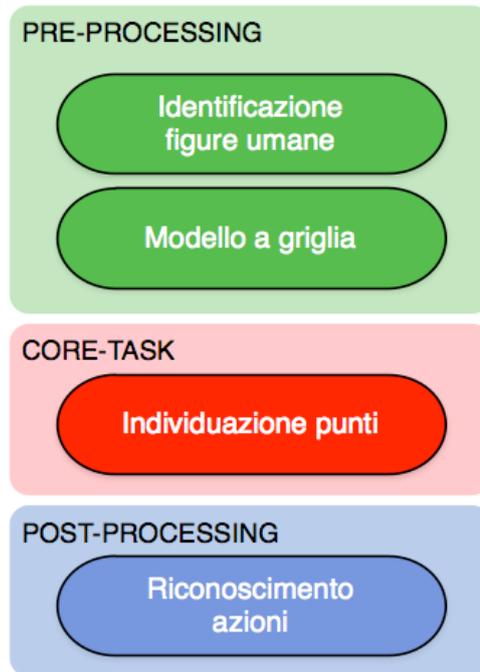


Figura 4.1: Fasi principali del sistema.

durante lo sviluppo, preceduti da una breve premessa sulle scelte progettuali affrontate.

4.1 Scelte progettuali

Il software realizzato ha lo scopo di analizzare un flusso video, il sistema hardware utilizzato è quindi un normalissimo personal computer dotato di una webcam.

I flussi video solitamente hanno un frame rate di circa 25 fotogrammi per secondo, ovvero in un secondo vengono visualizzate 25 immagini in successione. Scendendo al di sotto di tale limite, l'occhio umano comincia a percepire un flusso video in maniera non fluida. Se volessimo analizzare tutti i frame in tempo reale, il sistema di elaborazione dovrebbe essere in grado di analizzare un frame ogni 40 millisecondi.

La complessità di analizzare un frame è influenzata principalmente dalla sua risoluzione. In questo caso si prende in considerazione una risoluzione di 640x480 pixel.

Per implementare il software di identificazione e riconoscimento dei volti, si è deciso di utilizzare come linguaggio di programmazione *C++* in ambiente Mac Os X (il sorgente è comunque compilabile agevolmente in altri ambienti), e come editor Eclipse.

La decisione di utilizzare *C++* è stata dettata principalmente da due motivazioni:

- il software sviluppato in tale linguaggio garantisce una **capacità computazionale** maggiore rispetto ad altri linguaggi come Java,
- esistono delle **librerie complete ed affidabili** che contengono implementate un insieme di funzioni destinate all'elaborazione e all'acquisizione di immagini, oltre ad una vasta gamma di algoritmi ottimizzati ed efficienti.

Più nello specifico, in riferimento all'ultimo punto sopracitato, è stata utilizzata la libreria OpenCV, descritta in breve nel prossimo paragrafo.

4.1.1 OpenCV

Dalla sua introduzione nel 1999 (sviluppata inizialmente sotto Intel), è stata largamente adottata come strumento di sviluppo principale da parte della comunità di ricercatori e sviluppatori in computer vision. In questo campo, è infatti, al momento, una delle migliori librerie disponibili.

OpenCv è multiplatforma, open source e fornisce nativamente anche un wrapper in Python, infine è rilasciata sotto licenza BSD, quindi liberamente utilizzabile per scopi sia accademici che commerciali.

La libreria vanta più di 2500 algoritmi ottimizzati per l'immagine analysis [55].

La figura 4.2 mostra la struttura della libreria, la componente CV contiene funzioni per l'elaborazione base delle immagini e di algoritmi ad alto livello per la computer vision, ML è la libreria di machine learning (classificatori, algoritmi per reti neurali). HighGUI contiene le routine di I/O e le funzioni per il salvataggio e il caricamento di video e immagini, infine CXCore contiene le strutture dati di base.

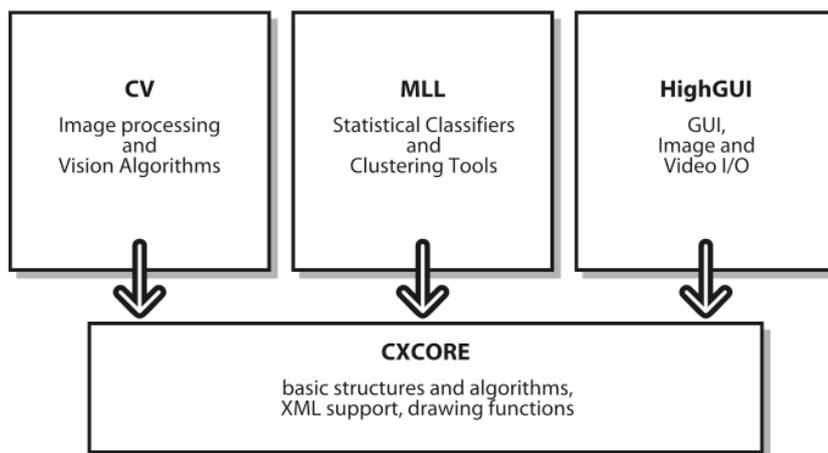


Figura 4.2: Architettura della libreria OpenCV.

Poiché le funzioni operano con immagini e con matrici numeriche, le dimensioni seguono rispettivamente la convenzione larghezza-altezza (X, Y) comune nella grafica e l'ordine righe-colonne (R, C) . La matrice di un'immagine è contenuta in un oggetto `cv::Mat()`, mentre un punto (che può rappresentare un pixel di un immagine) è contenuto nell'oggetto `cv::Point(x, y)`.

4.2 Segmentazione

Come si evince nell'immagine 4.3 l'operazione di segmentazione consta di tre metodi che permettono di ottenere, separatamente, le figure in primo piano.

4.2.1 Filtri di smoothing

Filtrare un immagine significa eseguire alcune operazioni in modo da esaltare o attenuare alcune sue caratteristiche.

I filtri di smoothing sono usati per il blurring dell'immagine e per la riduzione del rumore. L'operazione di blurring è normalmente utilizzata in fase di pre-elaborazione, allo scopo di eliminare piccoli dettagli inutili o addirittura dannosi per le successive elaborazioni, ovvero di compensare piccole imperfezioni quali le interruzioni che spesso si verificano nelle linee di contorno.

Queste imperfezioni sono solitamente generate dalla stessa sorgente video che, se troppo sensibile o deteriorata, può produrre del rumore o degli artefatti nell'immagine. Nella figura 4.4 si possono notare due tipi di rumore, additivo ed impulsivo, che possono essere prodotti [56].

Utilizzare un filtro, significa effettuare un'operazione di *convoluzione* definita nel dominio discreto e bidimensionale, che consta di due parametri in input: la **matrice dell'immagine** ed il filtro, rappresentato da una matrice più piccola detta **maschera di convoluzione**¹. La convoluzione fra matrici mira a calcolare il nuovo valore di ogni pixel dell'immagine originale, sovrapponendo ad ognuno di essi la matrice di convoluzione (con il centro sul punto in questione) ed eseguendo la sommatoria dei prodotti fra ogni pixel ed il

¹In genere è una matrice quadrata di ordine dispari

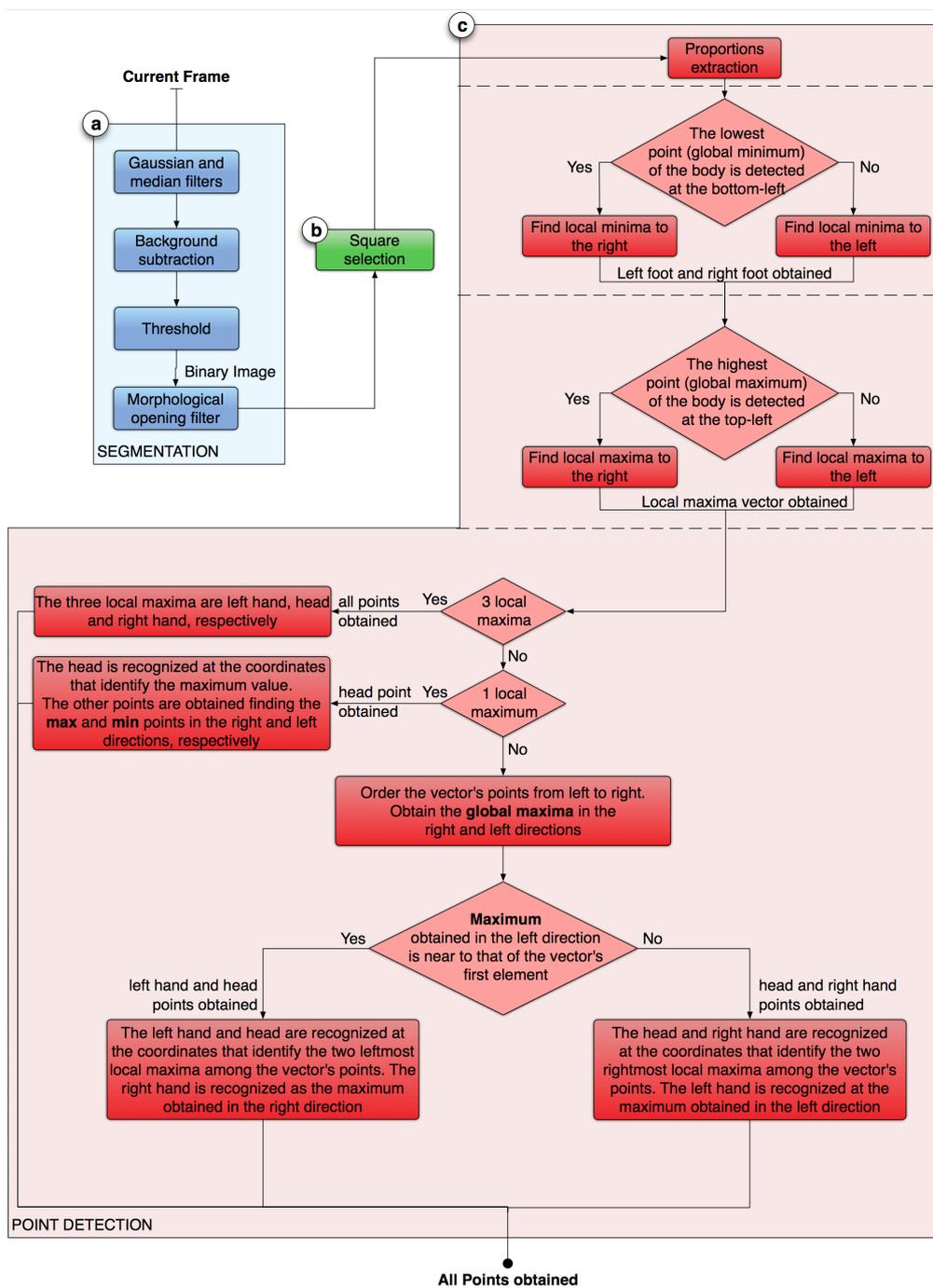


Figura 4.3: Diagramma di flusso per rilevare i punti.



