SAPIENZA
UNIVERSITÀ DI ROMA

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica

Corso di Laurea Magistrale in Ingegneria delle Comunicazioni

KINECT PEOPLE DETECTION

Relatore:

Prof. Gianni Orlandi

Candidati:

Antonio Montesano
Pietro Saulini

Sommario

1. Introduzione

Oggigiorno le tecniche commerciali e di marketing delle aziende si basano ormai quasi esclusivamente sulla raccolta dei dati degli individui in modo da conoscerne le abitudini, i comportamenti, i gusti e gli interessi creando così pubblicità mirate. Questo con l'obiettivo di attirare sempre più utenti e cercare di creare e vendere prodotti adatti ad un numero di persone sempre maggiore. Ad esempio è diventato del tutto normale visualizzare, su siti social e non solo, banner pubblicitari riguardanti oggetti di nostro interesse che abbiamo appena visto su un altro sito web.

Si è pensato di applicare questa analisi anche al di fuori della rete, nel mondo reale, ad esempio analizzando il comportamento di individui di fronte a prodotti commerciali di un qualche tipo, in ambienti come ad esempio negozi o luoghi commerciali in generale.

Questo lavoro si basa sull'idea di implementare la struttura di base di questo tema, vale a dire l'individuazione e la determinazione della posizione di persone in un determinato ambiente, sfruttando l'elaborazione in tempo reale dei dati ottenuti da sensori video.

2. Stato dell'arte

Ad oggi, lo sviluppo e il miglioramento delle tecniche per l'individuazione automatica in real time di soggetti umani, fermi o in movimento, risulta essere un argomento essenziale, non solo per quanto riguarda temi come la sicurezza e la sorveglianza, ma anche in ambienti come l'automotive, il gaming, settori commerciali e di marketing, ricerca medica e molti altri. Le tecnologie utilizzate per questo scopo sono di varia natura come ad esempio l'impiego di sensori acustici, sensori pressure-sensitive e sensori chimici, ma anche l'image recognition, l'utilizzo di sistemi radar ecc.

In particolare nell'image recognition, che poi è il settore nel quale rientra lo sviluppo di questo lavoro, si è arrivati a sviluppare tecniche di detection, recognition e tracking, che sfruttano l'intelligenza artificiale, basate su algoritmi di machine learning, deep learning, pattern recognition anche molto avanzati. Fondamentalmente, in quest'ambito, è previsto l'utilizzo di sensori video in un numero più o meno elevato (array di sensori), posti in un ambiente, confinato o meno, secondo precisi schemi. Ovviamente, il numero di questi sensori dipenderà da quelli che sono l'obiettivo finale e la potenza computazionale a disposizione.

I sistemi di ultima generazione sono in grado di riconoscere simultaneamente in real time, centinaia di caratteristiche riguardanti il sesso, l'età, il colore e il tipo di abbigliamento ecc.; fare distinzione tra categorie come ad esempio persone che camminano, che corrono, che saltano ecc.; di tracciare il tragitto percorso dagli individui e di prevedere, con buona approssimazione quello che percorreranno e il loro comportamento. Tutto questo è ovviamente possibile, in real time, solo sfruttando algoritmi che riducono notevolmente i tempi di calcolo.

Figura 1 - people tracking in ambiente aperto (Hitachi)

3. Obiettivo

Questa tesina si propone l'obiettivo di realizzare una GUI sviluppata utilizzando Matlab, che esegua un monitoraggio video in tempo reale di un ambiente, individuando la presenza e la posizione di persone nella zona di interesse. Lo sviluppo sarà basato sull'utilizzo dei due sensori video di un Kinect che verrà posto in maniera fissa in un determinato ambiente e interfacciato al software Matlab sviluppato. Quest'ultimo elaborerà i dati ricevuti dai sensori e restituirà informazioni sull'eventuale presenza di persone e sulla loro posizione rispetto a questi.

4. Strumentazione

Notebook Toshiba Satellite C855 con processore Intel core i5 3230M quadcore da 2.6GHz, 16GB di RAM, SSD, scheda grafica AMD Radeon HD 7500 e sistema operativo Microsoft Windows 10.

Notebook HP Pavilion g6 processore Intel core i5-2419M dualcore @2.30GHz, 8 GB RAM, SSD, scheda grafica AMD Radeon HD 7400 , sistema operativo Microsoft Windows 10.

1.1. Microsoft Kinect xbox 360

Il Kinect è un dispositivo dotato di telecamera RGB, doppio sensore di profondità a raggi infrarossi composto da uno scanner laser a infrarossi e da una telecamera sensibile alla stessa banda. La telecamera RGB ha una risoluzione di 640 × 480 pixel, mentre quella a infrarossi permette di utilizzare tre diverse risoluzioni 640 × 480 pixel, 320 × 240 pixel e 80 x 60 pixel. Il Kinect dispone anche di un array di microfoni.

Figura 2 - Microsoft Kinect Xbox 360

Il software interno distingue gli oggetti grazie ad un modello che combina le immagini a colori con le immagini del sensore di profondità e per ottenere queste ultime usa un modello simile al radar: la camera invia un segnale luminoso nella banda dell'infrarosso e sfrutta il tempo di ritorno dei rimbalzi per calcolare la distanza degli oggetti su cui è avvenuta la collisione. Usare tale banda consente anche di risolvere parzialmente il problema della luce ambientale: visto che il sensore non è progettato per ricevere luce visibile, sarà molto poco propenso a falsi positivi.

