ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA

CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA E SCIENZE INFORMATICHE

HUMAN ACTIVITY RECOGNITION CON TELECAMERE DI PROFONDITÀ

Tesi in

VISIONE ARTIFICIALE E RICONOSCIMENTO

Relatore

ANNALISA FRANCO

Presentata da

ANDREA ZAGNOLI

Sessione III

Anno Accademico 2015/2016

Abstract

Lo studio presentato in questa Tesi si propone di elaborare, implementare e testare un algoritmo di Human Activity Recognition (HAR) basato su telecamere di profondità. Per HAR si intende quel settore del machine learning che mira a studiare tecniche che, tramite l'acquisizione di informazioni da sorgenti di diverso tipo, permettano ad una macchina di apprendere in modo autonomo un metodo di classificazione delle attività umane. In particolare l'algoritmo proposto sfrutta la tecnologia delle telecamere di profondità (il sensore utilizzato è il Microsoft Kinect) che a differenza delle tradizionali telecamere a colori proiettano un campo di luce infrarossa e, in base a come questa viene riflessa dagli oggetti nella stanza, è in grado di calcolare la distanza tra il sensore e l'oggetto. L'algoritmo implementato in ambiente .NET, è stato testato su due dataset raccolti dal Computer Science Department, Cornell University e su un nuovo dataset raccolto contestualmente a questo studio. I risultati sperimentali confermano l'efficacia dell'algoritmo su tutte le azioni raccolte nei diversi dataset.

Sommario

1 INTRODUZIONE

Lo studio presentato in questa Tesi si propone di elaborare, implementare e testare un algoritmo di Human Activity Recognition (HAR) basato su telecamere di profondità. Per HAR si intende quel settore del machine learning che mira a studiare tecniche che, tramite l'acquisizione di informazioni da sorgenti di diverso tipo, permettano ad una macchina di apprendere in modo autonomo un metodo di classificazione delle attività umane.

Le tecniche di Human Activity Recognition possono essere organizzate in diverse categorie a seconda della tipologia di sensori che vengono utilizzati per l'acquisizione dei dati necessari alla classificazione. Distinguiamo principalmente due categorie: la prima fa uso esclusivamente di sensori indossabili che registrano il movimento di un individuo (accelerometri, giroscopi, ecc.), la seconda fa uso di sensori ottici, come telecamere, per individuare una persona all'interno del campo visivo e quindi utilizzare la sagoma della figura umana per evincere informazioni sulle azioni che sta eseguendo. La tecnica riportata in questo studio fa parte proprio di questa seconda categoria; in particolare l'algoritmo proposto sfrutta la tecnologia delle telecamere di profondità (il sensore utilizzato è il Microsoft Kinect) che a differenza delle tradizionali telecamere a colori proiettano un campo di luce infrarossa e, in base a come questa viene riflessa dagli oggetti nella stanza, è in grado di calcolare la distanza tra il sensore e l'oggetto.

L'algoritmo implementato in ambiente .NET, è stato elaborato sulla base di quello proposto nello studio di Manzi *et al.* [1]. L'algoritmo sfrutta la posizione delle articolazioni dello scheletro costruito sulla sagoma umana tramite le API di Kinect e si articola in quattro fasi: normalizzazione della posizione delle articolazioni, individuazione delle pose chiave e compressione della sequenza video, estrazione dei descrittori per mezzo di una finestra scorrevole, addestramento di un classificatore SVM. Contestualmente all'implementazione del sistema di riconoscimento è stato anche realizzato un software per l'acquisizione di un nuovo dataset che

comprende 10 diversi soggetti (5 uomini e 5 donne) che eseguono 14 diverse azioni. L'algoritmo oggetto di studio in questa tesi è stato quindi testato su due dataset già esistenti e largamente diffusi nella comunità scientifica in ambito di Human Activity Recognition entrambi raccolti dal Computer Science Department, Cornell University.

Il documento di Tesi è diviso in cinque sezioni: una sezione introduttiva relativa alla descrizione della problematica della Human Activity Recognition [2 Human Activity Recognition] dove vengono anche descritti impieghi reali motivando il crescente interesse verso questa problematica nel settore del machine learning. Questa sezione è seguita da [3 Stato dell'Arte] una panoramica sugli studi condotti dalla comunità scientifica in questo ambito in modo da fornire una visione complessiva dello stato dell'arte in questo settore. Il documento prosegue con [4 Algoritmo di estrazione e classificazione dei descrittori] dove viene presentato in modo dettagliato l'algoritmo oggetto di studio e vengono forniti dettagli sull'implementazione [5 Implementazione]. In fine segue un'approfondita analisi [6 Risultati Sperimentali] dei risultati sperimentali ottenuti testando il sistema di classificazione sui tre sopracitati dataset.

2 HUMAN ACTIVITY RECOGNITION

Con il termine Human *Activity Recognition* si intende la disciplina che mira a riconoscere le azioni e/o gli obiettivi che cercano di raggiugere uno o più agenti[1] in una scena a partire da una serie di osservazioni sugli agenti stessi e sull'ambiente in cui sono immersi. Per attività si intende una serie più o meno complessa di movimenti del corpo che identificano delle azioni precise. A seconda di quanto articolata e complessa è un'azione possiamo distinguere diversi tipi di attività. Definiamo *gestures* movimenti basilari del corpo, quelle che possono essere chiamate "azioni atomiche", come possono essere allungare un braccio, muovere una gamba ecc. Per *actions* si intendono invece movimenti composti da più gesture che implicano il movimento di diverse parti del corpo, ad esempio camminare, sedersi, saltare, salutare ecc. Per *interazioni* invece si intendono tutte quelle azioni che prevedono l'interazione con un oggetto dell'ambiente circostante o un'altra persona come possono essere mangiare, pulire, abbracciare, ecc. In fine, la categoria delle azioni più complesse e articolate, è quella delle g*roup activities* dove un gruppo di persone numeroso interagisce sia con gli altri membri del gruppo sia con oggetti dell'ambiente; fanno parte di questa categoria attività come: un gruppo di persone che marcia o un gruppo di persone che assiste ad uno spettacolo ecc.

Sin dagli anni '80 questo campo di ricerca ha coinvolto diverse comunità scientifiche grazie alle opportunità che offre nel fornire un supporto personalizzato per diverse tipologie di applicazioni e alle diverse affinità con diversi settori come la medicina, la sociologia, la sicurezza, l'interazione uomo-macchina e lo sport. L'activity recognition diventa un importante argomento di ricerca soprattutto in relazione alla realizzazione di *smart-pervasive environments* nei quali il comportamento di un attore[1] e il resto dell'ambiente sono monitorati e analizzati al fine di inferire quali sono le attività che il soggetto sta compiendo e a quale scopo. Solitamente un sistema di human activity recognition (HAR) comprende la

[1] Con *agente* o *attore* si intendono soggetti senzienti, solitamente umani, che compiono azioni autonome e indipendenti al fine di raggiungere uno scopo o di soddisfare un bisogno.

modellazione delle attività e comportamenti, il monitoraggio dell'ambiente, data processing e pattern recongition, queste componenti vengono solitamente organizzate in architetture a tre strati [Figura 1]: uno strato di basso livello dove risiede la sensoristica del sistema. Un livello intermedio di computazione dove viene eseguito il *pre-processing:* eliminando il rumore e le ridondanze, gestendo i vuoti di informazione nel segnale, la *segmentation* dove vengono selezionati i segmenti di informazioni rilevanti, la *feature extraction* al fine di trasformare i dati in descrittori utili alla classificazione e la *dimensional reduction* dove viene diminuito il numero di features per preservare quelle più significative. Infine, nel livello superiore risiedono gli algoritmi di machine learning, reasoning e data-mining che permettono di inferire quali sono le azioni svolte.

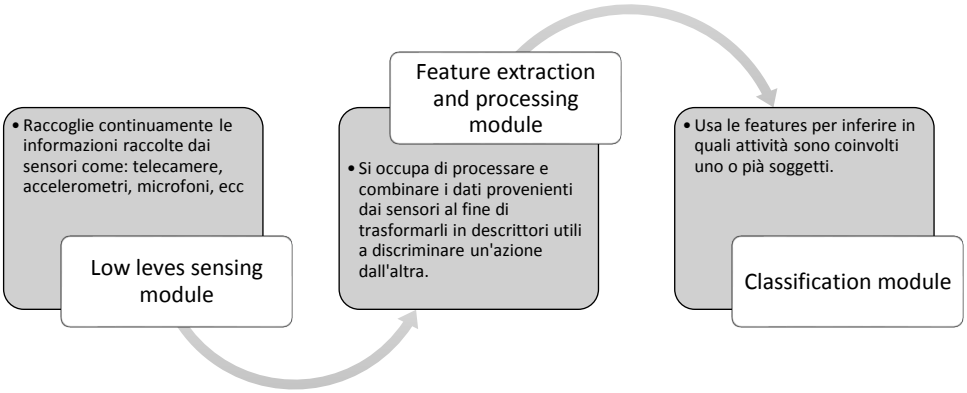

Figura 1 – Descrizione delle componenti di un sistema di Activity recognition scomposto in tre sub unità funzionali.

2.1 Approcci all'activity recognition

A seconda dell'impiego per il quale si intendono adottare tecniche di activity recognition e quindi a seconda di quale sia l'obiettivo di un HAR sono state proposte dalla comunità scientifica diversi approcci, che differiscono principalmente per il tipo di segnale che si intende utilizzare, grazie ai quali è possibile acquisire dati sull'ambiente e sugli agenti che compiono le azioni. I diversi approcci descritti qui di seguito sono: *sensor-based* e *vision-based.*

2.1.1 Sensor-based activity recognition

Questa tipologia di approcci fa uso di una rete di sensori per monitorare il comportamento umano. Gran parte dei sensori di questa categoria è costituita da quelli che vengono chiamati *wearable sensors*, ovvero sensori indossabili come giroscopi o accelerometri che i soggetti monitorati indossano costantemente e che, tramite canali wireless, trasmettono i dati al sistema che poi li elaborati. Il principale svantaggio di questa tipologia di sensori è dovuto alla durata di vita della batteria, essendo dispositivi mobili necessitano di un'alimentazione tramite batterie che ovviamente devono essere ricaricate. Questo è un dettaglio da non sottovalutare per una serie di applicazioni quali, ad esempio, gli elder care HAR systems [2.2 Impieghi reali – Health care monitoring] dove l'utilizzatore potrebbe dimenticare di indossare il dispositivo o di ricaricarne la batteria. Inoltre uno svantaggio secondario può essere la praticità di utilizzo, costringere l'utente a indossare costantemente dei sensori è infatti un approccio abbastanza invasivo. Un considerevole vantaggio di questo tipo di approcci sta nel fatto che maggior parte degli smartphones di nuova generazione hanno al loro interno molti sensori di movimento come quelli elencati in precedenza; inoltre, sono dotati inoltre di un hardware sufficientemente potente da poter eseguire gli algoritmi di pre-processing e feature extraction direttamente all'interno del dispositivo. In fine data la precisione dei wearable sensor e della qualità dei dati raccolti con questo tipo di approcci si ottengono ottimi risultati in termini di performance e accuracy [2]. Data la popolarità degli smartphone e altri dispositivi mobili di questo tipo sono state studiate numerosissime tecniche e sviluppati altrettanti sistemi che adottano questo approccio [3].

2.1.2 Vision-based activity recognition approach

Le tecniche che utilizzano telecamere e sensori ottici per fare activity recognition fanno parte di questa categoria di approcci. Facendo uso di immagini e video come input, gli algoritmi utilizzati, fanno parte della branca della visione artificiale. Dal momento che i sensori utilizzati in questo caso inquadrano l'intero ambiente e non solamente il soggetto che compie l'azione gli step necessari al fine di eseguire operazioni di activity

recognition sono: l'individuazione del soggetto umano (solitamente eseguito con tecniche di background subtraction) e il tracking del soggetto nel tempo; solo successivamente si possono applicare le tecniche di feature extraction. Sicuramente uno svantaggio di questa tipologia di approcci è dovuto alla necessità di un maggior numero di operazioni di pre-processing da eseguire prima della vera e propria estrazione dei descrittori e della classificazione dell'azione. Come vantaggio è importante sottolineare che i vision-based HAR system non necessitano che i soggetti indossino dei sensori e quindi questi sistemi possono essere utilizzati anche in campo di sicurezza e videosorveglianza dove i soggetti che eseguono le attività non sono collaborativi e vengono analizzati "passivamente". Anche nel caso degli elder-care system questi sistemi risultano essere più robusti in questo senso dal momento che non è necessario ricordarsi di caricare la batteria o di indossare il dispositivo stesso. I vision-based HAR systems nascono proprio con lo scopo di creare degli smart environment poco invasivi ed estremamente trasparenti che mettano gli utenti in condizione di eseguire le proprie attività come se il sistema non esistesse nemmeno.

Come affermato inizialmente gli HAR systems di questo tipo facevano uso di tradizionali telecamere RGB e tramite algoritmi di background subtraction andavano isolare le regioni dove era presente il soggetto umano e quindi calcolare la posizione delle articolazioni. Come è facile intuire, nonostante in ricerca gli algoritmi abbiano fatto enormi progressi, risulta difficile ridurre al minimo l'errore commesso nel calcolo della posizione degli arti a causa dell'alta variabilità di illuminazione, ombre e contrasto nel tempo (soprattutto per quanto riguarda gli ambienti outdoor). Negli ultimi anni è diventato molto popolare, anche nella comunità scientifica, l'uso di telecamere di profondità, la più famosa è sicuramente il Kinect prodotto da Microsoft.

Figura 2 - A sinistra (a) Kinect 1 a destra (b) Kinect 2

Sensor overview

Qui di seguito viene presentato in breve il Microsoft Kinect e la tecnologia delle telecamere di profondità in genere.

Kinect, inizialmente conosciuto come Project Natal, è un accessorio inizialmente realizzato per la console Xbox 360 di Microsoft. La distribuzione sul mercato di Kinect è iniziata nel novembre 2010, sebbene all'inizialmente Microsoft commercializzasse solamente una versione esclusiva solamente per console a partire dal febbraio 2012 venne realizzata anche una versione compatibile per PC con sistema operativo Windows 7 e Windows 8.

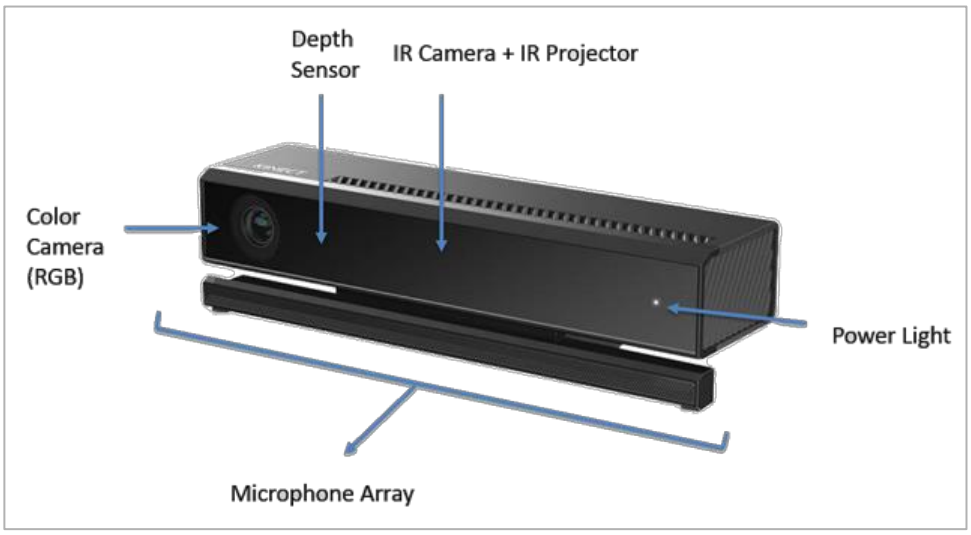

Figura 3 - Kinect 2 hardware

La sensoristica hardware del Kinect [Figura 3] comprende *proiettore di infrarossi* (IR), una *telecamera a infrarossi* e una tradizionale telecamera a colori. Il sensore di profondità comprende quindi sia il proiettore che la telecamera a infrarossi.

Il funzionamento del sensore di profondità funziona nel modo seguente:

1) Il proiettore IR emette un pattern a macchie puntiformi, invisibile all'occhio umano, nello spazio 3D.
2) La telecamera a infrarossi cattura il riflesso del pattern proiettato.
3) È possibile determinare la profondità di ogni punto proiettato grazie alla relativa traslazione destra o sinistra del punto a partire da un pattern standard che viene calibrato sul sensore. La traslazione del punto è direttamente proporzionale alla distanza del punto dal sensore.

Scheda tecnica Kinect 1

RGB Camera
• È una telecamera tradizionale che cattura le 3 componenti principali di colore e opera a 30 fps con una risoluzione di 320x240 pixel con 8 bit per canale. Il Kinect permette anche di abbassare gli fps a 10 per aumentare la risoluzione fino a 1280x1024. Ha un campo visivo di 57°orizzontalmente e 43° verticalmente.
3D Depth sensor
• 3D Depth sensor comprende sia il proiettore che la telecamera a infrarossi, insieme questi operano per creare quella che viene chiamata mappa di profondità. Il sensore funziona in modo ottimale a partire dalla distanza di 0.8m fino a 3.5m producendo un output video a 30 fps con una risoluzione di 320x240 pixels. Ha un campo visivo di 57° orizzontalmente e di 43° verticalmente.
Motorized tilt
• Sia la telecamera RGB che il sensore di profondità sono installati su un supporto motorizzato regolabile che permette di regolare l'inclinazione verticale di ±27°.

Tabella 1 - Scheda tecnica componenti Hardware Kinect 1

Scheda tecnica kinect 2

RGB Camera	
• La nuova telecamera ha un campo visivo maggiore (70° orizzontalmente e 60° verticalmente) ed opera a 30 fps in caso di illuminazione ottimale e a 15 fps in caso di scarsa illuminazione. La risoluzione è di 1920x1080 pixel per fotogramma (1080p video).	
3D Depth sensor	
• Anche il sensore di profondità è stato migliorato ora acquisisce fotogrammi con risoluzione 512x424 sempre ad una frequenza di 30 fps. Ora Kinect è in grado di individuare e tracciare contemporaneamente 6 persone (solamente 2 nella versione precedente). Lo scheletro è composto da 25 articolazioni contro le 15 di Kinect 1.	
Microphone array	
• Il dispositivo di inclinazione motorizzato è stato rimosso dato il miglioramento del campo visivo della telecamera RGB e infrarosso; ha lasciato spazio ad un array di microfoni per registrare il suono e determinare la direzione dalla quale provengono le onde sonore.	