Figura 4.4: Tipi di rumore: (a) immagine originale, (b) rumore additivo, (c) rumore impulsivo (salt-and-pepper)

corrispettivo elemento nella matrice di convoluzione, la figura 4.5.

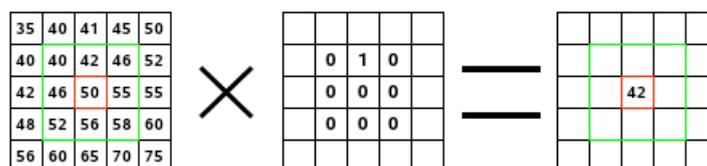


Figura 4.5: Alla sinistra la matrice dell'immagine: ogni pixel è marcato con il suo valore. Il pixel iniziale ha un bordo rosso. L'area di azione del kernel è quella con il bordo verde. Al centro il kernel e a destra il risultato della convoluzione.

In base alla tipologia del rumore, si utilizzano diversi filtri per la correzione dell'immagine. Per il sistema si è scelto di utilizzare un filtro gaussiano e mediano. Nelle sezioni seguenti si vedrà come sono stati applicati.

Filtro Gaussiano

Il filtro gaussiano è un filtro basato sulla funzione gaussiana che assegna un peso tanto maggiore quanto più il pixel è vicino all'origine, ovvero più è

larga la “campana” della funzione e maggiore sarà l’effetto di smoothing. Il nuovo valore del pixel sarà dato dalla media pesata dei valori nel suo vicinato.

Questa tipologia di filtro è abbastanza efficace nella rimozione di strappi nell’immagine (come graffi o dettagli inutili), e per la rimozione dell’artefatto JPG (dato che alcune webcam ne sono soggette), la figura 4.6 permette di apprezzarne l’efficacia. Essendo inoltre un filtro lineare² è sicuramente efficiente.

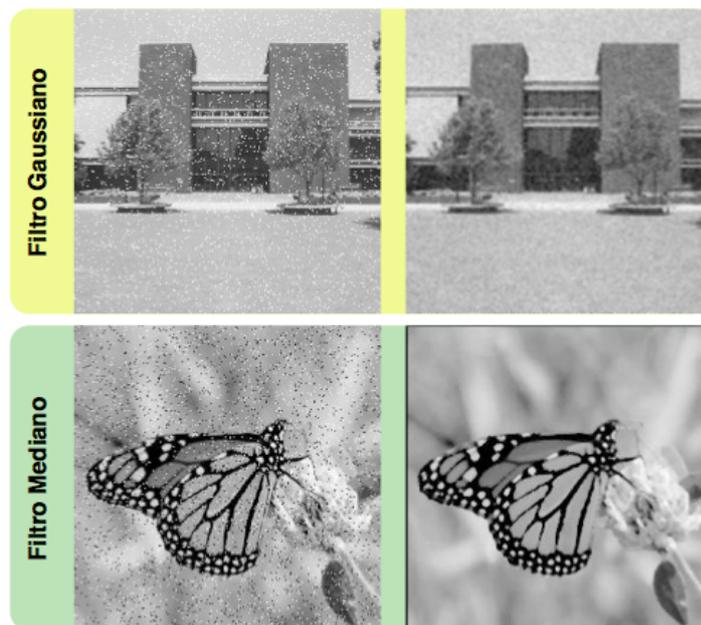


Figura 4.6: Filtri Gaussiano e Mediano all’opera: a sinistra l’immagine perturbata, a destra l’immagine filtrata.

Per la sua implementazione si è scelto di utilizzare una finestra (ovvero la grandezza della matrice di convoluzione del filtro) di 9x9 pixel.

```
1 cv::GaussianBlur(image, result, cv::Size(9,9));
```

²Un filtro di tipo lineare utilizza i pesi assegnati nella matrice di convoluzione ed effettua una semplice operazione aritmetica.

Filtro Mediano

Il principio di funzionamento di questo filtro è quello di lavorare su singolo campione, andandolo a sostituire con il valore mediano³ dei suoi vicini, rappresentati dalla matrice di convoluzione, con al centro il valore da sostituire, un esempio immediato del suo funzionamento è mostrato in figura 4.7. Appartiene alla classe dei filtri non lineari.

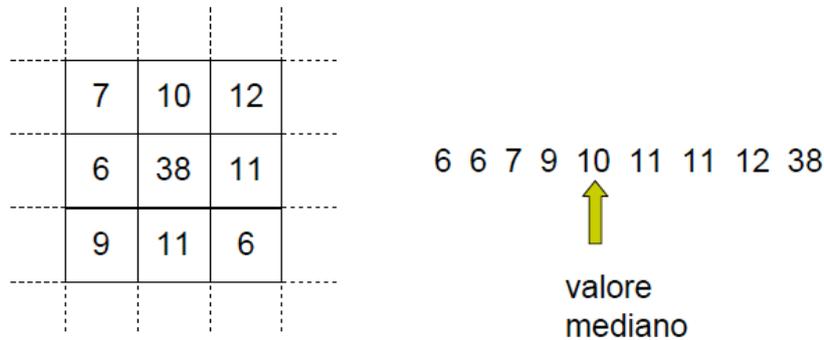


Figura 4.7: A sinistra la finestra (3x3) corrente nell'immagine analizzata: i valori di intensità vengono ordinati e viene scelto il valore mediano, che si andrà a sostituire al pixel centrale della maschera (ovvero, corrispondente al valore 38).

Quindi l'effetto del filtro mediano è di forzare i pixel ad assumere un valore uguale a quello di uno dei pixel circostanti, eliminando così eventuali spike isolati di intensità, infatti, viene spesso utilizzato per rimuovere tipologie di rumore come *salt-and-pepper*. Un esempio della sua efficacia si può osservare nella figura 4.6 dove l'immagine viene quasi completamente ricostruita.

Per la sua implementazione si è scelto di utilizzare una finestra (ovvero la grandezza della matrice di convoluzione del filtro) di 3x3 pixel, data la sua tendenza a distorcere gli elementi dell'immagine, e a uniformare le zone con intensità simile, oltre ad abbassare il costo computazionale.

³Il mediano M di un insieme di valori è tale che metà dei valori sono minori di M e metà dei valori sono maggiori di M .

```
1 cv::medianBlur(image, result, 3);
```

4.2.2 Sottrazione dello sfondo

La sottrazione dello sfondo comprende un insieme di tecniche atte a confrontare un'immagine osservata (il frame corrente, che si sta analizzando) con una stima dell'immagine che non contiene oggetti di interesse. Le zone dell'immagine, in cui vi è una significativa differenza tra l'immagine osservata e la stima, indicano la posizione degli oggetti di interesse.

Considerando di non sapere a priori l'ambiente dell'installazione e quindi l'eterogeneità dello sfondo, si deve assumere che le immagini catturate sono soggette a variazioni di luminosità, quindi durante la differenza, si potrebbero produrre degli artefatti. Inoltre, una scorretta illuminazione potrebbe proiettare delle ombre che, se di intensità elevata, sarebbero rilevate anch'esse come elemento in primo piano.

Per ottenere una buona immagine di foreground, quindi, non basta una semplice sottrazione dello sfondo ottenuto da un solo frame, ma occorre prevedere un modello di esso, dove certe variazioni nei pixel possono essere accettabili (sotto una certa soglia).

Si è quindi utilizzato un metodo che utilizza un modello adattivo gaussiano, implementato recentemente con la classe `BackgroundSubtractorMOG2` fornita da `OpenCv`.

Questo metodo permette di aggiornare dinamicamente il modello dello sfondo, includendo quelle variazioni ricorrenti tra i fotogrammi (ad esempio, se un certo elemento nello sfondo riflette la luce, potrebbe far variare l'intensità dei pixel che lo compongono). L'aggiornamento dinamico però deve essere disabilitato (in modo autonomo) una volta che un visitatore inizia ad utilizzare il sistema.

Per fare ciò, basta cambiare i parametri dell'operatore fornito da `BackgroundSubtractorMOG2`, ovvero:

- **nframes** indica quanti fotogrammi utilizzare nel modello;
- **history** stabilisce quanto deve essere lunga la storia, ovvero ogni quanti frame aggiornare il modello

Di seguito si propone un esempio di come utilizzare l'operatore:

```
1 BackgroundSubtractorMOG2 mog;
2 ...
3 for (;;) { // ad ogni iterazione viene catturato un frame dalla sorgente
4     ...
5     mog(frame, foreground, -1); // inizializzo l'operatore
6     mog.nframes = 1000;
7     mog.history = 800;
8     mog.getBackgroundImage(bgimg); // ottengo il modello dello sfondo
9     ...
10 }
```

Nel codice sia `frame` che `foreground` sono matrici delle immagini, ovvero oggetti di tipo `cv::Mat()`, la prima rappresenta il frame corrente, la seconda l'immagine dove sono presenti solo le figure in primo piano.

Alla fine di questa operazione, si effettua un'operazione di sogliatura⁴ sull'immagine di `foreground` ottenendo un'immagine binaria, dove i pixel bianchi rappresentano gli elementi di interesse (la figura umana) e quelli neri tutto il resto.

```
1 threshold(foreground, foreground, 128, 255, THRESH_BINARY);
```

4.2.3 Filtro morfologico: Apertura

I filtri morfologici definiscono una serie di operatori che trasformano un'immagine sondandola con un elemento di forma predefinita (allo stes-

⁴Permette di separare un istogramma in due regioni di interesse: una che include tutti i pixel con livello di grigio inferiore a quello cercato e l'altra contenente i pixel rimanenti.

so modo di una matrice di convoluzione). Il modo in cui questo elemento interseca la vicinanza di un pixel determina il risultato dell'operazione [55].

Le trasformazioni morfologiche di base sono chiamate *dilatazione* ed *erosione* e si presentano in una grande varietà di contesti, ma, utilizzati su un'immagine binaria, permettono di eliminare o assottigliare gli oggetti di scena (blobs) oppure accrescere gli oggetti o riempire delle aree vuote.

Erosione

L'erosione di una immagine binaria A è definita formalmente in termini di traslazione dell'elemento strutturante B :

$$A \ominus B = \{z \mid (B_z) \subseteq A\}$$

Dove, $A \ominus B$ è l'insieme dei punti z dell'immagine tale che B traslato in z sia contenuto interamente in A . L'effetto della trasformazione è quello di contrarre le regioni corrispondenti agli oggetti. Ad esempio, questo tipo di operazione si può eseguire sulle immagini ottenute da una binarizzazione nel caso in cui oggetti che dovrebbero essere separati risultano erroneamente connessi.

Nell'esempio in figura 4.8, viene utilizzato un elemento strutturante quadrato 3×3 e permette di rimuovere dall'oggetto tutti i punti che hanno almeno un vicino appartenente allo sfondo. In pratica, i contorni dell'oggetto vengono contratti di un pixel in tutte le direzioni.

