



UNIVERSITÀ POLITECNICA DELLE  
MARCHE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA E  
DELL'AUTOMAZIONE

**Monitoraggio Respiratorio Senza Contatto:  
Applicazioni delle Telecamere di Profondità**

**Contactless Respiratory Monitoring: Applications of Depth  
Cameras**

Relatore:

**Prof. Gambi Ennio**

Candidato:

**Carloni Michele**

Correlatrice:

**Prof.ssa Senigagliesi Linda**

---

ANNO ACCADEMICO 2023/2024



## Abstract

Lo scopo di questa tesi è quello di estrarre il segnale respiratorio di un soggetto umano, attraverso l'utilizzo di una telecamera in grado di leggere il valore di profondità associato ad ogni pixel di ogni frame. La telecamera presa in considerazione in questo studio è l'Intel Realsense D455, accompagnata dal sensore a contatto fisico PolarH10. Il polarH10 estrarrà il segnale respiratorio corretto del soggetto il quale verrà confrontato con il segnale catturato dalla telecamera per capire se l'estrazione del segnale è avvenuta con successo. Il segnale respiratorio ottenuto dalla telecamera si basa sul movimento della distanza di una parte del corpo relativa al soggetto. Le aree di interesse sono state estratte attraverso il processamento dei frame, utilizzando il modello di machine learning YOLO v8n-pose, in grado di riconoscere una persona all'interno di un frame. Attraverso vari algoritmi implementati è stato possibile estrarre un segnale per ogni area del corpo considerata, calcolando la distanza dell'area dalla telecamera per ogni frame processato. I vari soggetti sono stati testati in condizioni statiche (in posizione seduto frontale e in piedi ruotato di  $45^\circ$ ) e dinamiche (movimento rotazionale del torso e movimento libero nello spazio inquadrato dalla telecamera). Come indice di valutazione è stato utilizzato il MAE (mean absolute error) il quale definisce l'errore medio in Hertz che si ha tra la frequenza respiratoria reale e quella predetta. Le predizioni effettuate nei casi statici hanno un MAE molto basso con un massimo di 0.04 Hz. Nel caso dinamico con la rotazione del torso del soggetto si è riscontrato un MAE di 0.008 Hz e 0.005 Hz per due algoritmi in particolare, definendo quindi un segnale respiratorio predetto molto accurato. Infine, con un MAE tra 0.053Hz e 0.0173 Hz, il caso dinamico di movimento libero da parte della persona non ha avuto il successo sperato.

# Indice

<b>Introduzione</b>	<b>8</b>
<b>1 Materiali</b>	<b>13</b>
1.1 POLAR H10 . . . . .	13
1.2 Camera Intel Realsense D455 . . . . .	14
1.3 Setup sperimentale e dataset . . . . .	17
<b>2 Metodi</b>	<b>19</b>
2.1 Algoritmi acquisizione della depth . . . . .	19
2.2 Algoritmi acquisizione dal Polar H10 . . . . .	20
2.3 Sincronizzazione dei segnali . . . . .	23
2.4 Elaborazione dei dati . . . . .	23
2.4.1 Processamento dati telecamera Realsense . . . . .	24
2.4.2 Estrazione del respiro da singolo punto . . . . .	25
2.4.3 Estrazione del respiro dal poligono dell'addome . . . . .	26
2.4.4 Estrazione del respiro dal poligono del petto . . . . .	27
2.4.5 Estrazione del respiro di tutto il torso . . . . .	27
2.4.6 Estrazione del punto centrale tra spalle . . . . .	28
2.4.7 Estrazione del punto centrale del collo . . . . .	29
2.4.8 Segnale respiratorio estratto dal PolarH10 . . . . .	29
2.4.9 Segnale ECG estratto dal PolarH10 . . . . .	32
2.4.10 Segnale accelerometrico estratto dal PolarH10 . . . . .	34
2.5 Elaborazione dei dati nel caso statico . . . . .	36
2.5.1 Segnale respiratorio estratto dalla telecamera . . . . .	36
2.6 Elaborazione dei dati caso dinamico . . . . .	37
2.6.1 Segnale respiratorio rispetto al centro tra spalle . . . . .	37
2.6.2 Segnale respiratorio rispetto al centro del collo . . . . .	38
2.6.3 Segnale respiratorio rispetto al poligono dell'addome . . . . .	38
<b>3 Risultati e discussione</b>	<b>41</b>
3.1 Confronto dei dati in condizioni statiche . . . . .	41
3.1.1 Soggetto seduto . . . . .	42
3.1.2 Soggetto ruotato . . . . .	44
3.2 Confronto dei dati in condizioni dinamiche . . . . .	46

3.2.1	Soggetto in movimento frontale . . . . .	48
3.2.2	Soggetto con rotazione del torso . . . . .	53
3.2.3	Soggetto movimento libero nello spazio . . . . .	58
<b>4</b>	<b>Sviluppi futuri</b>	<b>64</b>
<b>5</b>	<b>Conclusione</b>	<b>66</b>

# Elenco delle figure

1.1	PolarH10 indossato da un atleta . . . . .	14
1.2	Intel Realsense D455 . . . . .	15
1.3	Struttura interna della telecamera . . . . .	16
1.4	Setup sperimentale . . . . .	17
2.1	Rappresentazione del numpy array . . . . .	20
2.2	disposizione degli assi cartesiani del PolarH10 . . . . .	22
2.3	keypoints . . . . .	25
2.4	Punto Centrale . . . . .	26
2.5	Poligono dell'addome . . . . .	26
2.6	Poligono del torso . . . . .	27
2.7	Poligono dell'busto . . . . .	28
2.8	Punto centrale delle spalle . . . . .	28
2.9	Punto del collo . . . . .	29
2.10	Segnale nel tempo ed in frequenza . . . . .	31
2.11	Segnale respiratorio nel tempo catturato dal PolarH10 . . . . .	32
2.12	Segnale respiratorio in frequenza catturato dal PolarH10 . . . . .	32
2.13	grafico del segnale . . . . .	34
2.14	asse Z - grafico nel tempo . . . . .	35
2.15	asse Z - grafico filtrato . . . . .	35
2.16	asse Z - grafico in frequenza . . . . .	35
2.17	singolo punto - grafico nel tempo . . . . .	36
2.18	singolo punto - grafico filtrato . . . . .	36
2.19	singolo punto - grafico in frequenza . . . . .	37
2.20	grafico del movimento . . . . .	38
3.1	posizione seduto frontale . . . . .	42
3.2	posizione ruotato alla sinistra della telecamera . . . . .	45
3.3	posizione ruotato alla destra della telecamera . . . . .	45
3.4	Soggetto 4 - singolo punto frequenza . . . . .	46
3.5	Soggetto 4 - poligono addome frequenza . . . . .	46
3.6	Soggetto 4 - poligono petto frequenza . . . . .	46
3.7	Soggetto 4 - poligono torso frequenza . . . . .	46
3.8	camminata frontale . . . . .	48

3.9	Dinamico - singolo punto rispetto a centro spalle . . . . .	49
3.10	Dinamico - poligono addome rispetto a centro spalle . . . . .	49
3.11	Dinamico - singolo punto rispetto a centro collo . . . . .	50
3.12	Dinamico - poligono addome rispetto a centro collo . . . . .	50
3.13	Dinamico - singolo punto rispetto a poligono addome . . . . .	51
3.14	Dinamico - poligono petto rispetto a poligono addome . . . . .	51
3.15	Dinamico - poligono torso rispetto a poligono addome . . . . .	52
3.16	rotazione torso . . . . .	53
3.17	soggetto 3 - singolo punto rispetto a poligono addome . . . . .	55
3.18	soggetto 1 - frequenza del movimento dell'algorithmo poligono torso . .	56
3.19	soggetto 1 - poligono torso rispetto a poligono addome . . . . .	56
3.20	soggetto 2 - singolo punto rispetto a centro spalle . . . . .	57
3.21	soggetto 2 - singolo punto rispetto a centro spalle nel dominio del tempo	57
3.22	movimento libero . . . . .	58
3.23	soggetto 2 - singolo punto rispetto a centro collo (frequenza) . . . . .	60
3.24	soggetto 2 - singolo punto rispetto a centro collo (tempo) . . . . .	61
3.25	soggetto 3 - poligono addome rispetto a centro spalle (frequenza) . . .	61
3.26	soggetto 3 - poligono addome rispetto a centro spalle (tempo) . . . .	61

# Introduzione

Il segnale respiratorio, un indicatore vitale fondamentale, riveste un ruolo cruciale nella valutazione dello stato di salute di un individuo. Al giorno d'oggi molte delle diagnosi effettuate nel campo della medicina prendono in considerazione il segnale respiratorio del paziente per monitorare il funzionamento dei principali organi vitali (Riferimento [12]).

Con lo sviluppo della tecnologia negli ultimi decenni varie aziende hanno cominciato a produrre dei dispositivi dotati di sensori in grado di misurare il battito cardiaco e la respirazione del soggetto. Tra i più utilizzati ci sono sicuramente i sensori indossabili sul petto o sul polso che registrano le variazioni nel movimento toracico o addominale durante la respirazione. Pensiamo ad esempio ai vari orologi utilizzati durante sessioni di allenamento che monitorano il battito, il respiro e l'accelerazione cardiaca in tempo reale attraverso l'implementazione di sensori ottici poggiati sul polso.

Una sottocategoria di sensori indossabili sono le fasce cardio le quali, diversamente dagli smart watch, vengono poste attorno al torace in maniera aderente al corpo. Ogni volta che il cuore si contrae si genera un segnale elettrico. Questo segnale si propaga attraverso il cuore e si diffonde nel corpo. La variazione del potenziale elettrico che si verifica durante un battito cardiaco può essere misurata sulla superficie della pelle, attraverso gli elettrodi che vanno a rilevare la differenza di potenziale. Nonostante ciò la presenza fisica di un sensore sul proprio corpo al momento di una visita o di una sessione di allenamento può comportare alcune problematiche. Pensiamo ad esempio a tutte quelle persone con una pelle molto delicata che potrebbero riscontrare prurito e irritazione dopo alcuni minuti che il dispositivo è stato indossato (Riferimento [9]). Oltre a ciò, potrebbe risultare complesso interfacciarsi fisicamente con il paziente come è stato durante il periodo del Covid-19, durante il quale, con l'introduzione di nuove regole per la sicurezza in ambito sanitario, bisognava igienizzare il dispositivo prima e dopo del suo utilizzo ed evitare il contatto fisico con il paziente. Un altro fattore da tenere a mente è la tipologia di persona presa in considerazione, infatti nel caso di bambini o anziani la persona potrebbe fare fatica a cooperare con il medico durante la visita. Inoltre, l'utilizzo di sensori contactless permette a persone non specializzate di poter condurre delle misurazioni personalmente senza dover essere a conoscenza di procedure mediche per la lettura del respiro usando un sensore a contatto fisico. Con il progresso scientifico e soprattutto tecnologico, ad oggi esiste la possibilità di poter effettuare un'analisi

accurata del segnale respiratorio con l'utilizzo di sensori che non necessitano del contatto fisico con la persona. Un esempio è l'impiego di video camere in grado di leggere un segnale di tipo RGB-D (colori e profondità) in modo da poter elaborare anche movimenti relativi allo spostamento del torace del soggetto al momento del respiro. Nella letteratura scientifica corrente sono stati effettuati vari esperimenti utilizzando diversi soggetti, dispositivi ed algoritmi di misura per il controllo del respiro attraverso una video camera in grado di leggere il segnale dei colori e della profondità. La tabella 1 raccoglie le informazioni principali di 10 articoli di ricerca degli anni più recenti.

Le telecamere utilizzate per catturare l'immagine RGB ed il segnale di profondità sono state: telecamera Intel Realsense modello SR300, modello D415 e telecamera kinect. L'impiego di queste telecamere è stato accompagnato dall'uso di altri sensori come lo spirometro e cinture di respirazione per l'estrazione dei dati reali. In questo modo è stato poi possibile confrontare i dati acquisiti dalla telecamera con quelli rilevati dai sensori. Ogni esperimento scientifico citato fa utilizzo di un dataset variegato, includendo soggetti di entrambi i sessi su un ampio raggio di età. Così facendo è possibile definire delle performance più precise in quanto, se testate su un ampio dataset con soggetti differenti tra loro, risulteranno più affidabili. Per il processamento dei segnali catturati sono stati usati principalmente algoritmi di machine learning come Convulsional Neural Network, decision trees, random trees e Support Vector Machine. Il loro utilizzo ha diversi vantaggi tra i quali l'analisi e la gestione di dati complessi e rumorosi in modo più efficace rispetto ai metodi tradizionali, migliorando l'accuratezza e la precisione del rilevamento e della misurazione del segnale, e sono in grado di riconoscere pattern complessi nei dati respiratori che potrebbero non essere facilmente individuabili con metodi convenzionali. Infine, osservando i risultati ottenuti dalle varie sperimentazioni, si può affermare che la lettura del segnale respiratorio con una depth camera, facendo uso dei giusti algoritmi, risulta abbastanza precisa, e svolgendo acquisizioni per un periodo di tempo relativamente lungo il segnale risulterà corretto, con una percentuale di errore insignificativamente piccola.

Realizzato da	Riferimenti	Anno	Soggetti	Condizioni	Sensori	Algoritmo	Performance
University of Siegen, Germany	[10]	2021	19 Partecipanti (12 maschi, 7 femmine)	Ogni soggetto è stato registrato in 3 differenti situazioni: seduto, in piedi, in piedi con self occlusion (girato, torso poco visibile...)	Xbox kinect camera	Principal Component Analysis	La misurazione sul petto riporta un 93,7% di precisione e sull'addome 73,2%
University of Ottawa, Canada	[7]	2022	6 Soggetti	Spazio chiuso simile ad un ufficio. Testato con 1/2 persone presenti nel campo visivo della camera	Xbox kinect camera, 3 radar IR-UWB e 2 cinture di respirazione	Classificazione effettuata con SVM, decisions trees e random trees	Precisione complessiva del RR (Respiration Rate) pari circa al 90%
Membri del IEEE	[22]	2020	30 Soggetti in diverse condizioni di respirazione	Spazio chiuso, distanza tra 0.5m - 3m	Kinect V2 camera	Algoritmo BL-AT-GRU in combinazione con RNN (Recurrent Neural Network)	Precisione del RR pari a 93.6%
Carleton University, Canada	[11]	2021	5 neonati vestiti	Camera che inquadra l'incubatore di ogni soggetto	Intel RealSense SR300	2 CNN (Convolutional Neural Network)	Errore di 3.5 respiri/min per 82% delle volte
Carleton University, Canada	[15]	2022	9 Soggetti (4 femmine con età tra 16-42 e 5 maschi con età tra 18-34)	Telecamera posta ad una distanza di 0.3m dal soggetto su un'altezza di 1.3m	Intel RealSense D415	Indicatori: ICA, CCA. Algoritmo: Bland Altman	Errore di 2.89 respiri/min per CCA, errore di 10.42 respiri/min per ICA
Technopole Center, UK	[2]	2023	7 pazienti sani	Tutti i Pazienti sono stati testati con una frequenza di respiro tra 4-40 respiri/min con un incremento di 4 respiri al minuto alla volta.	Intel RealSense D415	RMSD (Root Mean Square Deviation)	Errore di 0.69 respiri/min
Membri del IEEE	[24]	2020	11 Soggetti	Totali di 66 registrazioni, 3 diversi tipi di respirazione (respirazione profonda, respirazione normale, respirazione lieve)	RGB CMOS camera USB M2ST036-H	CNN, ZCA (Zero-phase Component Analysis)	ZCA supera CNN con un'accuratezza del 65.5%
Sensors	[8]	2020	39 Soggetti (14 maschi e 25 femmine tra 65 - 75 anni)	2 differenti condizioni: con maglia, senza maglia	Xbox kinect camera, spirometro	Indicatori: tidal volume (TV), RR. Algoritmo: Bland-Altman	Errore TV 10.7%, Errore RR 0.2% (5 respiri su 206)
Sensors	[1]	2021	67 pazienti sani	Camera posta ad una distanza di 0.9m	Kinect V2 Camera	RMSD	Precisione del RR pari a 94.8%
Carleton University, Canada	[6]	2020	4 pazienti neonati	Camera posizionata sopra il letto del paziente in modo da inquadrare tutti gli angoli	Intel RealSense SR300	Perspective transformation, Automatic ROI Selection, Respiratory Rate estimation	Precisione del RR pari a 87%

Tabella 1: Tabella delle ricerche sui dispositivi di monitoraggio del segnale respiratorio.