Tabella 2 - Scheda tecnica componenti Hardware Kinect 2

Come detto in precedenza, data la possibilità di poter interfacciare il device con un pc e dato il suo modesto prezzo, il Kinect è diventato immediatamente una tecnologia utile anche alla ricerca e alla portata di tutti; offre infatti oltre al suo developer kit ufficiale rilasciato da Microsoft, il Microsoft Kinct SDK [4], ma anche diverse librerie open source sviluppate da terzi come OpenNL [5] e OpenKinect [6].

Le performance del Kinect, nonostante il prezzo non elevato, non sono affatto da sottovalutare; infatti, l'accuratezza del sensore a infrarossi è piuttosto elevata. In condizioni ottimali l'errore commesso è inferiore a un centimetro, mentre in condizioni "comuni" l'errore sul calcolo della profondità varia da 1 a 3 cm. L'errore che deve essere minimizzato maggiormente, però, non è tanto quello sul calcolo della profondità ma quanto quello sulla precisione della stima della posizione dell'arto dello scheletro che viene costruito sulla sagoma umana; in questo caso diventa

più complicato non commettere errori essendo molteplici le variabili in gioco: oggetti che coprono parte della sagoma umana, posizioni difficili da riconoscere, ecc. Si stima che l'errore medio sul tracciamento dello scheletro è di circa 10 cm; fortunatamente quando si parla di riconoscimento di attività ciò che realmente importa non è la stima esatta di una posa del soggetto inquadrato quanto quali sono i cambiamenti, le transizioni tra una posa e l'altra in questo senso l'errore sul calcolo della posizione diventa meno rilevante in quanto ciò che importa è la differenza tra la posizione precedente e quella successiva.

Un altro fattore da non sottovalutare quando si fa uso delle immagini RGB-D del Kinect è il *preprocessing*, dal momento che spesso le scene catturate sono complesse e contengo effetti riflettenti che possono distorcere i raggi IR proiettati e quindi falsare la misura della profondità è importante applicare delle operazioni di elaborazione per migliorare la qualità dell'immagine. Normalmente le operazioni da applicare sono le seguenti:

- Al fine di utilizzare correttamente la porzione di informazione RGB in combinazione con la componente della profondità sono necessarie operazioni di allineamento dell'output. Queste operazioni fanno parte di una serie di altre operazioni di calibrazione dell'hardware Kinect.
- Come accennato in precedenza, la scena può presentare numerosi punti di rumore (holes) dovuti a riflessi dell'infrarosso su oggetti trasparenti o semiriflettenti; questi punti di rumore possono determinare letture della profondità errate o mancanti, diventa quindi necessario adottare tecniche per chiudere o correggere questi holes prima che le immagini vengano utilizzate.
- Infine, per facilitare l'estrazione dello scheletro umano vengono applicati degli algoritmi di segmentazione dell'immagine, data la natura delle immagini di profondità questo tipo di operazioni risultano essere molto più semplici e accurate rispetto ai risultati che comunemente si ottengono con le tradizionali immagini RGB o grayscale. Come si può notare dalla Figura 4 è molto più semplice

isolare la figura umana dall'immagine di profondità (immagini nella seconda riga) rispetto a quelle RGB.

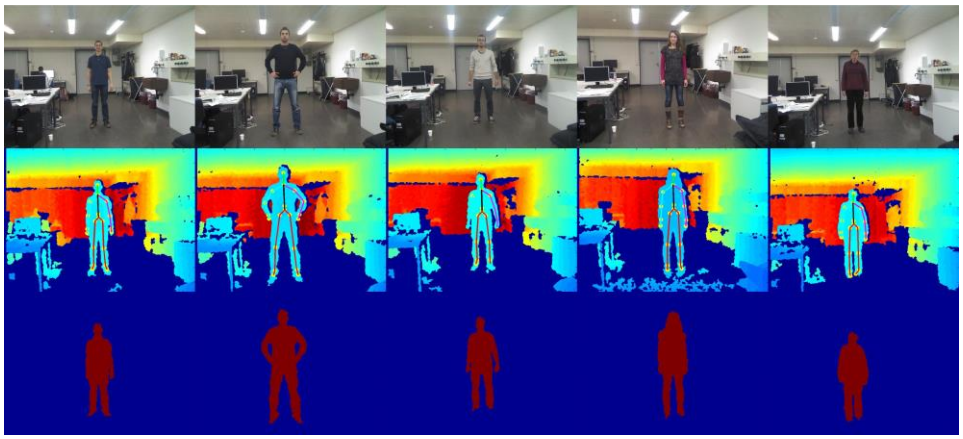

Figura 4 - Esempio di immagini RGB-D acquisite tramite Kinect, nella prima riga è possibile vedere la componente RGB dell'immagine, nella seconda sono invece presentate le immagini di profondità ricolorate con una scala cromatica che parte dal rosso (oggetti più lontani) e arriva all'azzurro (oggetti più vicini). In questo caso nell'immagine viene già riportato lo scheletro umano stilizzato nelle varie articolazioni corporee ottenuto con tecniche di pose estimation. In fine nella ultima riga sono presenti le sagome umane segmentate e isolate dal background a partire dalle immagini profondità.

2.2 Impieghi reali

Come accennato in precedenza questo settore di ricerca sta avendo particolare successo negli ultimi anni. Tale popolarità è dovuta al range di impieghi che possono avere tecniche di questo tipo. In particolare la possibilità di apprendere e riconoscere determinate azioni umane può permettere la realizzazione di sistemi video interattivi innovativi che possono essere realizzati sia a scopo puramente interattivo, e quindi utilizzati dalle grandi softwarehouse che realizzano videogiochi si per console che per PC, si a scopo educativo, in questo caso si può pensare a sistemi interattivi all'interno di musei, sia a scopo commerciale e pubblicitario, dove possono essere installati sistemi dotati di Kinect per invogliare (magari facendo uso della gamification) il potenziale cliente a conoscere un brand un prodotto interagendo tramite gesti, in fine un importante settore dove si possono utilizzare questo tipo di applicazioni è la sicurezza. Riguardo a questo topic è importante spendere qualche parola in più in quanto al momento la ricerca si sta muovendo soprattutto in questa direzione. In ambito di sicurezza si può pensare alla realizzazione di sistemi

di video sorveglianza automatizzati che possono inviare autonomamente segnalazioni di allarme, alle forze dell'ordine o a persone specifiche, nel caso si verifichino delle situazioni inaspettate. In questo senso sistemi di questo tipo possono essere impiegati sia a livello industriale, per monitorare impianti produttivi al fine di allertare il personale in caso di guasti o malfunzionamenti, al livello di sicurezza in luoghi pubblici nel caso di eventi, fiere o concerti ma anche come miglioramento delle attuali telecamere a circuito chiuso che potrebbero inviare segnalazioni in caso ad esempio di furto avvertendo le forze dell'ordine nel caso qualcuno cercasse di scappare correndo via da un negozio ed in fine possono essere utilizzati anche per uso domestico soprattutto per allertare i parenti di anziani che vivono da soli e potrebbero sentirsi male, questi sistemi, che vengono chiamati fall detection system, sono in grado di riconoscere se una persona cade a terra.

Il principale obiettivo dei sistemi HAR sono osservare e analizzare le attività umane e interpretare gli eventi che accadono correttamente. Sia che questi facciano uso di sensori visivi e non, gli HAR systems ricevono processano dati (ambientali, temporali, spaziali, ecc.) per capire il comportamento umano. Solitamente bisogni che si intende soddisfare con la realizzazione di un HAR system fanno parte della *human-sensing taxonomy* [7], ovvero tutta la serie di processi che mira a estrarre qualunque tipo di informazione riguardante persone che si trovano in un determinato ambiente. Nello specifico:

- *Presenza:* verificare se è presente almeno un soggetto umano nell'ambiente monitorato.
- *Numero:* contare il numero di persone che popolano un'ambiente, questo è molto utile soprattutto negli impianti di videosorveglianza.
- *Posizione:* determinare la posizione della persona nello spazio 3D.
- *Tracking:* determinare gli spostamenti della persona, compresa la postura che assume, al variare del tempo.
- *Identità:* verificare l'identità di una persona, questo banalmente potrebbe essere eseguito aggiungendo un software basato sul

riconoscimento del volto oppure, in modo meno semplice, riconoscendo la postura, il modo in cui cammina o si muove. Si parla infatti di *Biometric motion Signature.*

Esistono diversi settori in cui possono essere utilizzati i sistemi HAR tra i quali possiamo distinguere in particolare: *Ambient Assisted Living* (AAL), applicazioni di monitoraggio e assistenza sanitaria, monitoraggio e sorveglianza per ambienti indoor e outdoor, *tele-immersion applications,* ambienti assistiti per l'industra manifatturiera.

Ambient Assisted Living per smart home

Ambient Assisted Living (AAL) è il termine coniato nei primi anni 2000 per descrivere un insieme di soluzioni tecnologiche non intrusive destinate a rendere attivo, intelligente e cooperativo l'ambiente nel quale viviamo, efficace nel sostenere la vita indipendente, capace di fornire maggiore sicurezza, semplicità, benessere e soddisfazione nello svolgimento delle attività della vita quotidiana. I principali obiettivi dichiarati dall'Associazione Europea AAL sono:

- Estendere il periodo in cui le persone possono vivere nel loro ambiente preferito, aumentando la loro autonomia, autosufficienza e mobilità;
- Aiutare a mantenere la salute e le capacità funzionali delle persone anziane;
- Promuovere stili di vita migliori e più salutari per le persone a rischio;
- Aumentare la sicurezza, prevenire l'esclusione sociale e mantenere la rete relazionale delle persone;
- Sostenere gli operatori, i familiari e le organizzazioni dell'assistenza;
- Migliorare l'efficienza e la produttività delle risorse nella società che invecchia.

In questo settore è molto importante il concetto di *Activity of Daily Life* (ADL) ovvero l'insieme delle azioni comunemente svolte in ambiente domestico durante il giorno, costruire un sistema efficiente che permetta di inferire e prevedere azioni di questo tipo consente la realizzazione di una

serie di algoritmi di reasoning che possono assistere gli utenti del sistema in tempo reale.

Health care monitoring

Lo sviluppo della scienza medica e della tecnologia ha innalzato considerevolmente la qualità della vita dei pazienti. Alcuno studi [8] affermano che entro il 2050 la popolazione mondiale aumenterà notevolmente con la conseguenza che circa il 30% in più delle persone avrà un'età superiore ai 60 anni. Questo comporta una più alta richiesta di personale medico e paramedico che, nell'immediato futuro, non è soddisfattibile. Con i sistemi HAR si cerca di innalzare il livello di autonomia e completezza degli attuali sistemi di monitoraggio dei pazienti in modo rendere meno necessario l'intervento di un operatore umano.

Di base un sistema di questo tipo è composto da un insieme di componenti che si occupano di riconoscere diverse attività o situazioni e fornire assistenza come: l'identificazione se una persona cade (fall detection), assistere persone con problemi cognitivi, tracciare gli spostamenti di una persona, avvertire il personale specializzato in caso di comportamenti anomali o situazioni pericolose in modo da garantire un rapido intervento.

Sicurezza e sorveglianza

I sistemi di sorveglianza tradizionale vengono monitorati da operatori umani che segnalano le situazioni di allarme e devono prestare costante attenzione alle registrazioni delle telecamere. Spesso gli addetti sono tenuti a monitorare un grosso numero di ambienti per un lungo numero di ore, questo incide notevolmente sull'aumento dello stress e la stanchezza degli operatori causando un calo drastico calo dell'attenzione. Al fine di alleggerire il carico di lavoro degli addetti alla sicurezza si intende realizzare dei sistemi intelligenti che automatizzano il meccanismo di monitoraggio. In campo di sicurezza è impossibile pensare di utilizzare i classici algoritmi per il riconoscimento degli oggetti per ricostruire la scena in quanto spesso la cista è occlusa da oggetti, viene inquadrato un ampio campo visivo e la qualità dei dettagli è bassa. Di conseguenza è importante

che i sistemi di videosorveglianza utilizzino tecniche capaci di aggirare le più comuni problematiche dovute a fattori ambientali come rumore, occlusione, contrasto, ombre. Inoltre va comunque sottolineato che un sistema realizzato per degli spazi indoor come una banca, un aeroporto, un museo o un centro commerciale non possono essere utilizzati anche per ambienti esterni come piazze o stadi senza almeno effettuare un'accurata opera di tuning e testing.

Tele-immersion (TT) applications

La Tele-immersion è una tecnica che permette all'utente di comunicare ad un software la sua posizione e i suoi movimenti all'interno di un ambiente virtuale e interagire con oggetti virtuali o con altri utenti. Questi sistemi ovviamente necessitano di una notevole potenza di calcolo in quanto necessitano di elaborare un'enorme quantità di dati e restituire l'output in real-time.

Industry manufacturing assisting

Le tecniche di activity recognition possono anche assistere i lavoratori nelle loro mansioni giornaliere. Questi sistemi possono comprendere sia Wearable Sensor che amplificano le capacità fisiche e sensoriali degli operai e aumentandone l'efficienza produttiva. Possono, invece, costituiti da telecamere a circuito chiuso che permettono di monitorare le attività lavorative e segnalare eventuali problemi, guasti, situazioni anomale e infortuni. In ambito industriale non è necessario che il soggetto dell'azione sia per forza un lavoratore umano, si possono creare anche sistemi di assistenza ai robot ad esempio nella produzione di automobili.

Mobile Activity Recognition applications

Come affermato in precedenza, il sempre più popolare uso degli smartphones, consente di portare con sè un device che contiene una serie di sensori hardware utili per fare Activity Recognition. Sono innumerevoli gli studi che hanno l'obiettivo di utilizzare questi device mobili per creare degli HAR system sia per la realizzazione di applicazioni per l'utente finale sia per aumentare le performance dello stesso dispositivo; ad esempio Hermann

et al. [9] hanno dimostrato come con operazioni di *activity context aware tuning* sia possibile aumentare l'efficienza della durata della batteria di un telefono fino a cinque volte.

I principali settori per i quali sono state sviluppate applicazioni di activity recognition tramite cellulari sono: fitness tracking, health care, fall detection, context aware behaviour, home and work automation, self-managing systems, targeted advertising e molte ancora [3].

3 STATO DELL'ARTE

Dal momento che lo studio proposto in questa tesi mira all'implementazione, al testing e al miglioramento di un algoritmo [Capitolo 4] proposto in letteratura che sfrutta un approccio vision-based per l'activity recognition in questa sezione vengono proposti alcuni popolari studi condotti dalla comunità scientifica e costituiscono a grandi linee quello che è lo stato dell'arte per quanto riguarda i Vision-based HAR algorithms.

3.1 Tecniche basate su telecamere RGB

Da tempo la comunità scientifica indaga per trovare delle soluzioni efficienti in questo settore, sono state proposte molteplici metodologie (utilizzano come input sequenze video RGB) che differiscono per il tipo di approccio utilizzato. Sommariamente possiamo distinguere due grosse categorie di approcci: *single-layer* e *hierarchical.* Gli approcci single-layer solitamente vengono utilizzati per classificare e riconosce attività semplici come gesture e action, gli approcci gerarchici, al contrario, sono in grado di classificare attività più complesse data la loro capacità di scomporre l'attività stessa in più sotto unità, dette *sub-events*. Gli approcci gerarchici tentano quindi di riconoscere i sub-events e ricostruire quindi l'attività più complessa. Queste due macro categorie a loro volta possono essere frammentate in più sotto classi a seconda dell'approccio algoritmico adottato.

La tassonomia proposta in [10] permette di raggruppare a seconda delle loro caratteristiche gli algoritmi similari [Figura 5]. Qui di seguito verranno descritti sommariamente gli approcci adottati a seconda delle diverse categorie e presentato a titolo di esempio qualche algoritmo.

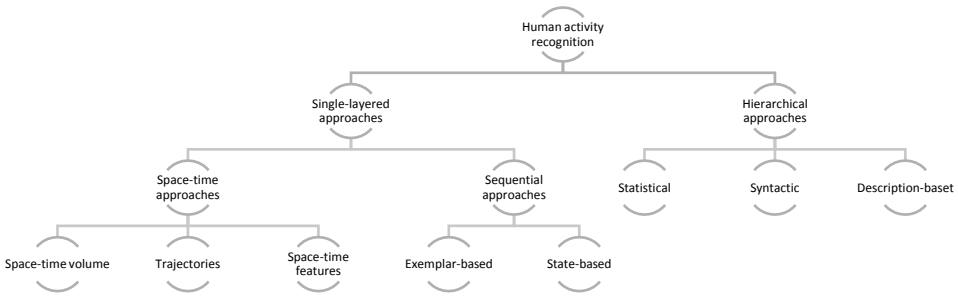

Figura 5 - Tassonomia degli algoritmi proposti basandosi unicamente su sequenze video RGB

3.1.1 Single-layerd approaches

Fanno parte di questa categoria tutti gli algoritmi che classificano un'attività direttamente dai fotogrammi di una sequenza video. Dal momento che è difficile determinare in quale istante del video inizi l'azione da riconoscere, la maggior parte degli approcci utilizza una sliding window che iterativamente tenta di riconoscere una sequenza in diversi istanti temporali del video. Gli approcci di questo tipo sono molto efficienti quando è possibile individuare un pattern sequenziale caratteristico all'interno dei video di training. Data la loro natura sono particolarmente adatti a riconoscere azioni semplici come camminare, salutare, saltare, fare determinati esercizi a corpo libero ecc.

Space-time approaches

Questi approcci riconoscono le attività umane analizzando il *space-time volume* delle attività nei video. Per capire cosa si intende con volume spazio-tempo definiamo un'immagine come una rappresentazione 2D nelle dimensioni XY di uno spazio 3D nelle dimensioni XYZ, di conseguenza un video è una serie di immagini 2D concatenate in ordine cronologico. Definiamo quindi un volume spazio-tempo come una concatenazione di immagini 2D creando una sequenza video 3D nelle dimensioni XYT. Tipicamente un approccio di questo, basandosi su dei video di training, tipo non fa altro che costruire un modello del volume spazio-tempo dell'attività. Quindi una volta che deve classificare un nuovo video usa tecniche di template matching per calcolare quali tra i vari modelli è quello più simile.