Dilatazione

La dilatazione di una immagine binaria A tramite un elemento strutturante B è definita come:

$$A \oplus B = \{z \mid (B'_z) \cap A \neq \emptyset\}$$

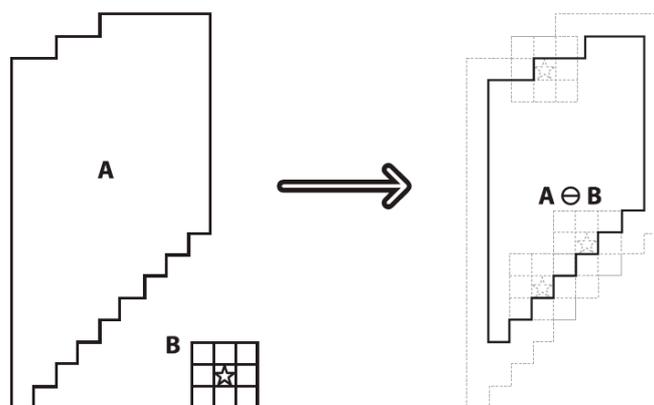


Figura 4.8: Filtro morfologico di erosione con un elemento strutturante 3x3 quadrato.

L'elemento strutturante viene riflesso rispetto alla sua origine e spostato di z posizioni mediante una traslazione. La Figura 4.9 mostra un esempio di dilatazione dell'insieme A tramite un elemento strutturante quadrato B simmetrico ($B' = B$). Il risultato è l'insieme di tutti i punti di posizione z per cui B' ed A si sovrappongono almeno in un punto. La trasformazione è estensiva poiché l'insieme originario è contenuto nell'insieme dilatato.

La dilatazione è utilizzata per migliorare la qualità delle immagini ottenute dalla sogliatura nei casi in cui delle regioni presentano lacune, oppure quando parti di un oggetto che dovrebbero essere connesse risultano invece frammentate.

Apertura

Le operazioni di erosione e dilatazione per uno stesso elemento strutturante permettono di definire operatori più complessi. Eseguendo l'erosione di un insieme A attraverso un elemento strutturante B e la dilatazione del risultato nuovamente tramite B , otteniamo l'operazione di *opening* (o apertura).

$$A \circ B = (A \ominus B) \oplus B$$

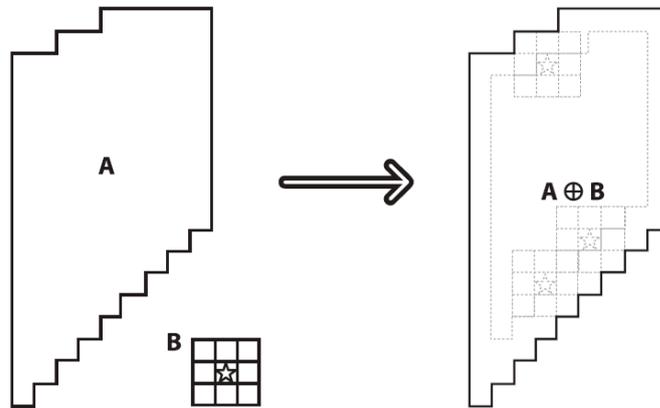


Figura 4.9: Filtro morfologico di dilatazione con un elemento strutturante 3x3 quadrato.

L'effetto è quello di preservare le regioni di forma simile all'elemento strutturante e di eliminare quelle differenti. L'operazione di apertura elimina i dettagli chiari più piccoli dell'elemento strutturante, rende più omogenei i contorni degli oggetti ed elimina le piccole interruzioni, la figura 4.10 rende l'idea di come funziona.

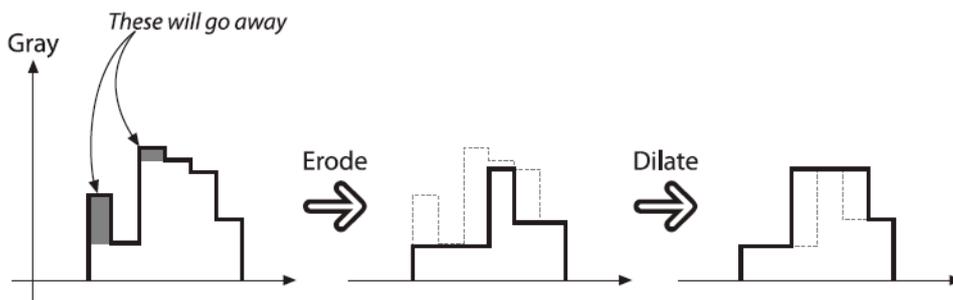


Figura 4.10: Filtro morfologico di apertura con un elemento strutturante 3x3 quadrato.

Poiché i filtri di smoothing applicati in precedenza possono distorcere (di solito aumentando la zona rilevata come foreground), si utilizza un filtro di

apertura per definire meglio la figura umana. Ad esempio, se una persona si posiziona con le braccia alzate, sopra la testa, quest'ultima potrebbe non essere rilevata.

A tal scopo si è prima definito un elemento strutturale a forma ellittica, di diametro 5x5, di seguito un esempio di come utilizzare questa operazione.

```
1 structElement = getStructuringElement(MORPH_ELLIPSE, Size(5, 5));  
2 morphologyEx(foreground, foreground, MORPH_OPEN, structElement);
```

Dove `foreground` è l'immagine binaria ottenuta dalla sogliatura.

4.2.4 Identificazione dei blobs

Una volta ottenuta un'immagine binaria pulita, dove i pixel bianchi sono considerati parte degli elementi in `foreground`, viene effettuata un'operazione di `labeling`⁵. Questa operazione consiste nello scorrere pixel per pixel la matrice dell'immagine binaria alla ricerca di componenti connesse⁶, che sono memorizzati in una struttura di OpenCV, che permettono di ottenere informazioni sui blobs trovati, in particolare l'area.

Successivamente, infatti, per ogni elemento della lista viene trovata l'area e se inferiore ad un certa soglia, viene eliminata. Questa operazione è utile per filtrare ulteriormente sia gli artefatti, sia elementi (piccoli) che possono cambiare in modo repentino dal modello di sfondo.

Si sono così ottenute le sagome dei visitatori.

Separazione delle figure

Durante l'azione può accadere che le persone si tocchino, o che siano abbastanza vicini facendo sì che la segmentazione individui un solo blob.

⁵Il `labeling` delle di un'immagine binaria è un'operazione mediante la quale a ciascun blob viene associata un'etichetta (`label`) distinta

⁶Una componente connessa di un'immagine binaria è una regione connessa di dimensione massima costituita da pixel marcati come oggetto.

Assumendo che, inizialmente, i giocatori vengano individuati separatamente, si può arginare il problema attuando due approcci:

- **Ulteriore erosione:** se i giocatori sono sovrapposti leggermente, o ad esempio, con un solo arto, applicando un filtro di erosione si possono riottenere due blobs separati.
- **Considerare la storia:** mantenendo un buffer di qualche frame precedente al contatto, è possibile stimare (in modo molto approssimato) attraverso una media, la posizione degli arti di ogni singolo blob.

In questi casi, però, ottenere dei punti validi è difficile, si preferisce quindi visualizzare un messaggio che interrompe temporaneamente l'azione e che chiede ad entrambi di fare, ad esempio, un passo nella direzione opposta all'altro.

4.3 Conversione in griglia

Questa fase permette di convertire ogni blob ottenuto in una sua rappresentazione, a grana più grossa, che permette sia di accedere in modo più agevole al contorno della figura, sia di abbattere il costo computazionale dato che non si deve scorrere ogni volta tutta la matrice dell'immagine, ma solo un suo sottoinsieme.

Questa idea, proposta inizialmente da Roccetti et al. 2010 [35], e modificata per questo sistema, permette di ottenere una sorta di modello delle figure in foreground, come si evince in figura 3.2.

Per costruirla, si divide l'immagine binaria di foreground in una griglia statica, dove ogni cella ha una dimensione M , tale dimensione può essere cambiata in base alla tipologia di installazione in cui il sistema può essere applicato.

Successivamente si memorizzano, per ogni cella, il numero di occorrenze dei pixel appartenenti ai blobs (come se si sovrapponesse la griglia all'immagine binaria) e vi si calcola un indice K (ovvero la percentuale, per ogni cella, di quanti pixel bianchi occupano la cella), se superiore ad una certa soglia, la cella viene considerata "attiva" (o colorata).

Nei test effettuati, mantenendo una distanza di circa 4-5 metri, è stato stimato un valore ottimale di M pari a 5×5 pixel, ed un valore di K pari a 0.4.

4.4 Estrazione dei punti

Prima di addentrarsi nella descrizione dell'algoritmo per la definizione dei punti, è importante fare una premessa su come il sistema distingue i vari punti trovati.

Successivamente si esporrà in dettaglio i passaggi che si possono osservare nel punto (c) del diagramma in figura 4.3.

4.4.1 Classi di punti

Il sistema è in grado di distinguere tra due classi di punti:

- **Precisi:** rappresentano i punti identificati con un ottimo valore di precisione (ad esempio, il caso più favorevole si ha quando per ogni direzione si trova un solo massimo)
- **Approssimati:** rappresentano invece quei punti che non sono stati individuati con precisione, ma si presume siano in una determinata posizione, utilizzando le considerazioni sulle proporzioni (già accennate precedentemente, ma di cui si parlerà nella sezione successiva).

La maggior parte delle volte, vengono individuati tutti punti appartenenti alla prima categoria.

In alcuni casi, come si può notare in figura 4.11, anche definendo un *punto approssimato*, si ottiene una posizione non corretta; questo accade soprattutto quando una, od entrambe mani sono vicine o davanti al busto. Per evitare questi errori di approssimazione, si è introdotta una sorta di “area insensibile”, leggermente più larga del busto, dove qualsiasi punto trovato al suo interno non viene riconosciuto.

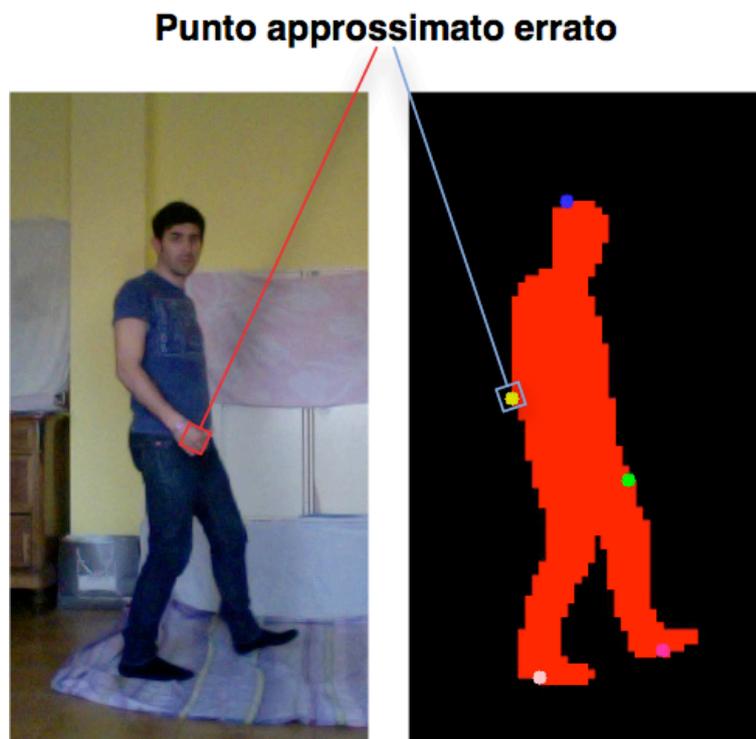


Figura 4.11: Punti approssimati: esempio di rilevazione errata.