In questa tesi proponiamo uno studio riguardo l'analisi del segnale RGB-D acquisito attraverso una telecamera Intel Realsense D455 per la lettura del segnale respiratorio senza utilizzo di sensori a contatto fisico. I dati raccolti dalla telecamera verranno poi confrontati con i dati derivanti dal sensore Polar H10 applicato ad ogni soggetto per definire l'accuratezza ed il margine di errore ottenuti.

Il resto della tesi è organizzato nel seguente modo:

Il capitolo 1 descrive tutti i materiali utilizzati durante la fase di sperimentazione. Raccoglie le informazioni principali riguardo ai due sensori Polar H10 e camera IntelRealsense D455 fondamentali per la cattura e il confronto dei dati spiegato nel capitolo 3. A seguire viene presentato il capitolo 2 dove viene descritto nel dettaglio tutto il processo di sperimentazione effettuato. In particolare verranno spiegati quali algoritmi sono stati utilizzati per l'acquisizione della depth, gli algoritmi di estrazione del segnale dal sensore Polar, il metodo di sincronizzazione dei 2 sensori ed infine come abbiamo elaborato questi segnali attraverso la loro rappresentazione grafica nel dominio del tempo e della frequenza. Procedendo con il capitolo 3, vengono raccolti e discussi tutti i dati estratti dai due sensori utilizzati. Sono mostrati i grafici di maggior interesse e le tabelle relative alle diverse situazioni sperimentali per confrontare gli algoritmi testati su ogni soggetto. Nel capitolo 4 vengono forniti alcuni spunti ed idee per possibili sviluppi futuri. Si riassumono i successi raggiunti con questo esperimento ed i punti problematici non ancora risolti. In Conclusione, il capitolo 5 richiamerà i punti principali toccati in questa tesi, mettendo in evidenza cosa è stato fatto e quali sono stati i risultati ottenuti.



# Capitolo 1

## Materiali

In questo capitolo verranno descritti i sensori utilizzati, riportando le informazioni principali ed il perché sono stati scelti per eseguire questo studio. In seguito verrà illustrato il setup sperimentale con le situazioni analizzate (tipologia di movimenti e distanza) ed il dataset utilizzato.

### 1.1 POLAR H10

Il Polar H10 è un cardiofrequenzimetro che si indossa sul petto tramite una fascia elastica, dotata di elettrodi avanzati per rilevazione accurata della frequenza cardiaca, riducendo al minimo le interferenze e migliorando la precisione delle misurazioni. Dispone di una tecnologia Bluetooth che consente di integrarlo facilmente con smartphone, smartwatch, computer da bicicletta, attrezzature da palestra e altri strumenti di monitoraggio della performance. In ambito clinico, il Polar H10 viene utilizzato per monitorare la frequenza cardiaca dei pazienti durante test di sforzo, riabilitazione cardiaca e altri programmi di monitoraggio medico (Riferimento [18]). Abbiamo scelto di utilizzare il Polar H10 per questo studio grazie alla sua comprovata accuratezza ed efficienza nella cattura e nell'elaborazione di segnali come il battito cardiaco e la frequenza cardiaca e dati reattivi all'elettrocardiogramma. In particolare, il sensore è stato impiegato contemporaneamente alla registrazione del segnale RGB-D della camera Intel RealSense, con l'obiettivo di definire i valori effettivi del segnale respiratorio. Grazie alla precisione dei dati forniti dal Polar H10, possiamo confrontare questi valori con quelli estratti dalla telecamera, verificando così l'affidabilità del segnale respiratorio registrato dalla Intel Realsense. Questo confronto ci permetterà di stabilire se i valori del segnale respiratorio estratti dalla telecamera corrispondono effettivamente a quelli registrati dal Polar H10, garantendo la validità dei risultati del nostro esperimento. In Figura 1.1 viene mostrato il PolarH10 indossato in fase di allenamento.



Figura 1.1: PolarH10 indossato da un atleta

Dal punto di vista tecnico il PolarH10 viene utilizzato principalmente per campionare il segnale dell'elettrocardiogramma e l'accelerazione che si genera con il movimento stesso del sensore. Di base la frequenza di campionamento dell'elettrocardiogramma risulta pari a 130 Hz mentre per l'accelerazione è di 200 Hz. In particolare l'accelerazione viene campionata sui tre assi, con i riferimenti posti come rappresentato in figura 2.2.

## 1.2 Camera Intel Realsense D455

La telecamera Intel RealSense D455 rappresenta un'innovazione significativa nel campo della percezione visiva e della tecnologia dei sensori ottici. La sua progettazione si basa su un sistema stereoscopico sofisticato che utilizza due sensori ottici posti agli estremi della telecamera. Questa configurazione simula la visione binoculare umana, permettendo alla telecamera di calcolare la profondità degli oggetti sulla base della disparità tra le immagini catturate dai due sensori. Questo approccio non solo consente alla D455 di catturare immagini ad alta risoluzione fino a 1280x800 pixel a 30 fotogrammi al secondo, ma anche di calcolare con precisione

la distanza dagli oggetti grazie al supporto di un emettitore infrarosso strutturato (Con risoluzione massima della depth pari a 1280x720). Questa tecnologia è particolarmente vantaggiosa in condizioni di luce variabile o in ambienti con superfici riflettenti, dove altri sensori potrebbero incontrare difficoltà. La RealSense D455 trova applicazioni in una vasta gamma di settori, dall'automazione industriale alla robotica avanzata [20], passando per la sanità [5] e l'interfaccia uomo-macchina. Per esempio, è utilizzata per la navigazione autonoma dei robot [3], il controllo di qualità automatizzato nei processi industriali [14], il monitoraggio del movimento per applicazioni sanitarie e l'interazione gestuale per l'interfaccia utente [17]. Il suo sviluppo mirato a migliorare l'interazione tra l'uomo e la macchina ha aperto nuove frontiere tecnologiche, rendendo possibili scenari operativi più intelligenti, sicuri ed efficienti. La telecamera Intel RealSense D455 è stata applicata per registrare segnali RGB-D durante le acquisizioni su vari soggetti. L'obiettivo è confrontare il segnale respiratorio estratto dal segnale di profondità con quello registrato dal Polar H10, valutando così l'affidabilità della RealSense nel rilevare il segnale respiratorio con precisione. La Figura 1.2 mostra un'immagine della telecamera in fase di utilizzo ed in Figura 1.3 è riportata la sua struttura interna.



Figura 1.2: Intel Realsense D455

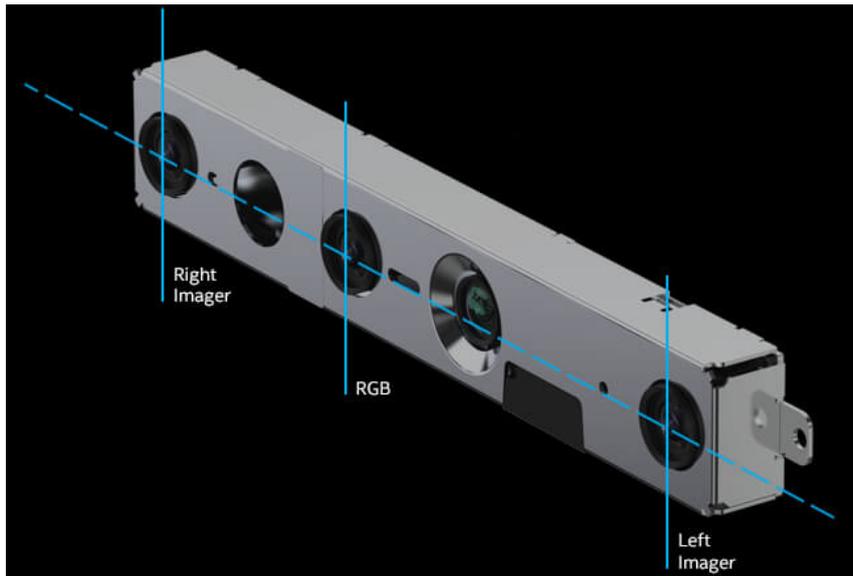


Figura 1.3: Struttura interna della telecamera

### 1.3 Setup sperimentale e dataset

Lo studio è stato condotto su un campione di 5 soggetti maschi, con un'età compresa tra i 20 e i 30 anni. L'obiettivo era testare la respirazione dei partecipanti sia in condizioni statiche che dinamiche. Per le condizioni statiche, i soggetti sono stati posizionati in due modalità: seduti frontalmente e in piedi con un'orientazione di 45 gradi rispetto alla camera. In particolare, la posizione seduta è stata testata a tre diverse distanze dalla telecamera: 1 metro, 2 metri e 3 metri. Per le condizioni dinamiche, sono state considerate tre tipologie di movimento: camminata frontale avanti e indietro, movimenti leggeri del torso e movimenti casuali in tutte le direzioni. È importante notare che non tutte le acquisizioni sono state effettuate con tutti i soggetti partecipanti, il che implica una variazione nel numero di prove per ciascuna configurazione. Questo approccio ha permesso di valutare l'affidabilità della telecamera Intel RealSense D455 in una varietà di scenari realistici, confrontando i dati raccolti con quelli del dispositivo PolarH10 per verificare la precisione della misurazione del segnale respiratorio. La Figura 1.4 mostra il setup sperimentale dove sono state effettuate tutte le acquisizioni.



Figura 1.4: Setup sperimentale



# Capitolo 2

## Metodi

In questo capitolo verranno descritti i metodi utilizzati per l'acquisizione, la sincronizzazione e l'elaborazione dei dati raccolti durante lo studio. Verranno dettagliati gli algoritmi impiegati per l'acquisizione delle immagini di profondità mediante la telecamera Intel RealSense D455 e per la registrazione dei segnali respiratori tramite il dispositivo PolarH10. Successivamente, sarà illustrato il processo di sincronizzazione dei segnali provenienti dai due dispositivi, fondamentale per garantire un confronto accurato dei dati. Infine, verranno spiegate le tecniche di elaborazione dei dati utilizzando grafici, sia per il caso statico sia per il caso dinamico.

### 2.1 Algoritmi acquisizione della depth

Per eseguire le acquisizioni del segnale di profondità attraverso la telecamera Intel RealSense D455, abbiamo sviluppato uno script in Python utilizzando la libreria `pyrealsense2`. L'obiettivo principale era catturare il massimo numero di frame al secondo, garantendo così una registrazione dettagliata e fluida del segnale. A tal fine, abbiamo implementato una funzione asincrona in Python che apre uno stream di dati e legge ogni frame in arrivo. Durante questo processo, due array principali, `'color_data'` e `'depth_data'`, vengono dichiarati per immagazzinare rispettivamente i dati relativi ai segnali di colore e profondità.

In dettaglio, `'color_data'` è un array in cui ogni elemento è un array tridimensionale di NumPy. Ogni elemento di questo array tridimensionale rappresenta tre matrici, ciascuna dedicata a uno dei canali RGB (Red, Green, Blue). Queste matrici hanno una larghezza e un'altezza pari alla risoluzione della telecamera, che in questo caso è 640x480 pixel. D'altra parte, `'depth_data'` è un array in cui ogni elemento corrisponde a una matrice singola, dove ogni valore rappresenta la profondità di un determinato pixel. Sebbene l'uso di array tridimensionali di NumPy possa sembrare complesso a prima vista, la Figura 2.1 fornisce una rappresentazione grafica di come funzionano questi array multidimensionali, facilitando la comprensione del loro utilizzo.

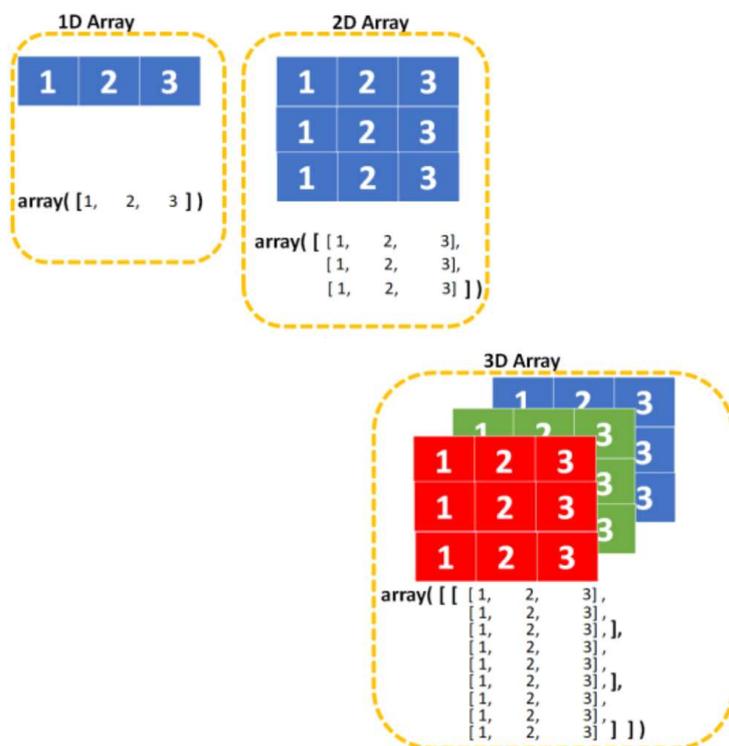


Figura 2.1: Rappresentazione del numpy array

Una volta che l'algoritmo all'interno della funzione ha processato tutti i frame, vengono restituiti gli array 'color\_data' e 'depth\_data', insieme all'array timestamp, che contiene i valori temporali relativi a ogni frame. Questa scelta progettuale permette di calcolare successivamente i dati di profondità per una persona individuata in ogni frame e associare questi dati a un preciso istante temporale preso dall'array dei timestamp. Per facilitare il riutilizzo di queste variabili in altri script, abbiamo deciso di salvarle nei file 'color\_data.npy', 'depth\_data.npy' e 'timestamps\_realsense.npy'. In questo modo, i dati acquisiti possono essere facilmente caricati e analizzati in successivi passaggi del progetto.

## 2.2 Algoritmi acquisizione dal Polar H10

All'interno dello stesso script in linguaggio python per l'estrazione del segnale dalla telecamera intel Realsense D455, abbiamo definito una seconda funzione per l'estrazione del segnale dal sensore PolarH10. Affinché l'algoritmo implementato vada a buon fine, sono state dichiarate 5 variabili globali. I primi tre sono codici esadecimali che prendono il nome di UUID (Universally Unique Identifier). Questi UUID sono utilizzati per identificare in modo univoco i servizi e le caratteristiche Bluetooth Low Energy (BLE) del dispositivo Polar H10. Mentre le variabili ECG\_WRITE e ACC\_WRITE sono entrambe array di byte (bytearray) utilizzate per inviare co-

mandi specifici al dispositivo. Questi comandi sono scritti in una forma binaria che il dispositivo Polar H10 può interpretare per attivare determinate funzionalità, come la registrazione dell'ECG (elettrocardiogramma) e dell'accelerometro.