Spesso basarsi solamente sul volume attività non risulta essere molto efficiente per questo sono state ideate diverse varianti che forniscono una diversa rappresentazione dello spazio-tempo. Ad esempio si possono utilizzare le *traiettorie* compiute dal corpo durante l'azione per costruire un modello. Per fare questo è necessario fornire una stima della posizione delle articolazioni del corpo all'interno della scena e quindi tracciarne lo spostamento nei frame successivi creandone le traiettorie. Un'altra possibilità è quella data dall'utilizzo di *features* estratte dal volume o dalle traiettorie aggiungendo quindi un ulteriore livello di astrazione prima di costruire il modello su cui fare template matching.

In fine questi algoritmi variano a seconda della tecnica di matching utilizzata, infatti, sono state proposte metologie con *Neighbor-based matching* o *statistical model algorithms.*

Sequential approaches

Gli approcci sequenziali sono approcci a single-layer che riconoscono le attività umane analizzando una sequenza di features. A tale scopo un video in input viene trasforamto in una sequenza di feature vectors e l'algoritmo è in gradi di affermare se in tale video è presente una determinata azione se è in grado di individuare una determinata sequenza di features presente anche nelle classi sui cui è stato fatto training. Anche in questo esistono due principali categorie: gli *exemplar-based recognition approaches*, rappresentano le attività mantenendo una sequenza di features che funge da template o un insieme di sequenze campione; quando un nuovo video deve essere classificato vengono comparate le feature estratte da quest'ultimo con i template acquisiti in fase di training, i *model-based recongition approaches,* rappresentano un'attività come in modello composto da un insieme di stati, il modello viene generato in fase di training spesso un approccio statistico; per calcolare la verosimiglianza tra video in input e modello viene valutata per ogni modello la probabilità di generare la sequenza di feature vectors osservata.

Hidden Markov models (HMMs) e le reti bayesiane dinamiche (DBNs) sono stati ampiamente utilizzati per algoritmi di questo tipo; in entrambi i casi un'attività viene rappresentata come un insieme di stati "nascosti", si assume che chi svolge l'azione si trovi in uno stato in ogni frame, quindi si calcola la probabilità di transitare da uno stato a quello successivo frame per frame. Una volta che le transizioni e le relative probabilità sono state modellate in fase di training si calcola la probabilità che un determinato modello generi la sequenza di features osservate nei video da classificare quindi si sceglie quello con probabilità più alta. La prima volta che venne utilizzato HHMs per activity recognition risale al 1992 presentato in [11].

3.1.2 Hierarchical approaches

L'idea di base che ha portato all'uso degli approcci gerarchici per riconoscere attività complesse e di alto livello è quella di suddividere le attività complesse in sub-events più semplici e basarsi sul riconoscimento di questi per riuscire a classificare l'azione complessiva. Nella maggior parte di questi metodi si combinano gli approcci single-layered presentati in precedenza [Figura 6]. Concettualmente è possibile ipotizzare che per addestrare un classificatore con un algoritmo gerarchico richiede meno campioni di esempio perché i sub-events si possono ripeter più volte nella stessa attività o anche in attività diverse. Da questa scomposizione in sotto-attività si traggono diversi vantaggi: in primo luogo è più semplice classificare correttamente un'azione non articolata e inoltre gli algoritmi diventano meno complessi e concettualmente comprensibili permettendo di introdurre più facilmente la conoscenza umana e riducendone anche il costo computazionale. In tal senso se si cerca di classificare attività articolate con degli approcci non gerarchici si tende generare strutture complesse e features che non sono interpretabili, escludendo la possibilità di introdurre della conoscenza pregressa da parte degli sviluppatori; al contrario con un modello gerarchico è possibile organizzare le attività come una

composizione di sotto-eventi semanticamente interpretabili incorporando all'interno dell'algoritmo le conoscenze pregresse molto facilmente.

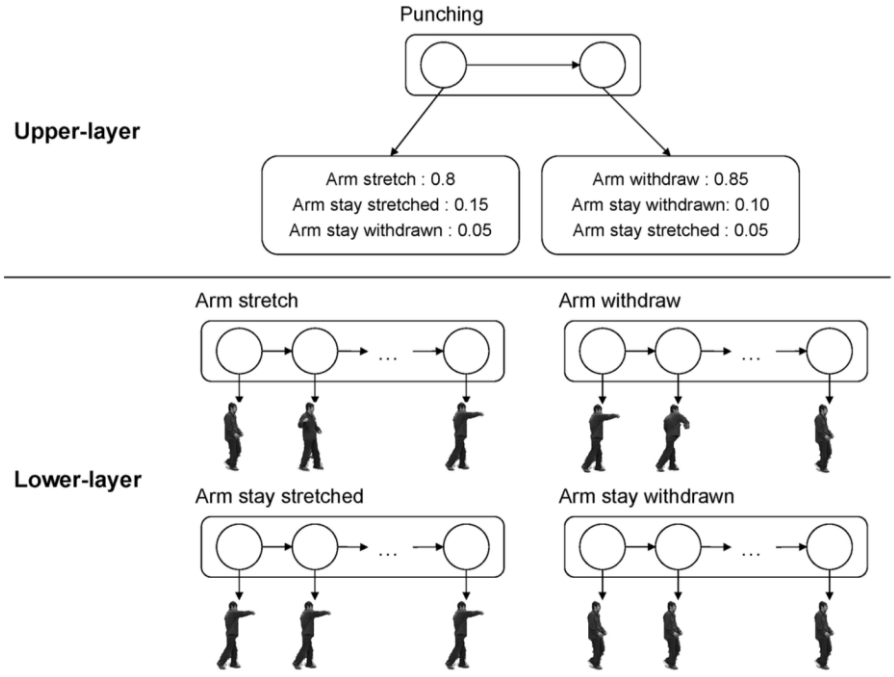

Figura 6 - Esempio di struttura a più strati, nel lato inferiore sono presenti dei classificatori semplici che permettono di riconoscere le fasi intermedie, nello strato superiore invece vengono sintetizzate le informazioni calcolate dal livello sottostante per poter classificare l'intera azione.

Statistical approaches

Fanno uso di modelli generati su base statistica per riconoscere le attività, solitamente si fa uso su due livelli di modelli state-based come HMMs e DBNs per riconoscere le sotto-sequenze [Figura 6]. Quindi i gesti base vengono classificati nello strato inferiore i gesti base che fornisce in input allo strato soprastante una serie di azioni atomiche allineate temporalmente. Questo layer si comporta come lo strato sottostante ma ad un livello di astrazione più alto, infatti, lo score di verosimiglianza viene calcolata come probabilità che il modello generi la sequenza di azioni base osservata (ovvero quella prodotta dal primo livello). Come stima dello score vengono utilizzati la *maximum likelihood estimation* (MLE) [12] o la *maximum a posteriori probability* (MAP) [13].

Una delle forme fondamentali dell'approccio gerarchico statistico è il *layered HMMs* (LHMMs) introdotto nel 2002 da Oliver *et al.* in [14] che

implementa l'HMMs sia per riconoscere le azioni atomiche sia nello strato superiore per riconoscere, in base alla concatenazione di quest'ultime, l'azione composta.

Gli algoritmi gerarchici di impronta statistica sono molto robusti quando si cerca di riconoscere attività sequenziali e, se si addestrano i classificatori con un dataset di esempi sufficientemente ampio, si riescono a ottenere buoni risultati anche in caso di input imperfetti e con rumore. La più grossa limitazione di questo tipo di algoritmi si riscontra quando si tenta di classificare due azioni che avvengono quasi contemporaneamente o che comunque si sovrappongono nel tempo, in questo caso spesso viene commesso un errore.

Syntactic approaches

Questo tipo di algoritmi mappa le azioni umane come una stringa di simboli, un simbolo corrisponde ad un'azione atomica generata dal livello base tramite i soliti algoritmi mono-livello. Un'azione è quindi modellata come una come una *production rule* che genera la stringa di azioni atomiche e vengono riconosciute con le tecniche di *parsing* provenienti dal campo dei linguaggi di programmazione. Questi metodi usano le *context-free grammars* e le *stochastic context-free grammars* come modelli per rappresentare le azioni composte.

Una delle limitazioni ancora una volta sta nel fatto che anche questo tipo di algoritmi non è in grado di classificare correttamente attività non sequenziali ma concorrenti; questo è dovuto alla loro stessa natura in quanto essendo le azioni rappresentate come stringhe di sotto-eventi concatenati è impossibile esprimere concetti di parallelismo. Un altro problema in cui si incorre adottando questo approccio sta nel fatto che dal momento che le azioni vengono classificate con tecniche di parsing è necessario definire una serie di production rules che mappi l'intero dominio di sequenze di sotto-eventi.

Description-based approaches

I description-based approaches mantengono esplicitamente le strutture spaziotemporali che compongono le azioni complesse. Anch'essi come gli altri approcci gerarchici scompongono le azioni composte in azioni atomiche descrivendone però, in questo caso, i legami temporali, spaziali e logici che intercorrono tra esse. Di conseguenza le attività vengono rappresentate come una struttura ad albero dove le foglie sono le azioni atomiche riconoscibili per mezzo di single-layer approaches e la radice rappresenta l'azione rappresentata nella sequenza video. Classificare un'attività con un approccio description-based significa individuare in una sequenza osservata le azioni atomiche che soddisfano la struttura che identifica una classe.

A differenza delle due tipologie gerarchiche precedenti, data la loro natura, questi algoritmi sono in grado di classificare azioni parallele e sovrapposte. Uno svantaggio dell'uso di queste tecniche, invece, è dovuto al fatto che non sono in grado di sopperire agli errori commessi negli strati inferiori, e quindi alla mancata individuazione o all'errata classificazione delle sotto-azioni.

3.2 Stato dell'arte con sensori di profondità, il formato RGB-D

Questo settore di ricerca associato alle depth map acquisite tramite sensori Kinect sta avendo particolare successo negli ultimi anni. Tale popolarità è dovuta al range di impieghi che possono avere tecniche di questo tipo. In particolare la possibilità di apprendere e riconoscere determinate azioni umane può permettere la realizzazione di sistemi video interattivi innovativi che possono essere realizzati sia a scopo puramente interattivo, e quindi utilizzati dalle grandi softwarehouse che realizzano videogiochi sia per console che per PC, sia a scopo educativo, in questo caso si può pensare a sistemi interattivi all'interno di musei, sia a scopo commerciale e pubblicitario, dove possono essere installati sistemi dotati di Kinect per

invogliare (magari facendo uso della gamification) il potenziale cliente a conoscere un brand un prodotto interagendo tramite gesti.

3.2.1 Object detection and tracking

Una delle applicazioni più ovvie sulle quali si può essere sfruttata la tecnologia delle telecamere di profondità è l'individuazione di oggetti nella scena e il tracking di essi nei fotogrammi successivi. Questa attività, in assenza di sensori ad infrarosso, viene eseguita tramite sequenze di immagini RGB con la tecnica di *background subtraction* (sottrazione dello sfondo) per isolare il foreground; questa tecnica risulta efficace nel caso in cui la telecamera rimanga fissa e la scena resti più o meno costante (limitare il più possibile variazioni di luminosità, ecc.), mentre nel caso contrario l'estrazione degli oggetti "in primo piano" diventa molto più difficoltosa e lo stato dell'arte prevede l'utilizzo di modelli più avanzati per la sottrazione del background anche se comunque resta difficile ottenere dei buoni risultati.

L'utilizzo di telecamere di profondità come Kinect rende questo task molto più semplice dal momento che sono invarianti per cambiamenti di luminosità e contrasto. Sono diversi gli studi che si sono proposti di sviluppare tecniche che sfruttano le immagini di profondità realizzare sistemi per riconoscere oggetti e tracciare la loro posizione all'interno della scena, le strategie presentate nei diversi studi adottano due diversi approcci: in un primo caso si fa uso solamente delle immagini registrate dal sensore di profondità, in alternativa invece sono state sviluppati algoritmi che cercano di trarre vantaggio dalla combinazione dell'output generato dalla telecamera a infrarossi sia dall'output generato dalla tradizionale telecamera RGB.

L'algoritmo presentato in [15] utilizza esclusivamente delle sequenze video rilevate da sensori di profondità in ambienti indoor per individuare le persone nella scena. Come prima cosa vengono individuate le aree potenziali in cui possono trovarsi delle persone con una *2-D chamfer distance matching scans* dopo di che ogni regione selezionata viene analizzata con un *3-D head model* in fine tramite un algoritmo di region

growing viene individuata la sagoma dell'intero corpo e il viene estratto il contorno.

Al contrario nello studio presentato in [16] invece sfrutta sia sequenze video acquisite attraverso sensori di profondità che sequenze video ottenute con tradizionali fotocamere RGB. La tecnica sfrutta le informazioni estratte dalle telecamere di profondità per individuare più facilmente e con maggior precisione le persone all'interno della scena successivamente il perimetro della sagoma umana viene utilizzato per segmentare il corrispondente fotogramma della sequenza video RGB in modo poi da estrarre delle features visuali che permettono di effettuare il tracking del corpo all'interno della scena fotogramma per fotogramma. L'operazione di tracciamento risulterebbe molto complicata se si disponesse solo della sequenza dell'immagine di profondità; al contrario risulta banale utilizzando riferimenti visivi estratti dalle immagini RGB.

In entrambi gli esempi portati si dimostra che il contributo informativo apportato dai sensori di profondità è notevole e spesso semplifica il lavoro e migliora considerevolmente le prestazioni. Allo stesso modo, però, si evince che le fotocamere a infrarosso non sono la soluzione ad ogni problema ma spesso resta indispensabile accoppiarle a telecamere tradizionali per avere uno spettro di informazioni sufficiente per affrontare problemi di questo tipo.

3.2.2 Object and scene recognition

L'object recognition si distingue dall'object detection per il fatto che la detection individua una o più occorrenze di un oggetto di interesse e le localizza (determinandone la posizione), recognition suppone che l'oggetto sia già localizzato e isolato al fine di individuarne la categoria di appartenenza. Per scene recognition invece si intende una naturale estensione dell'object recognition all'intera scena. Convenzionalmente le tecniche di object and scene recongition fanno affidamento sull'estrazione da immagini RGB di colori, texture, movimenti o la combinazione di queste per rappresentare un oggetto. Quindi viene addestrato un classificatore per determinare se tale oggetto è presente in altre scene. Anche in questo caso,

il basso costo del Kinect, offre una buona opportunità per combinare features RGB a features su depth-images.

Per affrontare questa problematica nell'articolo [17] viene proposto un dataset dove vengono accoppiate immagini RGB e relative immagini di profondità che ritraggono diverse scene che comprendono diversi oggetti organizzati secondo una struttura gerarchica. Successivamente viene dimostrato come è possibile aumentare le performance di algoritmi di object recognition solo combinando tecniche di background subtraction tramite depth-images.

Nell'algoritmo presentato in [18] gli autori migliorano un loro precedente studio aggiungendo ad una tecnica da loro sviluppata, che faceva già uso delle immagini di profondità al fine di etichettare gli oggetti di una scena, il *contestual modelling* combinando tecniche di segmentazione ad albero con l'algoritmo per costruire superpixel *Markov random fields* (*MRFs*).

Una nuova tecnica innovativa è stata proposta in [19]. La particolarità dell'algoritmo presentato sta nel cercare di estrarre caratteristiche geometriche tridimensionali degli oggetti ritratti nella scena; gli autori hanno sviluppato a tal fine una nuova tecnica di feature extraction chiamata HONV. Il principio fondante di questa tecnica è che è possibile riconoscere un oggetto conoscendo la sua superfice 3D senza di fatto conoscerne il colore o la texture. Per estrarre queste features si calcolano le orientazioni dei piani tangenti alla superficie dell'oggetto (questo può essere fatto come concatenazione degli istogrammi locali degli angoli azimutali e degli angloli zenitali). Lo studio ha dimostrato come questa tecnica si dimostri significativamente superiore rispetto agli istogrammi dell'orientazione del gradiente (HOG).

3.2.3 Human activity analysis

Questo settore della visione artificiale sta assumendo una notevole importanza negli ultimi anni grazie alle numerose possibilità di impiego che spaziano dalla sicurezza intesa sia come sorveglianza di luoghi pubblici sia come monitoraggio di luoghi domestici (ad esempio nel caso di anziani che

vivono da soli e potrebbero ferirsi o cadere, in questi casi il sistema provvederebbe a riconoscere l'accaduto e a inviare un allarme) all'intrattenimento dove in questo caso si può pensare sia allo sviluppo di videogames veri e propri sia di usare queste tecniche in applicazioni a fini pubblicitari e/o educativo-interattivi con l'introduzione di concetti di gamification. A dimostrare la sua popolarità negli ultimi dieci anni è stata pubblicata una notevole quantità di articoli scientifici a riguardo i cui studi sono stati finanziati sia in ambito accademico sia da agenzie di consumo e di sicurezza sia in ambito industriale. La maggior parte di queste pubblicazioni però prevedeva l'uso delle tradizionali sequenze video basate su canali RGB, l'introduzione dei sensori di profondità diventano quindi un ottimo strumento per ipotizzare un considerevole miglioramento di queste tecniche.

Quando si parla di analisi di attività umane i settori di ricerca sono due: il primo comunemente chiamato *pose estimation* ha l'obiettivo di individuare nel modo più veloce e con la maggior precisione possibile la posizione degli arti e del corpo umano presente nella scena 3D; nel caso dell'*activity recognition* invece quello che si cerca di fare è estrarre e comprendere la semantica delle attività che sta eseguendo la persona o le persone ritratte nella scena e classificarle. In parole povere l'obiettivo della pose estimation è quello di fornire una posizione 3D dello scheletro umano stilizzato mentre quando si parla di activity recognition invece si cerca di capire cosa la persona sta facendo analizzando dei pattern temporali in funzione della posizione degli arti e del corpo.