In realtà una soluzione più robusta, proposta successivamente nel capitolo 5, consiste nell’aggiungere una webcam posizionata sopra i visitatori.

4.4.2 Popolazione delle liste di riferimento

Per “*liste di riferimento*” si intendono due liste, rispettivamente popolate nelle due direzioni del piano cartesiano e che contengono le zone che forniscono un “risultato positivo” della griglia di aree sensibili, cioè quei settori in cui è stata accertata una figura in primo piano.

La griglia viene ciclata sia per righe che per colonne e memorizzato il punto corrispondente per ciascuna area considerata occupata da una figura in primo piano.

In questo modo si può avere un accesso diretto (e in un tempo costante) in ciascun punto che costituisce il contorno della figura; inoltre, con una semplice differenza si può valutare la distanza di ciascun punto dalla sua adiacente, e in ogni direzione.

```
1 POPULATE_ROW_REFERENCED_LIST(grid, rowsList)
2 begin
3   for i:=0 to GRID_HEIGHT do begin
4     for j:=0 to GRID_WIDTH do begin
5       if (IS_ACTIVE(grid[i][j]))
6         rowsList.ADD(i,j);
7     end
8   end
9 end
```

4.4.3 Estrazione delle proporzioni

Questa fase permette di poter ottenere, per ogni step seguente, i punti necessari, senza che la definizione dei vari gap sia disturbata da altre parti del corpo diverse dalla regione di interesse (ROI).

In base alle proporzioni di cui si è parlato nella sezione 3.4.2, sono stati quindi definiti i seguenti limiti che serviranno allo scopo:

- La parte bassa del blob (**lowerBody**): ottenuta dividendo l'altezza totale del blob per 7.5, il valore così ottenuto moltiplicato per 5, usato come limite per definire le due parti del blob.

- La linea mediana della parte superiore del blob (**upperBodyMedian**): ottenuta prendendo il punto medio della larghezza della parte superiore del blob diviso utilizzando **lowerBody**.
- La linea mediana della parte inferiore del blob (**lowerBodyMedian**): ottenuta prendendo il punto medio della larghezza della parte inferiore del blob diviso sottraendo **lowerBody** all'altezza totale del blob.

4.4.4 Rilevazione dei piedi

Questa operazione viene eseguita trovando il minimo globale nella parte inferiore del blob, punto che rappresenterà il primo piede, mentre la ricerca per il secondo piede nelle regioni a destra o a sinistra, viene valutata a seconda di dove è stato trovato il minimo globale.

La procedura che segue viene utilizzata per trovare i piedi, quando il minimo assoluto si trova nella parte sinistra della linea mediana inferiore del corpo ($\text{min.x} < \text{lowerMedian}$).

In questo caso, vengono ciclati i quadrati (intese come celle della griglia) rispetto al profilo inferiore del corpo, e vengono quindi calcolati i picchi.

Nella procedura che segue si definiscono:

- **min**: il minimo assoluto;
- **righBound**: il valore rispetto l'asse dell'ascissa oltre il quale un punto non è considerato;
- **peaksValue**: vettore risultato contenente i valori dei picchi;
- **colsList**: la lista contenente i punti del contorno inferiore dell'immagine;
- **n**: numero di quadratini appartenenti al contorno inferiore dell'immagine.

Estrazione dei punti

```
1 FIND_MIN_PEAKE_FROM_LEFT(min, rightBound, peaksValues, colsList, n)
2   begin
3     minimum = min.y;
4     for i := min.x + 1 to n-1 do begin
5       if colsList[i][colsList[i].size() - 1].x < rightBound then
6         if ABS(MAX(colsList[i][colsList[i].size() - 1].y - minimum,
7                   0)) then
8           peaksValues[i] = colsList[i][colsList[i].size() - 1];
9         else
10          minimum = colsList[i][colsList[i].size() - 1].y;
11        endif
12      endif
13    end
14    return peaksValues;
15  end
```

La procedura restituisce un vettore di interi che contiene tutti i possibili punti candidati. Essi sono il risultato del massimo tra 0 e il valore ottenuto sottraendo il valore di y del punto corrente, al minimo (ovvero il valore di y dell'ultimo punto che non è decrementato).

Per ottenere il minimo relativo viene utilizzata la seguente procedura:

```
1 EXTRACT_RELATIVE_MIN_FROM_LEFT(peaksValues, tolerance, results, n)
2   begin
3     localMax, sequenceLength = 0;
4     for i:=0 to n-1 do begin
5       if peaksValues[i]>0 then
6         sequenceLength++;
7         if peaksValues[i] >= localMax then
8           localMax:=peaksValues[i];
9           localMax:=i;
10        endif
11        if i = n-1 then
12          results.ADD(localMax);
13        endif
14      else
15        if localMax > 0 then
16          if sequenceLength > tolerance then
17            results.ADD(localMax);
18          endif
19          localMax = 0;
20        endif
21      endif
22    end
23  end
```

```

22         endif
23     end
24 end

```

Dove `tolerance` indica quanto dovrebbe essere lunga una sequenza passi per iniziare il calcolo del minimo relativo, mentre `localMax` è il massimo locale trovato per ogni sequenza considerata valida.

Questa procedura è ripetuta in modo analogo se il minimo assoluto si trova a destra, cambiando l'intervallo del ciclo, da `min.x` a 0. Inoltre si dovrà considerare il limite (sull'asse orizzontale) a sinistra, invece di quello a destra.

Per ottenere i punti di piedi, vengono allineati i picchi in base al valore dell'ordinata, dal più piccolo al più grande. Il piede sinistro sarà il primo elemento, il piede destro sarà il secondo.

4.4.5 Rilevamento top-down

Il rilevamento top-down dei picchi è quella fase responsabile della ricerca, sfruttando nuovamente valori massimi e minimi, dei punti relativi che appartengono alla parte superiore del corpo (cioè, mani e testa).

La procedura che segue viene utilizzata per trovare i picchi, quando il massimo assoluto è posizionato nella parte destra della linea mediana superiore del corpo (`max.x > upperMedian`). Naturalmente, come per il passo precedente, la procedura proposta è analoga nel caso si trovi il massimo a sinistra, basta cambiare la direzione di ricerca.

```

1 FIND_MAX_PEAKE_FROM_RIGHT(max, leftBound, peaksValues, colsList, n)
2     begin
3         minimum = max.y;
4         for i := min.x+1 to 0 do begin
5             if colsList[i][0].x > leftBound then
6                 if ABS(MIN(colsList[i][0].y - minimum, 0))
7                     then
8                         peaksValues[i] := colsList[i][0];
9                 else

```

Estrazione dei punti

```
9             minimum := colsList[i][0].y;  
10             endif  
11         endif  
12     end  
13     return peaksValues;  
14 end
```

In figura 4.12 viene mostrato graficamente come viene effettuato il calcolo del massimo relativo.

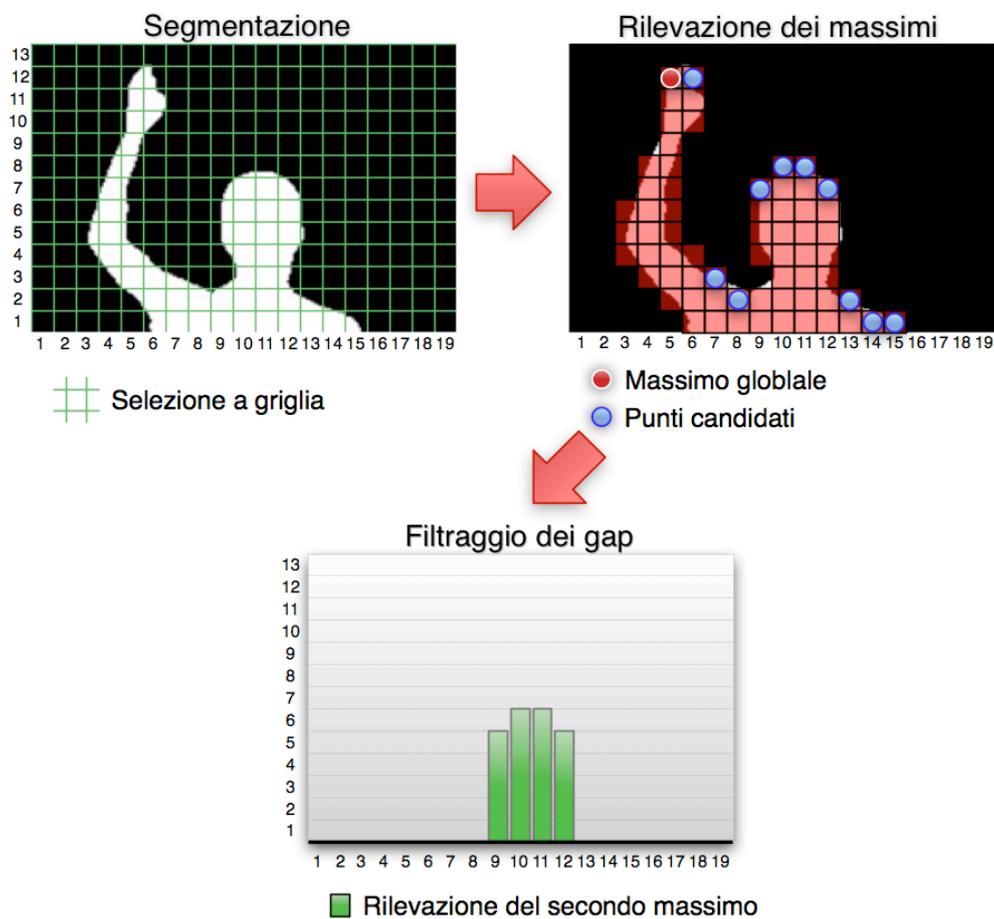


Figura 4.12: Come viene rilevata una mano e la testa.

La procedura restituisce un vettore di interi che consentono di ottenere i massimi relativi. Questi interi sono il risultato, in valore assoluto, del minimo

tra 0 e il valore ottenuto sottraendo il valore y del punto corrente al minimo, (ovvero il valore di y dell'ultimo punto che non è decrementato).