Le informazioni possono essere inoltre elaborate per generare immagini 3D, tali da consentire al Kinect di distinguere la profondità di oggetti fino ad 1cm.

1.2. Matlab 2015a

Matlab (**Mat**rix **Lab**oratory) è un ambiente di sviluppo, fortemente incentrato sul calcolo vettoriale, scritto in C che comprende anche l'omonimo linguaggio di programmazione. È molto versatile nell'interfacciarsi con altri programmi e creare interfacce utente. Nonostante sia specializzato nel calcolo numerico è anche possibile sfruttarlo per il calcolo simbolico grazie al motore di calcolo simbolico di Maple.

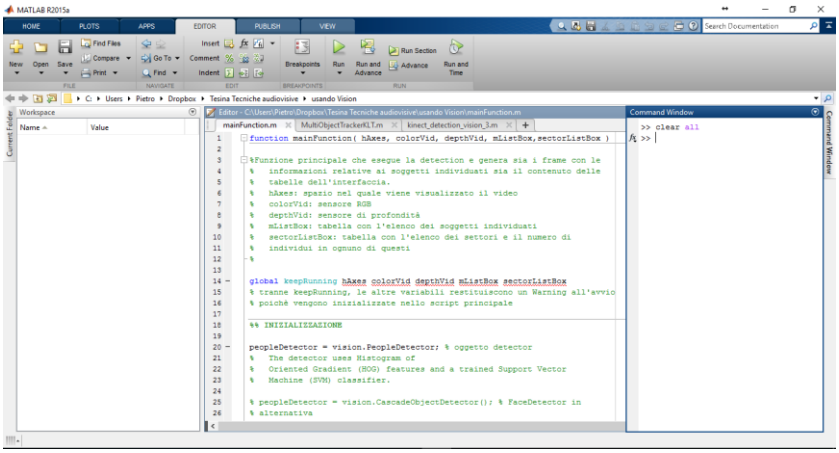

Figura 3 - Matlab R2015a

5. Connessione e interfacciamento Matlab-Kinect

Per il collegamento del dispositivo è necessario l'uso del kinect adaptor che fornisce al kinect la giusta alimentazione necessaria al suo funzionamento.

L'utilizzo del kinect xbox 360 richiede alcuni passaggi preliminari da eseguire sul pc per l'interfacciamento con il computer stesso e con matlab.

Il primo passo consiste nell'installazione, su matlab, del package "Kinect for Windows Sensor" v.15.1.0. Successivamente bisogna scaricare dal sito della Microsoft e installare:

Kinect for Windows Developer Toolkit v1.8.0
Kinect for Windows Drivers v1.8
Kinect for Windows Runtime v1.8
Kinect for Windows SDK v1.8

L'ultimo passo consiste nell'installazione dei software OpenNI e NITE.

Se la procedura è andata a buon fine, inserendo il seguente comando su matlab si otterrà:

```
>> hwInfo = imaqhwinfo('kinect')

hwInfo =

       AdaptorDllName: 'C:\MATLAB\SupportPackag...'
    AdaptorDllVersion: '4.9 (R2015a)'
          AdaptorName: 'kinect'
            DeviceIDs: {[1]  [2]}
           DeviceInfo: [1x2 struct]
```

5.1. Toolbox e librerie

5.1.1. Kinect® for Windows® from Image Acquisition Toolbox™

Questo toolbox aggiuntivo di Matlab consente di utilizzare le funzioni video del Kinect Xbox 360. In realtà nello sviluppo del nostro lavoro non abbiamo utilizzato questo toolbox ma vale la pena citarlo per l'eventuale sviluppo di altre applicazioni simili, come ad esempio lo skeletal tracking.

Questa scelta è dovuta al fatto che l'utilizzo del dispositivo kinect xbox 360 attraverso questo toolbox, permette la detection simultanea di sole due figure umane. Questa limitazione è stata di fatto in parte superata con successive versioni del kinect, ad esempio, il kinect v.2 è in grado di distinguere fino a 6 soggetti contemporaneamente.

5.1.2. Computer Vision System Toolbox

Questo toolbox contiene algoritmi , funzioni e applicazioni per la progettazione e simulazione per computer vision e sistemi di processing video. Permette di realizzare feature detection, extraction e matching; object detection e tracking; previsione del movimento e video processing. Proprio l'utilizzo delle sue funzioni ci permetterà di eseguire la detection.

5.2. Acquisizione dati

L'acquisizione dei dati dal kinect avviene sfruttando direttamente ciò che vedono i sensori. Nella funzione principale, ad ogni ciclo, vengono catturati degli snapshot, rappresentati come matrici 640x480 uint16 dello streaming video del sensore rgb e di quello ad infrarossi. I primi verranno passati al detector e utilizzati per comporre il video visualizzato nell'interfaccia, mentre i secondi per il calcolo della posizione.