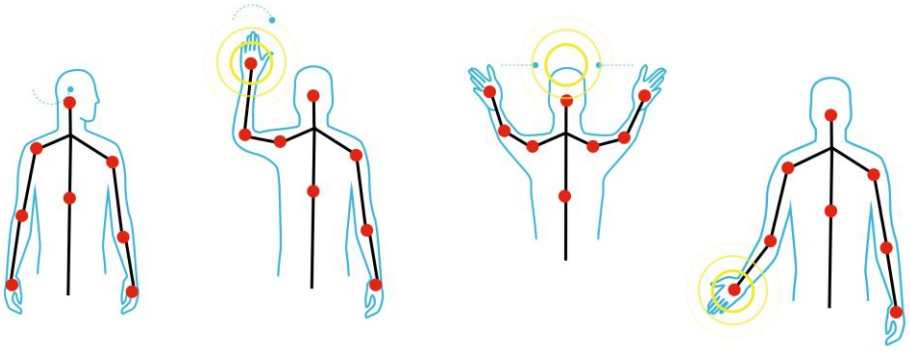

Figura 7 - Esempio di individuazione dello scheletro umano stilizzato in diverse pose del corpo

3.2.4 Pose estimation

In [20] Shotton et al. hanno presentato un metodo per la stima della posa basata sulla suddivisione dello scheletro nei differenti giunti articolari a partire da un singolo frame video, ovvero una singola immagine RGB-D. L'algoritmo da loro presentato è quindi totalmente indipendente da qualsiasi tipo di informazione temporale e quindi robusto per quanto riguarda l'eventuale perdita del tracciamento della posizione corrente. Nello studio hanno utilizzato un approccio di riconoscimento basato su *object recognition* individuando una rappresentazione intermedia delle parti del corpo al fine di semplificare il complesso problema della stima della posa in uno più semplice di classificazione per-pixel; in fine l'algoritmo produce un punteggio con confidenza per ogni diversa posizione 3D delle articolazioni retroproiettando il risultato della classificazione. Il metodo, testato su un dataset caratterizzato da una forte variabilità, si è dimostrato robusto in caso di cambiamenti di posa, struttura corporea (adulto, bambino, alto, basso, magro, robusto, ecc.) e tipologia di abbigliamento. Il contributo apportato da questo articolo sta nella possibilità individuare le singole parti del corpo al fine di avere una valutazione complessiva di stima della posa e, come descritto in un successivo studio [21], rendere questo tipo di algoritmo invariante per tipologia di struttura corporea posa e abbigliamento indossato allenando un classificatore su un vasto dataset che presenti un ampio spettro di variabilità. Inoltre in [21] si mostra come può essere parallelizzato questo tipo di task rendendo l'algoritmo fino a dieci

volte più veloce, aspetto da non trascurare nell'ambito della pose estimation.

Nello studio condotto in [22] è stato apportato un miglioramento alla tecnica di tracciamento dello scheletro umano già implementato dagli ideatori di Kinect, in particolare viene adottato un *offset vote regression approach* nel quale ogni pixel dell'immagine "vota" per la posizione delle diversa articolazioni del corpo (a differenza del metodo precedente dove i voti venivano usati per etichettare le varie parti del corpo); questo algoritmo si dimostra in grado di stimare la posizione degli arti anche se alcune parti del corpo non sono visibili a casa di oggetti che occludono l'inquadratura o a causa del limitato campo visivo del sensore.

Un altro metodo migliorativo è presentato in [23] dove gli autori estendono la tecnica di machine learning originale introducendo la possibilità di imparare a predire direttamente le corrispondenze tra i pixel dell'immagine e la traccia del modello 3D; è impiegata una tecnica di minimizzazione dell'energia per ottimizzare efficientemente la posa della traccia del modello 3D senza dover ricorrere alla tradizionale tecnica iterativa di *interrated closest point* (ICP).

Ye *et al.* in [24] e Shen *et al.* in [25] si propongono di trovare una tecnica di pose estimation altamente accurata e robusta a partire da una singola immagine di profondità, trascurando però l'elevata complessità computazionale dell'algoritmo impiegato. Entrambi i metodi riportati si basano su due step di esecuzione [Figura 8]: grazie a un algoritmo di pose estimation viene individuata una posa di partenza che poi viene migliorata con tecniche di raffinamento. Nel primo studio [24] viene presentato un nuovo sistema di pose estimation che acquisisce come input una singola immagine di profondità sulla quale viene fatto un matching tra una serie di immagini pre-acquisite al fine di generare sia una stima della configurazione del corpo sia un etichettamento semantico della nuvola di punti dell'immagine del sensore. La stima iniziale viene quindi rifinita in un secondo step adattando la configurazione del corpo predefinita con quella dell'immagine osservata in input. In aggiunta viene anche adottata

una tecnica di smoothing al fine di migliorare le depth map molto "rumorose". I risultati ottenuti su diversi dataset dimostrano come l'algoritmo sia molto più accurato rispetto ai precedenti metodi usati nello stato-dell'arte.

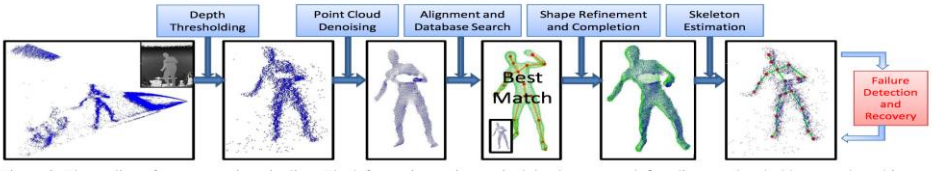

Figura 8 - Workflow dell'algortimo utilizzato in [24] i primi tre step si occupano della pose estimation mentre negli ultime tre viene eseguita la rifinitura.

Un'altra tecnica divisa in due fasi viene proposta in [25] dove nel secondo step la posa iniziale viene rifinita in modo da aumentare le performance. A differenza dell'algoritmo precedente si cerca di migliorare la stima della posa traendo vantaggio dalla *consistenza temporale dei movimenti* e dalla *distorsione sistematica*. Quello che si cerca di fare è filtrare tutte quelle che possono essere considerate stime "anomale" avendo come riferimento un modello di azione umano e cercando di mantenere la consistenza temporale tra le pose dei frame consecutivi ovvero, è improbabile che in un frame una persona sia ritratta con il braccio destro alzato e in quello successivo sia abbassato senza passare da una posizione intermedia.

3.2.5 Activity recognition

Queste tecniche utilizzano delle sequenze video acquisite tramite sensore di profondità per tracciare la diversa posizione delle articolazioni nei diversi frame e costruire un modello che rappresenti l'azione svolta.

Negli articoli [26] [27] [28] sono presentati delle feature compatte che permettano di classificare efficientemente diverse attività. In [26] gli autori presentano un nuovo tipo di descrittori, da loro nominati EigenJoints, basati sulla differenza della posizione dei joint e combinano le informazioni dell'azione si nelle pose statiche sia nei movimenti. Successivamente utilizzano un Naïve-Bayes-Nearest-Neighbor (NBNN) per la classificazione e la multi-classificazione delle azioni. L'algoritmo viene testato sul dataset *Microsoft Research (MSR) Action3D* dimostrandosi

superiori a quelli realizzati in precedenza; affinché questa tecnica risulti efficace possono anche essere utilizzati solamente 15-20 frame estratti dalla sequenza originale.

La tecnica presentata in [27] fa anch'essa uso degli Eigen joints ma in questo caso l'algoritmo viene rifinito introducendo *Accumulated Motion Energy (AME)* che permette di filtrare i frame dalla sequenza video iniziale in modo da eliminare quelli "rumorosi" riducendo così il costo computazionale. Anche in questo caso l'algoritmo si è dimostrato molto efficiente e permette di alleggerire il carico di lavoro sulle machine permettendo di analizzare fino al 60-70% dei frame in meno rispetto alla sequenza iniziale.

Un metodo differente ma sempre caratterizzato da descrittori compatti è stato presentato da Xia *et. Al* nel loro articolo [28] dove illustrano un nuovo metodo di action recongition basato si istogrammi delle posizioni 3D delle articolazioni chiamati HOJ3D. Questi descrittori vengono estratti dagli scheletri individuati tramite l'algoritmo a cui era stato fatto riferimento precedentemente in [20]. I descrittori sono quindi proiettati usando LDA poi clusterizzati in k cluster di *visual words* che rappresentano le pose prototipali dell'azione. L'evoluzione temporale della visual word viene modellata tramite il modello discreto *hidden Markov models* (HMMs). In aggiunta, grazie al modello del sistema a coordinate sferiche e al robusto metodo di stima dello scheletro, il metodo si è dimostrato invariante ai diversi punti di vista della scena. Il dataset su cui è stato testato è composto da 200 sequenze 3D di 10 attività indoor eseguite da 10 diversi individui ripresi da diverse inquadrature.

Un approccio differente viene usato in [29] dove, al contrario delle tecniche precedenti non viene utilizzato lo scheletro estratto dalle immagini di profondità ma l'algoritmo si basa direttamente sulle immagini RGB-D; con questo si intende sia la parte a colori sia la parte dello spettro di profondità. Inoltre gli autori specificano che, se altre tecniche di questo tipo cercano di estrarre informazioni prima dalla componente RGB per poi combinarle a quelle ottenute dalle depth map, il loro algoritmo estrae simultaneamente i

descrittori dall'intero frame RGB-D. Quella presentata è una metodologia adattativa che permette di estrarre automaticamente delle features spaziotemporali riconducendo il task a un problema di ottimizzazione a cui viene trovata una soluzione grazie a un approccio *graph-based genetic programming* (RGGP) dove un gruppo di operatori 3D sono casualmente assemblati in una soluzione basata su grafi e, tramite l'algoritmo genetico, questa viene progressivamente migliorata. In fine la combinazione che ottiene le migliori performance viene selezionata come rappresentazioni sub-ottima.

Altri studi [30] [31] [32] invece sono tutti accumunati dall'utilizzo dell'*hidden Markov model* (HMM), un modello grafico popolare usato comunemente negli algoritmi di riconoscimento delle attività basato su sequenze video RGB. In [30] la tecnica presentata si compone di una struttura a un solo strato mentre in [31] [32] si adotta una struttura a due strati che permette di mappare meglio le attività umane in quanto queste possono essere considerate come una serie di sotto-azioni che si susseguono nel tempo.

In [33] Rayes *et al.* introducono un nuovo approccio per il riconoscimento di gestualità basato su un innovativo *Feature Weighting approach* in aggiunta al *Dynamic Time Warping framework.* Le features estratte dalla posizione 3D delle articolazioni umane vengono comparate luna con l'altra all'interno dei vari fotogrammi della sequenza e i pesi vengono assegnati alle features in base alla loro variabilità interclasse. La tecnica sopra descritta viene anche applicata per riconosce l'inizio e la fine dei festi nelle sequenze video. Il risultato ottenuto supera in performance il metodo tradizionale.

Wang *et al.* [34] propongono una nuova feature chiamata *local occupancy pattern* che permette di rappresentare la "depth appearance" che è studiata per catturare la relazione tra il copro umano e gli oggetti circostanti con cui si interagisce. Inoltre definisce un *actionlet* una particolare congiunzione di caratteristiche per ogni sottoinsieme di joint costituendo una feature strutturata. Le actionlet sono utili per addestrare un classificatore a

riconoscere le azioni umane inoltre queste si dimostrano robuste in caso di variazioni di rumore, traslazione e disallineamento temporale.

Al fine di trovare un algoritmo che permettesse l'estrazione di features in tempro ridotto per permetterne l'utilizzo in sistemi interattivi low-latency nell'articolo [35] viene presentato un metodo che fa uso degli "*action point*". Questo innovativo concetto è definito come in preciso istante temporale nel quale la presenza di un'azione può essere facilmente identificata per tutte le istanze di quella determinata azione.

4 ALGORITMO DI ESTRAZIONE E CLASSIFICAZIONE DEI DESCRITTORI

L'idea di base dell'algoritmo presentato in questa tesi viene da una rivisitazione della tecnica proposta in [1] da Manzi *et al.* dove dopo una fase di normalizzazione delle posizioni dei joint dello scheletro si fa uso di un algoritmo di clustering per individuare le pose chiave dell'azione rappresentata nel video per poi segmentare la sequenza video compressa tramite una finestra scorrevole. L'algoritmo implementato nello studio qui presentato ricorda fortemente l'approccio proposto da Manzi *et al.* ma sono state apportate alcune modifiche come l'utilizzo di un metodo di clustering differente, inoltre è stato introdotto un meccanismo di majority voting quando si tenta di classificare una nuova sequenza. Nei seguenti paragrafi vi è una dettagliata descrizione dell'algoritmo e, nello specifico, verranno evidenziati in casi in cui questo differisce da quello di presentato in [1].

L'algoritmo si compone di 4 step [Figura 9] a partire dagli scheletri della sequenza video catturata dalla telecamera di profondità vengono selezionate delle features che costituiranno poi un training set per addestrare un classificatore SVM. Nello specifico ogni step opera come segue.

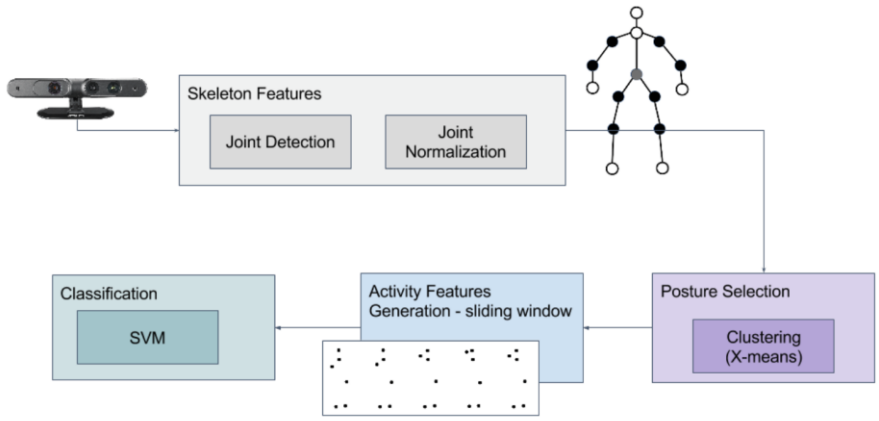

Figura 9 - Workflow operativo dell'algoritmo presentato nell'articolo [1]. L'algoritmo si compone di quattro step sequenziali, i primi tre permettono di raffinare man mano una feature che rappresenta l'azione che servirà poi nel quarto step ad addestrare un classificatore SVM.

4.1 Step 1: Posture Features

A seconda delle varie tecniche di individuazione e traking dello scheletro nei vari frame si costruiscono diversi modelli costituiti solitamente da 15 o 25 joint. Utilizzare l'insieme completo dei joint al fine di estrarre delle feature sulla postura per i diversi frame diventa ridondante (vengono usate troppe informazioni, molte delle quali non sono necessarie) è computazionalmente oneroso. Questo approccio scorretto potrebbe inoltre determinarne un calo dell'efficacia e di precisione dell'algoritmo stesso; a tale scopo vengono selezionati dei joint "chiave[2]" da utilizzare per la costruzione dei quelle che vengono chiamate *posture features*, ovvero dei vettori numerici che contengono le informazioni sulla postura del soggetto in ogni frame del video. Gli studi condotti dagli autori dell'articolo [1] hanno evidenziato che la migliore combinazione si ottiene con l'uso di 7 joint perimetrali (rispettivamente testa, collo, mani, piedi e torso, quest'ultimo usato da riferimento [Figura 10]). Dal momento che le posizioni spaziali nelle coordinate X,Y,Z vengono prese come distanza dell'articolazione dal sensore queste non possono essere direttamente utilizzate per due motivi principali: la persona non si troverà mai alla stessa distanza dal sensore, le persone possono avere una struttura corporea differente l'una dall'altra. Queste coordinate 3D devono essere quindi normalizzate; questa operazione viene fatta prendendo come riferimento joint denominato come *torso*. Selezionati n joint J_i appartenenti ad uno scheletr si definiscono quindi le *posture features* come il feature vector f

$$f = [j_1, j_2, \dots, j_{n-1}]$$

dove j_i è calcolato come:

$$j_i = \frac{J_i - J_0}{\|J_1 - J_0\|} \qquad i = 1, 2, \dots, n-1$$

dove J_1 è l'articolazione che identifica il collo e J_0 l'articolazione che identifica il torso. La lunghezza di f sarà quindi pari a $3(n-1)$ in quanto non

[2] Possiamo definire *joint chiave* i joint più significativi per il processo di classificazione delle posture.

viene calcolata la distanza euclidea tra le due posizioni dei joint ma vengono mantenute separatamente le distanze delle tre componenti dello spazio euclideo.

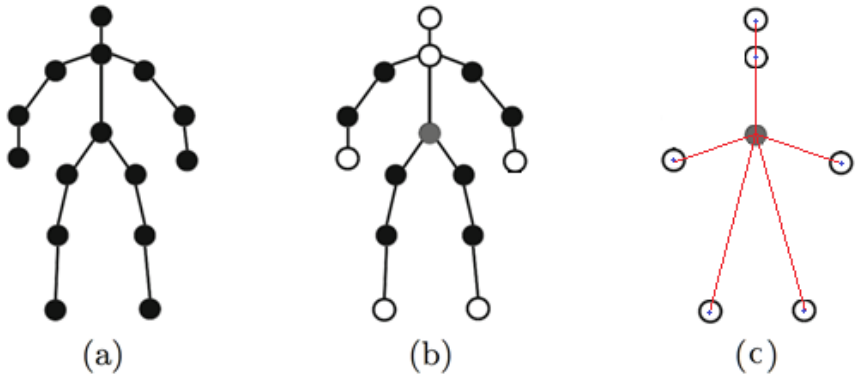

Figura 10 - Nella figura (a) possiamo vedere lo scheletro iniziale, l'input dell'algoritmo, nella figura (b) possiamo vedere colorati in nero i joint che verranno presi in considerazione per il calcolo delle features mentre in bianco quelli esclusi; il pallino colorato di grigio è il joint "torso" che viene usato come riferimento per la normalizzazione. (c) invece è la rappresentazione grafica delle posture features ovvero l'output finale del primo step di estrazione, ovviamente qui viene rappresentata la distanza euclidea, in realtà il vettore di feature mantiene le distanze scomposte nelle 3 dimensioni.

4.2 Step 2: Selection of postures with gesture clustering

Dal momento che azioni umane non sono discrete ma continue nel tempo e inoltre possono, in certi casi, mantenere una posizione costante per diversi fotogrammi è fondamentale individuare, nell'arco temporale in cui si svolge l'azione, quali sono le pose chiave che forniscono un maggiore contributo informativo per distinguere un'azione dall'altra. La comunità scientifica ha proposto innumerevoli varianti per determinare quali siano le *key poses*, nello studio [1] viene proposto di utilizzare una tecnica di *clustering* al fine di raggruppare tutte le pose simili in un certo numero di insiemi.