La procedura seguente è simile a quella della fase precedente, con la sola differenza che il *for* va dall'ultimo al primo elemento del vettore.

```

1 EXTRACT_RELATIVE_MAX_FROM_RIGHT(peaksValues, tolerance, results, n)
2     begin
3         localMax, sequenceLength = 0;
4         for i:=n-1 to 0 do begin
5             if peaksValues[i]>0 then
6                 sequenceLength++;
7                 if peaksValues[i] >= localMax then
8                     localMax:=peaksValues[i];
9                     localMax:=i;
10                endif
11                if i = 0 then
12                    results.ADD(localMax);
13                endif
14            else
15                if localMax > 0 then
16                    if sequenceLength > tolerance then
17                        results.ADD(localMax);
18                    endif
19                    localMax = 0;
20                    sequenceLength = 0;
21                endif
22            endif
23        end
24    end

```

4.4.6 Ottenimento dei punti rimanenti

Una volta trovati i picchi (ovvero i massimi relativi) della parte superiore del corpo, si può ricadere in tre possibili scenari:

- **un singolo massimo globale:** indica che è stata trovata solo la testa (ovvero, le braccia sono posizionate al di sotto delle spalle), in questo caso i due punti rimanenti saranno trovati nella fase successiva;

Estrazione dei punti

- **tre massimi locali:** indica il caso più fortunato, dove sono state trovate le posizioni di entrambe le mani e la testa (ovvero, le braccia si trovano entrambe sopra le spalle), basta ordinare il vettore utilizzando il valore dell'ascissa di ciascun punto trovato.
- **due massimi locali:** uno identifica chiaramente una mano, mentre l'altro la testa di un visitatore. In questo si deve definire se il primo picco, ordinando il vettore per il valore dell'ascissa di ciascun punto, corrisponde alla testa o alla mano sinistra. Una volta definiti, occorre solo calcolare il punto rimanente.

Per calcolare i punti rimanenti si usa una funzione che restituisce il massimo (o minimo) del piano cartesiano ruotato di 90° , in realtà si ha già tale valore nella prima lista popolata per colonne (`colsList`).

```
1 GET_LEFT_LIMB(colsList, n, topBodyY)
2     begin
3         for i:=0 to n-1 do begin
4             if colsList[i][0].x < topBodyY then
5                 return colsList[i][colsList[i].size-1]
6             else
7                 return GET_LAST_USEFUL_GAP(colsList[i]);
8     end
```

`topBodyY` viene utilizzato per escludere tutti i punti al di sotto di un certo valore (sull'asse delle ordinate) considerato come parte inferiore del corpo.

La funzione `GET_LAST_USEFUL_GAP` permette di assegnare un punto approssimato della mano (in modo da ottenere una migliore stima della posizione se, ad esempio, un visitatore ha le braccia parallele al corpo), restituendo il primo punto al di sopra di un gap (inteso come spazio) nella sequenza passata.

`GET_RIGHT_LIMB` è analoga, basta cambiare la direzione con cui si scorre `colsList` (dal primo all'ultimo elemento del vettore).

I confronti sui picchi rilevati dalla scansione top-down, di cui si è parlato sopra, vengono effettuati come segue:

```
1 tempRightLimb = GET_RIGHT_LIMB();
2 tempLeftLimb = GET_LEFT_LIMB();
3
4 if peaks.size = 1 then
5
6     rightLimb = tempRightLimb;
7     leftLimb = tempLeftLimb;
8     head = peaks[0];
9
10 else if peaks.size() = 2 then
11     // ordino da sinistra a destra(asse x)
12     SORT(peaks , COMPARE_X);
13     // il primo picco e' la mano sinistra
14     if ARE_NEIGHBOUR(peaks[0],tempLeftLimb then
15
16         leftLimb = peaks[0];
17         head = peaks[1];
18         rightLimb = tempRightLimb;
19
20     else // il primo picco e' la testa
21
22         leftLimb = tempLeftLimb;
23         head = peaks[0];
24         rightLimb = peaks[1];
25     endif
26
27 else
28
29     SORT(peaks , COMPARE_X);
30     leftLimb = peaks[0];
31     head = peaks[1];
32     rightLimb = peaks[2];
33
34 endif
```

Dove `ARE_NEIGHBOUR()` è una funzione che verifica se il valore, rispetto l'ascissa dei punti passati, diverge al massimo di 5 quadratini; restituisce `true` se vengono considerati lo stesso punto (o un'approssimazione della stessa mano), `false` altrimenti.



Figura 4.13: (a) Frame sorgente. (b) Blob corrispondente. (c) Risultato della trasformazioni in griglia e punti finali riconosciuti (blu=testa, verde=mano sinistra, giallo=mano destra, bianco=piede destro, rosa=piede sinistro).

Una rappresentazione dei punti che si trovano alla fine del processo è data dall'immagine (c) della figura 4.13.

4.5 Riconoscimento movimenti

Una volta ottenute le coordinate dei cinque punti che caratterizzano le parti del corpo di ogni giocatore, si è in grado di riconoscere un determinato movimento. Per fare questo occorre sapere a priori come e in quanti modi potrebbe essere riprodotto e rilevato.

Come già accennato nel cap 2.5, sono stati presi in considerazione due tipologie di approcci diversi, che a seconda del movimento specifico da valutare, possono essere utilizzati sia separatamente che congiuntamente.

In questo capitolo verranno mostrati i due diversi approcci, facendo una breve premessa su come si è affrontata la necessità di normalizzare una serie (catturata tra diversi frame) di punti, utilizzando il filtro di Kalman.

4.5.1 Filtro di Kalman discreto

Nel 1960, R. E. Kalman pubblicò un articolo dove veniva descritta una soluzione ricorsiva per il problema del filtraggio lineare di dati discreti.

Il *filtro di Kalman* è un insieme di equazioni matematiche che forniscono una soluzione (ricorsiva) computazionalmente efficiente del metodo dei minimi quadrati⁷.

Permette di determinare stime degli stati passati, presenti e futuri e questi risultati possono essere conseguiti persino quando la natura del sistema modellato non è conosciuta con precisione. Per questo motivo può essere particolarmente indicato (ed efficiente) per normalizzare i punti rilevati dal sistema quando, ad esempio, da un frame all'altro il set di punti ottenuto (da una sola parte del corpo) subisce delle variazioni inaspettate, dovute ad un errore.

Le equazioni del filtro di Kalman ricadono in due gruppi:

- **time update:** responsabili della previsione dello stato attuale e della covarianza dell'errore, valutate per ottenere una stima a priori per lo step successivo; possono anche essere pensate come equazioni di *predizione*.
- **measurement update:** responsabili del feedback e vengono impiegate per unire una nuova misurazione con la stima a priori al fine di ottenere una migliore stima a posteriori; rappresentano le equazioni di *correzione*.

In figura 4.14 sono rappresentate graficamente le due equazioni, che si richiamano in procedura ricorsiva.

⁷Tecnica di ottimizzazione che permette di trovare una **curva di regressione** (ovvero una funzione), che si avvicini il più possibile ad un insieme di punti del piano cartesiano.

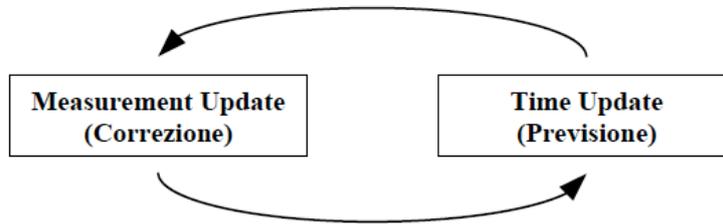


Figura 4.14: L'algoritmo del filtro di Kalman visto come una procedura ricorsiva in cui le equazioni di *time update* forniscono una previsione e quelle di *measurement update* determinano un miglioramento della stima introducendo l'informazione contenuta nella misurazione.

Alla fine dell'algoritmo, in uscita si ha un valore di x che è solamente approssimato al valore reale. Un esempio di come funziona a livello grafico lo si può osservare in figura 4.15.

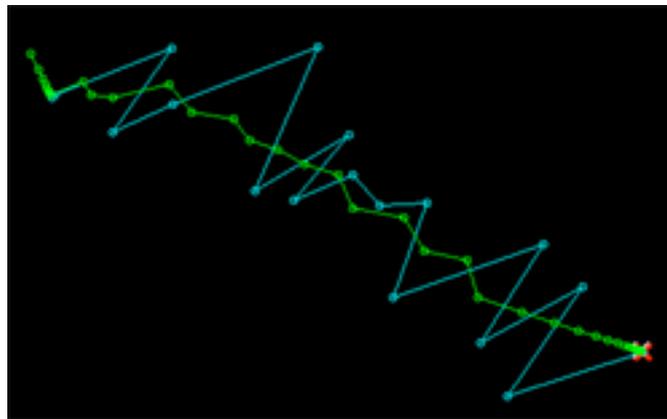


Figura 4.15: Applicazione filtro di Kalman: linee e punti azzurri sono i punti catturati, linee e punti verdi sono i punti normalizzati dal filtro.

OpenCV fornisce una classe per implementare il filtro di Kalman, segue un esempio di applicazione dell'algoritmo.

```
1 /* param1: dimensione degli stati  
2 * param2: dimensione delle misure
```

Descrizione del sistema: dettagli implementativi

```
3  * param3: dimensione del vettore di controllo
4  */
5  KalmanFilter KF(4, 2, 0);
6
7  /* (x, y, Vx, Vy), rappresentano gli stati */
8  Mat_<float> state(4, 1);
9  /* indica che il processo e' con disturbo */
10 Mat processNoise(4, 1, CV\_32F);
11
12 /* Matrice delle misurazioni */
13 Mat\_<float> measurement(2, 1);
14 measurement.setTo(Scalar(0));
15
16 /* popolazione delle prime due misure
17  * rispettivamente con la coordinata X
18  * e Y del mouse, le restanti sono settate
19  * a 0
20  */
21 KF.statePre.at<float>(0) = mouse_info.x;
22 KF.statePre.at<float>(1) = mouse_info.y;
23 KF.statePre.at<float>(2) = 0;
24 KF.statePre.at<float>(3) = 0;
25
26 /* matrice di transizione */
27 KF.transitionMatrix = *(Mat_<float>(4, 4) << 1, 0, 1, 0,
28                                     0, 1, 0, 1,
29                                     0, 0, 1, 0,
30                                     0, 0, 0, 1);
31
32 /* setta le matrici identita' */
33 setIdentity(KF.measurementMatrix);
34 setIdentity(KF.processNoiseCov, Scalar::all(1e-2));
35 setIdentity(KF.measurementNoiseCov, Scalar::all(1e-1));
36 setIdentity(KF.errorCovPost, Scalar::all(.1));
37
38 for (;;) {
39     /* calcola lo stato e il punto predetto */
40     Mat prediction = KF.predict();
41     Point predictPt(prediction.at<float>(0),
42                    prediction.at<float>(1));
43
44     /* setta le nuove misure (x e y) */
45     measurement(0) = mouse_info.x;
46     measurement(1) = mouse_info.y;
```

```
47
48     /* crea un punto stimato dalle misure precedenti*/
49     Point measPt(measurement(0), measurement(1));
50     mousev.push_back(measPt);
51
52     /* aggiorna lo stato stimato dalla misura */
53     Mat estimated = KF.correct(measurement);
54
55     /* converte la matrice in punto e lo inserisce nel vettore di stato */
56     Point statePt(estimated.at<float>(0),
57                  estimated.at<float>(1));
58     kalmanv.push_back(statePt);
59 }
```

4.5.2 Zone sensibili

Inizialmente proposto da Rocchetti et. al [54], si basa sull'idea che la maggior parte dei movimenti richiedono al giocatore di muovere (nel loro caso le mani) una determinata parte del corpo (o più parti) da una zona di **partenza** ad una di **destinazione**, lungo una **traiettoria** che le collega. Ognuna di queste zone può essere rappresentata come un'area sensibile e i punti interessati al movimento devono transitare dalla area sensibile di origine a quella di destinazione, attraversando una traiettoria rappresentata da più zone sensibili che si attivano/disattivano al passaggio del punto (figura 4.16).