Fondamentale è il fatto che la matrice che descrive l'immagine proveniente dal sensore di profondità ha per elementi le distanze in millimetri, cioè ogni elemento colora un pixel in base alla distanza a cui si trova quel preciso punto. Di fatto quello che otteniamo è un'immagine in scala di grigi con valori che vanno da 0 a 4000. Per valori inferiori a 800 e superiori a 4000 il colore nell'immagine è lo stesso (nero).

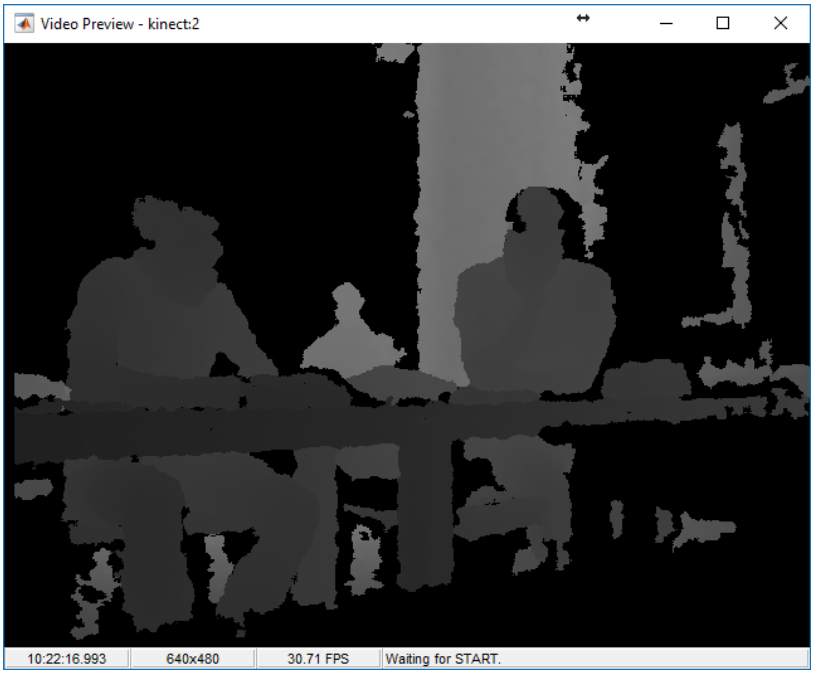

Figura 4 - Fotogramma del sensore ad infrarossi del kinect

L'immagine ottenuta da questo sensore risulta però molto rumorosa sui bordi. Nelle successive versioni del kinect questo aspetto è stato migliorato ottenendo un'immagine molto più dettagliata e stabile.

6. Detection

La detection è effettuata utilizzando su ogni frame, la funzione vision.PeopleDetector. il detector usa Histogram of Oriented Gradient (HOG) features e un classificatore Support Vector Machine già allenato. Il detector restituisce un array contenente i punti (x, y, w, h) delle boundary boxes che circondano il soggetto identificato e che verranno successivamente rappresentate sui frame con dei rettangoli di colore giallo.

6.1. Histogram of Oriented Gradient (HOG)

L'istogramma dei gradienti orientati è un descrittore usato in computer vision e image processing con lo scopo di rilevare oggetti.

La tecnica conta le occorrenze dell'orientazione del gradiente in porzioni di imagine localizzate.

Il concetto fondamentale su cui si basa il descrittore HOG è che l'aspetto e la forma di un oggetto in un'immagine possono essere descritti dai gradienti della distribuzione di intensità (dei pixel) o dalle direzioni del bordo. L'immagine viene suddivisa in regioni chiamate celle; i pixel relativi ad ogni cella vengono usati per compilare un istogramma delle direzioni del gradiente. Il vettore descrittore è la concatenazione di tali istogrammi. Per aumentare l'accuratezza, gli istogrammi locali vengono normalizzati, in modo da avere più contrasto, calcolando una misura dell'intensità su una regione più grande della singola cella, chiamata blocco. Questa regione viene usata per normalizzare tutte le celle interne.

Il descrittore HOG presenta qualche vantaggio rispetto agli altri descrittori. Dato che opera su celle locali, risulta invariante a trasformazioni geometriche e fotometriche, eccetto per l'orientazione degli oggetti. Tali cambiamenti apparirebbero solo in regioni dello spazio più grandi. Inoltre è stato dimostrato da Dalal e Triggs che il campionamento lasco dello spazio, il campionamento raffinato sull'orientazione e una forte normalizzazzione fotometrica permettono di individuare corpi umani anche se in movimento, finchè essi mantengono una posizione più o meno eretta. Per questo motivo il descrittore HOG è particolarmente indicato per il rilevamento di umani nelle immagini.

6.1.1. Implementazione dell'algoritmo

6.1.1.1. Calcolo del gradiente

Solitamente il primo passo in molti descrittori è un pre-processing che assicuri valori normalizzati per colore e gamma. Questo passo può essere omesso con l'HOG in quanto la normalizzazione effettuata dal descrittore ottiene essenzialmente gli stessi risultati, risparmiando quindi un leggero impatto sulla performance. Il primo passo quindi sarà il calcolo dei valori del gradiente. Il metodo più comune è quello di applicare la maschera delle derivate del primo ordine lungo la direzione orizzontale o verticale o entrambe.