Di seguito sono descritte nel dettaglio le variabili dichiarate:

- `PMD_SERVICE = "FB005C80-02E7-F387-1CAD-8ACD2D8DF0C8"`  
Gestisce il servizio principale del PolarH10, essenziale per le operazioni di monitoraggio della frequenza cardiaca e di altri dati biometrici.
- `PMD_CONTROL = "FB005C81-02E7-F387-1CAD-8ACD2D8DF0C8"`  
Rappresenta una caratteristica di controllo all'interno del servizio principale, utilizzata per inviare comandi al dispositivo, come avviare o interrompere la registrazione dei dati.
- `PMD_DATA = "FB005C82-02E7-F387-1CAD-8ACD2D8DF0C8"`  
Rappresenta una caratteristica per la trasmissione dei dati, utilizzata per ricevere i dati effettivi dal dispositivo, come i valori della frequenza cardiaca misurati.
- `ECG_WRITE = bytearray([0x02, 0x00, 0x00, 0x01, 0x82, 0x00, 0x01, 0x01, 0x0E, 0x00])`  
Variabile che contiene un array di byte (`bytearray`) specifico utilizzato per inviare un comando al dispositivo Polar H10. Questo comando attiva la registrazione dei dati dell'elettrocardiogramma (ECG) dal dispositivo. I byte nell'array rappresentano un messaggio formattato secondo il protocollo di comunicazione del Polar H10, includendo informazioni come il tipo di comando, parametri di configurazione e identificatori necessari per avviare la registrazione ECG. Un parametro definito da questo messaggio è la frequenza di campionamento, che per l'ECG è pari a 130 Hz.
- `ACC_WRITE = bytearray([0x02, 0x02, 0x00, 0x01, 0xC8, 0x00, 0x01, 0x01, 0x10, 0x00, 0x02, 0x01, 0x08, 0x00])`  
Anche questa variabile contiene un array di byte che serve per inviare un comando al dispositivo al fine di attivare la registrazione dei dati dell'accelerometro. Il messaggio formattato contiene specifiche quali la tipologia del messaggio e parametri di configurazione, inclusa la frequenza di campionamento, che per l'accelerometro è fissata a 200 Hz.

Avendo dichiarato tutte e cinque le variabili necessarie, all'interno della funzione sono stati definiti 4 array: 'ecg\_session\_data', 'ecg\_session\_time', 'acc\_session\_data', 'acc\_session\_time'. Le prime due servono per immagazzinare i dati e i tempi di cattura relativi all'elettrocardiogramma, mentre le successive due variabili conservano i dati ed i tempi relativi all'accelerometro. Per riempire queste variabili vengono chiamate le funzioni 'ecg\_data\_conv' e 'acc\_data\_conv'. Nella prima funzione si esegue una lettura dei dati relativi all'ecg e tutti i valori che vengono letti vengono poi immagazzinati all'interno dell'array 'ecg\_session\_data' attraverso il comando `append()`, allo stesso modo avviene per l'array 'ecg\_session\_time' che invece colleziona tutti gli istanti di tempo in cui la misura dell'ecg è stata prelevata. Nella seconda funzione, vengono letti i dati relativi all'accelerazione, facendo un accorgimento su quella che è la struttura dei dati ricevuti, infatti in questo caso la registrazione verrà fatta a 3 byte alla volta, in cui ogni byte rappresenta rispettivamente l'accelerazione sull'asse x, y, z. In Figura 2.2 è mostrato lo schema di disposizione degli assi dell'accelerazione sullo spazio tridimensionale.



Figura 2.2: disposizione degli assi cartesiani del PolarH10

Alla fine dell'esecuzione della funzione, verranno ritornati i 4 array 'ecg\_session\_data', 'ecg\_session\_time', 'acc\_session\_data', 'acc\_session\_time' nel codice main dello script. Questi 4 array verranno poi salvati nei rispettivi file numpy 'polar\_ecg\_time.npy', 'polar\_ecg\_data.npy', 'polar\_acc\_time.npy', 'polar\_acc\_data.npy' per essere riutilizzati nei successivi script di elaborazione dei dati e implementazione di grafici, che verrà spiegato nella sezione 2.4.

## 2.3 Sincronizzazione dei segnali

Dal punto di vista pratico il soggetto viene posto davanti alla telecamera già con il sensore Polar indossato ad una distanza variabile tra 0.5 e 4 metri. In questo caso la telecamera Realsense è stata impostata ad una risoluzione di 640x480 per avere una lettura più fluida dei frame e meno pixel da elaborare. Avviando lo script vengono eseguiti tutti i settaggi per la configurazione della telecamera Realsense e viene lanciata la funzione "connect\_to\_polar()" che ritorna un valore booleano se il polar risulta attivo e connesso al computer in cui è presente lo script per cominciare la trasmissione dei dati. Al momento della connessione del dispositivo vengono lanciate le due funzioni 'acquire\_realsense\_data' e 'acquire\_polar\_data' su due thread differenti, facendo uso della libreria 'asyncio' per la gestione di funzioni asincrone. Il comando utilizzato è il seguente:

```
result_intel, result_polar = await asyncio.gather(acquire_realsense_data(config, st_time, align), acquire_polar_data(client, st_time))
```

Questo permette di avere un campionamento sincrono dei frame RGB-D catturati con la telecamera Intel e i dati dell'ECG ed accelerazione registrati dal PolarH10. Alla termine delle esecuzioni delle 2 funzioni verranno ritornate le variabili array di interesse che verranno salvate nei file '.npy' corrispondenti come già spiegato in precedenza.

## 2.4 Elaborazione dei dati

In questa sezione verranno descritti i processi successivi all'acquisizione e al salvataggio dei dati. Nella sotto sezione 2.4.1, verrà illustrato il processamento dei frame catturati utilizzando l'algoritmo di deep learning YOLO (You Only Look Once) per stimare la posa del soggetto e calcolare un valore di distanza associato al respiro. Questo algoritmo consente di analizzare i dati acquisiti dalla telecamera Intel RealSense D455, identificando e misurando i cambiamenti di distanza in una zona di interesse causati dal movimento respiratorio. Si seguirà con la spiegazione di altri algoritmi implementati per ricavare il segnale respiratorio, definito dal cambiamento di distanza, relativo a diverse aree di interesse del corpo del soggetto di riferimento.

### 2.4.1 Processamento dati telecamera Realsense

Attraverso il processamento dei frame catturati con la telecamera Intel RealSense, lo scopo principale è quello di estrarre il segnale respiratorio basandosi sul movimento del busto del soggetto, definendo un valore di distanza per ogni frame, che varierà a seconda se il soggetto sta inspirando o espirando. La prima operazione da compiere è individuare la sagoma della persona ed estrarla dal resto dello sfondo. A tal fine, abbiamo utilizzato l'algoritmo di Deep Learning YOLO (You Only Look Once) nella sua versione v8n-pose, che è in grado di riconoscere e analizzare le figure presenti all'interno di un'immagine. L'algoritmo YOLO v8n-pose è una delle ultime evoluzioni della famiglia YOLO, progettata per il rilevamento di oggetti in tempo reale con un'enfasi particolare sul riconoscimento accurato delle pose umane. Questo algoritmo combina l'efficienza e la velocità di YOLO con capacità avanzate di analisi delle pose, rendendolo particolarmente adatto per applicazioni in cui è necessario rilevare e tracciare i movimenti del corpo umano. YOLO v8n-pose funziona suddividendo l'immagine in una griglia e, in ogni cella della griglia, predice la presenza di oggetti e le loro pose. Questa versione specifica dell'algoritmo è ottimizzata per riconoscere le articolazioni del corpo umano e ricostruire la posizione del soggetto in modo preciso. L'algoritmo è addestrato su grandi dataset di immagini annotate, che gli consentono di apprendere le caratteristiche distintive delle pose umane e di distinguere con alta precisione tra diverse parti del corpo. Una volta che l'algoritmo identifica il soggetto all'interno del frame, è possibile estrarre le coordinate delle articolazioni chiave del busto, come le spalle e il torace. Queste coordinate vengono poi utilizzate per calcolare la distanza relativa del busto rispetto alla telecamera. Misurando i cambiamenti di queste distanze frame per frame, possiamo determinare se il soggetto sta inspirando o espirando. L'efficienza di YOLO v8n-pose permette di eseguire questa analisi in tempo reale, il che è essenziale per applicazioni che richiedono un monitoraggio continuo del respiro. Inoltre, l'algoritmo è abbastanza robusto da funzionare in vari contesti come condizioni di bassa illuminazione (Riferimento [23]), garantendo un'accuratezza elevata nella rilevazione delle pose umane. In sintesi, l'uso di YOLO v8n-pose per l'analisi dei frame catturati dalla telecamera Intel RealSense ci consente di estrarre in modo preciso e affidabile il segnale respiratorio basato sul movimento del busto del soggetto, migliorando l'accuratezza complessiva del nostro esperimento. Per l'estrazione del respiro, abbiamo deciso di implementare 6 funzioni diverse, ciascuna progettata per estrarre la distanza di un punto preciso all'interno della sagoma del soggetto, oltre a 6 array distinti in grado di immagazzinare i valori delle distanze relative a ciascuna funzione. Inizialmente vengono letti i file 'color\_data.npy' e 'depth\_data.npy', immagazzinando i loro valori nelle variabili numpy array 'color\_data' e 'depth\_data'. In seguito si ha l'implementazione di un algoritmo che comincia con un ciclo for che itera sulla lista contenente tutti i valori dei pixel relativi ai frame, elaborando così un frame alla volta. Per ogni frame viene rilevata la sagoma della persona e ad essa vengono associati vari keypoints. I keypoints sono punti del corpo del soggetto che l'algoritmo YOLO calcola automaticamente. In particolare, i keypoints di maggior interesse

che vengono definiti includono: naso, occhi, orecchie, spalle, gomiti, bacino, gambe e piedi (Figura 2.3). Una volta estratti questi keypoints, le 6 funzioni vengono chiamate una dopo l'altra e, per ognuna, si calcola un valore della distanza di un'area di interesse che verrà poi inserito all'interno dell'array rappresentativo di quella funzione. Di seguito è riportata la spiegazione della logica di funzionamento di tutte le funzioni implementate.

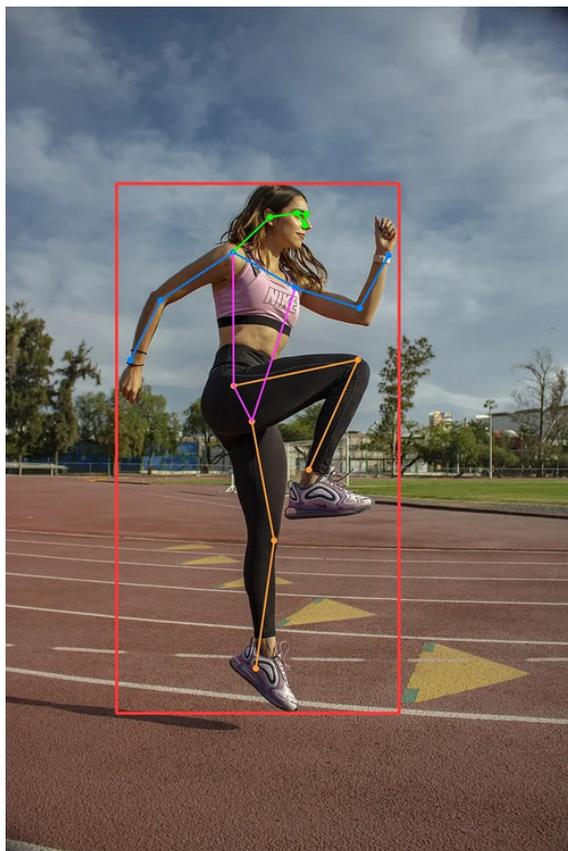


Figura 2.3: keypoints

## 2.4.2 Estrazione del respiro da singolo punto

Questa funzione riceve in input i due keypoints delle spalle e i due keypoints del bacino. Viene fatta una media tra questi 4 keypoints, trovando quindi le coordinate del punto centrale al bacino, e poi viene calcolata una seconda media tra i due keypoints delle spalle e il punto appena trovato (Figura 2.4). Alla fine come risultato otterremo quindi un punto posizionato centralmente sul petto, ed utilizzando la variabile 'depth\_data' è possibile ricavare la distanza passandogli come indici le coordinate del punto trovato. La funzione ritornerà poi il valore della distanza che verrà immagazzinato nella lista 'depth\_data\_list\_singolo\_punto'.

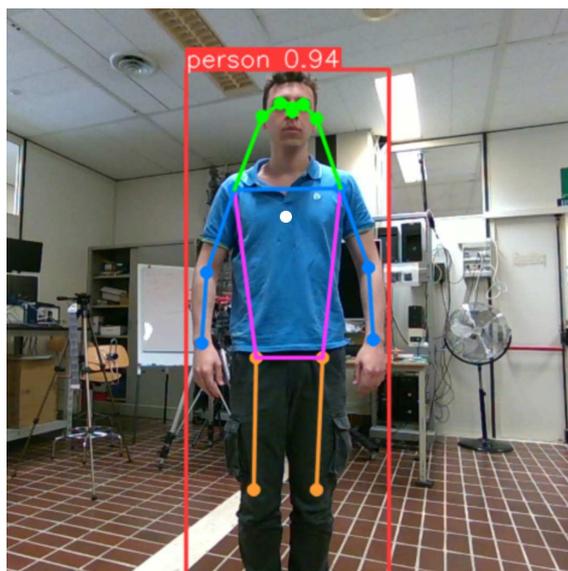


Figura 2.4: Punto Centrale

### 2.4.3 Estrazione del respiro dal poligono dell'addome

In questa funzione viene costruito il poligono che comprende solo l'addome, ovvero dalla fine della gabbia toracica fino al bacino. In seguito, per ogni punto appartenente al poligono viene calcolata la distanza dalla telecamera facendo poi una media delle stesse distanze. Alla fine viene ritornato il valore per essere inserito nella lista corrispondente. In Figura 2.5 viene mostrata un immagine del poligono considerato.

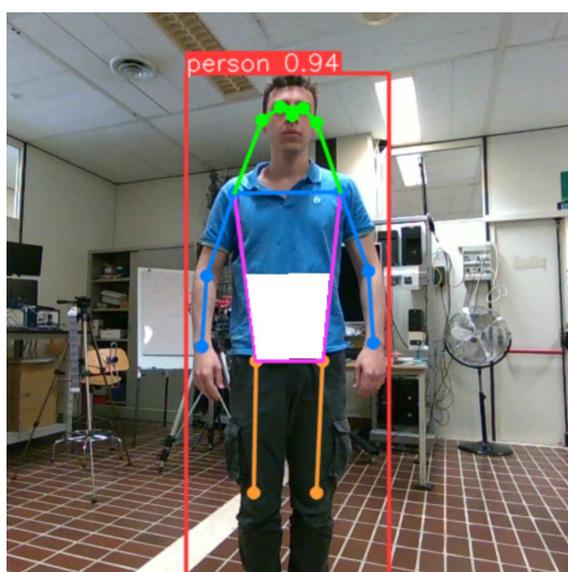


Figura 2.5: Poligono dell'addome

#### 2.4.4 Estrazione del respiro dal poligono del petto

In questa funzione si parte dai 4 keypoints, 2 per il bacino e 2 per la spalla, ottenendo il poligono del petto. In seguito si procede facendo la media delle distanze di ogni punto contenuto nel poligono ed il risultato viene ritornato per essere immagazzinato nell'array relativo alla funzione. La Figura 2.6 mostra un immagine del poligono di riferimento.

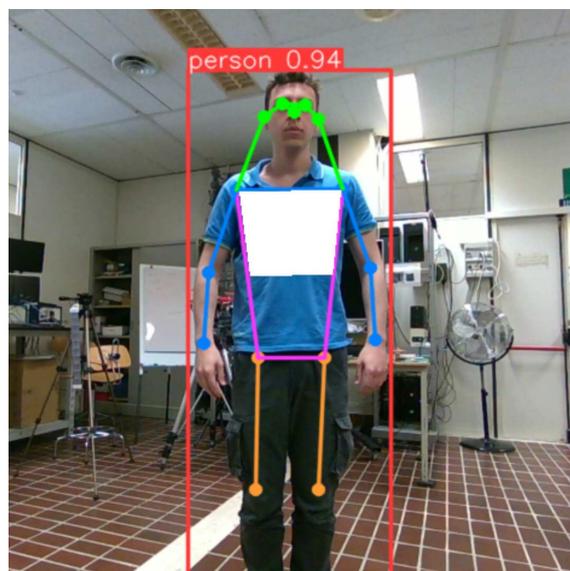


Figura 2.6: Poligono del torso

#### 2.4.5 Estrazione del respiro di tutto il torso

Allo stesso modo delle precedenti 2 funzioni, qui viene creato il poligono che copre tutto il torso (Figura 2.7). Per tutti i punti appartenenti al poligono viene calcolata la distanza dalla telecamera ed in seguito si procede facendo una media di tutti quei punti. In fine la funzione restituirà il valore della distanza ottenuta da inserire nella lista corrispondente.