Questo approccio permette soprattutto, in modo agevole, di valutare anche la velocità di un movimento, nel caso questo richieda di essere compiuto in un certo tempo; mentre rimane più limitativo quando i movimenti richiedono l'interazione con più punti o movimenti più complessi (o che non possono essere modellati con delle aree sensibili).

4.5.3 One dollar

Proposto da Wobbrock et al. [53], usa un metodo leggero e performante (consta di quattro steps ed è scritto in meno di 100 righe di codice). Permette

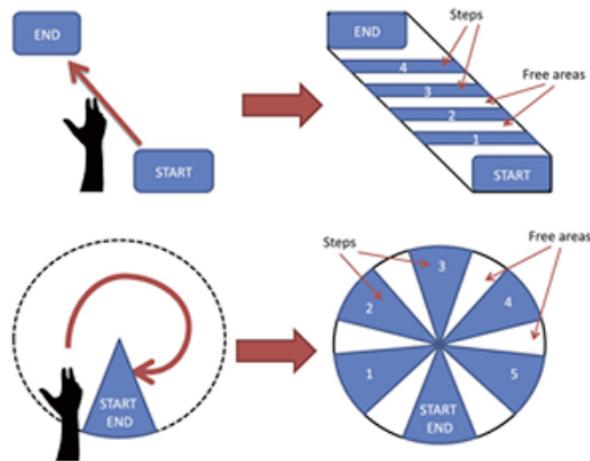


Figura 4.16: Riconoscimento tramite zone sensibili: esempi di applicazione.

di riconoscere un gesto valutandone il tracciato dei punti ottenuto durante l'esecuzione dello stesso (ovvero la traiettoria) e confrontandolo con un set di punti template.

I quattro steps sopracitati possono essere riassunti come segue:

- (a) Per ogni punto ricevuto si campiona un modello con 64 punti.
- (b) Si trova il centroide e ruota il path, in modo che il punto in cui è iniziato il movimento sia a 0° .
- (c) Riduce il path ad un rettangolo di riferimento.
- (d) Scorre i punti del path e trova l'angolazione giusta per avere un miglior riscontro con il modello.

La figura 4.17 mostra a livello visuale ogni step.

Il sistema permette, ovviamente, anche la definizione di nuovi template e non utilizza procedure matematiche complesse.

Questo approccio, però, siccome normalizza il set di punti ricevuto in input, non valuta il tempo, quindi non può essere utilizzato, così come viene

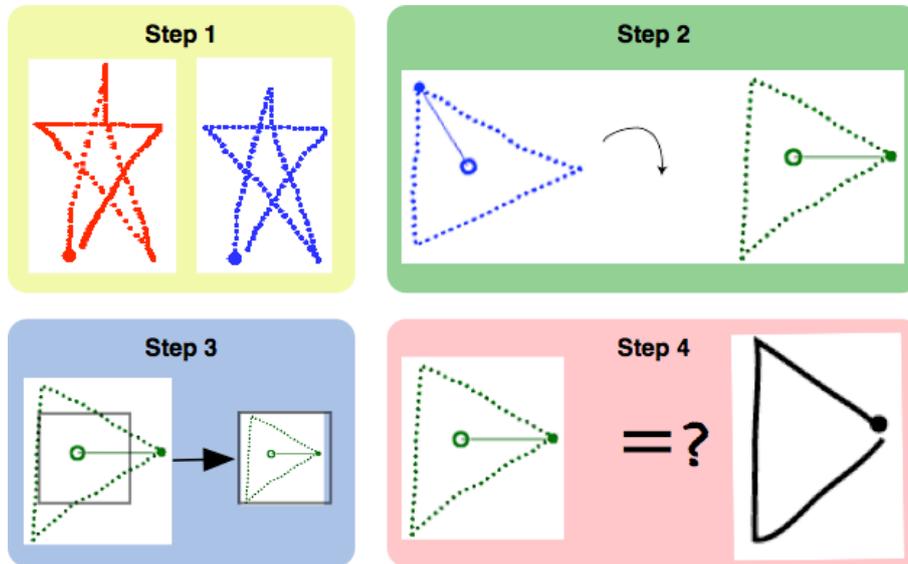


Figura 4.17: \$1 Dollar: steps dell'algorithm.

proposto, con gesti basati sulla velocità; inoltre il valore di tolleranza che viene assegnato ad ogni gesto è molto sensibile, se viene settata, ad esempio, ad un valore troppo basso possono essere riconosciuti come validi, anche gesti “perturbati”.

Capitolo 5

Risultati dei test ed analisi

Nonostante l'algoritmo non sia stato sperimentato in un reale contesto museale, si è cercato di testarlo in un ambiente che presentava caratteristiche simili ad un'installazione, ovvero:

- *ambiente controllato*: solo le persone che vogliono interagire con il sistema sono presenti nel campo visivo della webcam;
- *sfondo eterogeneo*: per ogni prova effettuata, lo sfondo preso in considerazione ha delle caratteristiche eterogenee (colori diversi, sfumature, ombre);
- *distanze contenute*: la distanza tra il giocatore e la locazione della webcam (e dello schermo) non supera i 6 metri¹.

In questo capitolo saranno presentati i risultati dei test effettuati, in particolare, per testarne (e rappresentarne) l'efficacia, si sono presi 57 fotogrammi differenti dove una persona partendo da una posizione statica (con le braccia conserte) inizia ad eseguire delle azioni.

¹Questo parametro dipende anche dalla risoluzione della webcam presa in esame o semplicemente dalla qualità/tipologia delle lenti che monta.

È importante specificare che per questi test si è disattivata la funzione che definisce “l’area insensibile”, leggermente più larga del busto, dove qualsiasi punto trovato al suo interno non viene riconosciuto.

Una parte del corpo è stata classificata come *rilevata* quando l’algoritmo restituiva un punto su di esso o ad un intorno stretto di esso; se ad esempio il punto destinato ad una delle due mani veniva assegnato all’avambraccio, questo non è stato ritenuto valido.

Nelle prossime sezioni saranno presentati i risultati dei test effettuati, i problemi riscontrati e le limitazioni del sistema.

5.1 Prestazioni

Nei risultati ottenuti in prima istanza, effettuando i test sui 57 frames, come si può notare nella tabella 5.1, salta subito all’occhio che in tutte le situazioni, la testa e i due piedi sono sempre rilevati correttamente, ottenendo una precisione perfetta.

Non è perfetta, né ottimale, invece, la precisione con cui l’algoritmo rileva gli arti superiori nel frameset analizzato.

La mano sinistra, infatti, non viene rilevata in quasi un fotogramma su tre, mentre la mano destra una volta su quattro.

	Rilevati	Tasso rilevati	Mancati	Tasso mancati
Testa	57	100%	0	0%
Mano sinistra	41	72%	16	28%
Mano destra	43	75%	14	25%
Piede sinistro	57	100%	0	0%
Piede destro	57	100%	0	0%

Tabella 5.1: Performance dell’algoritmo in tutto il frameset

Prestazioni

Sulla base di questi risultati, si è deciso di analizzare più in profondità, per quali frames l’algoritmo si rivelava efficace, e per quali invece non lo era. Tuttavia, in questo modo, si è anche osservato che l’algoritmo deve essere valutato più attentamente in tutte quelle situazioni in cui una persona esegue un’azione che sia utile ai fini del riconoscimento (ovvero, per tutte quelle azioni che possano essere valutate in un sistema di riconoscimento).

Perciò si sono divisi i 57 frames in due gruppi, dove, il primo gruppo include tutte quei fotogrammi che sono stati catturati mentre avveniva un qualche tipo di azione, mentre il secondo includeva anche i restanti. Nella figura 5.2, si può notare quali dei 57 frames appartengono al primo gruppo (sfondo verde), mentre i restanti (sfondo rosso) sono quelli che si sono esclusi.

Eseguendo nuovamente i test sul nuovo pool di 39 frames, si sono ottenuti i risultati riportati in tabella 5.2, che confermano che, quando viene effettuato un certo tipo di azione, la rilevazione delle parti del corpo è molto più precisa, ottenendo un “tasso di punti mancati” minore della metà rispetto al test precedente.

	Rilevati	Tasso rilevati	Mancati	Tasso mancati
Testa	39	100%	0	0%
Mano sinistra	34	87%	5	13%
Mano destra	33	85%	6	15%
Piede sinistro	39	100%	0	0%
Piede destro	39	100%	0	0%

Tabella 5.2: Performance dell’algoritmo quando viene effettuata un’azione

Questo può essere facilmente spiegato osservando le differenze tra i fotogrammi appartenenti al primo gruppo e quelli che appartengono al secondo (dove non si sta eseguendo alcuna attività).

Si noti come quasi tutti i frame che appartengono al secondo gruppo, infatti, hanno il blob dove la persona presenta le braccia conserte o rilassate, in prossimità del tronco. I frame che sono del primo gruppo, invece, nella grande maggioranza dei casi, rivelano che la persona è intenta in un'azione, estendendo le estremità (braccia e gambe).

5.2 Limiti e possibili errori

Data la semplicità della metodologia nell'ottenere i punti (seppur precisi), il sistema non è esente da limitazioni ed errori:

- La metodologia applicata ai fotogrammi catturati da una webcam frontale non trova sempre in modo preciso le posizioni delle mani, dato che è impossibile con un approccio del genere individuare i punti degli arti che si trovano vicino al corpo (se paralleli, con le braccia conserte, o se vengono sporte le mani avanti al corpo).
- Se una persona sporge di poco i gomiti, o mette le mani sui fianchi, i gomiti stessi vengono rilevati come arti superiori 5.1.

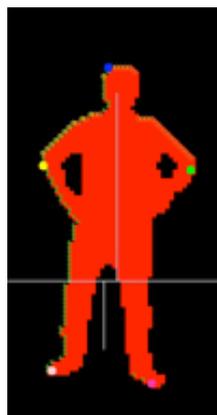


Figura 5.1: Esempio di errore.

Limiti e possibili errori

- Se vi sono più di due giocatori nella scena, si rilevano più di due blobs separati l'efficienza diminuisce notevolmente (questo è stato rilevato provando l'algoritmo con un normale notebook).