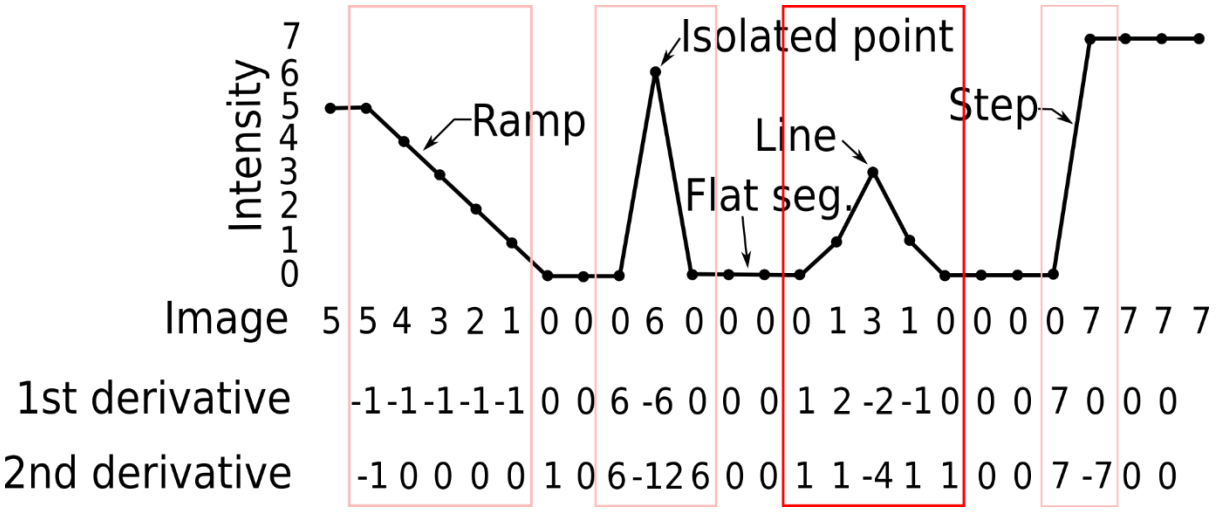

Figura 5 - Maschera delle derivate di una funzione (intensità) di un'immagine

6.1.1.2. Orientation binning

Il secondo passo consiste nel creare l'istogramma della cella. Ogni pixel all'interno della cella assegna un peso per un canale di istogramma basato sull'orientazione calcolato sui valori trovati nel calcolo del gradiente. Le celle possono avere forma rettangolare o radiale, i canali dell'istogramma possono spaziare tra 0° e 180° o tra 0° e 360°, a seconda che il gradiente sia un "unsigned" o "signed". Dalal and Triggs trovarono che gradienti unsigned usati in concomitanza con 9 canali di istogramma arrivavano al miglior risultato per i loro esperimenti sul rilevamento di umani. Riguardo i pesi, il contributo dei pixel può essere dato sia dall'ampiezza del gradiente stesso, sia da qualche funzione dell'ampiezza. Nei test, utilizzare l'ampiezza del gradiente porta ai risultati migliori.

6.1.1.3. Descrizione dei blocchi

Per provvedere a cambi di illuminazione e contrasto, i picchi del gradiente devono essere normalizzati localmente, ciò richiede di raggruppare le celle in blocchi spazialmente connessi più grandi. Il descrittore HOG è quindi il vettore concatenato delle componenti dell'istogramma della cella normalizzata da tutte le regioni del blocco. I blocchi sono tipicamente sovrapposti, ciò significa che ogni cella contribuisce più di una volta nel descrittore finale. Esistono due geometrie principali di blocco: a blocchi rettangolari R-HOG e a blocchi circolari C-HOG. Gli R-HOG generalmente sono griglie quadrate rappresentate da tre parametri: numero di celle per blocco, numero di pixel per cella e numero di canali per istogramma di cella.

Il passo finale nell'object recognition usando HOG è di inserire i descrittori in un qualche sistema di recognition basato sull'apprendimento supervisionato.

6.2. Support vector Machine (SVM)

In machine learning, l'SVM è un modello classificazione binario supervisionato che classifica i dati trovando il miglior iperpiano che separa tutti i punti di una classe da quelli dell'altra. Una volta allenato su immagini contenenti oggetti particolari, il classificatore SVM può prendere decisioni riguardanti la presenza di un oggetto o di persone.

Per miglior iperpiano, per un SVM, si intende quello che ha la maggiore distanza da entrambe le classi. Si viene a creare così una regione nella quale non ci sono punti appartenenti ai dati analizzati. I support vectors sono i punti che si trovano più vicini all'iperpiano di separazione e definiscono i confini della regione vuota tra le due zone.