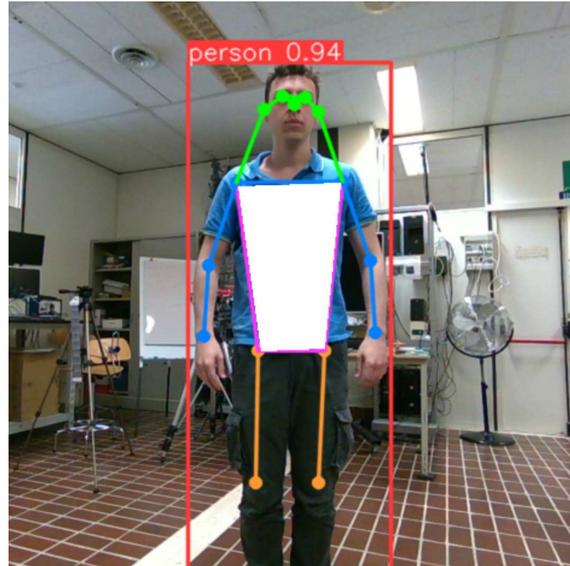


Figura 2.7: Poligono dell'busto

#### 2.4.6 Estrazione del punto centrale tra spalle

In questa funzione viene effettuata una media delle coordinate  $x$  ed  $y$  dei due key-points delle spalle in modo da trovare il punto centrale. Verrà poi calcolata la distanza del punto centrale per essere restituita a termine della funzione. In Figura 2.8 è rappresentata l'immagine del punto centrale delle spalle.

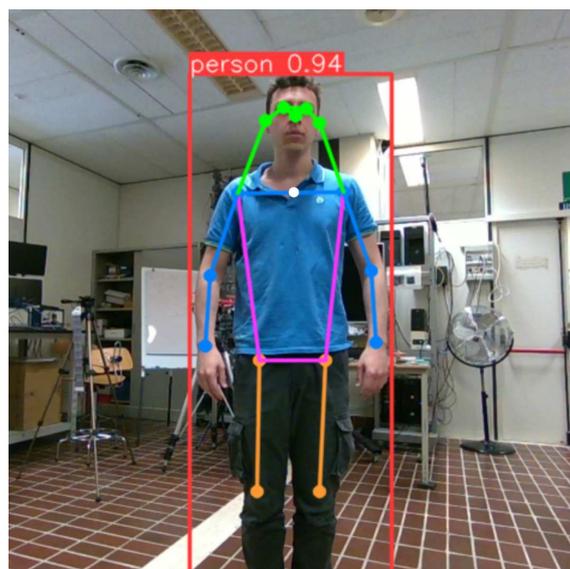


Figura 2.8: Punto centrale delle spalle

## 2.4.7 Estrazione del punto centrale del collo

Quest'ultima funzione si occupa di calcolare la distanza del collo dalla telecamera. Inizialmente viene fatta la media tra il keypoint del naso ed il punto centrale delle spalle trovato nella funzione precedente. Trovato il nuovo punto viene calcolata la distanza che esso ha dalla telecamera ed il suo valore viene ritornato dalla funzione. La figura 2.9 mostra il punto calcolato del collo

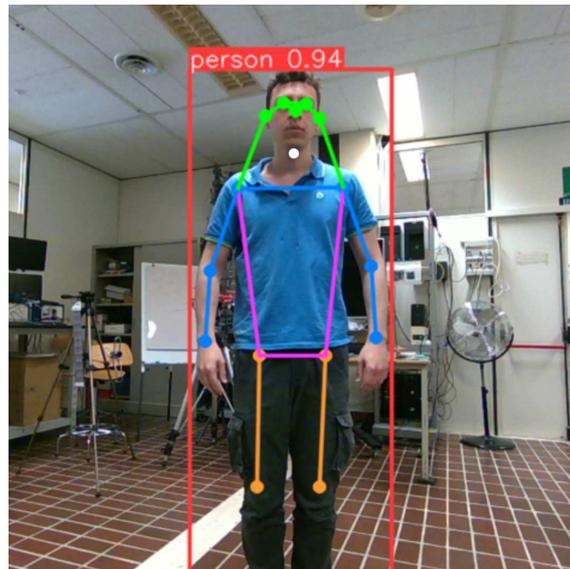


Figura 2.9: Punto del collo

Le distanze relative al punto centrale tra spalle ed il punto dentrale del collo saranno poi utili in seguito quando verranno implementati gli algoritmi per la rappresentazione del segnale respiratorio catturato con la telecamera nel caso dinamico. Dopo che il ciclo for avrà iterato tutti i frame catturati, le 6 liste verranno salvate nei seguenti file: 'depth\_data\_yolo1.npy', 'depth\_data\_yolo2.npy', 'depth\_data\_yolo3.npy', 'depth\_data\_yolo4.npy', 'depth\_data\_yolo\_spalla.npy', 'depth\_data\_yolo\_collo.npy'.

## 2.4.8 Segnale respiratorio estratto dal PolarH10

Per l'estrazione del segnale respiratorio catturato attraverso il PolarH10 viene considerato il segnale dell'elettrocardiogramma. Inizialmente vengono letti i 2 file numpy 'polar\_ecg\_data.npy', 'polar\_ecg\_time.npy'. Avendo notato che in tutti i soggetti il valore dell'ECG non risulta corrispondente alla realtà per i primi secondi di campionamento, allora abbiamo deciso di scartare tutti i valori campionati nei primi 5 secondi della misurazione. A questo punto, avremo 2 array 'ecg\_data' e 'ecg\_time' contenenti rispettivamente i valori dell'ecg e del timestamp attendibili. La differenza tra l'ultimo ed il primo valore di 'ecg\_time' risulterà pari al tempo complessivo di durata del campionamento e verrà memorizzato nella variabile 'time\_diff'. Grazie a questo valore possiamo calcolare la frequenza di campionamento dividendo la

lunghezza di `'ecg_data'` (il numero complessivo di campioni) per `'time_diff'` ovvero l'effettivo tempo di campionamento. Utilizzando la libreria di python `'neurokit2'`, atta all'analisi e la visualizzazione dei segnali fisiologici, possiamo calcolarci la frequenza cardiaca passando alla funzione `'ecg_rate()'` la lista `'ecg_data'`. A questo punto, non rimane altro che ricavarsi il segnale respiratorio che verrà inserito nel numpy array chiamato `'edr'` (ECG-Derived Respiration). Per far ciò la libreria `'neurokit2'` (Riferimento [13]) mette a disposizione la funzione `'ecg_rsp()'` che prende in input `'ecg_rate'` (frequenza cardiaca), la frequenza di campionamento pari a 130 Hz e un metodo di campionamento. Tra i 4 metodi di campionamento `'vangent2019'` ([21]), `'sarkar2015'` ([19]), `'charlton2016'` ([4]), `'soni2019'` ([16]), è stato scelto il metodo `'vangent2019'`. Questo metodo, sviluppato da Van Gent et al. nel 2019, si basa sull'osservazione che le variazioni respiratorie influenzano l'ampiezza e la posizione dei picchi R nel tracciato ECG. Durante la respirazione, i movimenti toracici e i cambiamenti nella pressione intratoracica alterano il ritorno venoso al cuore, provocando variazioni ritmiche nell'ampiezza delle onde R. Questo fenomeno, noto come modulazione respiratoria del segnale ECG, viene sfruttato dal metodo `'vangent2019'` per dedurre il ritmo respiratorio. Il processo di estrazione del segnale respiratorio tramite `'nk.ecg_rsp'` con il metodo `'vangent2019'` inizia con la rilevazione dei picchi R nel segnale ECG. Una volta individuati i picchi R, il metodo analizza le variazioni nell'ampiezza e nell'intervallo tra i picchi per costruire un segnale che rappresenta l'andamento respiratorio. La funzione implementa algoritmi avanzati di elaborazione del segnale per isolare le componenti del segnale ECG influenzate dalla respirazione, garantendo una stima accurata dell'attività respiratoria. Questo approccio presenta diversi vantaggi. Innanzitutto, permette di monitorare contemporaneamente il battito cardiaco e la respirazione utilizzando un solo tipo di sensore, riducendo la necessità di apparecchiature aggiuntive e semplificando l'acquisizione dei dati. Inoltre, l'accuratezza del metodo `'vangent2019'` nella stima del segnale respiratorio lo rende particolarmente utile in contesti clinici e di ricerca, dove è essenziale ottenere misure precise dell'attività fisiologica. Inoltre, la funzione `'nk.ecg_rsp'` della libreria `'neurokit2'` è progettata per essere user-friendly e altamente configurabile, permettendo agli utenti di applicare facilmente il metodo `'vangent2019'` ai loro dati ECG. Gli utenti possono personalizzare i parametri del filtro e le impostazioni di rilevamento dei picchi per adattarsi meglio alle specificità dei loro dati e alle esigenze della loro analisi. In conclusione, il metodo di campionamento `'vangent2019'` rappresenta un'importante innovazione nell'estrazione del segnale respiratorio dai dati ECG. Integrato nella funzione `'nk.ecg_rsp'` di `'neurokit2'`, offre una soluzione potente e versatile per il monitoraggio simultaneo del battito cardiaco e della respirazione, con applicazioni potenziali in una vasta gamma di studi fisiologici e clinici. Le informazioni riportate sono state prese dal seguente riferimento: [21]. Per il segnale respiratorio estratto dal PolarH10 vengono rappresentati due grafici: uno nel dominio del tempo e uno nel dominio della frequenza. Entrambi i grafici sono realizzati utilizzando la funzione `'plot()'` della libreria `'matplotlib'`. Il grafico nel dominio del tempo prende come input l'intervallo di tempo da visualizzare lungo l'asse x, mentre la variabile `'edr'`, che

rappresenta il segnale respiratorio, viene mostrata lungo l'asse  $y$ . Questo consente di ottenere un segnale sinusoidale in cui ogni oscillazione corrisponde a un respiro. Idealmente, contando il numero di respiri nel segnale temporale e confrontandolo con il numero di respiri rilevati con la telecamera Intel Realsense, si potrebbe valutare la qualità del segnale catturato con la telecamera. Tuttavia, questo approccio presenta alcune difficoltà: con acquisizioni di lunga durata, contare manualmente tutte le oscillazioni diventerebbe scomodo e aumenterebbe il rischio di errori. Pertanto, risulta più pratico determinare la frequenza respiratoria del soggetto, un valore unico espresso in Hz, che può essere facilmente confrontato con altre frequenze rilevate in altri grafici. Per realizzare ciò, è necessario applicare la trasformata di Fourier al segnale respiratorio espresso nel dominio del tempo. La trasformata di Fourier è una tecnica matematica utilizzata per trasformare un segnale dal dominio del tempo al dominio della frequenza. La trasformata di Fourier si basa sull'idea che un segnale periodico possa essere rappresentato come una combinazione lineare di onde sinusoidali di diverse frequenze. In Figura 2.10 viene mostrato un segnale continuo nel dominio del tempo (segnale colorato rosso), ed il corrispettivo segnale nel dominio della frequenza (Segnale blu). Come possiamo osservare il segnale nel tempo è scomponibile in 3 sinusoidi principali, ovvero quelle con ampiezza maggiore. Ogni sinusoide ha una frequenza diversa l'una dalle altra, per questo, nel grafico in frequenza possiamo distinguere 3 picchi più prominenti rispetto a tutti gli altri.

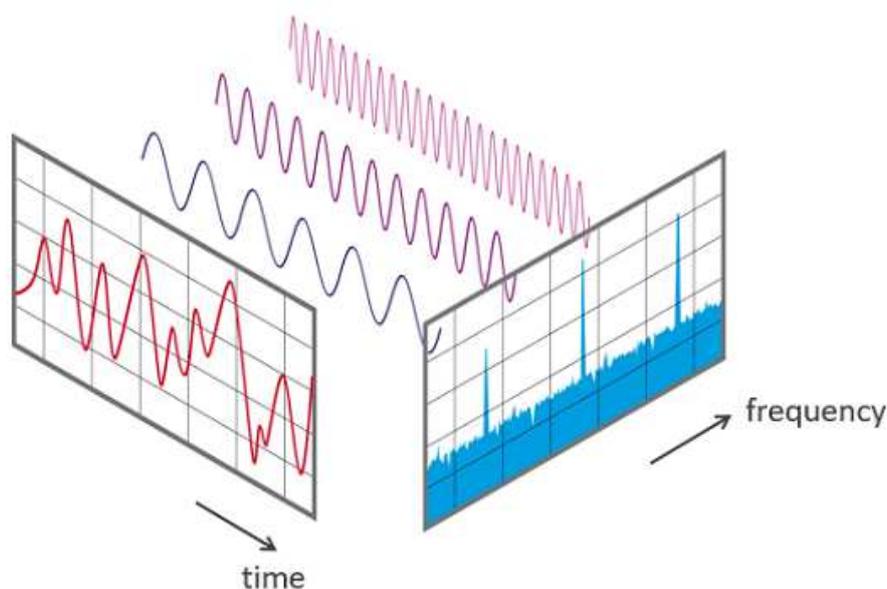


Figura 2.10: Segnale nel tempo ed in frequenza

Dal punto di vista pratico, la libreria 'numpy' introduce il metodo 'fft' che prende in input la variabile di riferimento del segnale respiratorio (in questo caso 'edr') ed esegue la trasformata di fourier. A questo punto viene applicata una maschera per selezionare solo il range di frequenze che ci interessa vedere, selezionando in

questo caso le frequenze comprese tra 0.1 Hz e 1 Hz. Per mostrare i risultati a schermo viene utilizzata sempre la funzione 'plot()' che prende in input tutte le frequenze selezionate attraverso la maschera. In Figura 2.11 e Figura 2.12 vengono mostrati rispettivamente i grafici del segnale respiratorio catturato con il polar nel dominio del tempo e nel dominio della frequenza. Come è possibile notare il grafico in frequenza ha un utilizzo più semplice rispetto al grafico nel dominio del tempo perché mostra la frequenza maggiore, ovvero quella della respirazione, che poi può essere confrontata con quella di altri segnali respiratori registrati dalla telecamera intel.