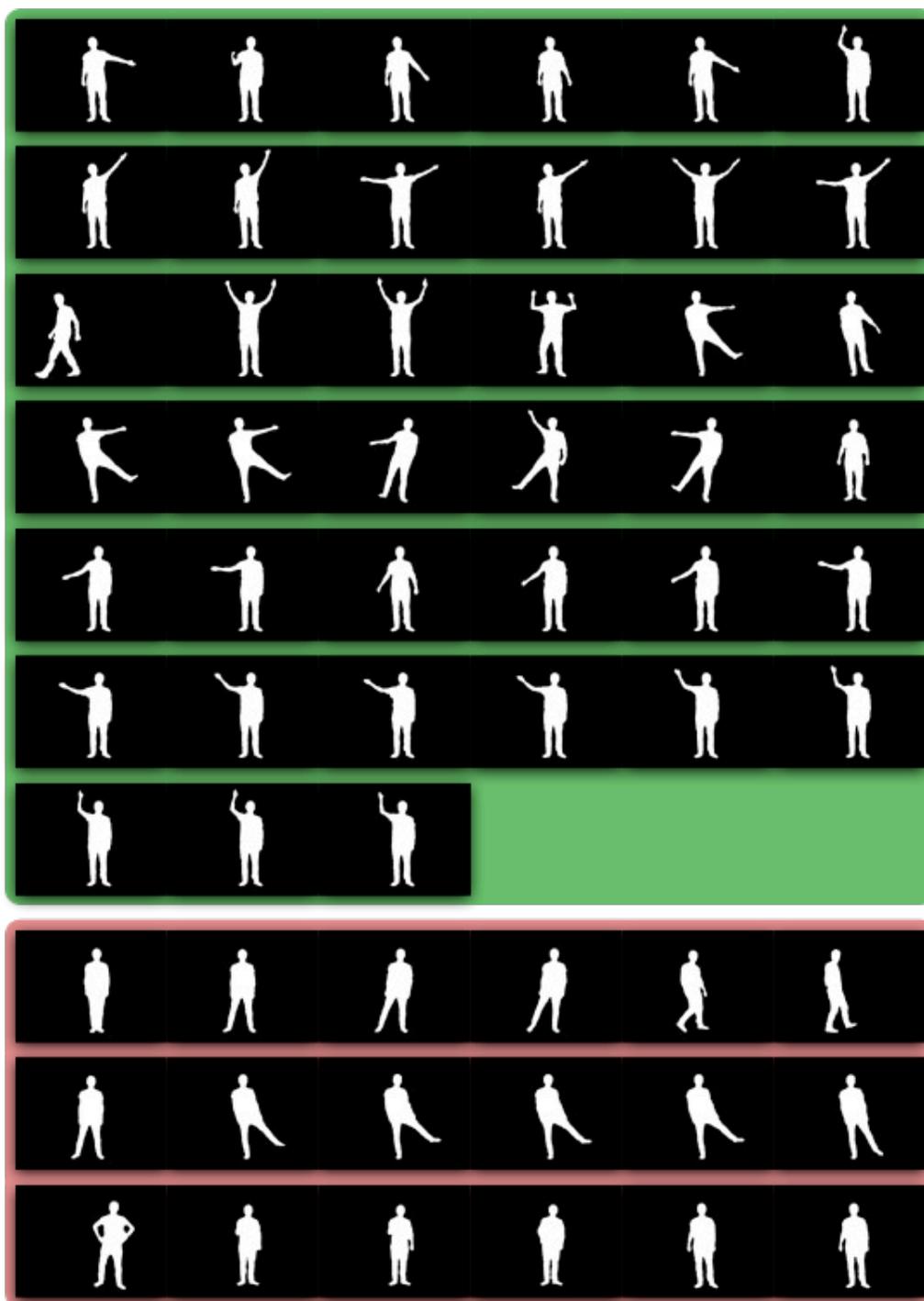


Figura 5.2: Il frameset analizzato.

Conclusioni

In un museo si possono incontrare ambientazioni eterogenee e diverse a seconda degli artefatti esposti e del luogo in cui sono collocati. Si è mostrato come le nuove tecnologie che utilizzano interfacce naturali permettono interazioni interessanti tra i visitatori e le opere; sebbene siano stati proposti molti sistemi, nella maggioranza dei casi appaiono come adattamenti, anziché approcci mirati al particolare contesto in esame.

Si è spiegato come la maggior parte delle sistemi di computer vision utilizzati in un contesto museale ricadono in una delle seguenti categorie: (a) rilevazione, (b) monitoraggio e (c) riconoscimento delle azioni dei visitatori.

Alla luce di quanto detto si è valutato che una valida soluzione dovrebbe essere il più possibile flessibile e non invasiva. Si è dimostrato che per rispettare queste caratteristiche non occorre richiedere ai visitatori l'utilizzo di un qualsiasi dispositivo hardware, ma è sufficiente una semplice webcam e una serie di algoritmi originali.

Si è descritto dapprima l'idea come estensione del “metodo dei massimi e dei minimi” proposto anche in occasione dell'installazione Tortellino X-Perience e rassegnata una prima panoramica, successivamente si è mostrata la parte implementativa.

Infine, si ha dimostrato la validità della metodologia, soprattutto sotto il profilo dell'efficienza, fornendo ottimi risultati dei test.

Sviluppi futuri

Grazie agli ambiziosi obiettivi del progetto, sono possibili numerosi sviluppi del lavoro svolto finora; alcuni sono mirati al miglioramento e al consolidamento di quanto già realizzato; altri sono estensioni del sistema al fine di risolvere le problematiche sopracitate (e considerare quelle che non sono ancora state affrontate).

Segue una rassegna dei principali sviluppi possibili:

- Introdurre una migliore classificazione delle proporzioni del corpo, non solo per definire le linee generali dei blobs trovati, ma anche come vincoli per il miglior posizionamento di punti approssimati. Inoltre si potrebbe, sempre partendo dalle proporzioni, definire ulteriori punti interessanti (ad esempio, potrebbe essere interessante rilevare i punti delle principali articolazioni, per capire l'esatta postura del visitatore).
- Analizzare le celle geometricamente all'interno della griglia della sagoma umana (o meglio di un bounding-box della griglia), che non sono "accese", potrebbe fornire ulteriori informazioni sulla postura, quindi una migliore approssimazione dei punti.
- Introdurre una seconda webcam, sopra la testa dei visitatori (ovvero che riprenda dall'altro verso il basso), permetterebbe di avere una visione tridimensionale dello scenario, quindi valutare la profondità e riconoscere gesti effettuati nelle tre dimensioni. Inoltre risolverebbe in parte uno dei problemi sopracitati, permettendo di riconoscere le mani quando, ad esempio, un giocatore pone le mani davanti al corpo.
- Aumentare la velocità computazionale, sia migliorando le funzioni attuali, sia utilizzando il calcolo parallelo tramite GPU per l'applicazione dei filtri sull'immagine. OpenCV attualmente integra il calcolo paral-

lelo tramite GPU e la collaborazione con Nvidia ha portato ad una veloce integrazione della libreria CUDA con GPU.

Ringraziamenti

Prima di tutto vorrei ringraziare il mio relatore, il Professor Marco Rocchetti, per la sua guida attenta durante il progetto riuscendo a proteggere la mia autonomia, sostenendo il mio lavoro. Da lui ho imparato che avere una visione più ampia dei problemi, è più importante che concepire algoritmi complessi. In secondo luogo, i miei ringraziamenti vanno al Dott. Gustavo Marfia per le discussioni costruttive sul mio lavoro e per i consigli che mi ha dato durante lo sviluppo che mi hanno permesso di non “divergere troppo” dall’obiettivo.

Inoltre, vorrei esprimere la mia sincera gratitudine ai miei compagni di corso, in particolare a Catia per la pazienza dimostrata in tante occasioni e per le sue inimitabili battute, a Mattia per i consigli preziosi e per il suo aiuto provvidenziale durante i diversi progetti, a Giovanni per la sua costanza e gli appunti, a Antonio per la sua compagnia soprattutto durante quest’ultimo periodo e a Andrea, per la sua completa disponibilità.

Ringrazio sentitamente la mia ragazza, Beatrice, senza le sue preziose parole di incoraggiamento non sarei riuscito a laurearmi in questa sessione, e senza le sue correzioni questo documento sarebbe un campo minato di errori ortografici.

Infine, ringrazio i miei famigliari, per la pazienza e per il sostegno datomi durante il mio percorso.

Bibliografia

- [1] Baklouti, M., Monacelli, E., Guitteny, V. & Couvet, S., “*Intelligent assistive exoskeleton with vision based interface*,” in Proceedings of the 6th international conference on Smart Homes and Health Telematics, Vol. 5120 of Lecture Notes In Computer Science, Springer-Verlag, Berlin, Heidelberg, 2008.
- [2] Nickel, K. & Stiefelhagen, R., “*Visual recognition of pointing gestures for human-robot interaction*”, Image Vision Computing 25, 2007.
- [3] Kevin, N.Y.Y.; Ranganath, S.; Ghosh, D.; , “*Trajectory modeling in gesture recognition using CyberGloves®and magnetic trackers*” TENCON 2004. 2004 IEEE Region 10 Conference , vol.A, no., pp. 571- 574 Vol. 1, 21-24 Nov. 2004
- [4] Schlömer, T., Poppinga, B., Henze, N., & Boll, S. (2008).“*Gesture recognition with a Wii controller.*” (A. C. M. Press, Ed.)Proceedings of the 2nd international conference on Tangible and embedded interaction TEI 08, 11.
- [5] Sato, M., Poupyrev, I, and Harrison, C. “*Touché: Enhancing Touch Interaction on Humans, Screens, Liquids, and Everyday Objects.*” In Proceedings of CHI’12. 2012. ACM.
- [6] Microsoft Kinect “*Kinect for Windows - Developers*”. <<http://www.microsoft.com/en-us/kinectforwindows/develop/new.aspx>>.

- [7] Keller, C.G.; Enzweiler, M.; Rohrbach, M.; Llorca, D.F.; Schnorr, C.; Gavrilu, D.M.; , “*The Benefits of Dense Stereo for Pedestrian Detection*” Intelligent Transportation Systems, IEEE Transactions on , vol.12, no.4, pp.1096-1106, Dec. 2011
- [8] Fang, G., Gao, W. & Zhao, D. (2007), “*Large-vocabulary continuous sign language recognition based on transition-movement models*”, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 37(1), 1–9.
- [9] Keskin, C., Balci, K., Aran, O., Sankur, B. & Akarun, L. (2007), “*A multimodal 3d healthcare communication system*”, in ‘IEEE 3DTV Conference’, IEEE Computer Society Press, pp. 1–4.
- [10] Lokavee, S.; Watthanawisuth, N.; Mensing, J.P.; Kerdcharoen, T.; , “*Sensor pillow system: Monitoring cardio-respiratory and posture movements during sleep*”, Biomedical Engineering International Conference (BMEiCON), 2011 , vol., no., pp.71-75, 29-31 Jan. 2012
- [11] Hiroshi, K., Makito, S., Kazuhiko, S., Ken-ichi, T. & Kazuo, K. (2006), “*Pattern recognition for video surveillance and physical security*”, Technical Report 375, The Institute of Electronics, Information and Communication Engineers.
- [12] Cohen, C.J.; Morelli, F.; Scott, K.A.; , “*A Surveillance System for the Recognition of Intent within Individuals and Crowds*”, Technologies for Homeland Security, 2008 IEEE Conference on , vol., no., pp.559-565, 12-13 May 2008
- [13] Bay H., Fasel B. and Van Gool L., “*Interactive Museum Guide: Fast and Robust Recognition of Museum Objects*,” in Proc. of the First International Workshop on Mobile Vision, Graz, 2006.