Con il termine Clustering (in italiano «raggruppamento») si denota una famiglia di metodi di classificazione non supervisionata in grado di individuare raggruppamenti intrinsechi (cluster) di pattern nello spazio multidimensionale, e di definire in corrispondenza di tali raggruppamenti le

classi (incognite)[3]. In poche parole dato un insieme di feature vector come input, un algoritmo di clustering, restituisce un partizionamento dell'insieme di partenza, raggruppando tra loro i vettori che hanno caratteristiche simili. Esistono moltissimi algoritmi di clustering ognuno dei quali con numerose varianti, possiamo distinguerne due principali categorie: gli algoritmi *agglomerativi* che adottano un approccio bottom-up ovvero che gradualmente aggregano cluster più piccoli per formarne di più grandi, gli algoritmi *disgiuntivi* che a partire da un insieme unico gradualmente lo dividono in sotto insiemi. Un'altra classificazione che è possibile fare è quella tra gli algoritmi che in input necessitano di conoscere quanti cluster devono creare e quelli che invece non lo richiedono.

Manzi *et al.* utilizzano l'algoritmo X-mean [36] che permette di determinare in modo automatico il numero di cluster più adatto per quell'insieme di dati e inoltre si dimostra essere meno sensibile del tradizionale K-mean nel caso l'algoritmo individui un minimo locale.

Nell'implementazione fatta durante questo studio invece si è preferito utilizzare il tradizinale K-mean per due principali ragioni: la prima era quella di verificare era davvero necessario l'uso di un algoritmo più articolato come l'X-mean per il clustering delle posizioni, la seconda invece è per garantire che la lunghezza dei vettori prodotti in questa fase sia sempre maggiore dell'ampiezza della sliding window applicata nello step successivo [4.3 Step 3: Activity features with Sliding Window].

A questo punto per ciascun frame viene sostituita la posture feature con le coordinate del centroide del cluster di appartenenza ottenuto calcolando la distanza minima come segue:

$$\min_{C} \sum_{i=1}^{k} \sum_{f_i \in C_j} \|f_i - \mu_j\|^2$$

dove: f_i è l'i-esimo posture feature vector mentre μ_j è il centroide del j-esimo cluster C_j.

[3] http://bias.csr.unibo.it/maltoni/ml/DispensePDF/6_ML_Clustering.pdf

L'output finale di questo step della computazione è una sequenza del tipo [C_1, C_1, C_2, C_2, C_2, ... , Cn, Cn] di dimensione pari a quanti sono i frame della sequenza video iniziale.

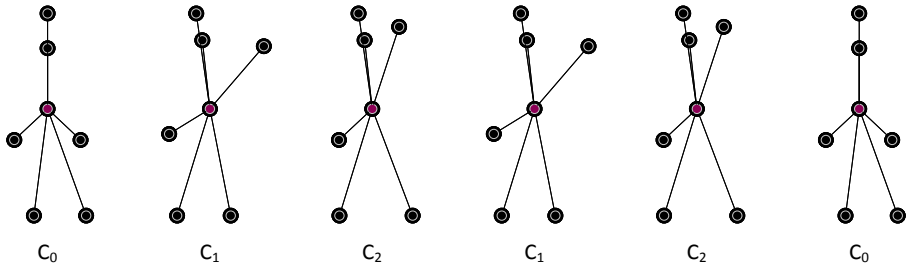

Figura 11 – Esempio di sequenza compressa, alla fine dello Step due si avrà una sequenza simile con la differenza che in questo caso sono stati eliminati tutti i frame adiacenti con lo stesso centroide.

4.3 Step 3: Activity features with Sliding Window

In questa fase della computazione ci troviamo con una sequenza di centroidi temporalmente allineati che contiene sicuramente molteplici centroidi adiacenti uguali, per questo motivo vanno eliminate tutte le occorrenze multiple adiacenti in modo da ottenere una sequenza compressa [Figura 11] dove sono presenti solo le transizioni da un cluster ad un altro. La sequenza compressa non ha una dimensione fissa questo dipende da quanto è ripetitiva l'attività dalla quale è stata estratta; un'azione molto ripetitiva avrà una sequenza compressa più lunga in quanto si alterneranno numerose volte gli stessi centroidi, mentre un'azione meno ripetitiva avrà una sequenza compressa più corta dovuto al fatto che mancherà quell'alternanza tra centroidi tipica delle azioni cicliche.

Una sequenza di output può essere ad esempio [C_1, C_3, C_2, C_4, C_5, C_3, C_2, C_1, C_5] (vedi [Figura 11]). Dal momento che un'azione umana non ha un momento di inizio e di fine ben precisa sulla sequenza compressa ottenuta viene utilizzata una *sliding window* di lunghezza *w* (che costituisce uno dei parametri del sistema) che, scorrendo di una posizione alla volta, genera una serie di features lunghe *w* che identificano una determinata azione. Tutte le sequenze duplicate vengono tralasciate.

Questo porta a due vantaggi:

- Il primo è uniformare la lunghezza dei feature vector, dal momento che, come affermato in precedenza, la lunghezza della sequenza compressa è variabile, applicando una finestra scorrevole si producono diverse sotto-sequenze di lunghezza costante *w*.
- Il secondo è di produrre, per ogni singola sequenza video, più activity feature vectors che rappresentano la sequenza identificando come momento di inizio e di fine istanti temporali diversi in modo da coprire ogni eventualità.

Un esempio dell'output finale è il seguente:

$$F_1 = [C_1, C_3, C_2, C_4, C_5, C_3],$$

$$F_2 = [C_3, C_2, C_4, C_5, C_3, C_2],$$

$$F_3 = [C_2, C_4, C_5, C_3, C_2, C_1],$$

$$F_4 = [C_4, C_5, C_3, C_2, C_1, C_5]$$

4.4 Step 4: Training e classificazione

Con lo step precedente si è conclusa la fase di estrazione dei descrittori dalle sequenze video, in questo paragrafo viene descritto come vengono utilizzati i descrittori estratti in precedenza al fine di addestrare un classificatore e come questo viene utilizzato per attribuire una categoria a delle sequenze video dove la classe è ignota.

Per quanto riguarda lo studio di Manzi *et al.* [1] le feature generate nello step precedente vengono usate per addestrare un classificatore SVM multiclasse (nello specifico viene usato l'algoritmo Sequential Minimum Optimization SMO). Dal momento che SVM nasce come classificatore binario viene usata una strategia one-vs-one per ottenere il risultato di un classificatore multi-classe. Nello specifico nell'articolo di riferimento non vengono aggiunte ulteriori informazioni riguardo a questo step dell'algoritmo. Qui di seguito verrà presentato più in dettaglio l'approccio utilizzato nello studio condotto per questa tesi.

Si sceglie di usare come classificatore le *Support Vector Machine* (SVM), la teoria che governa i meccanismi di funzionamento di SVM è stata introdotta da Vapnik a partire dal 1965 (statistical learning theory), e perfezionata più recentemente (1995) dallo stesso Vapnik e altri. SVM è uno degli strumenti più utilizzati per la classificazione di pattern. Invece di stimare le densità di probabilità delle classi, Vapnik suggerisce di risolvere direttamente il problema di interesse (che considera più semplice), ovvero determinare le superfici decisionali tra le classi (classification boundaries). Date due classi di pattern multidimensionali linearmente separabili, tra tutti i possibili iperpiani di separazione, SVM determina quello in grado di separare le classi con il maggior margine possibile. Il margine è la distanza minima di punti delle due classi nel training set dall'iperpiano individuato.

Una volta addestrato il classificatore con tutti i feature vector prodotti da tutte le sequenze appartenenti all'insieme dei dati di test per classificare un nuovo video che ritrae un'attività la cui classe è ignota si procede come segue:

a) Si estraggono i descrittori dell'attività seguendo i precedenti Step (Step 1, Step2, Step 3).
b) Si demanda al classificatore l'attribuzione della classe ad ognuno dei vettori estratti nel passo (a).
c) Utilizzando un meccanismo di *majority voting,* si attribuisce alla sequenza video la classe che appare il maggior numero di volte nei descrittori classificati nel passo (b).

4.5 Rappresentazione in pseudocodice

```
SVMClassifier = initNewSVMClassifier()

trainFromSequences(Sequence[] SubjectsSequences, int K, int W,
int[]SelectedJoints)
{
        TrainingSet = initTrainingSet();

        foreach(Sequence in SubjectsSequences)
        {
                features = computeGestureFeatures(Sequence,
                   SelectedJoints)
                compressedSequence =
                   clusterizeAndCompressSequence(Sequence, features, K)
                Sequence.ActivityFeatures =
                   applySlidingWindow(compressedSequence, W)

                TrainingSet.Add(Sequence.ActivityFeatures)
                SVMClassifier.Train(TrainingSet)
        }
}

classifyNewSequence(Sequence UnknownSequence, int K, int W,
int[]SelectedJoints)
{
        features = computeGestureFeatures(UnknownSequence,
            SelectedJoints)
        compressedSequence =
            clusterizeAndCompressSequence(UnknownSequence, features, K)
        UnknownSequence.ActivityFeatures =
            applySlidingWindow(compressedSequence, W)

        SVMClassifier.Classify(UnknownSequence.ActivityFeatures)
        Classes = initList()
        foreach(c in UnknownSequence.ActivityFeatures.getClass())
                Classes.Add(c)
        estimatedClass = majorityVoting(Classes)
}

computeGestureFeatures(Sequence, int[] SelectedJoints)
{
        Features = initFeatureList();
        foreach(frame in Sequence)
        {
                skeleton = frame.skeleton
                foreach(joint in selectedJoints)
                {
                        Features.Add(distance(skeleton[joint],
                            skeleton[torso]))
                }

        }
        return Features
}

clusterizeAndCompressSequence(Sequence, feature, k)
{
        clusters[] = KmeanClustering(feature);
        compressedSequence = sequenceWithNearestCluster(sequence,
             clusters)
        compressedSequence = removeDuplicates(compressedSequence);
        return compressedSequence
}
```

5 IMPLEMENTAZIONE

Si è scelto di implementare l'algoritmo nel linguaggio C# dal momento che non è tra le priorità di questo studio l'intenzione di testarne la rapidità e ottimizzarne il costo computazionale. Inoltre, usare il C# come linguaggio di programmazione ha permesso l'uso della libreria BioLab, creata dal Biometric System Laboratory[4] dell'Università di Bologna attivo presso il Department of Computer Science and Engineering (DISI). Questa libreria contiene l'implementazione di una serie di strutture dati e algoritmi machine learning, classificazione supervisiona, clustering, pattern recognition e image processing utili in ambito della visione artificiale.

In aggiunta, al fine di rendere semplice la manutenzione e l'aggiornamento del software stesso, sono state realizzate delle strutture dati preposte alla gestione dei frame video e delle informazioni degli scheletri estratti da ognuno di questi. In [Figura 13] è rappresentata tramite UML Class Diagram l'organizzazione delle classi per gestire la struttura dei frame.

Per quanto riguarda la struttura adottata per l'implementazione degli algoritmi di estrazione delle features è stata mantenuta la stessa struttura utilizzata all'interno della libreria BioLab; ogni step dell'algoritmo descritto precedentemente [Capitolo 4] è stato implementato con una classe C# a sua volta estende la classe astratta *Algorithm* [Figura 14].

Sono state molto utili anche tutte le classi, già presenti in BioLab, che gestiscono i vettori di descrittori come *FeatureVector* e *FeatureVectorSet.*

Per quanto riguarda l'implementazione del classificatore SVM, BioLab implementa già una sua versione dell'algoritmo di training e di classificazione appoggiandosi a sua volta alla libreria *libSVM.*[5] L'obiettivo Chih-Chung Chang e Chih-Jen Lin [37], i creatori della libreria, è quello di fornire agli utenti una libreria che implementi le Support Vector Machine in modo efficiente, semplice e intuitivo all'utilizzo. La libreria fornisce una

[4] http://biolab.csr.unibo.it/Home.asp
[5] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

semplice interfaccia alla quale gli utenti si possono facilmente collegare; le principali caratteristiche di libSVM sono:

- Formulazioni di SVM diverse
- Classificazione multi-classe efficiente
- Modello di selezione per training e test con Cross Validation
- Diversi tipi di Kernel
- Stima della probabilità
- SVM pesato per dati sbilanciati
- Sorgenti C++ e Java
- Interfacce per Python, R, MATLAB, Perl, Ruby, Weka, Common LISP, CLISP, Haskell, OCaml, LabVIEW, PHP e C# .NET con estensioni per CUDA.
- GUI per la rappresentazione grafica del funzionamento delle SVM
- Ricca documentazione e approfondita guida pratica introduttiva[6].

In fine è stata realizzata una semplice Windows Form Application utile per eseguire i test; attraverso la GUI [Figura 12] si possono scegliere in modo rapido le configurazioni dell'algoritmo (quanti e quali joint usare, il valore di K per l'algoritmo di clustering, la dimensione della window size W) e su quale dataset eseguire i test.

[6] https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

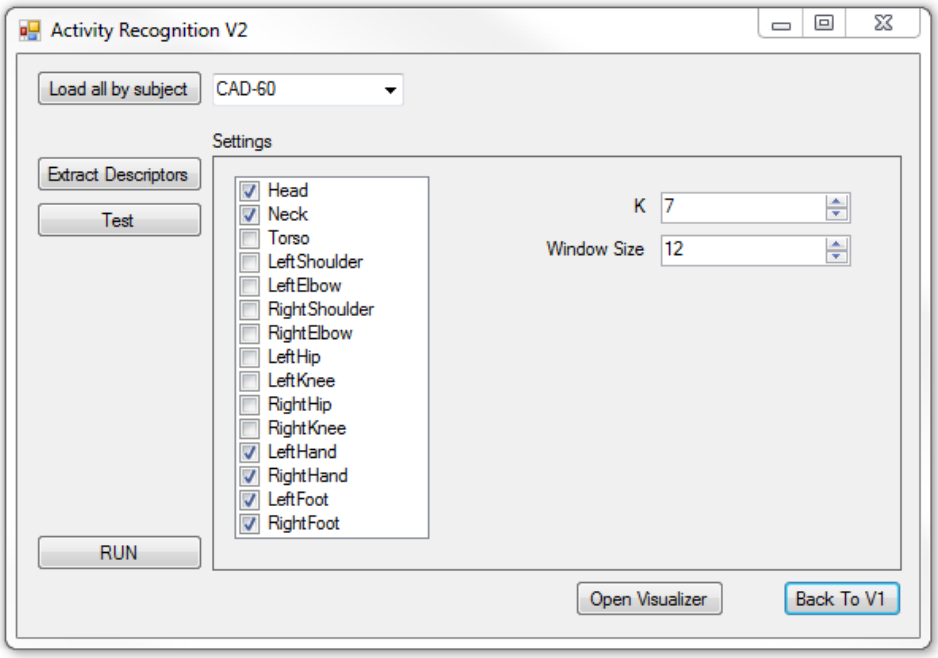

Figura 12 - Windows Form GUI realizzata per eseguire i test su diversi dataset usando setup differenti

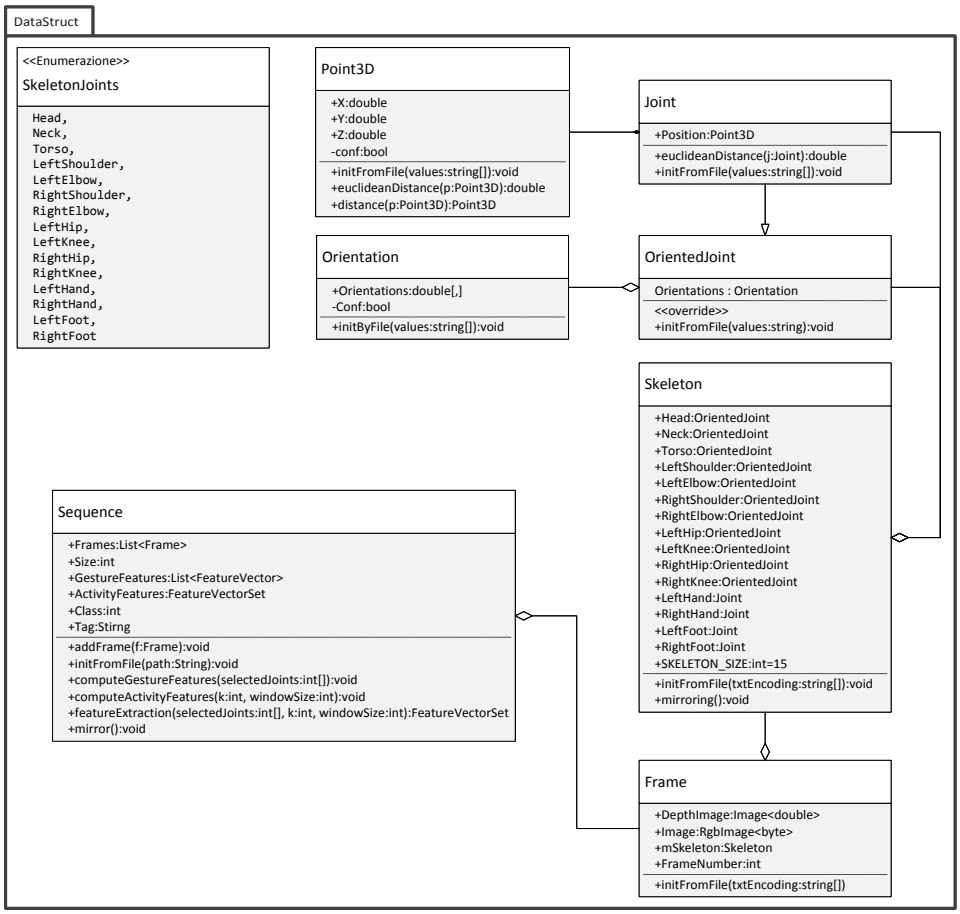

Figura 13 – UML Class Diagram relativo alle classi usate per la gestione delle sequenze video, dei frame e delle informazioni sugli scheletri tracciati.