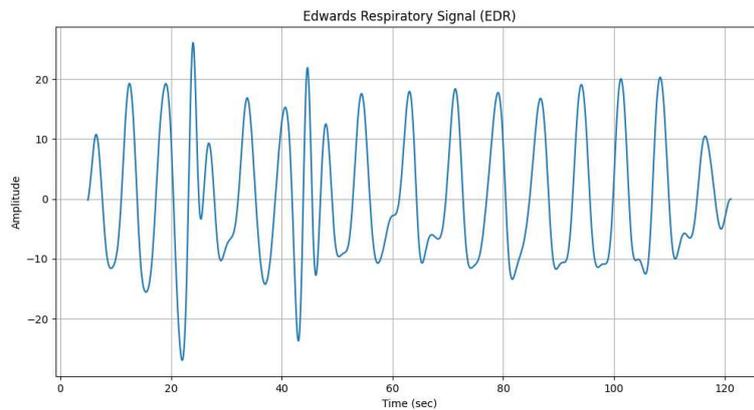


Figura 2.11: Segnale respiratorio nel tempo catturato dal PolarH10

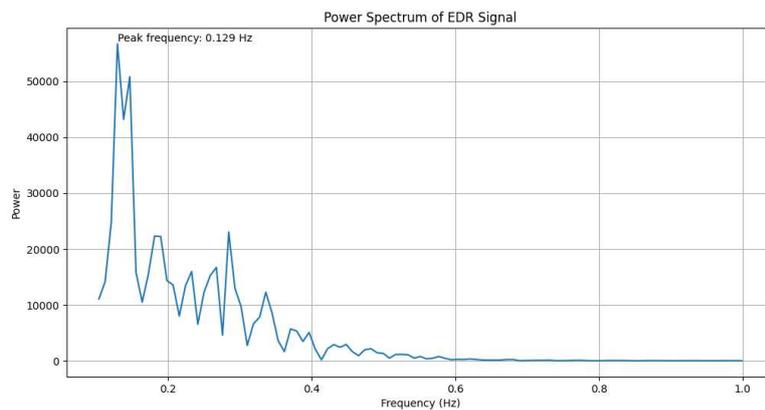


Figura 2.12: Segnale respiratorio in frequenza catturato dal PolarH10

## 2.4.9 Segnale ECG estratto dal PolarH10

Il segnale dell'elettrocardiogramma (ECG) è essenziale per monitorare il ritmo cardiaco. Nell'ambito di questo esperimento, l'analisi inizia con l'importazione dei dati dai file 'polar\_ecg\_data.npy' e 'polar\_ecg\_time.npy', memorizzati rispettivamente

nei numpy array 'ecg\_data' e 'ecg\_time'. Per garantire un segnale pulito e affidabile, i primi 5 secondi di registrazione vengono esclusi dall'analisi, essendo valori sporchi e lontani dal valore reale. Successivamente, viene calcolata la frequenza di campionamento, determinata dal rapporto tra il numero di campioni e il periodo di campionamento. Un parametro cruciale è il BPM (Battiti Al Minuto), che riflette il ritmo cardiaco del soggetto. Questo valore è derivato dall'analisi dei complessi QRS, onde elettrocardiografiche che rappresentano la depolarizzazione dei ventricoli del cuore. Il complesso QRS è una parte essenziale dell'elettrocardiogramma (ECG), il tracciato che registra l'attività elettrica del cuore. Questo complesso è composto da tre onde distintive: la prima, chiamata onda Q, rappresenta la depolarizzazione del setto interventricolare. Subito dopo, l'onda R indica la depolarizzazione della massa principale dei ventricoli, seguita dall'onda S, che mostra la depolarizzazione delle parti terminali dei ventricoli. L'analisi del complesso QRS è fondamentale per valutare la salute cardiaca e diagnosticare varie condizioni. La durata normale del QRS è inferiore a 0,12 secondi, mentre anomalie nella forma o nella durata del complesso possono indicare problemi come blocchi di branca o infarto miocardico. La morfologia del QRS, insieme alla sua ampiezza, fornisce importanti informazioni cliniche sulla funzione cardiaca del paziente. Nella pratica, il rilevamento dei picchi R nel segnale ECG è cruciale per determinare la frequenza cardiaca e identificare eventuali aritmie. Utilizzando strumenti come la libreria 'biosppy' in Python, è possibile automatizzare questo processo, migliorando l'efficienza e la precisione dell'analisi. Eseguendo il rapporto tra il numero complessivo di picchi ricavato dal segnale ECG ed il tempo di campionamento, è possibile ottenere il valore BPM. Ovviamente l'intenzione è quella di graficare i dati, per questo si usufruisce della libreria 'matplotlib' che mette a disposizione vari metodi per la creazione di vari tipi di grafici. il metodo 'plot()' prende in input la lista dei tempi sull'asse delle X, i valori campionati dell'ecg sull'asse delle Y e rinomina entrambi gli assi con le funzioni 'xlabel()' e 'ylabel()'. In Figura 2.13 viene mostrato il grafico ottenuto rappresentante il segnale ECG.

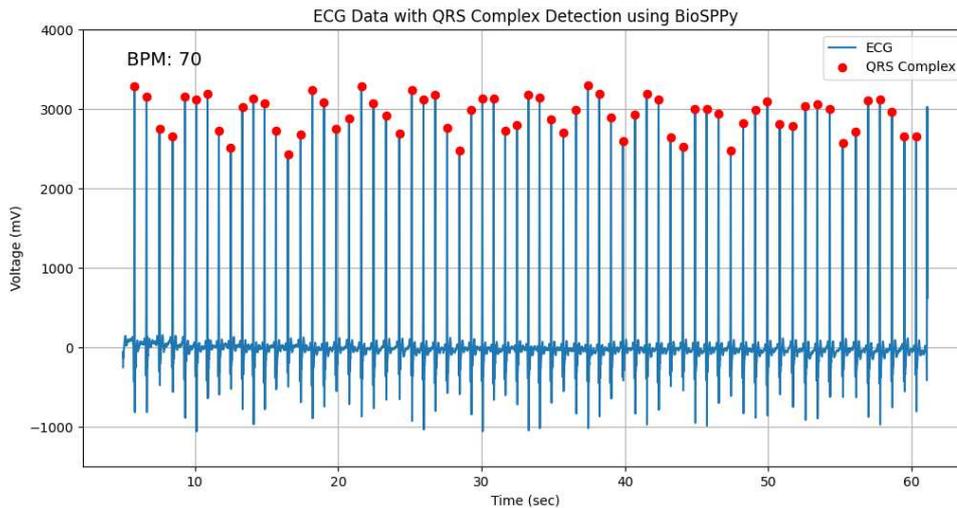


Figura 2.13: grafico del segnale

#### 2.4.10 Segnale accelerometrico estratto dal PolarH10

Tra tutti gli assi cartesiani, l'asse  $Z$  riveste un'importanza particolare nel contesto del nostro studio. Come evidenziato nella Figura 2.2, questo asse rappresenta l'accelerazione frontale del torso del soggetto. Pertanto, l'asse  $Z$  costituisce un parametro fondamentale per estrarre il segnale respiratorio effettivo della persona. Analogamente al segnale respiratorio ottenuto dall'ECG, come descritto nella Sezione 2.4.8, il segnale dall'asse  $Z$  sarà utilizzato per confrontare i dati e valutare l'affidabilità del segnale respiratorio estratto dalla telecamera Intel RealSense. L'analisi dell'asse  $Z$  inizia attraverso la lettura dei due file `'polar_acc_data.npy'`, `'polar_acc_time.npy'` rispettivamente per i dati dell'accelerazione e gli istanti di campionamento delle stesse. I dati estratti contengono i valori delle accelerazioni su tutti e 3 gli assi cartesiani, di conseguenza viene definita una lista contenente solo i valori riguardanti l'asse  $Z$  convertiti in  $m/s^2$ . Si procede con il calcolo della frequenza di campionamento attraverso il rapporto tra il numero di campioni ed il tempo di campionamento. Per quanto riguarda l'accelerazione abbiamo deciso di mostrare i dati attraverso 3 grafici: dominio del tempo, dominio del tempo con filtraggio e dominio della frequenza. Attraverso la funzione `'plot()'` della libreria `'matplotlib'` viene mostrato a schermo il grafico nel dominio del tempo, considerando i valori dell'asse  $Z$  e i valori degli istanti di tempo. Come possiamo vedere dalla Figura 2.14 è possibile distinguere a primo impatto le oscillazioni del segnale, ognuna delle quali rappresenta un respiro. Nonostante ciò il segnale risulta comunque rumoroso, quindi attraverso il package `'signal'` della libreria `'scipy'` viene applicato un filtro passa banda con due frequenze di taglio, a 0.1 Hz e 0.4 Hz. Con questa procedura il segnale avrà un andamento sinusoidale chiaro senza il rumore sovrapposto ed i respiri saranno maggiormente distinguibili (Figura 2.15). Infine, attraverso il metodo `'fft()'` definito dalla libreria `'scipy'` è vie-

ne effettuata la trasformata di fourier del segnale relativo all'asse Z, applicando una maschera tra la frequenza 0.1 Hz ed 1 Hz in modo da eliminare tutte le frequenze esterne all'intervallo che non comprenderebbero la frequenza respiratoria (Figura 2.16). È importante condiderare che l'accelerazione individuata sull'asse Z risulta leggermente amplificata. Nel caso statico ideale infatti il valore dell'accelerazione dovrebbe risultare minore dell'ordine dei  $\text{cm/s}^2$  rilevando solamente il movimento del petto causato dal respiro. Nel caso statico reale, il movimento del petto causato dal respiro viene bilanciato dal movimento del corpo, causando quindi una leggera amplificazione del segnale, che porta il valore dell'accelerazione nell'ordine dei  $\text{m/s}^2$ .

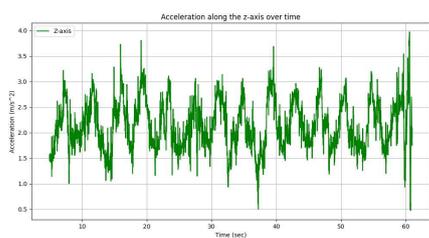


Figura 2.14: asse Z - grafico nel tempo

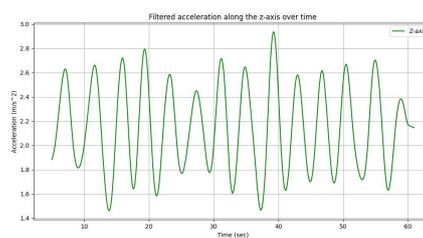


Figura 2.15: asse Z - grafico filtrato

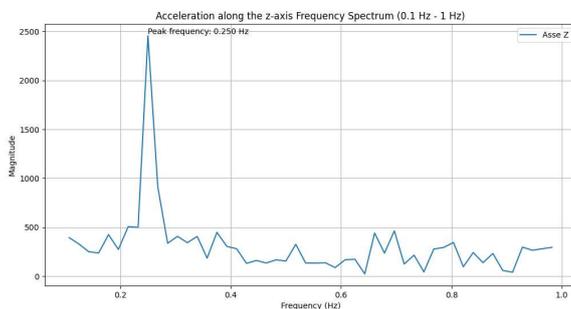


Figura 2.16: asse Z - grafico in frequenza

## 2.5 Elaborazione dei dati nel caso statico

In questa sezione verranno illustrati gli algoritmi per la definizione del segnale respiratorio nel caso statico. Si farà riferimento solo alle 4 liste contenenti le distanze rispettivamente del singolo punto del petto (2.4.2), poligono dell'addome (2.4.3), poligono del petto (2.4.4) e poligono del torso (2.4.5).

### 2.5.1 Segnale respiratorio estratto dalla telecamera

È stato creato lo stesso script in python per graficare tutti e 4 i file ottenuti dal processamento degli algoritmi YOLO. Per semplicità inizialmente vengono letti solamente 2 file: un file numpy contenente le distanze processate da un algoritmo YOLO, e il file contenente gli istanti di tempo di campionamento. Per ogni lista di distanze verranno elaborati 3 grafici: un grafico nel tempo, uno nel tempo con segnale filtrato ed uno nel dominio della frequenza. Il grafico nel tempo viene creato usando la funzione 'plot()' della libreria matplotlib, passando come argomenti la lista delle distanze dell'algoritmo YOLO di riferimento e la lista degli istanti di tempo. A seguire viene creato il grafico nel tempo filtrando le frequenze con un filtro passa banda che ha frequenze di taglio a 0.1 Hz e 0.8 Hz. In questo modo verrà visualizzato il segnale filtrato con un andamento sinusoidale, distinguendo i respiri più chiaramente. L'ultimo grafico riguarda la rappresentazione del segnale nel dominio della frequenza, realizzato attraverso la trasformata di fourier ed applicando una maschera su di esso per selezionare solamente le frequenze da 0.1 Hz e 1 Hz, essendo molto improbabile che un soggetto abbia una frequenza di respirazione al di fuori di questo range. Come già sottolineato, questa procedura deve essere eseguita per tutte e 4 le liste di distanze ottenute dagli algoritmi YOLO, in modo da confrontare i diversi segnali respiratori ricavati in base al tipo di algoritmo per capire quale è il più attendibile. In Figura 2.17, 2.18 e 2.19 sono riportati rispettivamente il grafico nel tempo, nel tempo filtrato e in frequenza di un segnale relativo al movimento del 'singolo punto' (Sezione 2.4.4).

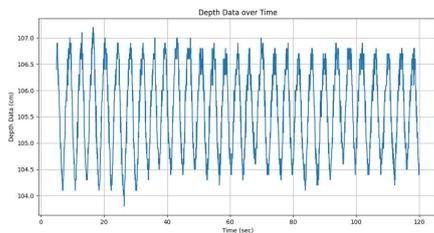


Figura 2.17: singolo punto - grafico nel tempo

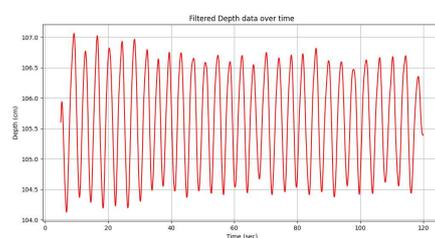


Figura 2.18: singolo punto - grafico filtrato

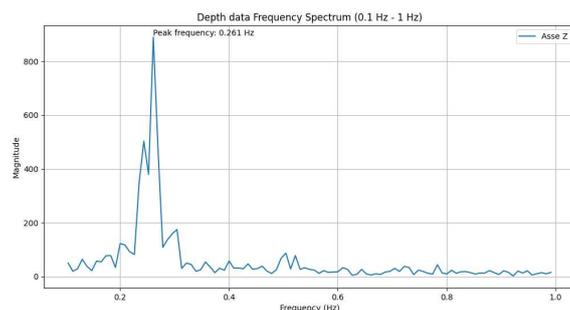


Figura 2.19: singolo punto - grafico in frequenza

## 2.6 Elaborazione dei dati caso dinamico

In questa sezione verranno spiegati gli algoritmi implementati per la determinazione del segnale respiratorio nel caso dinamico. Sarà quindi necessario fare utilizzo di tutte e sei le liste ricavate dall'algoritmo YOLO.

### 2.6.1 Segnale respiratorio rispetto al centro tra spalle

In questo algoritmo utilizziamo il centro tra le due spalle come punto di riferimento statico rispetto all'andamento respiratorio durante una situazione dinamica. Idealmente possiamo pensare il punto centrale tra le due spalle come un punto che è soggetto solo al movimento del corpo durante la camminata, ma non è soggetto al movimento del respiro. Questa considerazione è molto utile perché sottraendo le distanze del punto centrale ad una delle 4 liste di distanze (singolo punto del petto, poligono addome, poligono petto, poligono torso), ciò che ottengo è un movimento oscillatorio che rappresenterà l'andamento del respiro, senza il movimento oscillatorio della camminata del soggetto. L'algoritmo comincia con la lettura dei tre file: il file delle distanze dell'algoritmo YOLO, il file delle distanze del punto tra spalle ed il file degli istanti temporali. Facendo la sottrazione tra i 2 numpy array, uno delle distanze dello YOLO e l'altro contenente le distanze del punto centrale della spalla otterremo un array che rappresenterà il segnale respiratorio. Per questo segnale vengono realizzati i grafici nel tempo, nel tempo filtrato ed in frequenza come spiegato nella sotto sezione 2.5.1. In questo caso viene anche mostrato il grafico del segnale preso dall'array delle distanze dell'algoritmo YOLO senza aver sottratto l'array delle distanze del punto tra spalle, in modo da osservare delle oscillazioni che rappresentano il movimento del soggetto nel dominio del tempo (Figura 2.20). Ricordiamo che questo algoritmo va applicato per tutte e 4 le tipologie di processamenti YOLO.