- [14] Stock O., Zancanaro Z., Busetta P., Callaway C., Kruger A., Kruppa M., Kuflik T., Not W. and Rocchi C., “*Adaptive, intelligent presentation of information for the museum visitor in PEACH*,” User Modeling and User-Adapted Interaction, Springer, vol. 17, n. 3, pp. 257-304, 2007.
- [15] Papagiannakis G., Singh G. and Magnenat-Thalmann N., “*A survey of mobile and wireless technologies for augmented reality systems*,” Comput. Animat. Virtual Worlds, vol. 19, n. 1, pp. 3-22, 2008.
- [16] Rocchetti M., Marfia G., Amoroso A., Caraceni S. and Varni A., “*Augmenting Augmented Reality with Pairwise Interactions: The Case of Count Luigi Ferdinando Marsili Shooting Game*,” in Proc. of the 4th IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology (DENVECT’12) - 9th IEEE Communications and Networking Conference (CCNC 2012), Las Vegas, 2012.
- [17] Reis, T., de Sa, M. and Carrico, L., “*Multimodal Artefact Manipulation: Evaluation in Real Contexts*,” in Proc. of Third International Conference on Pervasive Computing and Applications, Istanbul, 2008, pp. 570-575.
- [18] Arnab S., Petridis P., Dunwell I. and de Freitas S., “*Enhancing learning in distributed virtual worlds through touch: a browser-based architecture for haptic interaction*,” Serious Games and Edutainment Applications, Springer, pp. 149-167, 2011.
- [19] Brave S., and Dahley A.. “*inTouch: a medium for haptic interpersonal communication*,” in Proc. of the CHI ’97 extended abstracts on Human factors in computing systems: looking to the future. Atlanta, 1997, pp. 363-364.
- [20] Milosevic B., Farella E. and Benini L., “*Continuous Gesture Recognition for Resource Constrained Smart Objects*,” in Proc. of the International

- Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Florence, 2010.
- [21] Bergamasco M., “*Le Musee del Formes Pures*,” in Proc. of the 8th IEEE International Workshop on Robot and Human Interaction, Pisa, 1999.
- [22] Maxence Parache, “*Hyper(reality)*”. <<http://maxenceparache.blogspot.it/>>.
- [23] Zabulis, X., Baltzakis and H., Argyros, A.. “*Vision-based Hand Gesture Recognition for Human-Computer Interaction*,” The Universal Access Handbook, Human Factors and Ergonomics, page 34.1 – 34.30. Lawrence Erlbaum Associates, Inc. (LEA), June 2009.
- [24] Fujiyoshi, H. and Lipton, A.J., “*Real-time human motion analysis by image skeletonization*,” in Proc. of the Fourth IEEE Workshop on Applications of Computer Vision, 1998. WACV '98. Proceedings, Princeton, 1998, pp.15-21.
- [25] Kehl, R. and Van Gool, L., “*Real-time pointing gesture recognition for an immersive environment*,” in Proc. of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Southampton, 2004, pp. 577- 582.
- [26] Malerczyk, C., “*Interactive Museum Exhibit Using Pointing Gesture Recognition*,” in Proc. of the 12-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Plze-Bory, 2004, pp. 165–171.
- [27] Manresa C., Varona J., Mas R. and Perales F.J., “*Hand Tracking and Gesture Recognition for Human-Computer Interaction*,” Electronic Letters on Computer Vision and Image Analysis, vol. 5, n, 3, pp. 96-104, 2005.

- [28] Wang R.Y. and Popovic J., “*Real-time Hand-Tracking with a Color Glove*,” ACM Trans. Graph., vol. 28, n. 3, pp. 1-8, 2009.
- [29] Yu, E. and Aggarwal, J.K., “*Human action recognition with extremities as semantic posture representation*,” in Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Miami, 2009, pp.1-8.
- [30] Moeslund T.B., Hilton A. and Krüger V., “*A Survey of Advances in Vision-based Human Motion Capture and Analysis*,” Comput. Vis. Image Underst., Elsevier, New York, vol. 104, n. 2-3, pp. 90-126, 2006.
- [31] Mitra S. and Acharaya T., “*Gesture recognition: a Survey*,” Trans. On Sys., Man and Cyb., IEEE, New York, vol. 37, n. 3, pp. 311-324, 2007.
- [32] Freeman W. T. Anderson, D.B., Dodge, C.N., Roth, M. , Weissman, C.D. , Yerazunis, W.S. , Kage, H. , Kyuma, I. , Miyake, Y. , Tanaka, K. , “*Computer Vision for Interactive Computer Graphics*,” IEEE Computer Graphics and Applications, vol. 18, no. 3, pp. 42-53, 1998.
- [33] Snibbe S.S. and Raffle H.S., “*Social immersive media: pursuing best practices for multi-user interactive camera/projector exhibits*,” in Proceedings of the 27th international conference on Human factors in computing systems, 2009.
- [34] Simpson, Z. “*Shadow Garden*,” in SIGGRAPH 2002 Electronic Art and Animation Catalog, 2002.
- [35] Roccetti M., Marfia G. and Zanichelli M., “*The Art and Craft of Making the Tortellino: Playing with a Digital Gesture Recognizer for Preparing Pasta Culinary Recipes*,” ACM Computers in Entertainment, ACM, vol. 8, n. 4, 2010.

- [36] Carrozzino M., Bergamasco M., “*Beyond virtual museums: Experiencing immersive virtual reality in real museums,*” *Journal of Cultural Heritage*, Elsevier, vol. 11, pp. 452-458, 2010.
- [37] S. M. Bopalkar, P. Talwai, and B. H. Parmar, “*Body parts detection in gesture recognition using color information,*” in *Proceedings of the International Conference & Workshop on Emerging Trends in Technology (ICWET '11)*. ACM, New York, NY, USA, 149-152, 2011.
- [38] Michael J. Jones and James M. Rehg, “*Statistical Color Models with Application to Skin Detection*” in *International Journal of Computer Vision*, vol. 46, no. 1, 81-96, 2002.
- [39] K. Mikolajczyk, C. Schmid, and A. Zisserman, “*Human detection based on a probabilistic assembly of robust part detectors,*” *The 8th ECCV*, Prague, Czech Republic, volume I, pages 69-81, 2004.
- [40] Haritaoglu, I., Harwood, D., Davis, L.S., “*Ghost: a human body part labeling system using silhouettes,*” *Pattern Recognition*, 1998. Proceedings. Fourteenth International Conference on , vol.1, no., pp.77-82 vol.1, 16-20, Aug 1998.
- [41] Chi-Hung Chuang; Jun-Wei Hsieh; Luo-Wei Tsai; Kuo-Chin Fan; , “*Human Action Recognition Using Star Templates and Delaunay Triangulation,*” *Intelligent Information Hiding and Multimedia Signal Processing*, 2008. IIHMSP '08 International Conference on , vol., no., pp.179-182, 15-17 Aug. 2008
- [42] Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen, Suh-Yin Lee, “*Human action recognition using star skeleton,*” in: *Proceedings of the International Workshop on Video Surveillance and Sensor Networks (VSSN'06)*, Santa Barbara, CA, October 2006, pp. 171–178.

- [43] Tran, K.N., Kakadiaris, I.A., Shah, S.K., “*Modeling motion of body parts for action recognition*,” in: BMVC, 2011.
- [44] Roccetti M., Marfia G. and Bertuccioli C., “*Day and Night at the Museum: Intangible Computer Interfaces for Public Exhibitions*”, submitted for publication, April 2012.
- [45] Marfia G., Roccetti M., Varni A. and Zanichelli M., “*Mercator Atlas Robot: Bridging the Gap between Ancient Maps and Modern Travelers with Gestural Mixed Reality*”, Proc. 21st IEEE International Conference on Computer Communication Networks (ICCCN 2012) - 8th International Workshop on Networking Issues in Multimedia Entertainment (NIME 2012), Munich, July, 2012.
- [46] Roccetti M. and Marfia G., “*Recognizing Intuitive Pre-defined Gestures for Cultural Specific Interactions: An Image-based Approach*,” in Proc. 3rd IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology, Las Vegas, 2011.
- [47] M. Roccetti, G. Matteucci, A. Marcomini, “*Technoculture of Handcraft: Fine Gesture Recognition for Haute Couture Skills Preservation and Transfer in Italy*”, Proc. 39th ACM International Conference and Exhibition on Computer Graphics and Interactive Techniques Posters, (SIGGRAPH 2012), Los Angeles, August, 2012.
- [48] Wren C., Azarbayejani A., Darrell T. and Pentland A., “*Pfinder: Real-time Tracking of the Human Body*,” IEEE Trans. on Patt. Anal. and Machine Intell., vol. 19, no. 7, pp. 780-785, 1997.
- [49] Han B., Comaniciu D. and Davis L., “*Sequential kernel density approximation through mode propagation: applications to background modeling*,” in Proc. of the Asian Conference on Computer Vision, 2004.

- [50] Z.Zivkovic, “*Improved adaptive Gaussian mixture model for background subtraction*”, International Conference Pattern Recognition, UK, August, 2004.
- [51] American Genetic Association, “*Neoteny body proportion heterochrony human*”, Journal of Heredity, vol. 12, 1921.
- [52] M. Roccetti, G. Marfia, “*Recognizing Intuitive Pre-defined Gestures for Cultural Specific Interactions: An Image-based Approach*”, Proc. 3rd IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology (DENVECT’11) - 8th IEEE Communications and Networking Conference (CCNC 2011), Las Vegas (USA), IEEE Communications Society, January 2011.
- [53] Wobbrock, J.O., Wilson, A.D. and Li, Y., “*Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes*,” Proceedings of the ACM Symposium on User Interface Software and Technology (UIST ’07), Newport, Rhode Island, October 2007.
- [54] M. Roccetti, G. Marfia, A. Semeraro, “*Playing into the Wild: A Gesture-based Interface for Gaming in Public Spaces*”, Journal of Visual Communication and Image Representation, Elsevier, Vol. 23, n. 3, 2012.
- [55] Bradski G., Kaehler A., “*Learning OpenCV: Computer Vision with the OpenCV Library*”, Pub. O’Reilly Media, September 2008.
- [56] Gonzalez R. C., Woods R. E. “*Digital Image Processing (3rd Edition)*”, Pub. Pearson-Prentice Hall, 2008.
- [57] Valle M., “*Filtro di Kalman discreto*”.<http://www.micro.dibe.unige.it/maurizio_valle/Elettronica_Industriale_2/>.