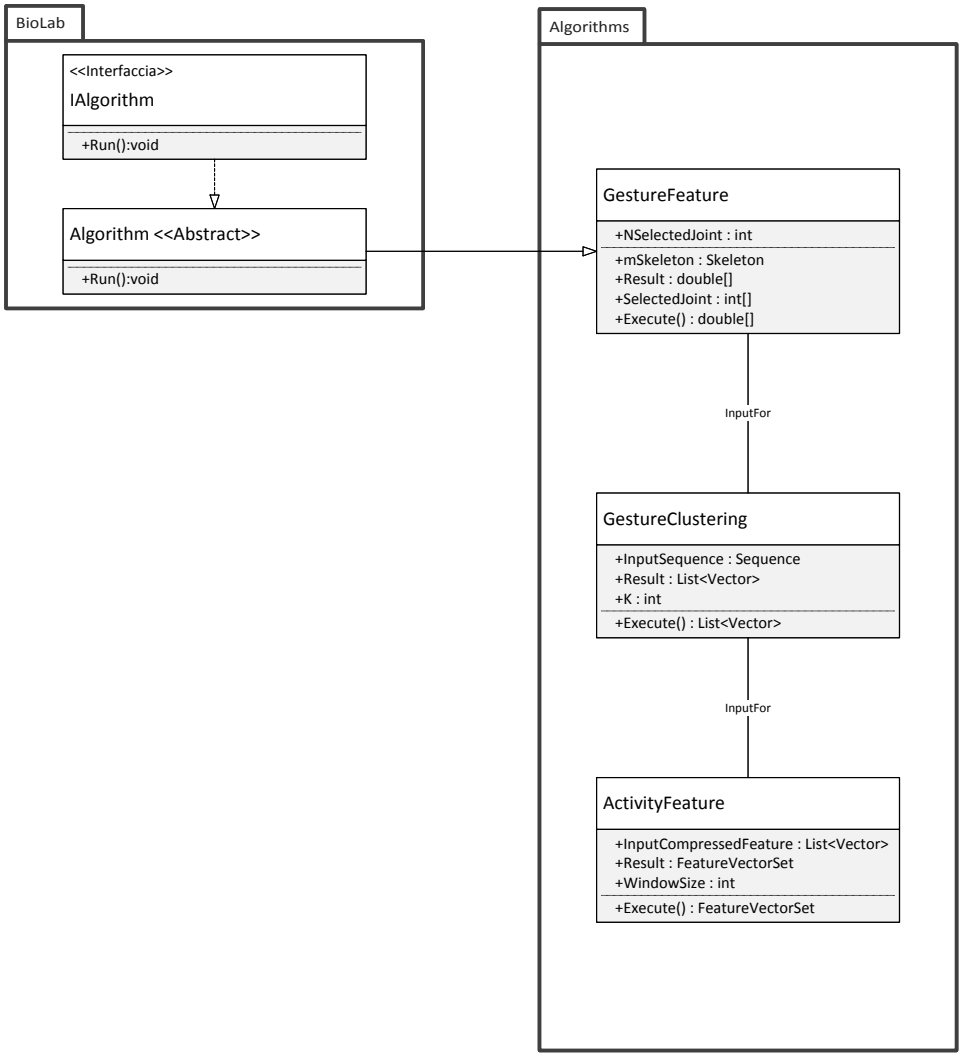

Figura 14 - UML Class Diagram relativo alla struttura dell'implementazione degli algoritmi di feature estraction illustrati precedentemente [Capitolo 4 Algoritmo di estrazione e classificazione dei descrittori].

6 RISULTATI SPERIMENTALI

L'algoritmo è stato testato su diversi dataset: il primo, ampiamente utilizzato dalla comunità scientifica in ambito di avtivity recognition, è il Cornell Acrivity Dataset (CAD-60), il secondo, costruito dagli stessi autori, è CAD-120, un dataset più strutturato che contiene ulteriori descrittori al fine di classificare anche l'interazione del soggetto umano con gli oggetti. Inoltre contestualmente all'implementazione dell'algoritmo oggetto di questa Tesi è stato realizzato un software per l'acquisizione di un nuovo dataset chiamato UniBo Human Activity Dataset (UHAD) [vedi 6.3 UniBo Human Activity Dataset (UHAD)] utilizzando come sensore il nuovo Kinect 2 che ha permesso di tracciare scheletri a 25 joint contro i 15 joint dei dataset precedentemente elencati. L'algoritmo, con le opportune modifiche (dovute al cambiamento del numero di articolazioni negli scheletri) è stato quindi testato anche su quest'ultimo dataset.

Nei paragrafi seguenti sono descritti più nello specifico i dataset utilizzati, i protocolli di test adottati, i risultati ottenuti e confrontati con quelli presentati nel sito ufficiale del Computer Science Department, Cornell University. Come indicatori per quantificare le performance e quindi avere dei termini di paragone tra i risultati dei vari algoritmi sono stati usati i valori di: Precision, Recall, Accuracy e F-Measure. Qui di seguito viene descritto il modus operandi per il calcolo degli indicatori.

Al fine di avere una stima complessiva dei risultati è necessario tradurre il problema da multi-classe a problema di classificazione binario, per fare questo viene presa volta per volta una delle classi come riferimento etichettandola come classe *positive* mentre, tutte le rimanenti, come *negative* e viene creata la seguente matrice di confusione.

Popolazione Totale	**Classe predetta *positve***	**Classe predetta *negative***
Classe reale *positive*	True Positive (TP)	False Negative (FN)
Classe reale *negative*	False Positive (FP)	True Negative (TN)

Tabella 3 - Matrice di confusione per un problema di classificazione binaria.

Avendo come riferimento i valori contenuti nella Tabella 3 si calcolano gli indicatori di performance come:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{P + N}$$

$$F - Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Iterativamente si raccolgono allo stesso modo i valori per tutte le classi, gli score finali vengono calcolati come media di tutti i valori precedentemente calcolati.

6.1 Cad-60

Il dataset CAD-60[7] raccolto dal Computer Science Department, Cornell University[8], contiene 60 sequenze video acquisite tramite Kinect (sensore di profondità e telecamera RGB) che ritraggono azioni svolte nella vita quotidiana. Le attività sono eseguite da quattro differenti individui, due uomini, 2 donne di cui solamente un mancino e i rimanenti destrorsi. I video sono stati ripresi in ambiente domestico indoor, nella scena sono presenti altri oggetti statici, a diverse distanze dal sensore di profondità, e il soggetto che compie l'azione. Sono stati registrati 12 diversi tipi di azioni etichettati come: parlare al telefono, scrivere su una lavagna, bere acqua, risciacquarsi la bocca con l'acqua, lavarsi i denti, mettere le lenti a contato, parlare sul divano, rilassarsi sul divano, cucinare (tagliare), cucinare (mescolare), aprire un contenitore di pillole e lavorare al computer. Ogni soggetto compie la stessa azione due volte, quindi per ogni azione abbiamo 8 diverse sequenze (due per ogni soggetto). In aggiunta il dataset oltre alle sequenze

[7] http://pr.cs.cornell.edu/humanactivities/data.php
[8] https://www.cs.cornell.edu/

video fornisce la struttura dello scheletro comprendente la posizione e le orientazioni di 15 articolazioni estratti frame per frame.

Il test, che fa riferimento al protocollo "new person" proposto dai creatori del dataset, è stato condotto utilizzando il seguente approccio:

- Si opera con la tecnica leave-one-out usando a rotazione, come testing, i dati relativi a uno dei quattro soggetti del dataset.
- Il dataset viene duplicato specchiando[9] lungo l'asse X le copie. Questa operazione ci permette al dataset di essere invariante rispetto ad azioni compiute da soggetti mancini e destrorsi. Dal momento che il dataset contiene solo un mancino e tre destrorsi, questa operazione ci permette avere un dataset simmetrico.
- Seguendo le direttive dei creatori del dataset [38]i test vengono condotti classificando le azioni per stanza. Nello specifico non tutte le azioni possono essere svolte in tutti gli ambienti di casa, per questo motivo la classificazione viene separata.

In [Tabella 4] sono riportate le performance in termini di precision e recall sul dataset CAD-60 di tutti gli studi precedenti. Per maggiori informazioni e riferimenti sugli algoritmi riportati in tabella è possibile consultare direttamente il sito[10] ufficiale del Computer Science Department, Cornell University.

[9] l'operazione di specchiatura è stata realizzata sostituendo alla componente delle ascisse di ogni joint dello scheletro il suo inverso, ovvero, $x_j = -x_j \quad \forall\, Joint\ j \in S_k \quad k = 0, \dots, n$ dove S_k è il k-esimo scheletro di una sequenza video lunga n.

Algoritmo	Precision (%)	Recall (%)
Sung et al., AAAI PAIR 2011, ICRA 2012	67.9	55.5
Koppula, Gupta, Saxena, IJRR 2012	80.8	71.4
Zhang, Tian, NWPJ 2012	86	84
Ni, Moulin, Yan, ECCV 2012	Accur: 65.32	-
Yang, Tian, JVCIR 2013	71.9	66.6
Piyathilaka, Kodagoda, ICIEA 2013	70*	78*
Ni et al., Cybernetics 2013	75.9	69.5
Gupta, Chia, Rajan, MM 2013	78.1	75.4
Wang et al., PAMI 2013	Accur: 74.70	-
Zhu, Chen, Guo, IVC 2014	93.2	84.6
Faria, Premebida, Nunes, RO-MAN 2014	91.1	91.9
Shan, Akella, ARSO 2014	93.8	94.5
Gaglio, Lo Re, Morana, HMS 2014	77.3	76.7
Parisi, Weber, Wermter, Front. Neurobot. 2015	91.9	90.2
Cippitelli, CIN 2016	93.9	93.5
Risultati ottenuti in questo studio	**100**	**100**

Tabella 4 - Risultati ottenuti sul dataset CAD-60 riportati nel sito[10] ufficiale del Computer Science Department, Cornell University.

I risultati qui riportati [Tabella 15] mostrano i risultati dei test eseguiti applicando il protocollo sopra citato. La tabella è strutturata come segue:

- La tabella è divisa in 20 "quadranti" verticalmente sono riportati i soggetti che sono stati utilizzati come test; a rotazione viene lasciato escluso un soggetto dalla fase di training per poi essere utilizzato come test. Orizzontalmente invece sono riportate le stanze, alcune azioni possono essere svolte in più luoghi (ad esempio parlare al telefono può essere fatto sia in ufficio che in camera da letto o in salotto) per questo vengono ripetute.
- Ogni quadrante contiene una matrice di confusione relativa alla classificazione delle azioni svolte nella stanza *x* dal soggetto *y*. Nelle righe della matrice di confusione sono riportate le classi attese (le classi reali) mentre nelle colonne sono riportate le classi predette.
- È stata aggiunta un'ulteriore colonna (denominata Overall) dove vengono riportati in percentuale i valori medi di classificazione osservando i test condotti su tutti e quattro i soggetti. In particolare: denominiamo $Test_{x,y}$ la matrice dei test del soggetto *y* nella stanza *x*, allora i valori della matrice Overall vengono calcolati come:

[10] http://pr.cs.cornell.edu/humanactivities/results.php

$$ovaerall_{x,y}(i,j) = \frac{\sum_{y=1}^{4} Test_{x,y}(i,j)}{\sum_{y=1}^{4}\sum_{k=1}^{t} Test_{x,y}(i,k)}$$

$$dove\ t = numero\ azioni\ ammesse\ in\ x\ e\ i,j \in \mathbb{N}, 0 < i,j \leq t$$

In particolare i risultati riportati nella seguente tabella sono stati ottenuti con una finestra scorrevole ampia 10 e eseguendo l'operazione di clustering con un valore di K pari a 7. Come si evince facilmente, le prestazioni dell'algoritmo sono ottime, introducendo il meccanismo di voto maggioritario non si commettono errori, tutte le sequenze video vengono classificate correttamente. Nella Tabella 5 sono riportati i valori di precision, recall accuracy e f-measure riassumendo i valori delle matrici di confusione utilizzando gli indicatori di prestazione suddetti.

	brushing teeth	*rinsing mouth with water*	*wearing contact*	*cooking chopping*	*cooking stirring*	*water drinking*	*opening pill container*	*talking on the phone*	*talking on couch*	*relaxing on couch*	*writing on whiteboard*	*working on computer*
Precision	1	1	1	1	1	1	1	1	1	1	1	1
Recall	1	1	1	1	1	1	1	1	1	1	1	1
Accuracy	1	1	1	1	1	1	1	1	1	1	1	1
F-Measure	1	1	1	1	1	1	1	1	1	1	1	1

Tabella 5 – Tabella riassuntiva degli indici di performance relativi ai risultati dei test ottenuti con W = 10 e K = 7 riportati in Tabella 15.

Indicatore	**μ**
Precision	100%
Recall	100%
Accuracy	100%
F-Measure	100%

Tabella 6 – μ indica i valori espressi in percentuale di di precision, recall, accuracy, f-measure ottenuta come media di t

Subject 1				Subject 2				Subject 3 (Left handed)				Subject 4				Overall		
Bathroom				Bathroom				Bathroom				Bathroom				Bathroom		
brushing_teeth	1	0	0	brushing_teeth	1	0	0	brushing_teeth	1	0	0	brushing_teeth	1	0	0	brushing_teeth 100%	0%	0%
rinsing_mouth_with_water	0	1	0	rinsing_mouth_with_water	0	1	0	rinsing_mouth_with_water	0	1	0	rinsing_mouth_with_water	0	1	0	rinsing_mouth_with_water 0%	100%	0%
wearing_contact_lenses	0	0	2	wearing_contact_lenses	0	0	2	wearing_contact_lenses	0	0	2	wearing_contact_lenses	0	0	2	wearing_contact_lenses 0%	0%	100%
wearing_contact_lenses rinsing_mouth_with_water brushing_teeth								wearing_contact_lenses rinsing_mouth_with_water brushing_teeth				wearing_contact_lenses rinsing_mouth_with_water brushing_teeth						wearing_contact_lenses rinsing_mouth_with_water brushing_teeth
Kitchen				Kitchen				Kitchen				Kitchen				Kitchen		
cooking_chopping	1	0	0	cooking_chopping	1	0	0	cooking_chopping	1	0	0	cooking_chopping	1	0	0	cooking_chopping 100%	0%	0%
cooking_stirring	0	1	0	cooking_stirring	0	1	0	cooking_stirring	0	1	0	cooking_stirring	0	1	0	cooking_stirring 0%	100%	0%
drinking_water	0	0	1	drinking_water	0	0	1	drinking_water	0	0	1	drinking_water	0	0	1	drinking_water 0%	0%	100%
opening_pill_container	0	0	3	opening_pill_container	0	0	3	opening_pill_container	0	0	3	opening_pill_container	0	0	3	opening_pill_container 0%	0%	100%
opening_pill_container drinking_water cooking_stirring cooking_chopping				opening_pill_container drinking_water cooking_stirring cooking_chopping				opening_pill_container drinking_water cooking_stirring cooking_chopping				opening_pill_container drinking_water cooking_stirring cooking_chopping						opening_pill_container drinking_water cooking_stirring cooking_chopping
Bedroom				Bedroom				Bedroom				Bedroom				Bedroom		
talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone 100%	0%	0%
drinking_water	0	1	0	drinking_water	0	1	0	drinking_water	0	1	0	drinking_water	0	1	0	drinking_water 0%	100%	0%
opening_pill_container	0	0	3	opening_pill_container	0	0	3	opening_pill_container	0	0	3	opening_pill_container	0	0	3	opening_pill_container 0%	0%	100%
opening_pill_container drinking_water talking_on_the_phone				opening_pill_container drinking_water talking_on_the_phone				opening_pill_container drinking_water talking_on_the_phone				opening_pill_container drinking_water talking_on_the_phone						opening_pill_container drinking_water talking_on_the_phone
Livingroom				Livingroom				Livingroom				Livingroom				Livingroom		
talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone 100%	0%	0%
drinking_water	0	1	0	drinking_water	0	1	0	drinking_water	0	1	0	drinking_water	0	1	0	drinking_water 0%	100%	0%
talking_on_couch	0	0	1	talking_on_couch	0	0	1	talking_on_couch	0	0	1	talking_on_couch	0	0	1	talking_on_couch 0%	0%	100%
relaxing_on_couch	0	0	1	relaxing_on_couch	0	0	1	relaxing_on_couch	0	0	1	relaxing_on_couch	0	0	1	relaxing_on_couch 0%	0%	100%
talking_on_the_phone drinking_water talking_on_couch talking_on_the_phone				relaxing_on_couch talking_on_the_phone drinking_water talking_on_couch				relaxing_on_couch talking_on_the_phone drinking_water talking_on_couch				relaxing_on_couch talking_on_the_phone drinking_water talking_on_couch						relaxing_on_couch talking_on_the_phone drinking_water talking_on_couch
Office				Office				Office				Office				Office		
talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone	1	0	0	talking_on_the_phone 100%	0%	0%
writing_on_whiteboard	0	1	0	writing_on_whiteboard	0	1	0	writing_on_whiteboard	0	1	0	writing_on_whiteboard	0	1	0	writing_on_whiteboard 0%	100%	0%
drinking_water	0	0	1	drinking_water	0	0	1	drinking_water	0	0	1	drinking_water	0	0	1	drinking_water 0%	0%	100%
working_on_computer	0	0	1	working_on_computer	0	0	1	working_on_computer	0	0	1	working_on_computer	0	0	1	working_on_computer 0%	0%	100%
working_on_computer drinking_water writing_on_whiteboard talking_on_the_phone				working_on_computer drinking_water writing_on_whiteboard talking_on_the_phone				working_on_computer drinking_water writing_on_whiteboard talking_on_the_phone				working_on_computer drinking_water writing_on_whiteboard talking_on_the_phone						working_on_computer drinking_water writing_on_whiteboard talking_on_the_phone

Tabella 15 - Tisultati ottenuti con finestra scorrevole larga 10 e numero di cluster uguale a 7.

Come si evince sia da Tabella 5 e Tabella 15 le prestazioni dell'algoritmo su questo dataset sono davvero sorprendenti, introducendo il meccanismo di voto maggioritario si azzerano gli errori, tutte le sequenze video sono classificate correttamente. A tal proposito sono stati condotti ulteriori test per approfondire e verificare la bontà del classificatore: l'algoritmo è stato nuovamente eseguito utilizzando lo stesso modus operandi facendo uso ad ogni iterazione dei valori di W (finestra scorrevole) e K (numero di cluster) diversi in modo da individuare quale fosse il punto critico nel quale il sistema cominci a introdurre un qualsiasi tipo di errore. In particolare si è deciso di provare a utilizzare valori di W sempre più piccoli e di K sempre più grandi in modo da accorciare la lunghezza dei descrittori e quindi ridurre la quantità di memoria allocata nel processo di calcolo delle features. Il punto di "rottura" è stato individuato per valori di W inferiori a 4. Con dimensione troppo piccola della finestra scorrevole, infatti, il processo di classificazione comincia a commettere errori. Anche in questo caso i valori di Precision e Recall calano solamente di qualche punto percentuale.

Per concludere l'analisi dei risultati su questo dataset è possibile affermare che l'algoritmo rispecchia le caratteristiche quello proposto da Manzi *et al.* [1]. È possibile inoltre presupporre che il meccanismo di voto maggioritario, nella lunga durata vada a irrobustire la qualità del classificatore.