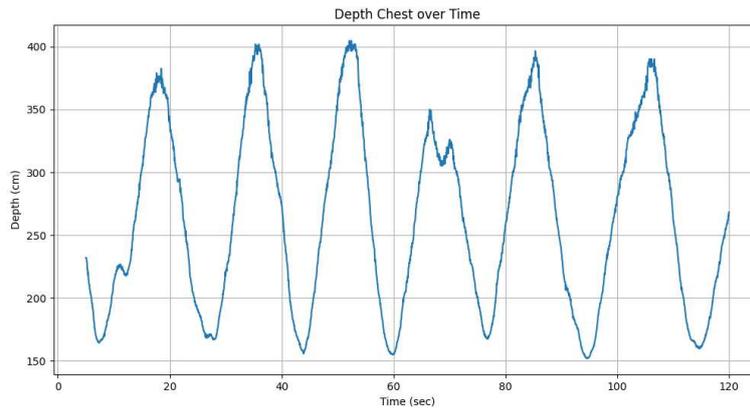


Figura 2.20: grafico del movimento

## 2.6.2 Segnale respiratorio rispetto al centro del collo

Questo algoritmo ha lo stesso scopo dell'algoritmo visto nella sezione 2.6.1, ovvero ricavare il segnale respiratorio in movimento trovando un punto che non è influenzato dal respiro. In questo caso abbiamo deciso di utilizzare un punto centrale del collo, con distanze calcolate come spiegato nella Sezione 2.4.7. L'algoritmo comincia leggendo i file relativi alle distanze di uno degli algoritmi YOLO, il file delle distanze del punto del collo ed il file con gli istanti di tempo. Di conseguenza si ottengono i 3 numpy array 'depth\_data', 'depth\_collo', 'timestamps' e si sottraggono 'depth\_data' e 'depth\_collo' ottenendo un nuovo numpy array che rappresenta il segnale respiratorio. Utilizzando sempre gli stessi algoritmi di costruzione dei grafici, come già spiegato nella sezione 2.6.1, vengono costruiti i grafici nel tempo del segnale rappresentativo del movimento, segnale respiratorio nel tempo, segnale respiratorio nel tempo filtrato con frequenze di taglio pari a 0.1 Hz e 0.8 Hz, segnale respiratorio in frequenza applicando il concetto di trasformata di fourier.

## 2.6.3 Segnale respiratorio rispetto al poligono dell'addome

L'ultimo algoritmo riguardante l'estrazione del segnale respiratorio in situazione di movimento utilizza la differenza tra l'array di distanze del singolo punto, poligono del petto o poligono del torso e l'array di distanze relativo a poligono dell'addome. Questo passaggio è stato fatto considerando che i punti dell'addome si muovono molto meno rispetto ai punti del torso, di conseguenza possiamo sottrarli per ricavare il segnale respiratorio.

Dal punto di vista pratico è stato effettuato sempre lo stesso passaggio di lettura dei tre file, ricavando i tre array 'depth\_data', 'depth\_addome', 'timestamps', e facendo la differenza tra 'depth\_data' e 'depth\_addome' ricavando quindi l'array

contenente i valori del segnale respiratorio. In seguito, attraverso le procedure già illustrate, verranno implementati il grafico del movimento nel dominio del tempo, il grafico del segnale respiratorio nel dominio del tempo, nel dominio del tempo con filtraggio e nel dominio della frequenza.



# Capitolo 3

## Risultati e discussione

In questo capitolo verranno analizzati e spiegati i dati raccolti dalle acquisizioni effettuate. Saranno introdotte le sezioni dedicate alle condizioni statiche e dinamiche. Nelle condizioni statiche, i soggetti sono stati posizionati sia in posizione seduta frontale che in piedi con una rotazione rispetto alla telecamera in modo da nascondere parzialmente il corpo. Per quanto riguarda le condizioni dinamiche, verranno esaminati i dati relativi a un soggetto in movimento frontale, a soggetti che compiono una leggera rotazione del torso, e infine a soggetti che eseguono movimenti casuali all'interno dell'ambiente inquadrato dalla telecamera Intel. È importante considerare che le diverse situazioni sperimentali sono caratterizzate da un database specifico e da condizioni di distanza variabili in ciascuna acquisizione.

### 3.1 Confronto dei dati in condizioni statiche

In questa sezione verranno analizzati i dati ricavati dalle condizioni statiche. I soggetti sono stati posti in 2 posizioni: seduti frontalmente e in piedi ruotati di un angolo minore di  $45^\circ$  rispetto alla telecamera.

I valori presenti su ogni tabella fanno riferimento al proprio attributo definito nella prima colonna. Nella prima riga, relativamente all'attributo 'BPM' sono presenti il numero di battiti al minuto registrati per quel soggetto. Il resto dei valori sono tutte frequenze, quindi l'unità di misura è Hz. Le ultime 4 righe rappresentano le frequenze dei segnali respiratori estratti con gli algoritmi descritti nelle sezioni 2.4.2, 2.4.3, 2.4.4, 2.4.5.

Di seguito è riportata la legenda di ogni tabella:

- BPM: battiti per minuto.
- Numero Respiri: numero di respiri effettuate per tutta la durata dell'acquisizione.
- Respiri ECG: frequenza del segnale respiratorio estratto dai valori dell'ECG.

- Respiri Asse Z: frequenza del segnale respiratorio estratto dai valori dell'asse z.
- singolo punto: frequenza del segnale respiratorio relativo al singolo punto del petto.
- poligono addome: frequenza del segnale respiratorio relativo al poligono dell'addome.
- poligono petto: frequenza del segnale respiratorio relativo al poligono del petto.
- poligono torso: frequenza del segnale respiratorio relativo al poligono del torso.

Come parametro per stabilire le performance è stato considerato il 'mean absolute error': una misura di accuratezza utilizzata per valutare le previsioni di un modello rispetto ai valori osservati. In altre parole, misura quanto in media le previsioni si discostano dai valori reali. è stato calcolato usando la seguente formula:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.1)$$

### 3.1.1 Soggetto seduto

I dati analizzati in questa sezione riguardano 3 soggetti che sono stati posti in posizione seduta rivolti frontalmente verso la telecamera (la posizione è illustrata in Figura 3.1). Sono state effettuate 3 acquisizioni per soggetto, ognuna ad una distanza rispettivamente di 1 metro, 2 metri, 3 metri. Tutte le acquisizioni sono state fatte per un tempo di 120 secondi.

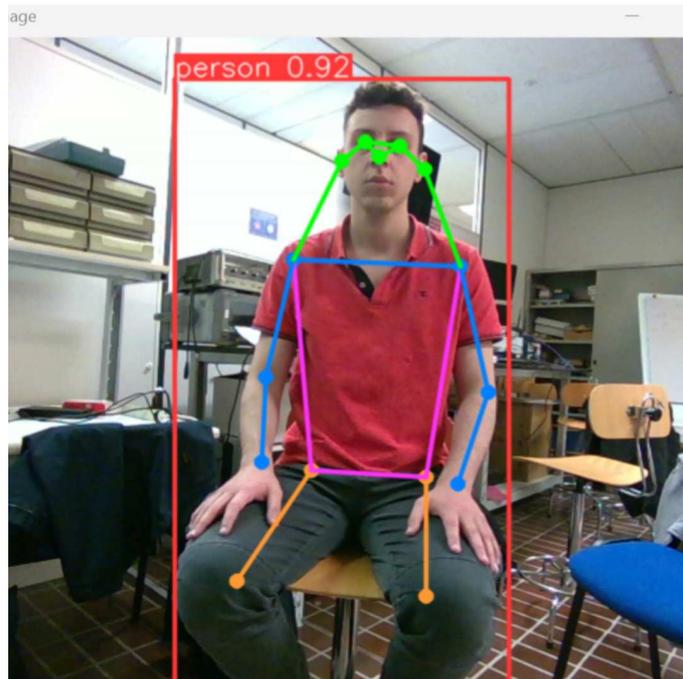


Figura 3.1: posizione seduto frontale

### Soggetto 1

<b>Distanza</b>	<b>1 metro</b>	<b>2 metri</b>	<b>3 metri</b>
BPM	96	89	85
Numero Respiri	24	24	25
Respiri ECG (Hz)	0.208	0.207	0.215
Respiri Asse Z (Hz)	0.207	0.207	0.215
singolo punto (Hz)	0.209	0.209	0.209
poligono addome (Hz)	0.209	0.418	0.209
poligono petto (Hz)	0.209	0.209	0.209
poligono torso (Hz)	0.209	0.200	0.209

Le frequenze dei respiri e le frequenze dell'asse z coincidono per ogni distanza misurata. Osserviamo come gli algoritmi 'singolo punto', 'poligono petto' e 'poligono torso' riportano frequenze più vicine al risultato reale. Con il secondo algoritmo 'poligono addome' la frequenza catturata a 2 metri di distanza risulta doppia rispetto al valore reale. In tutti e 4 gli algoritmi è presente un errore tra 0.001 - 0.007 Hz

### Soggetto 2

<b>Distanza</b>	<b>1 metro</b>	<b>2 metri</b>	<b>3 metri</b>
BPM	73	73	91
Numero Respiri	17	16	25
Respiri ECG (Hz)	0.146	0.138	0.216
Respiri Asse Z (Hz)	0.146	0.138	0.216
singolo punto (Hz)	0.148	0.130	0.217
poligono addome (Hz)	0.148	0.130	0.217
poligono petto (Hz)	0.148	0.130	0.217
poligono torso (Hz)	0.148	0.130	0.217

In questo caso tutti e 4 gli algoritmi riportano la frequenza respiratoria corretta per ogni distanza, con un leggero errore tra 0.001 - 0.008 Hz.

### Soggetto 3

<b>Distanza</b>	<b>1 metro</b>	<b>2 metri</b>	<b>3 metri</b>
BPM	66	56	60
Numero Respiri	30	22	21
Respiri ECG (Hz)	0.260	0.190	0.182
Respiri Asse Z (Hz)	0.259	0.190	0.182
singolo punto (Hz)	0.261	0.191	0.183
poligono addome (Hz)	0.244	0.191	0.183
poligono petto (Hz)	0.261	0.191	0.183
poligono torso (Hz)	0.261	0.191	0.183

Anche per il terzo soggetto le frequenze risultano attendibili, riscontrando sempre un errore tra 0.001 - 0.014 Hz, risultando massimo con la differenza tra 'Respiri ECG' e 'poligono addome' ad 1 metro di distanza. La tabella 3.1 riporta i valori del parametro 'mean absolute error' per ogni algoritmo relativamente ad ogni distanza. Come possiamo osservare tutti i valori risultano relativamente bassi, rendendo il segnale respiratorio, estratto da ogni algoritmo, affidabile in queste condizioni.

<b>Distanza</b>	<b>1 metro</b>	<b>2 metri</b>	<b>3 metri</b>
singolo punto (Hz)	0.001	0.004	0.003
poligono addome (Hz)	0.006	0.07	0.003
poligono petto (Hz)	0.001	0.004	0.003
poligono torso (Hz)	0.001	0.005	0.003

Tabella 3.1: MAE - posizione seduta frontale

#### 3.1.2 Soggetto ruotato

In questa sezione verranno analizzati i dati di 5 soggetti in piedi ruotati di un angolo minore di 45° rispetto alla destra o sinistra della telecamera (la posizione è illustrata nelle Figure 3.2, 3.3). La distanza di ogni acquisizione è di circa 1.8 metri. Tutte le acquisizioni sono state fatte per un tempo di 120 secondi.

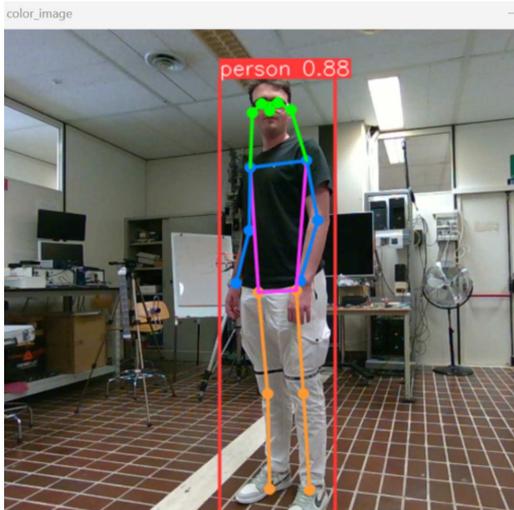


Figura 3.2: posizione ruotato alla sinistra della telecamera

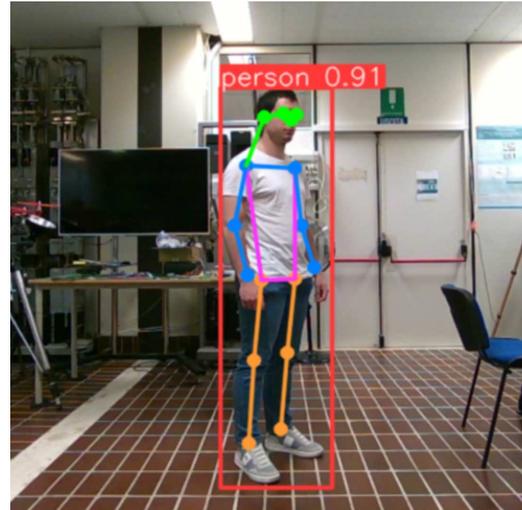


Figura 3.3: posizione ruotato alla destra della telecamera

## Dati

Soggetto	Soggetto 1	Soggetto 2	Soggetto 3	Soggetto 4	Soggetto 5
BPM	100	86	79	109	59
Numero Respiri	25	18	30	34	15
Respiri ECG (Hz)	0.216	0.156	0.260	0.294	0.129
Respiri Asse Z (Hz)	0.216	0.156	0.259	0.294	0.138
singolo punto (Hz)	0.217	0.157	0.261	0.244	0.131
poligono addome (Hz)	0.217	0.157	0.261	0.200	0.131
poligono petto (Hz)	0.217	0.157	0.261	0.113	0.131
poligono torso (Hz)	0.217	0.157	0.270	0.139	0.104

Anche nel caso statico con rotazione in generale tutti gli algoritmi sembrano piuttosto attendibili. Facendo delle osservazioni più nel dettaglio vediamo che tutti gli algoritmi hanno riportato il valore della frequenza correttamente per tutti i soggetti tranne per il soggetto 4. Le Figure 3.4, 3.5, 3.6, 3.7 mostrano i grafici associati al soggetto 4 rispettivamente per gli algoritmi di elaborazione del segnale respiratorio catturato con la telecamera. In tutti gli algoritmi è presente una frequenza attorno al valore 0.294 Hz, ma non è quella con l'ampiezza maggiore. Ogni algoritmo ha una frequenza molto diversa dalle altre e le frequenze della respirazione catturate con il PolarH10 coincidono. Questo significa che tutti e 4 gli algoritmi riportano frequenze basse molto influenti, dovute probabilmente da artefatti generati dai vari keypoints che vengono calcolati male in alcuni frame e di conseguenza l'andamento del segnale risulterà distorto.

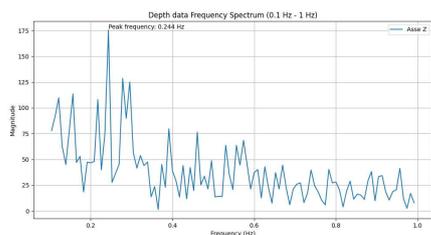


Figura 3.4: Soggetto 4 - singolo punto frequenza

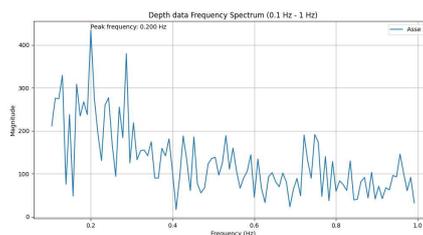


Figura 3.5: Soggetto 4 - poligono addome frequenza

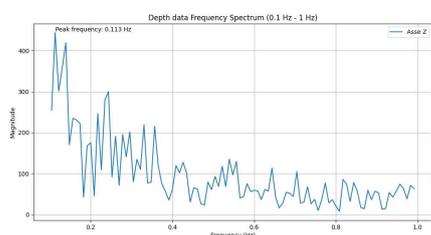


Figura 3.6: Soggetto 4 - poligono petto frequenza

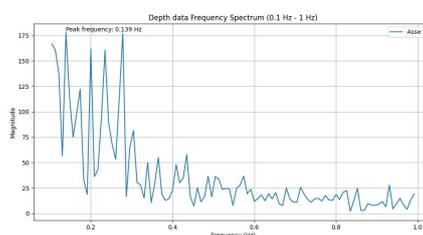


Figura 3.7: Soggetto 4 - poligono torso frequenza

Possiamo concludere quindi che in generale tutti gli algoritmi hanno riportato valori di frequenza attendibili per 4 soggetti su 5, quindi con un successo dell'80%. La tabella 3.2 mostra il 'mean absolute value' relativo ad ogni soggetto testato, il quale ci porta a concludere che anche in questo caso si ha un errore di massimo 0.04 Hz, tra le due frequenze respiratorie (quella reale calcolata con il polarH10 e quella predetta) praticamente coincidenti.