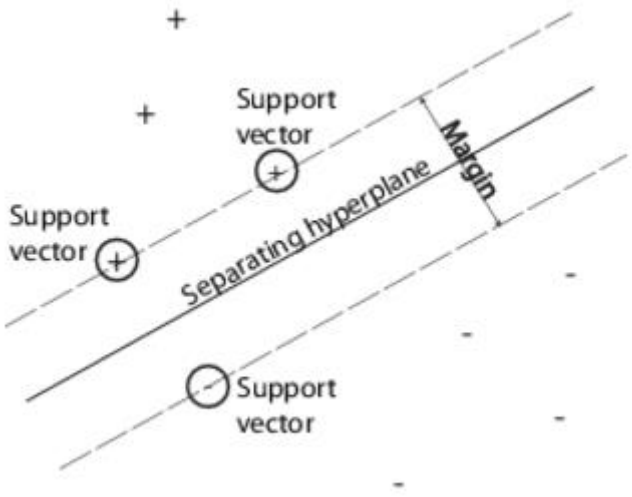

Figura 6 - Esempio grafico di Support Vector Machine

Per costruire il modello è necessario allenare l'algoritmo con un training set. Questo è un insieme di punti (un vettore) x_j, insieme alle loro categorie y_j.

Considerando $x_j \in R^d$ e $y_j = \pm 1$, l'equazione dell'iperpiano è

$$f(x) = x'\beta + b = 0$$

dove $\beta \in R^d$ e b è un numero reale.

Il seguente problema definisce il miglior iperpiano di separazione. Trovare β e *b* che minimizzano ||β|| per tutti i punti del set (x_j, y_j), $y_j f(x_j) \geq 1$.

I support vectors sono gli x_j sul limite, per i quali $y_j f(x_j) = 1$.
Per semplicità matematica, si considera spesso il problema equivalente di minimizzazione di ||β||, che non è altro che un problema di ottimizzazione quadratica.

La soluzione ottima $(\hat{\beta}, \hat{b})$ permette la classificazione di un vettore *z* nel modo seguente:

$$\text{class}(z) = \text{sign}\left(z'\hat{\beta} + \hat{b}\right) = \text{sign}\left(\hat{f}(z)\right)$$

$\hat{f}$ (z) è la *classification score* e rappresenta la distanza dal limite di decisione.

Risolvere il problema di ottimizzazione quadratica è computazionalmente più semplice. Per ottenere il duale prendiamo i moltiplicatori di Lagrange positivi α_j moltiplicati per il vincolo e sottratti dalla funzione obbiettivo:

$$L_P = \frac{1}{2}\beta'\beta - \sum_j \alpha_j\left(y_j\left(x_j'\beta + b\right) - 1\right)$$

Dove si cerca un punto stazionario di L_p su β e b. Ponendo il gradiente di L_p a zero otteniamo

$$\beta = \sum_j \alpha_j y_j x_j$$
$$0 = \sum_j \alpha_j y_j.$$

sostituendo in L_p, si trova il duale L_D:

$$L_D = \sum_j \alpha_j - \frac{1}{2}\sum_j \sum_k \alpha_j \alpha_k y_j y_k x_j' x_k,$$

Che massimizziamo per $\alpha_j \geq 0$.
Gli α_j diversi da zero nella soluzione del problema duale definiscono l'iperpiano. I punti x_j corrispondenti ai α_j diversi da zero sono i *support vectors*.

La derivata di L_D fatta rispetto agli α_j diversi da zero è pari a 0 all'ottimo

$$y_j f\left(x_j\right) - 1 = 0.$$

In particolare ci dà il valore di b alla soluzione, considerando gli α_j diversi da zero per ogni j.

Nella maggior parte dei casi reali i dati non sono separabili con un iperpiano. In questo caso, SVM può usare un soft margin, cioè un iperpiano che separa la maggior parte dei punti rappresentati i dati, ma non tutti.

6.3. Tracking

Per permettere al programma di individuare anche soggetti in movimento, si è reso necessario l'utilizzo di una funzione di tracking. Questa viene avviata successivamente all'individuazione dei soggetti (la detection viene effettuata ogni 10 frame)