6.2 Cad-120

Questo dataset[7] creato sempre dal Computer Science Department, Cornell University raccoglie 120 sequenze video registrate con Kinect. Ogni video mostra una azione complessa della vita quotidiana; sono state raccolte 12 diverse attività: *preparare i cereali per la colazione, prendere le medicine, impilare oggetti, prendere oggetti da una pila, preparare cibo al microonde, raccogliere oggetti, pulire oggetti, prendere del cibo, sistemare degli oggetti, mangiare un pasto*. Le azioni sono compiute da quattro diverse persone, due uomini e due donne, di cui un solo soggetto mancino. Ogni azione viene svolta tre volte dallo stesso soggetto. In totale sono stati

raccolti 61’585 RGB-D video frames. Anche in questo caso oltre alle immagini RGB e le mappe di profondità, il dataset comprende anche gli scheletri estratti frame per frame; in aggiunta, vengono anche fornite indicazioni per la segmentazione delle attività in modo da poter individuare delle sotto-azioni come: raggiungere, spostare, versare, mangiare, bere, aprire, posare, chiudere, lavare, ecc. Queste possono essere utilizzare per fare un’analisi multi-layer delle attività al fine di addestrare dei classificatori in grado di sfruttare le sotto-azioni per distinguere diverse attività ad alto livello [Sezione 3.1.2]. In fine, sempre all’interno del dataset, è possibile trovare le informazioni sugli spostamenti degli oggetti manipolati dalle persone, questo permette di utilizzare il dataset sia per il tracking degli oggetti sia per poter testare algoritmi di classificazione per le interazioni uomo-ambiente. Anche in questo caso i video sono stati registrati in un ambiente indoor domestico, la scena è una stanza ricca di oggetti statici ma non sono presenti altri oggetti o persone in movimento, non sono presenti oggetti che occludono anche solo parzialmente il soggetto che svolge l’azione.

In questo caso l’approccio per il testing è stato semplificato in quanto non sono state fornite delle direttive generali dai creatori del dataset [39] come nel caso precedente; per questo motivo l’algoritmo di testing si limita a:

- Operare con tecnica leave-one-out usando a rotazione, come testing, i dati relativi a uno dei quattro soggetti del dataset.
- Eseguire la clonazione di tutti i soggetti per specchiare le copie in modo di ottenere nuovamente una simmetria tra soggetti mancini e destrorsi.

Come nel paragrafo precedente [6.1 Cad-60] viene riportata qui di seguito la tabella [Tabella 7] con i risultati ottenuti in diversi studi su CAD-120, in questo caso non sono stati riportati tutti i dati della tabella presente nel sito ufficiale del Computer Science Department, Cornell University ma solamente quelli comparabili allo studio condotto in questa Tesi. Sono stati esclusi infatti i dati relativi alla classificazione delle sub-actions e al tracciamento degli oggetti nella scena.

Algoritmo	**Accuracy**	**Precision**	**Recall**
Koppula et al., IJRR 2013	84.7	85.3	84.2
Koppula, Saxena, ICML 2013	93.5	95.0	93.3
With-out ground-truth segmentation			
Koppula et al., RSS 2013	66.1	36.7	71.3
Koppula, Saxena, ICML 2013	*67.2*	*41.4*	*73.2*
Jiang, Saxena, RSS 2014	68.1	44.2	74.9
Risultati ottenuti in questo studio	**91**	**90**	**98**

Tabella 7 - Risultati ottenuti sul dataset CAD-120 riportati nel sito[11] ufficiale del Computer Science Department, Cornell University.

Come è stato fatto per il dataset precedente anche in questo caso i test sono stati ripetuti variando la dimensione della finestra scorrevole e il numero di cluster da considerare quando si individuano le pose chiave dell'attività. Qui di seguito sono riportate le tabelle che ritraggono i valori più significativi ottenuti durante la fase di testing. Seguiti da delle tabelle con gli indicatori di performance al fine di mettere a confronto l'algoritmo oggetto di studio con quelli presenti in Tabella 7, sia per verificare quale configurazione è la migliore per classificare queste azioni (ovvero quali sono i valori di *W* e *K* per i quali si ottengono le performance migliori).

Già da Tabella 8 e seguente si evince come le prestazioni dell'algoritmo calino, in questo caso, infatti, l'algoritmo commette degli errori, seppure pochi. In particolar modo è possibile notare come in ognuno dei quattro test eseguiti con i quattro soggetti differenti gli errori si focalizzino principalmente in un'attività specifica, diversa da soggetto a soggetto. Questo tipo di errori va sicuramente preso in considerazione per cercare di migliorare l'algoritmo ma, in realtà, data la peculiarità dell'errore stesso è possibile ipotizzare che l'algoritmo oggetto di studio trovi difficoltà a classificare quell'azione per quel determinato soggetto in quanto questa potrebbe essere eseguita dalla persona stessa in modo "anomalo" o comunque compromettente dal punto di vista dell'estrazione delle features. Questa considerazione porta alla necessità, al fine di validare la supposizione, di testare l'algoritmo su un dataset più ampio. Nello specifico sarebbe preferibile avere un dataset con un più alto numero di soggetti che

[11] http://pr.cs.cornell.edu/humanactivities/results.php

svolgono le azioni in modo da poter catturare un numero maggiore di sfumature differenti con le quali i soggetti eseguono le singole attività. Avendo uno spettro più ampio di soggetti, e quindi di modi diversi di eseguire un azione, per addestrare il classificatore si ipotizza di migliorare le performance ulteriormente; a questo scopo nel paragrafo successivo [6.3 UniBo Human Activity Dataset (UHAD)] sono riportati i risultati dei test dell'algoritmo su un nuovo dataset acquisito appositamente durante questo studio comprendente quattordici diverse azioni eseguite rispettivamente da dieci diversi soggetti.

Tabella 8 - Confusion matrix relative al testing dell'algoritmo sul dataset CAD-120 con parametri W = 15 e K =15. Il soggetto indicato nelle quattro tabelle è quello usato come test mentre i rimanenti tre sono stati usati per il training del classificatore. Nello specifico quelli riportati sono stati i migliori risultati ottenuti con l'algoritmo su questo dataset.

Overall										
arranging objects	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%
cleaning objects	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
having meal	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
making cereal	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
microwaving food	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%
picking objects	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
staching objects	0%	0%	0%	17%	0%	0%	58%	0%	0%	25%
taking food	0%	25%	0%	0%	0%	0%	0%	75%	0%	0%
taking medicine	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
unstacking objects	0%	0%	0%	17%	0%	0%	17%	0%	0%	67%
	arranging objects	cleaning objects	having meal	making cereal	microwaving food	picking objects	staching objects	taking food	taking medicine	unstacking objects

Tabella 9 - In questa matrice di confusione sono riportati i valori medi espressi in percentuale di tutti e quattro i test eseguiti ciascuna volta con un soggetto diverso (vedi tabella precedente).

In Tabella 10 sono riportati i valori di Precision, Recall, Accuracy e F-Measure relativi ai valori della Tabella 9, grazie a questi indicatori è possibile quantificare in modo migliore le effettive prestazioni dell'algoritmo. Tutti gli indicatori sopra citati riportano valori superiori al 90%, dato che sta a confermare la bontà dell'algoritmo che riesce a raggiungere ottime prestazioni anche su questo dataset.

	arranging objects	cleaning objects	having meal	making cereal	microwaving food	picking objects	staching objects	taking food	taking medicine	unstacking objects	**AVG**
Precision	1	0,8	1	0,8	1	1	0,78	1	1	0,73	**0,91**
Recall	1	1	1	1	1	1	0,58	0,75	1	0,67	**0,9**
Accuracy	1	0,98	1	0,97	1	1	0,94	0,98	1	0,94	**0,98**
F-Measure	1	0,89	1	0,89	1	1	0,67	0,86	1	0,7	**0,9**
TP	12	12	12	16	12	12	7	9	12	8	**11,2**
TN	112	109	112	104	112	112	110	112	112	109	**110**
FP	0	3	0	4	0	0	2	0	0	3	**1,2**
FN	0	0	0	0	0	0	5	3	0	4	**1,2**

Tabella 10 - Tabella riassuntiva di tutti gli indicatori di performance per i risultati ottenuti su CAD-120 con dimensione di finestra W = 15 e numero di cluster k= 15. Nelle colonne sono riportati i valori per le singole attività, nell'ultima colonna il valore medio delle precedenti.

Come anticipato precedentemente nella seguente tabella [Tabella 11] verranno riassunti i valori più significativi ottenuti variando i parametri *K* e *W*. Analizzando i dati si può affermare che, data la natura delle sequenze video molto lunghe e ritraenti azioni complesse e articolate, in generale l'algoritmo si comporta correttamente quando la dimensione della finestra scorrevole *W* e il numero di cluster *K* è sufficientemente elevato.

Dall'analisi dei dati si evince inoltre un'altra caratteristica dell'algoritmo: non è mai consigliabile utilizzare un'ampia dimensione della finestra scorrevole quando si utilizza un numero basso di cluster per la compressione della sequenza. Il motivo per il quale le prestazioni dell'algoritmo calano in queste condizioni è facilmente intuibile: utilizzando un numero di cluster basso implica un altrettanto basso numero di pose chiave con la conseguenza che anche pose intermedie che per la loro peculiarità andrebbero considerate indipendenti vengano accorpate nello stesso cluster. Questo fa sì che nella sequenza compressa si alternino continuamente gli stessi pochi cluster; usando poi una finestra scorrevole molto ampia si peggiora ulteriormente la situazione accorpando in un unico grande vettore di feature un insieme di pose chiave simili perdendo quindi tutte le peculiarità delle diverse azioni.

Infine è importante sottolineare che non sono stati riportati i risultati di test eseguiti con valori superiori a 15 di *K* e *W* dal momento che le prestazioni tornavano nuovamente a calare.

Fatte queste considerazioni è importante sottolineare quindi come per azioni lunghe e complesse sia importante fare uso di feature vector sufficientemente lunghi da modellare l'attività, il che comporta l'uso di una finestra scorrevole abbastanza grande. Inoltre, per quanto lunghe e complesse siano le attività è importante ricordare di non eccedere con la dimensione di *W* in quanto, da quello che è stato possibile analizzare nei dati, si rischia inutilmente di peggiorare la qualità dei descrittori.

	Num Cluster	Precision	Recall	Accuracy	F-Measure
Window size = 4	K = 5	0,834124	0,810417	0,962903	0,812537
	K = 7	0,842798	0,791667	0,959677	0,7819
	K = 12	0,872729	0,833333	0,967742	0,836396
	K = 15	0,881656	0,85	0,970968	0,847842
Window size = 9	K = 5	0,782565	0,7	0,941935	0,708089
	K = 7	0,832689	0,775	0,956452	0,777439
	K = 12	0,891232	0,858333	0,972581	0,861859
	K = 15	0,876667	0,85	0,970968	0,852535
Window size = 12	K = 5	0,736773	0,572917	0,916129	0,568082
	K = 7	0,818035	0,74375	0,95	0,748949
	K = 12	0,868316	0,85	0,970968	0,84637
	K = 15	0,892857	0,883333	0,977419	0,881869
Window size = 15	K = 5	0	0,414583	0,883871	---
	K = 7	0,755685	0,545833	0,91129	0,520596
	K = 12	0,842172	0,808333	0,962903	0,80599
	K = 15	**0,910505**	**0,9**	**0,980645**	**0,899724**

Tabella 11 -In questa tabella sono riportati i valori degli indicatori di performance (Precision, Recall, Accuracy e F-Measure) ottenuti eseguendo i test dell'algoritmo oggetto di sudio sul dataset CAD-120, i valori riportati si riferiscono alla media degli score ottenuti eseguendo i test sui quattro soggetti del dataset.In grassetto sono stati evidenziati i parametri con i quali si è ottenuto lo score migliore.

In conclusione un'ultima analisi che si può fare riguarda il paragone dell'algoritmo oggetto di studio in questa Tesi e gli altri algoritmi testati su questo stesso dataset: anche in questo caso le prestazioni dell'algoritmo implementato sorpassano quelle registrate negli studi precedenti confermando ancora una volta l'efficacia nel classificare sia azioni più semplici, raccolte nel dataset CAD-60 (vedi 6.1), sia attività più articolate e complesse come quelle di CAD-120.

6.3 UniBo Human Activity Dataset (UHAD)

Contestualmente alla realizzazione dell'algoritmo oggetto di studio di questa Tesi è stato realizzato un software per l'acquisizione di un nuovo dataset. L'intento è stato quello di voler acquisire un nuovo insieme di azioni e attività eseguito da un più ampio numero di persone. Come affermato in precedenza [6.2 Cad-120], dai risultati emersi dai test condotti sui dataset utilizzati dalla comunità scientifica, sembra essere necessario aumentare in numero di soggetti che eseguono le attività in quanto ogni persona ha un proprio modo peculiare per eseguire qualunque tipo di azione dalla più semplice alla più complessa. Aumentando il numero di esempi somministrati al classificatore in fase di training si aumentano le probabilità che lo stesso classificatore sia in grado di cogliere le peculiarità che contraddistinguono le diverse attività; inoltre avendo a disposizione più soggetti è più probabile inserire all'interno del dataset più casi limite (dovuti appunto alla soggettività con la quale ogni persona svolge un'azione) per i quali altrimenti sarebbe complicato classificare correttamente una determinata azione.

Scendendo più nel particolare il nuovo dataset denominato Unibo Human Activity Dataset (UHAD) è stato acquisito tramite un Kinect versione 2 che è in grado di tracciare fino a sei persone contemporaneamente con la possibilità di individuare venticinque joint per ciascun soggetto tracciato. La nuova versione dell'SDK di Kinect inoltre permette di specificare se la posizione di un determinato joint è stata individuata perché quest'ultimo è visibile oppure se questa è stata inferita data la posizione delle altre

articolazioni visibili oppure se il tracking è stato impossibile. Attualmente questo algoritmo non fa uso delle suddette informazioni ma queste potrebbero tornare utili per uno sviluppo futuro per questo si è deciso di salvarle nel dataset.

L'acquisizione delle attività è stata fatta in un ambiente indoor ben illuminato, la scena all'interno della quale le persone hanno eseguito le azioni è ricca di oggetti statici e talvolta qualche oggetto in movimento dovuto alla presenza di altre persone all'interno della stanza (l'unico soggetto tracciato rimane comunque il soggetto che sta eseguendo l'azione, le altre persone sono solamente visibili nell'immagine RGB del frame).

La struttura dei file testuali che rappresentano lo scheletro è stata volutamente mantenuta simile a quella degli altri dataset acquisiti dal Computer Science Department, Cornell University. Qui di seguito viene brevemente descritto come si è scelto di strutturare il dataset.

Composizione di UHAD

- In totale sono state acquisite le seguenti 14 diverse azioni: *bere, alzarsi da una sedia, sedersi su una sedia, prendere un oggetto da terra, versare una bevanda in un bicchiere, sfogliare le pagine di un libro, impilare degli oggetti, prendere oggetti da uno scaffale e appoggiarli su un tavolo, parlare al cellulare, gettare qualcosa in un cestino, salutare, indossare un cappotto, lavorare al computer, scrivere su un foglio di carta.*
- 10 soggetti, 5 di sesso maschile, 5 di sesso femminile.
- Ogni soggetto svolge tutte le azioni due volte.
- In totale sono state raccolte 280 sequenze video.

Struttura del dataset

- Sono presenti 10 directory una per ogni soggetto, il nome della cartella è strutturato come segue Subject_#_r/l dove r sta per destrorso e l per mancino.
- Le 10 directory contengono due sub-directory, una per ogni esecuzione delle azioni, nome della sub-directory Try_#.

- Ogni subdirectory contiene 14 file, dove 14 è il numero delle attività svolte dai soggetti, nome dei file activity_label.txt, activity_label è auto-esplicativo (si capisce facilmente a quale azione si riferisce). Nella stessa cartella sono presenti anche altre 14 cartelle che portano lo stesso nome dell'attività alla quale fanno riferimento. Queste cartelle contengono i fotogrammi delle singole azioni, i fotogrammi delle immagini di profondità sono chiamati Depth_#, mentre i fotogrammi della telecamera a colori RGB_# dove # è il numero del fotogramma.
- È presente un file che riassume tutti i nomi delle attività, nome del file label.txt situato nella directory più esterna.

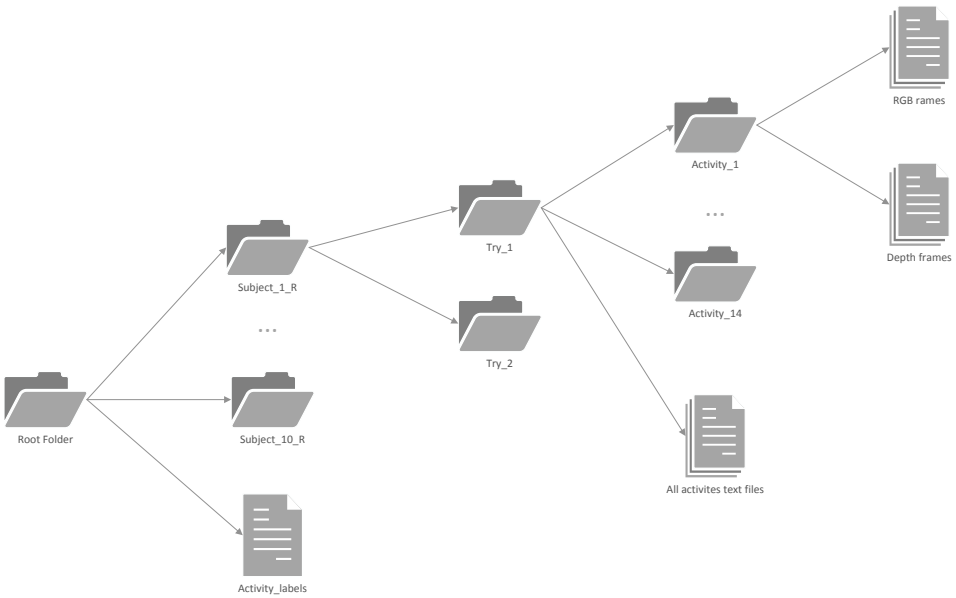

Figura 16 – Organizzazione gerarchica della struttura del dataset.