Algoritmo	MAE
singolo punto (Hz)	0.011
poligono addome (Hz)	0.02
poligono petto (Hz)	0.04
poligono torso (Hz)	0.04

Tabella 3.2: MAE - posizione in piedi ruotato

## 3.2 Confronto dei dati in condizioni dinamiche

Si ricorda che per ricavare il segnale respiratorio non è sufficiente leggere uno dei segnali catturati con gli algoritmi definiti nelle sezioni 2.4.2, 2.4.3, 2.4.4, 2.4.5, poiché in tal caso si otterrebbe il grafico del movimento complessivo, come mostrato in Figura 2.20. È importante notare che il movimento registrato presenta piccole

oscillazioni sulle creste dei picchi. Queste oscillazioni indicano che il segnale rappresentato include sia il movimento generale del corpo che il segnale respiratorio sovrapposto. Per estrarre esclusivamente il segnale respiratorio, è necessario considerare punti del corpo del soggetto la cui distanza dalla telecamera è influenzata solamente dal movimento e non dalla respirazione. Sottraendo la distanza di uno di questi punti dalla distanza di un punto influenzato anche dal respiro, si ottengono oscillazioni che rappresentano in modo isolato il segnale respiratorio. Questo metodo permette di separare il segnale del respiro dal movimento generale, fornendo una misurazione più accurata della respirazione del soggetto.

Come punti non influenzati dal movimento abbiamo considerato il punto centrale tra le due spalle, il punto centrale del collo e il poligono dell'addome, spiegati nelle sezioni 2.4.6, 2.4.7, 2.4.3. In seguito questi punti verranno sottratti dal segnale del movimento con il respiro sommato.

La sezione seguente analizzerà i dati ricavati dalle acquisizioni in condizioni dinamiche. I soggetti eseguiranno 3 movimenti diversi: movimento con camminata frontale, movimento con rotazione del torso e movimento libero nello spazio inquadrato dalla telecamera.

L'indice degli attributi è stato posto nella prima colonna e tutti i valori riportati, tranne quelli relativi al BPM e Numero Respiri, sono misurati in Hz, essendo delle frequenze.

Di seguito la legenda delle tabelle:

- BPM: battiti per minuto.
- Numero Respiri: numero di respiri effettuate per tutta la durata dell'acquisizione.
- Respiri ECG: frequenza del segnale respiratorio estratto dai valori dell'ECG.
- Respiri Asse Z: frequenza del segnale respiratorio estratto dai valori dell'asse z.
- singolo punto: frequenza del segnale respiratorio relativa al singolo punto del petto
- poligono addome: frequenza del segnale respiratorio relativa al poligono dell'addome.
- poligono petto: frequenza del segnale respiratorio relativa al poligono del petto.
- poligono torso: frequenza del segnale respiratorio relativa al poligono del torso.
- centro spalle: frequenza del segnale relativo al punto centrale tra le due spalle.
- centro collo: frequenza del segnale relativo al punto centrale del collo.

### 3.2.1 Soggetto in movimento frontale

In questa sezione vengono analizzati i dati ricavati da un soggetto che compie una camminata frontale rispetto alla telecamera, facendo due respiri nell'intervallo di tempo in cui va avanti e torna indietro nella posizione iniziale. L'acquisizione è stata fatta per un tempo di 120 secondi. In Figura 3.8 viene mostrata la camminata.

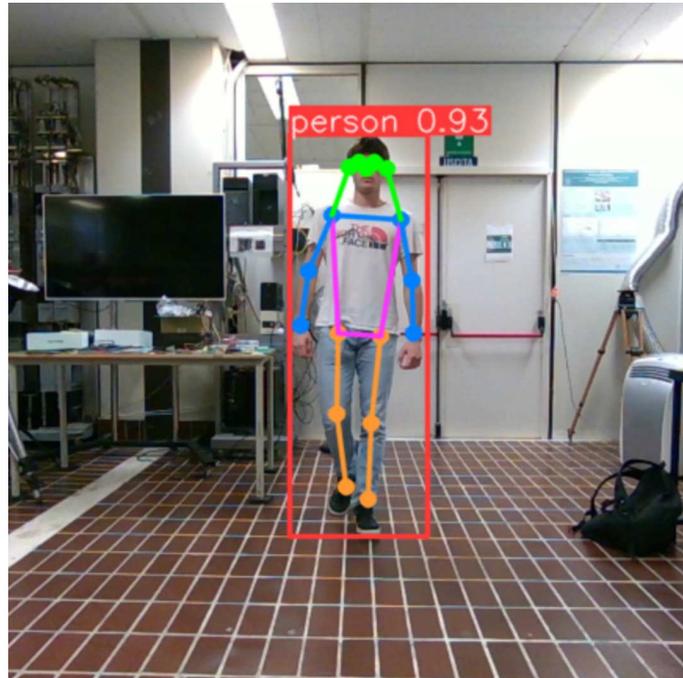


Figura 3.8: camminata frontale

#### Dati

BPM: 72

Numero Respiri: 32

Respiri ECG: 0.277

Respiri Asse Z: 0.277

Algoritmo	centro spalle	centro collo	poligono addome
singolo punto	0.200	0.278	0.278
poligono addome	0.278	0.139	/
poligono petto	0.200	0.278	0.278
poligono torso	0.278	0.139	0.278

Questa acquisizione è stata svolta come prova per verificare il funzionamento degli algoritmi, avendo effettuato la sperimentazione su un soggetto solo. Possiamo notare che le frequenze respiratorie estratte presenti nella prima colonna, sono riportate correttamente solamente per 'poligono addome' e 'poligono torso', mentre nella seconda colonna le frequenze risultano corrette solo per gli algoritmi 'singolo punto' e 'poligono petto'.

I grafici in Figura 3.9 e Figura 3.10 mostrano rispettivamente i risultati degli algoritmi 'singolo punto' e 'poligono addome' applicati al punto centrale delle spalle nel dominio della frequenza. Nel primo grafico, relativo all'algoritmo 'singolo punto', si osservano numerose frequenze con la stessa ampiezza, indicando che l'acquisizione è affetta da rumore. Nonostante la presenza di un picco a circa 0.277 Hz, esso non si distingue chiaramente dagli altri, rendendo difficile l'identificazione della frequenza respiratoria. Al contrario, nel secondo grafico, relativo all'algoritmo 'poligono addome', emerge una sola frequenza predominante che corrisponde alla frequenza di respirazione. Questo confronto evidenzia come, nel primo grafico, l'algoritmo 'singolo punto', anche sottraendo il punto centrale delle spalle, non riesca a eliminare il rumore sovrapposto al segnale. Invece, nel secondo grafico, l'algoritmo 'poligono addome' riesce a isolare correttamente il segnale respiratorio, dimostrando una maggiore efficacia nel filtrare il rumore.

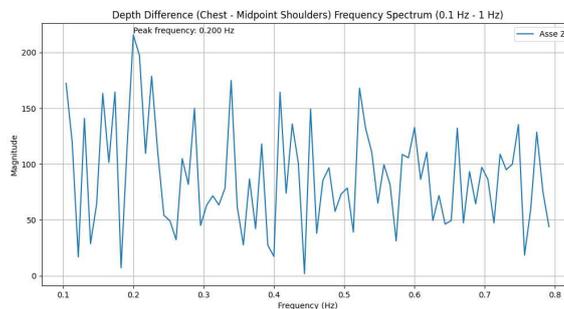


Figura 3.9: Dinamico - singolo punto rispetto a centro spalle

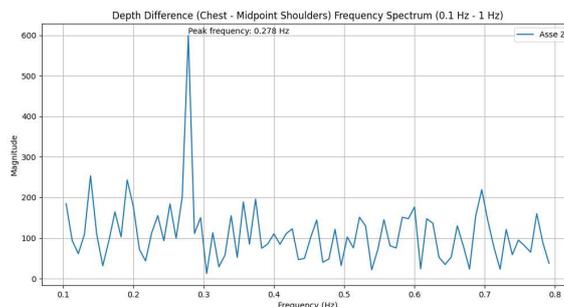


Figura 3.10: Dinamico - poligono addome rispetto a centro spalle

D'altro canto, i grafici mostrati in Figura 3.11 e Figura 3.12 illustrano rispettivamente i risultati degli algoritmi 'singolo punto' e 'poligono addome' applicati al punto centrale del collo nel dominio della frequenza. Anche in questo caso, il primo grafico, relativo all'algoritmo 'singolo punto', mostra la frequenza respiratoria corretta. Tuttavia, nel secondo grafico, relativo all'algoritmo 'poligono addome', viene visualizzata una frequenza pari alla metà della frequenza respiratoria corretta. Considerando che il soggetto ha compiuto un movimento frontale respirando due volte per ogni ciclo di spostamento avanti e indietro, si può concludere che la frequenza rilevata nel secondo grafico corrisponde alla frequenza del movimento. Questo indica che, in questo caso, l'algoritmo 'poligono addome' non riesce a estrarre correttamente il segnale respiratorio, confondendo la frequenza respiratoria con quella del movimento.

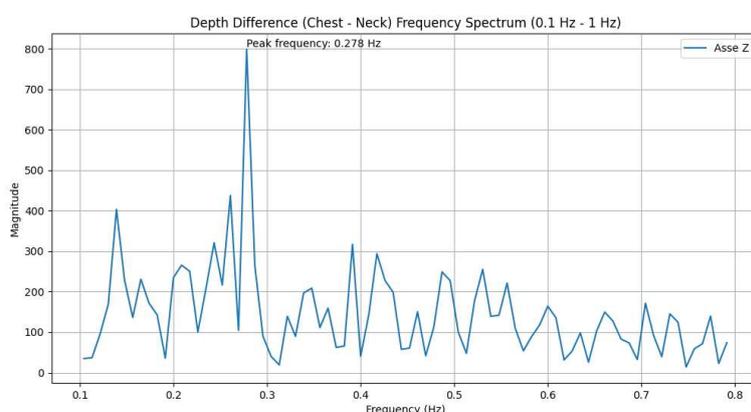


Figura 3.11: Dinamico - singolo punto rispetto a centro collo

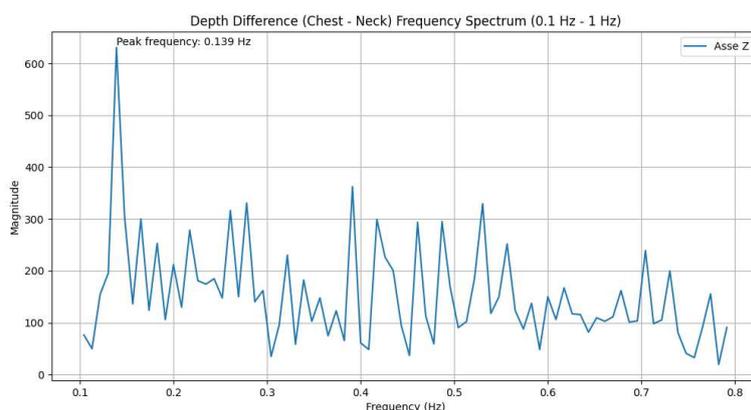


Figura 3.12: Dinamico - poligono addome rispetto a centro collo

L'ultimo caso analizza l'estrazione del segnale respiratorio confrontando i vari algoritmi con il segnale prodotto da 'poligono addome' sottratto ad essi. Ovviamente, non è stato considerato il caso in cui 'poligono addome' viene sottratto da se stesso, poiché il segnale risultante sarebbe nullo. La scelta di questo approccio è motivata dal fatto che, durante la respirazione, i punti appartenenti al quadrante dell'addome tendono a muoversi di meno. I grafici relativi ai tre algoritmi sono mostrati nelle Figure 3.13, 3.14 e 3.15. In tutti e tre i casi, emerge una sola frequenza prominente rispetto alle altre, che coincide con la frequenza respiratoria definita dal PolarH10. Questo risultato indica che il segnale 'poligono addome' sottratto ai segnali dei tre algoritmi consente di ricavare correttamente il segnale respiratorio per ciascuno di essi.

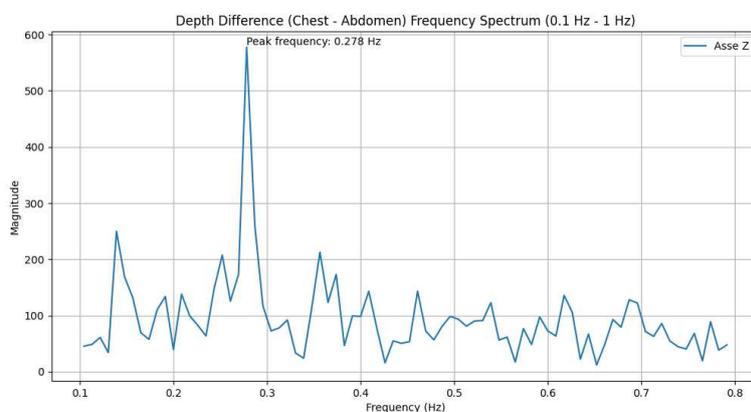


Figura 3.13: Dinamico - singolo punto rispetto a poligono addome

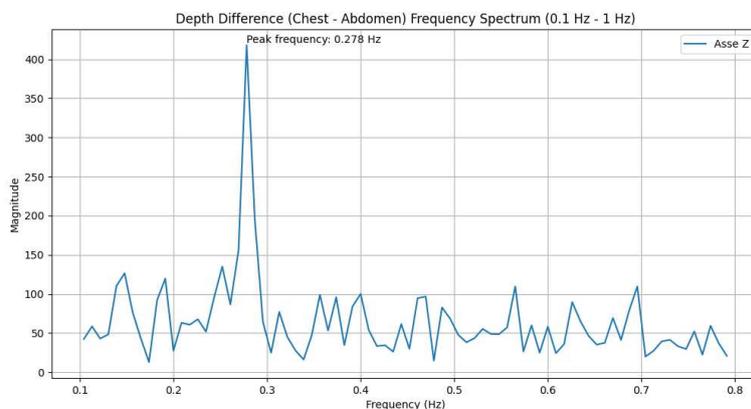


Figura 3.14: Dinamico - poligono petto rispetto a poligono addome

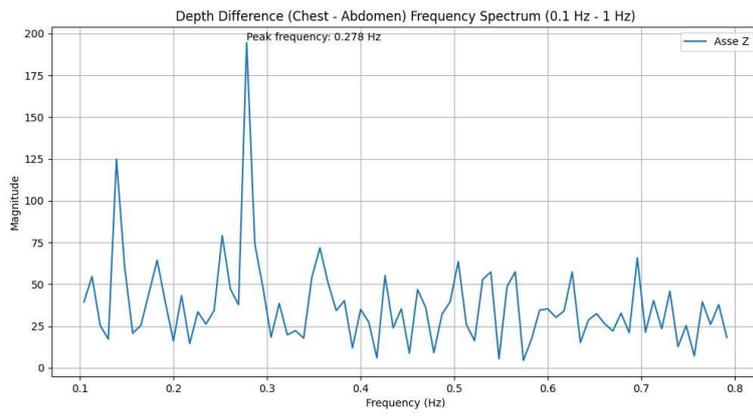


Figura 3.15: Dinamico - poligono torso rispetto a poligono addome

### 3.2.2 Soggetto con rotazione del torso

In questa sezione sono analizzati i dati di 4 soggetti che compiono un movimento rotazionale con il busto, senza effettuare alcuna camminata. Verranno mostrate le tabelle relative ad ogni soggetto e i risultati delle frequenze relative ad ogni algoritmo. Le acquisizioni sono state fatte tutte per un tempo di 120 secondi. La Figura 3.16 mostra il movimento compiuto.