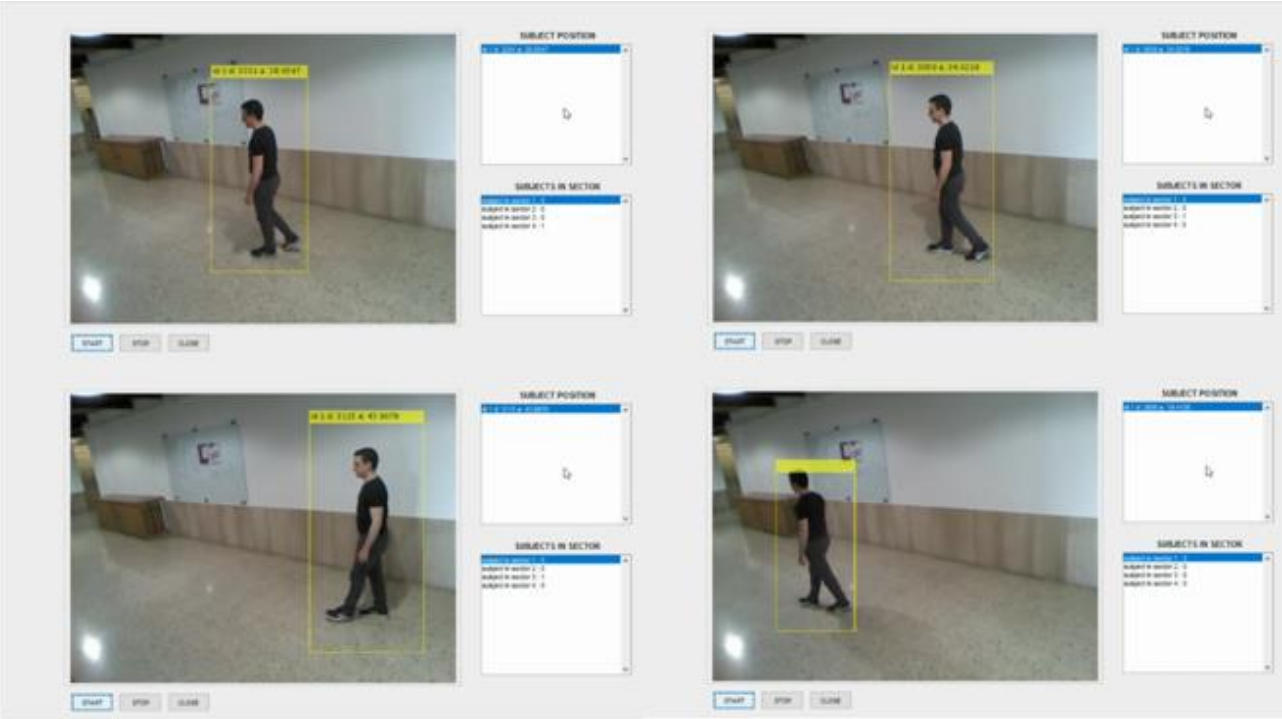

Figura 7 - prova con soggetto in movimento

Il tracking è realizzato utilizzando la funzione di Matlab MultiObjectTrackerKLT basata sull'algoritmo KLT (Kanade-Lucas-Tomasi).
I punti che vengono tracciati vengono calcolati usando detectMinEigenFeatures che trova i feature points attraverso il minimum eigenvalue algorithm (Shi-Tomasi).

Figura 8 – Esempio dei punti estratti dalla funzione detectMinEigenFeatures

In computer vision, il KLT feature tracker è una tecnica di estrazione delle caratteristiche. È stato proposto principalmente per affrontare il problema dell'alto costo computazionale delle tecniche tradizionali di image registration (processo di trasformazione di diversi set di dati in un sistema di coordinate). KLT utilizza le informazioni di intensità spaziale per dirigere la ricerca per la posizione

che produce la migliore corrispondenza e di fatto risulta essere molto più veloce delle tecniche tradizionali.

6.4. Calcolo della posizione

Per determinare la distanza a cui si trova un soggetto identificato, sfruttiamo i dati della matrice restituita dal sensore di profondità incrociandoli con l'array che descrive le boundary boxes.

Quello che andiamo a fare consiste nel determinare il punto centrale del box, descritto con gli indici di riga e colonna della matrice del frame rgb; nella stessa posizione della matrice del sensore di profondità avremo la distanza del punto.

Per semplicità di utilizzo abbiamo scelto di descrivere la posizione in coordinate polari. Considerando un angolo di apertura del campo visivo del sensore video del kinect di 57° (valore preso dalle specifiche del kinect xbox 360), per calcolare l'angolo a cui si trova il soggetto utilizziamo sempre la posizione nel frame del pixel centrale del box, nello specifico:

$$\beta = \frac{57° \cdot j}{640}$$

Con j indice di colonna del punto centrale del box.

Per quanto riguarda la distanza, il kinect ha un campo visivo che va dai 0.8m ai 4m.

7. Graphic User Interface

L'interfaccia realizzata consiste in una finestra non ridimensionabile, caratteristica utile per evitare problemi di proporzioni nel video.

Avviato il programma quello che compare è quanto mostrato nell'immagine

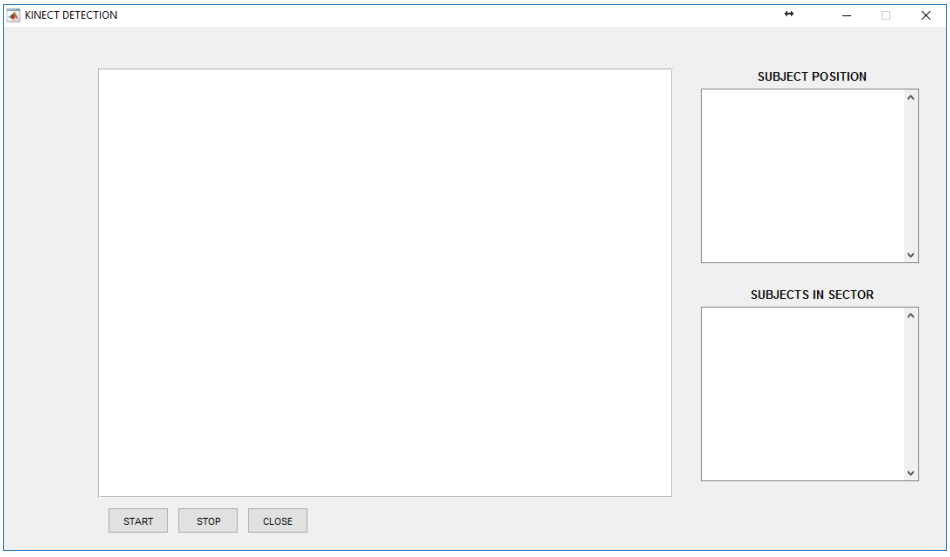

Figura 9 - Graphic User Interface dell'applicazione

Tramite il tasto START viene chiamata la funzione principale che avvia l'acquisizione dello streaming dei sensori e viene mostrato il video nel riquadro principale nel quale, i soggetti individuati vengono contrassegnati con dei riquadri colorati nei quali vengono riportate le informazioni sulla posizione.