Passando quindi all'analisi dei risultati verranno proposte qui di seguito le matrici di confusione relative ai risultati migliori ottenuti. In particolare verranno proposte le dieci differenti matrici relative al test con i dieci diversi soggetti seguite da una tabella riassuntiva che elenca i valori medi.

Il protocollo di testing rispecchia quello utilizzato su CAD-120 ovvero:

- Operare con tecnica leave-one-out usando a rotazione, come test, i dati relativi a uno dei dieci soggetti del dataset.
- Eseguire la clonazione di tutti i soggetti per specchiare le copie in modo di ottenere nuovamente una simmetria tra soggetti mancini e destrorsi.

- Essendo il dataset costituito da scheletri a 25 articolazioni anziché 15 si è deciso di utilizzare per i test le articolazioni equivalenti a quelle prese in esame nei dataset precedenti in modo da poterne paragonare i risultati.

Già dai dati riportati in Tabella 12 – 12 – 13 si può apprezzare la buona qualità della classificazione per tutti e dieci i soggetti del dataset. L'unico errore evidente viene commesso per il soggetto numero 7 dove l'azione di gettare un oggetto nel cestino viene classificata sempre in modo sbagliato. Sarà oggetto di studi futuri verificare se tale errore è dovuto all'algoritmo o a qualche sorta di anomalia dei dati nel dataset medesimo. Lo stesso identico errore si presenta anche in altri casi in cui l'algoritmo è stato eseguito con valori di K e W diversi, questo lascia supporre che l'attività registrata presenti qualche tipo di eccezione. Le suddette tabelle rappresentano i risultati dei test eseguiti con dimensione della finestra W pari a 5 e numero di pose chiave (cluster) K pari a 12 e rappresentano i risultati migliori ottenuti sul nuovo dataset. Risultati simili si ottengono con altre configurazioni, viene usato circa lo stesso numero di pose chiave ma una dimensione maggiore della finestra scorrevole. In Tabella 15, come nei risultati presentati per gli altri dataset, vengono riportati i valori medi di tutti e dieci i test sui diversi soggetti. Osservando questa tabella che riporta una valutazione complessiva dell'algoritmo si può già affermare che anche in questo caso il sistema risulta essere efficace, infatti, quasi la totalità delle azioni viene classificata correttamente, nel peggiore dei casi un'azione viene classificata correttamente l'85% delle volte, un risultato comunque soddisfacente.

Tabella 12 – Usando i soggetti numero 0-1-2-3; risultati ottenuti su UHAD con dimensione della finestra W pari a 5 e numero di pose chiave (cluster) K pari a 12.

Tabella 13v - Usando i soggetti numero 4-5-6-7; risultati ottenuti su UHAD con dimensione della finestra W pari a 5 e numero di pose chiave (cluster) K pari a 12.

Tabella 14 – Usando i soggetti numero 8-9; risultati ottenuti su UHAD con dimensione della finestra W pari a 5 e numero di pose chiave (cluster) K pari a 12.

	Overall													
drink	85%	0%	0%	0%	5%	0%	0%	0%	15%	0%	0%	0%	0%	0%
get up	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
grab object from ground	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
pour a drink	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
scroll book pages	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%
sit	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%	0%
stack items	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
take objects from shelf	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
talking on phone	5%	0%	0%	0%	0%	0%	0%	0%	95%	0%	0%	0%	0%	0%
throw somthing in bin	0%	0%	0%	0%	5%	5%	0%	0%	0%	85%	0%	5%	0%	0%
waving	0%	0%	0%	5%	0%	0%	0%	0%	0%	0%	95%	0%	0%	0%
wear coat	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%
work on computer	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
write on paper	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%
	drink	get up	grab object from ground	pour a drink	scroll book pages	sit	stack items	take objects from shelf	talking on phone	throw somthing in bin	waving	wear coat	work on computer	write on paper

Tabella 15 – Risultati ottenuti come media dei valori dei test su tutti e dieci i soggetti.

La conferma delle ottime prestazioni dell'algoritmo si ha osservando la Tabella 16 dove vengono riassunti gli indicatori di prestazioni per diversi test con parametri differenti. Approfondendo l'analisi possiamo affermare che si ottengono risultati migliori con dimensioni della sliding window più piccole rispetto a quelle utilizzate in precedenza. Essendo le sequenze video di questo dataset più corte rispetto a quelle degli altri dataset è possibile ipotizzare che delle finestre scorrevoli più piccole riescano a catturare meglio le caratteristiche delle azioni riprese. Per quanto riguarda il numero di cluster, ovvero il di pose chiave, anche questo dataset ci conferma che è conveniente mantenere il valore di K superiore a 10. Del resto aumentare esageratamente il numero di cluster non risulta essere buona pratica in quanto si rischia di incorrere in overfitting del dataset.

	W=4, K=7	**W=5, K = 6**	**W=5,K=12**	**W=8,K=12**	**W=12,K=12**
Precision	0,9590	0,8180	0,9760	0,9756	0,97
Recall	0,9571	0,7469	0,9750	0,9750	0,96
Accuracy	0,9939	0,9636	0,9964	0,9964	0,99
F-Measure	0,9574	0,7365	0,9747	0,9749	0,96
TP	19,14	14,64	19,50	19,50	19,29
TN	259,14	250,36	259,50	259,50	259,29
FP	0,86	5,00	0,50	0,50	0,71
FN	0,86	5,00	0,50	0,50	0,71

Tabella 16 – Tabella riassuntiva degli indicatori di prestazione dei valori più significativi raccolti testando l'algoritmo su UHAD con valori di K e W differenti.

7 CONCLUSIONI

In conclusione lo studio condotto in questa Tesi ha approfondito la tematica dello Human Activity Recognition in particolare ha permesso di analizzare quello che è al giorno d'oggi lo stato dell'arte in questo settore sia per quanto riguarda le tecniche basate su sensori sia per quelle basate su telecamere; è stato preso in esame lo studio di Manzi *et al.* al fine di implementare e migliorare il loro algoritmo di classificazioni. Sono infine stati condotti test su due dataset raccolti dal Computer Science Department, Cornell University, e su di un dataset acquisito contestualmente a questo studio.

In seguito ai risultati sperimentali ottenuti è possibile affermare che, in primo luogo, l'algoritmo si comporta egregiamente su tutti i dataset: le prestazioni sono sempre ottime, gli indicatori come Precision, Recall, Accuracy e F-Measure superano sempre il 90% in tutti i casi. Rimane comunque aperta la questione di una generalizzazione dei parametri W e K, che, per ottenere le prestazioni migliori, come si è dimostrato nei vari test, variano a seconda dei casi. In generale come già accennato in precedenza non è mai buona regola usare un numero di pose chiave molto inferiore alla dimensione della finestra scorrevole in quanto la diretta conseguenza è un drastico calo delle prestazioni. Con un numero troppo piccolo di pose chiave si tende ad accorpare all'interno di pochi cluster posizioni molto diverse tra loro ignorando le pose che, data la loro importanza semantica dovrebbero costituire dei cluster indipendenti. Con poche pose chiave le diverse attività diventano molto più omogenee tra di loro in quanto all'interno dello stesso cluster potrebbero trovarsi anche pose molto differenti tra loro. Calcolando quindi il centroide del cluster si ottengono valori intermedi che non sono più rappresentativi né di una posa né di altre, le diverse azioni tendono quindi ad uniformarsi diventando indistinguibili.

Dalle valutazioni fatte sull'analisi dei risultati emerge che è generalmente corretto utilizzare una dimensione della finestra scorrevole di ampiezza circa 10, le prestazioni migliori si trovano in un intorno di ±2, una finestra

troppo piccola diventa troppo peculiare e minuziosa mentre una finestra troppo ampia diventa troppo generica.

Ricordiamo inoltre che data la natura delle azioni si possono avere delle sequenze di centroidi compresse molto differenti a seconda dei casi, le attività ripetitive avranno sequenze compresse molto lunghe dove si alternano circa sempre nello stesso ordine gli stessi centroidi; mentre, azioni meno ripetitive, produrranno sequenze più corte dove sarà difficile trovare una sotto sequenza che si ripete. Tenendo presente queste caratteristiche è possibile pensare di fare tuning dell'algoritmo in modo da cogliere al meglio le peculiarità delle azioni che si intende classificare.

Gli studi hanno mostrato le ottime prestazioni dell'algoritmo, di conseguenza è facile pensare a diversi miglioramenti e sviluppi futuri:

- Al fine di migliorare la sensibilità dell'algoritmo di estrazione dei descrittori si potrebbe introdurre un meccanismo di pesatura basato sul *Tracking State* dei joint al fine di considerare più o meno affidabili le posizioni degli arti e quindi evitare di dare importanza a pose "irreali" dovute alle approssimazioni eseguite durante l'estrazione dello scheletro.
- Dal momento che, a seconda dei casi, si ottenevano risultati migliori con valori differenti di K e W si potrebbe pensare di estrarre un set di descrittori, ognuno dei quali ottenuto usando numero di pose chiave e dimensione della finestra scorrevole diversi, quindi utilizzare i diversi insiemi di feature per addestrare un multi-classificatore. Si può ipotizzare che questo meccanismo irrobustisca il metodo con il quale vengono predette le categorie delle azioni in quanto se solo una, o poche, componenti del multi-classificatore sbagliano il risultato sarà comunque corretto perché la maggioranza restituisce la classe giusta. Con questa tecnica si supera quindi la necessità di usare degli specifici valori di K e W in quanto le feature verrebbero estratte con un generico e predefinito set di valori di configurazione predefinito.
- Un'ulteriore tipo di analisi che deve essere eseguita sull'algoritmo presentato è quella di verificare come questo metodo si comporta in un

sistema real-time. Sicuramente la complessità computazionale non presenta un problema ma bisogna valutare quanto questo algoritmo risulti efficace in un'applicazione in tempo reale, ovvero, dopo quanti frame il classificatore è in grado di fare una stima corretta dell'azione? Per rispondere a questa domanda è necessario cambiare il protocollo di testing e simulare il comportamento del classificatore man mano che legge i frame dalle singole sequenze del dataset.

- Un'ulteriore possibile estensione consiste nel modificare l'algoritmo in modo da riconoscere non solo azioni svolte da più soggetti contemporaneamente ma anche attività che comportano l'interazione di uno o più soggetti, si tenga presente che la nuova versione di Microsoft Kinect è in grado di tracciare simultaneamente sei diverse persone.

Concludendo possiamo affermare che grazie a questi nuovi mezzi sarà possibile addestrare un'intelligenza artificiale a riconoscere le attività umane con una buona precisione senza fare uso di sensori indossati e quindi lasciando libero spazio agli utenti nei movimenti. Questi sistemi potranno essere in grado di assistere le persone nella vita quotidiana, al lavoro e in situazioni di emergenza rendendo gli ambienti interattivi e più sicuri.

8 BIBLIOGRAFIA

[1] A. Manzi, F. Cavallo e P. Dario, «A 3D Human Posture Approach for Activity Recognition Based on Depth Camera,» *Computer Vision -- ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II,* vol. 9914, pp. 432-447, 2016.

[2] O. D. Lara e M. A. Labrador, «A survey on Human Activity Recognition using Wearable Sensors,» *IEEE Communications Surveys & Tutorials,* vol. 15, n. 3, pp. 1192-1209, 2013.

[3] J. W. Lockhart, T. Pulikal e G. W. Weiss, «Application of Mobile Activity Recognition,» *Proceedings of the 2012 ACM Conference on Ubiquitous Computing,* pp. 1054--1058, 2012.

[4] Microsoft, «Kinect for Windows SDK,» Microsoft, [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=44561. [Consultato il giorno 2017].

[5] OpenNI consortium, «OpenNI,» [Online]. Available: http://openni.ru/index.html. [Consultato il giorno 2017].

[6] Kinect community, «OpenKinect,» [Online]. Available: http://openni.ru/index.html. [Consultato il giorno 2017].

[7] J. T. Sunny, S. M. George e J. J. Kizhakkethottam, «Application and Challenges of Human Activity Recognition using Sensors in a Samart Environment,» *International Journal,* vol. 2, n. 4, pp. 50-57, Settembre 2015.

[8] J. A. Goldstone, «The new population bomb, the four megatrends that will change the world,» *Foreign affairs,* pp. 31-43, 2010.

[9] R. Hermann, P. Zappi e T. S. Rosing, «Context aware management of mobile systems for sensing applications,» *Proc. 2n International Workshop of mobile sensing,* 2012.

[10] J. K. Aggarwal e M. S. Ryoo, «Human Activity Analysis: A Review,» *ACM Computer Survey,* vol. 43, n. 3, pp. 16:1-16:43, Aprile 2011.

[11] J. Yamato, J. Ohya e K. Ishii, «Recognizing human action in time-sequential images using hidden Markov model,» *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 379-385, 1992.

[12] J.-X. Pan e K.-T. Fang, «Maximum Likelihood Estimation,» in *Growth Curve Models and Statistical Diagnostics*, Springer New York, 2002, pp. 77-158.

[13] D. M. Greig, . B. T. Porteous e A. H. Seheult, «Exact Maximum A Posteriori Estimation for Binary Images,» *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 51, n. 2, pp. 271-279, 1989.

[14] N. O. Garg, E. Horvitz e A. Horvitz, «Layered representations for human activity recognition,» *Proceedings. Fourth IEEE International Conference on Multimodal Interfaces,* pp. 3-8, 2002.

[15] L. Xia, C.-C. Chen e J. K. Aggarwal, «Human Detection Using Depth Information by Kinect,» *Proc. Int. Workshop HAU3D,* pp. 15-22, Giugno 2011.

[16] J. Han, E. Pauwels, P. d. Zeeuw e P. d. Wit, «Employing a RGB-D Sensor for Real-Time Tracking of Humans across Multiple Re-Entries in a Smart Environment,» *IEEE Trans. Consumer Electron,* vol. 58, n. 2, pp. 255-263, Maggio 2012.

[17] K. Lai, X. R. L. Bo e D. Fox, «A large-scale hierarchical multi-view RGB-D object dataset,» *IEEE International Conference on Robotics and Automation,* pp. 1817-1824, 15 Agosto 2011.

[18] X. Ren, L. Bo e D. Fox, «RGB-(D) scene labeling: Features and algorithms,» *2012 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2759-2766, 2012.

[19] S. Tang, X. Wang, X. a. H. T. X. Lv, J. Keller, Z. He, M. Skubic e S. Lao, «Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor,» *Computer Vision – ACCV 2012,* vol. 7725, pp. 525-538, 2012.

[20] A. F. ,. A. B. A. K. M. F. R. M. T. S. Jamie Shotton, «Real-Time Human Pose Recognition in Parts from a Single Depth Image,» *CVPR,* June 2011.

[21] M. Finocchio, M. Budiu, J. Shotton e D. Murray, «Parallelizing the Training of the Kinect Body Parts Labeling Algorithm,» *BigLearn Workshop at NIPS,* Dicembre 2011.

[22] R. Girshick, J. Shotton, P. Kohli, A. Criminisi e A. Fitzgibbon, «Efficient Regression of General-activity Human Poses from Depth Images,» *Proceedings of the 2011 International Conference on Computer Vision,* pp. 415--422, 2011.

[23] J. T. Andrew Fitzgibbon, J. Shotton e T. Sharp, «The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation,» *Proc. CVPR,* 1 Giugno 2012.

[24] M. Ye, X. Wang, R. Yang, L. Ren e M. Pollefeys, «Accurate 3D Pose Estimation From a Single Depth Image,» *2011 International Conference on Computer Vision,* vol. IEEE, pp. 731-738, 2011.

[25] W. S. Tu, K. Deng, X. Bai, T. Leyvand e B. G. a. Z., «Exemplar-based human action pose correction and tagging,» *2012 IEEE Conference*

on Computer Vision and Pattern Recognition, pp. 1784-1791, Giugno 2012.

[26] T. X e Y. Y. L., «EigenJoints-based action recognition using Naïve-Bayes-Nearest-Neighbor,» *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 14-19, 2012.

[27] Y. T. Xiaodong Yang, «Effective 3D action recognition using EigenJoints,» *Journal of Visual Communication and Image Representation,* vol. 25, n. 1, pp. 2-11, Gennaio 2014.

[28] C. C. C. a. J. K. A. L. Xia, «View invariant human action recognition using histograms of 3D joints,» *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 20-27, 2012.

[29] L. a. S. L. Liu, «Learning Discriminative Representations from RGB-D Video Data,» *IJCAI '13,* pp. 1493--1500, 2013.

[30] A. a. L. S. a. K. J. T. a. K. T.-S. Jalal, «Human Activity Recognition via the Features of Labeled Depth Body Parts,» *Impact Analysis of Solutions for Chronic Disease Prevention and Management: 10th International Conference on Smart Homes and Health Telematics, ICOST 2012, Artiminio, Italy, June 12-15, 2012. Proceedings,* pp. 246-249.

[31] J. Sung, C. Ponce, B. Selman e A. Saxena, «Human Activity Detection from RGBD Images,» *Proceedings of the 16th AAAI Conference on Plan, Activity, and Intent Recognition,* pp. 47-55, 2011.

[32] J. Sung, C. Ponce, B. Selman e A. Saxena, «Unstructured human activity detection from RGBD images,» *2012 IEEE International Conference on Robotics and Automation,* pp. 842-849, 2012.

[33] M. Reyes, G. Domínguez e S. Escalera, «Featureweighting in dynamic timewarping for gesture recognition in depth data,» *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops),* pp. 1182-1188, 2011.

[34] J. Wang, Z. Li, Y. Wu e J. Yuan, «Mining actionlet ensemble for action recognition with depth cameras,» *2012 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1290-1297, 2012.

[35] S. Shotton e N. Jamie, «Action Points: A Representation for Low-latency Online Human Action Recognition,» July 2012.

[36] D. Pelleg e A. W. Moore, «X-means: Extending K-means with Efficient Estimation of the Number of Clusters,» *Proceedings of the Seventeenth International Conference on Machine Learning,* pp. 727-734, 2000.

[37] C.-C. Chang e C.-J. Lin, «LIBSVM : a library for support vector machines,» *ACM Transactions on Intelligent Systems and Technology,* pp. 2:27:1-27:27, 2011.

[38] S. Jaeyong, P. C, S. B e S. A, «Unstructured human activity detection from RGBD images,» *2012 IEEE International Conference on Robotics and Automation,* pp. 842-849, 2012.

[39] H. S. Koppula, R. Gupta e A. Saxena, «Learning Human Activities and Object Affordances from RGB-D Videos,» *Int. J. Rob. Res.,* vol. 8, n. 32, pp. 951-970, Luglio 2013.