Nel riquadro SUBJECT POSITION vengono riportate nuovamente le informazioni riguardanti la posizione, distanza (in mm) dal sensore e angolo (in gradi), dei soggetti rilevati, in modo da avere una più comoda visualizzazione dei dati utili.

Nell'ultimo riquadro, SUBJECT IN SECTOR, vengono contante le persone presenti nei vari settori in cui è stata suddivisa l'inquadratura.

Abbiamo preso in considerazione quest'ultimo aspetto tenendo conto dell'obbiettivo a cui mira questo lavoro, vale a dire l'applicazione in un ambiente commerciale. Abbiamo suddiviso la scena in quattro settori ipotizzando, in maniera semplificata, le posizioni di quattro espositori all'interno di un esercizio commerciale, per monitorare in tempo reale il comportamento dei clienti presenti al suo interno.

Oltre al tasto START abbiamo inserito anche lo STOP che permette di bloccare, oltre che il video sul frame attuale, anche le informazioni riportate nei riquadri laterali. Premendo nuovamente start riparte l'acquisizione dei dati.

Il tasto CLOSE chiude l'interfaccia e termina l'esecuzione del programma..

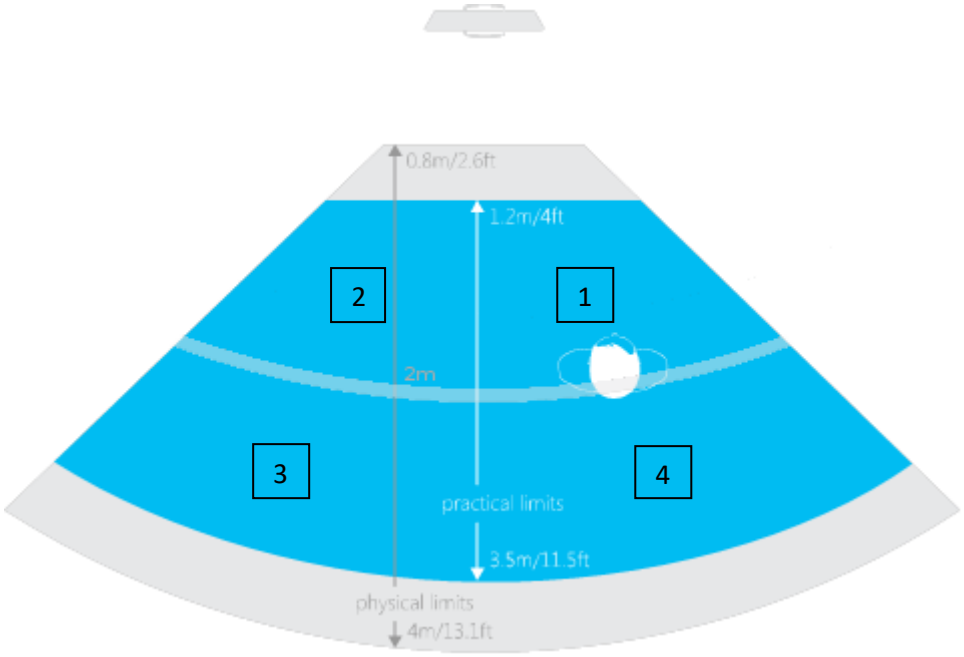

Figura 10 - rappresentazione campo visivo del kinect e sua suddivisione in 4 settori

8. Risultati

Le ripetute prove effettuate mostrano che l'applicativo sviluppato, riesce a rilevare fino a 5 soggetti fermi e in movimento (per motivi dovuti alle dimensioni del campo visivo non siamo riusciti ad andare oltre), definendo le rispettive posizioni con una buona accuratezza.

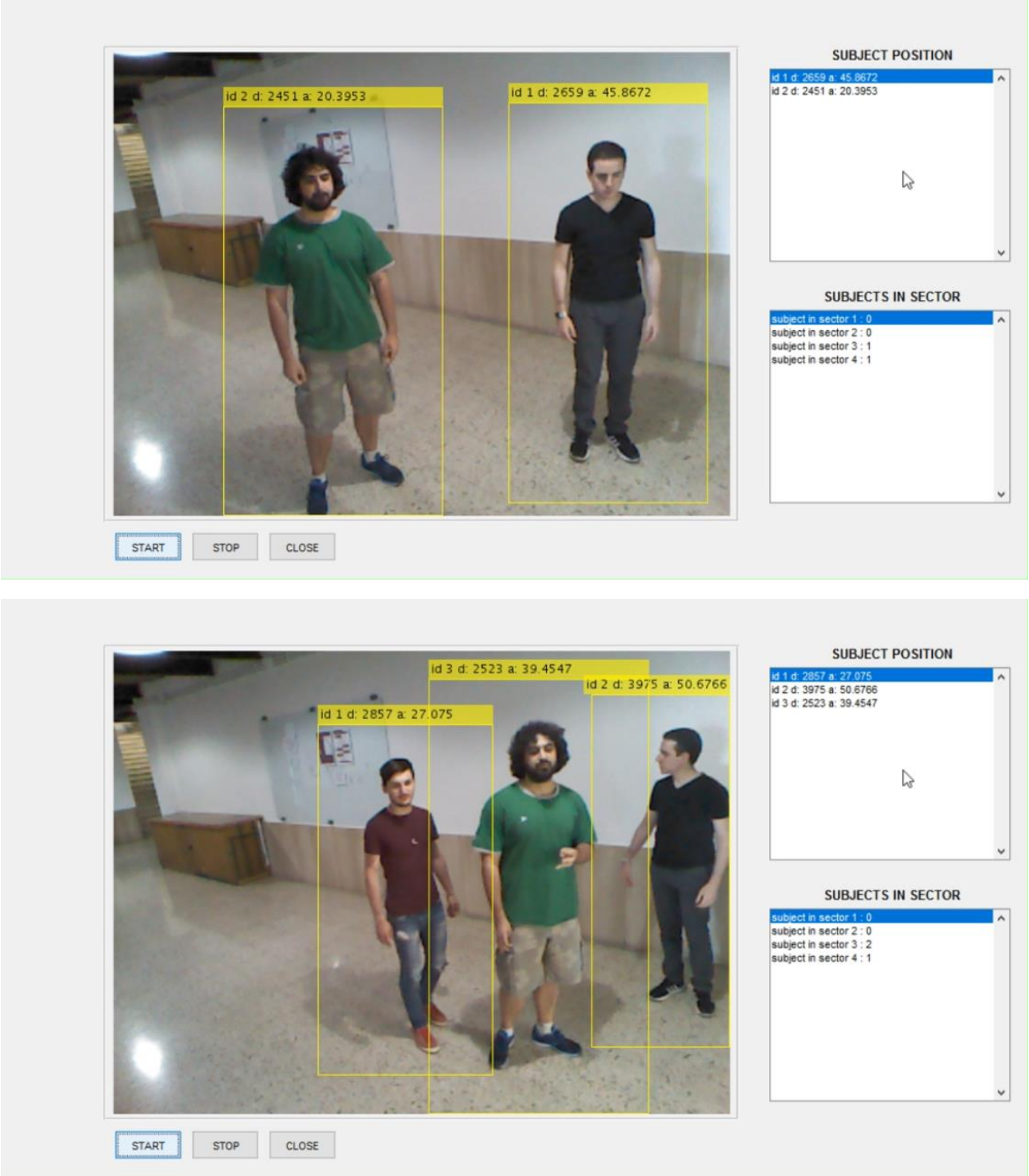

Figura 11-12 - Screenshot del programma in esecuzione con 2 e 3 soggetti

In ambienti esterni, o più in generale, in situazioni che presentano una forte luminosità naturale, il sensore RGB del kinect sembra non riuscire ad elaborare quanto ripreso in maniera corretta, restituendo delle immagini con una luminosità eccessiva che non permettono di distinguere le figure nella scena. Tuttavia probabilmente, questo è dovuto al fatto che il kinect xbox 360 è un dispositivo pensato per il gaming in ambienti chiusi.

Ovviamente la tecnica di detection utilizzata genera sia un certo numero di falsi positivi che di mancata rivelazione. Abbiamo notato questi tipi di errore specialmente in presenza di soggetti in movimento. Sicuramente con una capacità di elaborazione maggiore, è possibile elaborare più velocemente i singoli frame sia per ridurre le probabilità di errore sia per aumentare la fluidità del video.

9. Conclusioni

Utilizzando il toolbox vision, di fatto è possibile pensare di migliorare le prestazioni di questa applicazione sostituendo il kinect con un generico dispositivo dotato di fotocamera rgb e sensore ad infrarossi di qualità superiore. Aggiungendo poi un meccanismo di recognition dei soggetti nella scena diventerebbe possibile monitorare in maniera più accurata la zona di interesse potendo calcolare delle statistiche sui tempi di permanenza in una determinata posizione e sul comportamento dei singoli.

10. Bibliografia

Mathworks, “Computer Vision System Toolbox”, mathworks.com, 2017, Web, 7/04/2017.

Mathworks, “Support Vector Machines for Binary Classification”, mathworks.com, 2017, Web, 18/05/2017.

Navneet Dalal and Bill Triggs, “Histograms of Oriented Gradients for Human Detection”, INRIA Rhone-Alps.

Wikipedia, “Kanade–Lucas–Tomasi feature tracker”, en.wikipedia.org, 22 settembre 2014, Web, 19/05/2017.

Mathworks, “vision.PointTracker System object”, mathworks.com, 2016, Web, 19/05/2017.

Shi and Tomasi, “Good features to track”, IEEE Conference on Computer Vision and Pattern Recognition Seattle, June 1994.

John MacCormick, “How does the Kinect work?”, Web, 3/04/2017.

Support.xbox, support.xbox.com/it-IT/xbox-360/console/manual-specs, Web, 3/04/2017.

Hitachi, “Development of image-analysis technology with AI for real-time people-detection and tracking”, Tokyo, March 27, 2017.