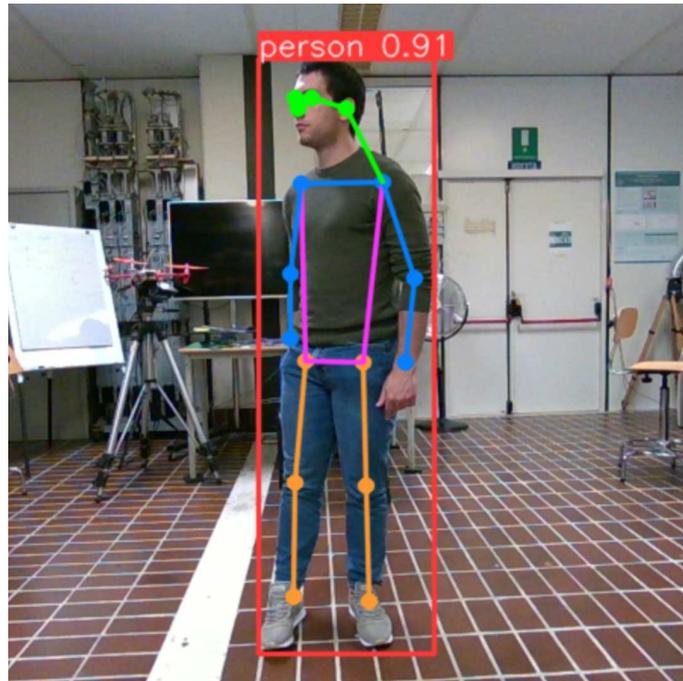


Figura 3.16: rotazione torso

#### Soggetto 1

BPM: 84

Numero Respiri: 39

Respiri ECG: 0.338

Respiri Asse Z: 0.328

Algoritmo	centro spalle	centro collo	poligono addome
singolo punto	0.165	0.157	0.165
poligono addome	0.322	0.487	/
poligono petto	0.165	0.165	0.331
poligono torso	0.165	0.165	0.165

## Soggetto 2

BPM: 106

Numero Respiri: 29

Respiri ECG: 0.250

Respiri Asse Z: 0.250

<b>Algoritmo</b>	<b>centro spalle</b>	<b>centro collo</b>	<b>poligono addome</b>
singolo punto	0.157	0.244	0.244
poligono addome	0.244	0.139	/
poligono petto	0.139	0.244	0.244
poligono torso	0.252	0.139	0.244

## Soggetto 3

BPM: 74

Numero Respiri: 17

Respiri ECG: 0.146

Respiri Asse Z: 0.146

<b>Algoritmo</b>	<b>centro spalle</b>	<b>centro collo</b>	<b>poligono addome</b>
singolo punto	0.139	0.139	0.148
poligono addome	0.139	0.139	/
poligono petto	0.122	0.122	0.139
poligono torso	0.139	0.139	0.139

## Soggetto 4

BPM: 94

Numero Respiri: 30

Respiri ECG: 0.260

Respiri Asse Z: 0.259

<b>Algoritmo</b>	<b>centro spalle</b>	<b>centro collo</b>	<b>poligono addome</b>
singolo punto	0.131	0.252	0.252
poligono addome	0.261	0.278	/
poligono petto	0.139	0.252	0.261
poligono torso	0.131	0.287	0.131

Nella tabella 3.3 sono riportati i valori del 'mean absolute value' per ogni algoritmo della prima colonna rispetto ai punti estratti presenti nella prima riga.

<b>Algoritmi</b>	<b>centro spalle</b>	<b>centro collo</b>	<b>poligono addome</b>
singolo punto (Hz)	0.101	0.051	0.047
poligono addome (Hz)	0.008	0.071	/
poligono petto (Hz)	0.107	0.053	0.005
poligono torso (Hz)	0.078	0.08	0.079

Tabella 3.3: MAE - rotazione torso

Osservando i parametri MAE di ogni algoritmo vediamo che nella maggior parte dei casi, l'errore risulta piuttosto elevato, avendo quindi delle frequenze predette diverse rispetto alla frequenza respiratoria reale. Solo nei casi di 'poligono addome' rispetto a 'centro spalle' e 'poligono petto' rispetto a 'poligono addome', l'errore risulta minimo, con una differenza massima di 0.2 Hz tra le due frequenze.

Analizzando i grafici possiamo concludere con diverse osservazioni. La Figura 3.17 mostra il grafico in frequenza del soggetto 3 per l'algoritmo 'singolo punto' rispetto a 'punto addome'. In questo caso c'è solo una frequenza con ampiezza maggiore rispetto alle altre che coincide con la frequenza di respirazione registrata dal sensore polar, questo vuol dire che il segnale respiratorio è stato estratto con successo, eliminando completamente il segnale del movimento.

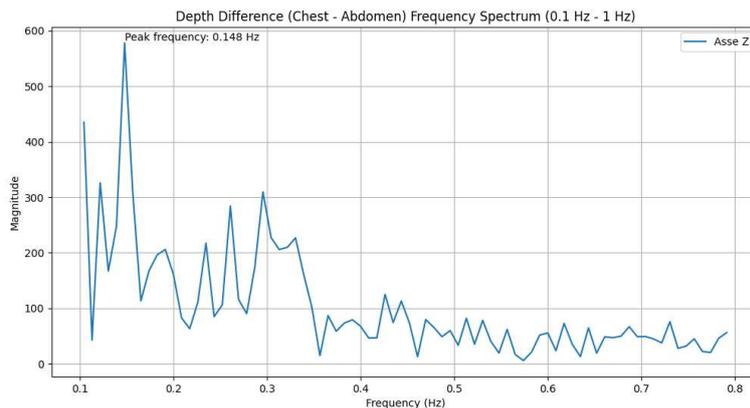


Figura 3.17: soggetto 3 - singolo punto rispetto a poligono addome

Nei grafici 3.18, 3.19 sono riportate le due frequenze relativamente al movimento del soggetto (catturato dall'algorithm 'poligono torso') e del respiro estratto dalla differenza tra 'poligono torso' e 'poligono addome'. Come possiamo notare i due grafici sono molto simili e nel secondo grafico riusciamo a distinguere una seconda frequenza attorno al valore di 0.33 Hz (frequenza del segnale respiratorio). Questo ci porta a concludere che, nonostante la differenza tra le distanze relative ai due algoritmi, il segnale del movimento non è stato eliminato completamente.

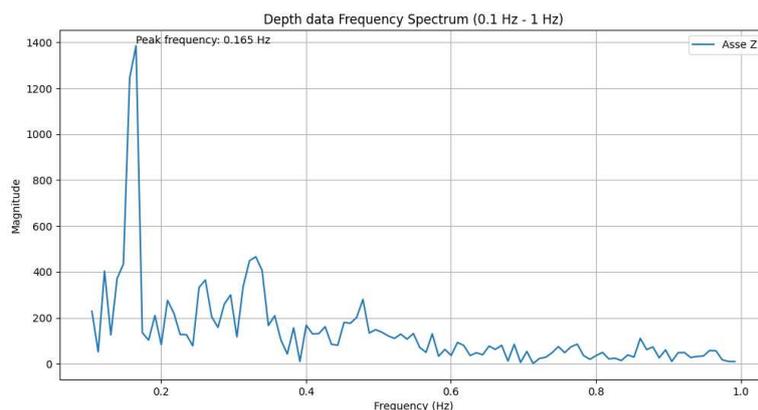


Figura 3.18: soggetto 1 - frequenza del movimento dell'algorithm poligono torso

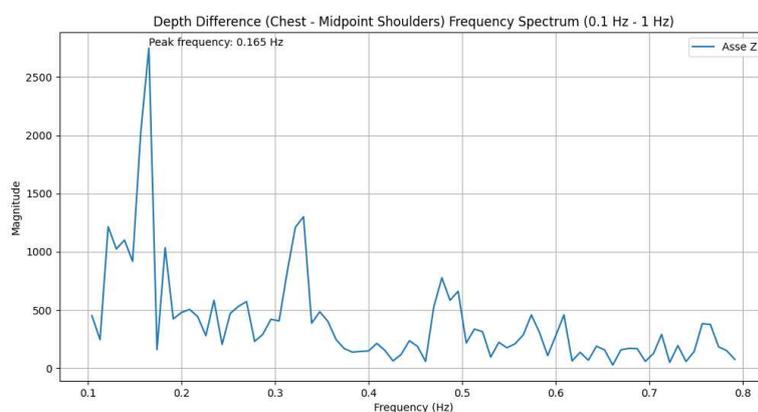


Figura 3.19: soggetto 1 - poligono torso rispetto a poligono addome

Il grafico 3.20 relativo alla differenza tra l'algorithm 'singolo punto' ed il punto centrale tra le spalle del soggetto 2 mostra un altro problema che è possibile riscontrare. In questo caso la frequenza con ampiezza maggiore risulta quella relativa al segnale del movimento (quindi anche qua la differenza non è stata efficace), ma sono presenti altre frequenze rilevanti. Questo ci porta a concludere che il segnale estratto è totalmente coperto dal rumore e non è possibile fare nessuna valutazione accurata per

quanto riguarda il segnale respiratorio. La Figura 3.21 mostra il grafico del segnale estratto dalla differenza nel dominio del tempo, e come possiamo vedere il segnale risulta completamente coperto dal rumore.

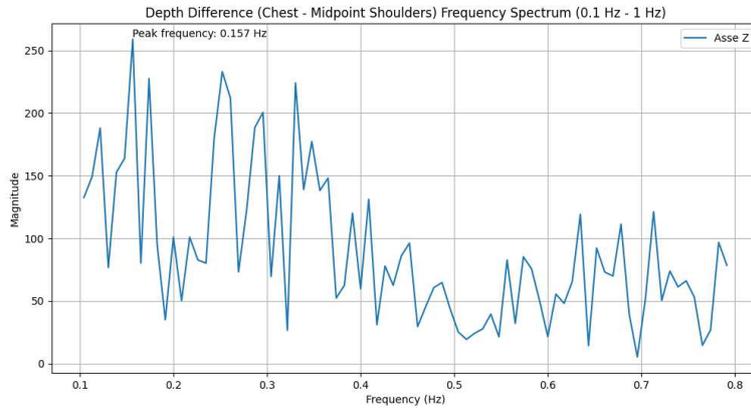


Figura 3.20: soggetto 2 - singolo punto rispetto a centro spalle

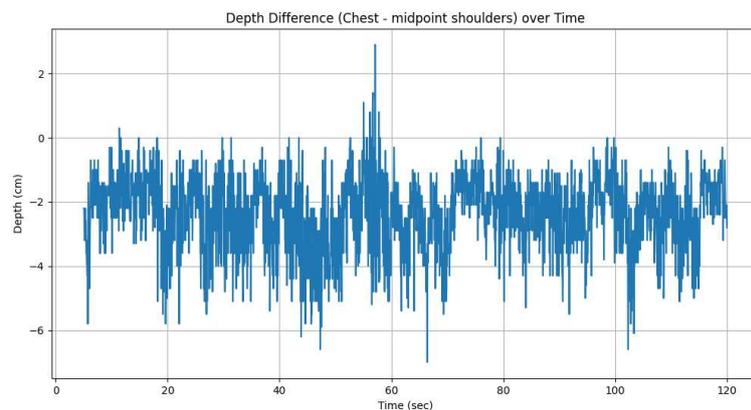


Figura 3.21: soggetto 2 - singolo punto rispetto a centro spalle nel dominio del tempo

In conclusione, possiamo affermare che gli algoritmi implementati presentano alcune difficoltà. Tra tutti, gli algoritmi che hanno ottenuto i migliori risultati sono stati 'poligono addome' rispetto a 'centro spalle' e 'poligono petto' rispetto a 'poligono addome'. Tuttavia, in alcuni casi si riscontrano imprecisioni significative, attribuibili principalmente a due problemi: la persistenza del segnale del movimento residuo dopo la differenza tra i due punti, e il rumore che rimane sovrapposto al segnale respiratorio. Questi fattori possono compromettere la precisione dell'estrazione del segnale respiratorio, evidenziando la necessità di ulteriori miglioramenti negli algoritmi per ridurre l'impatto di tali interferenze.

### 3.2.3 Soggetto movimento libero nello spazio

In questa sezione verranno analizzati i dati relativi a 4 soggetti che compiono un movimento libero nello spazio d' inquadratura della telecamera Intel Realsense. Per ogni soggetto verrà mostrata la tabella relativa degli algoritmi . Ogni acquisizione è stata fatta per un tempo di 120 secondi. La Figura 3.22 mostra il movimento compiuto da uno dei soggetti.

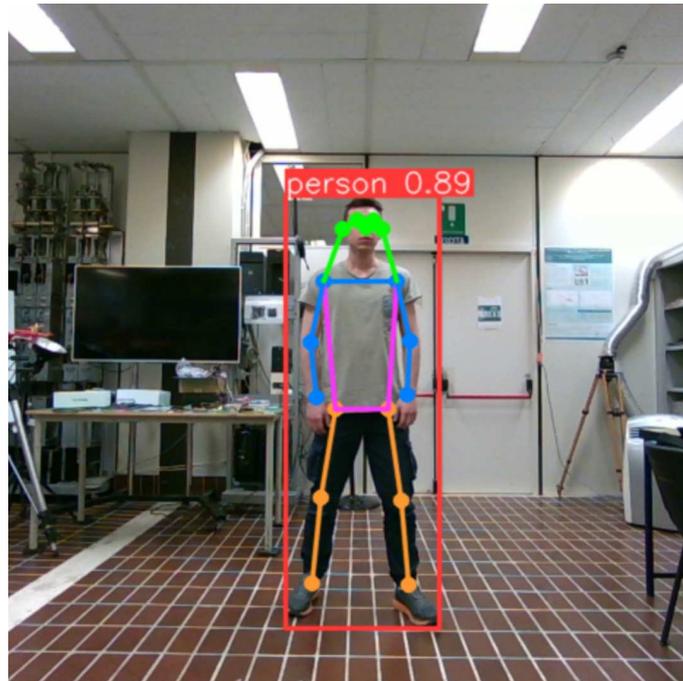


Figura 3.22: movimento libero

#### Soggetto 1

BPM: 69

Numero Respiri: 31

Respiri ECG: 0.268

Respiri Asse Z: 0.268

Algoritmi	centro spalle	centro collo	poligono addome
singolo punto (Hz)	0.261	0.270	0.139
poligono addome (Hz)	0.444	0.244	/
poligono petto (Hz)	0.122	0.270	0.444
poligono torso (Hz)	0.183	0.122	0.139

## Soggetto 2

BPM: 104

Numero Respiri: 35

Respiri ECG: 0.301

Respiri Asse Z: 0.302

<b>Algoritmi</b>	<b>centro spalle</b>	<b>centro collo</b>	<b>poligono addome</b>
singolo punto (Hz)	0.296	0.183	0.104
poligono addome (Hz)	0.104	0.104	/
poligono petto (Hz)	0.104	0.183	0.104
poligono torso (Hz)	0.104	0.104	0.104

## Soggetto 3

BPM: 65

Numero Respiri: 40

Respiri ECG: 0.344

Respiri Asse Z: 0.344

<b>Algoritmi</b>	<b>centro spalle</b>	<b>centro collo</b>	<b>poligono addome</b>
singolo punto (Hz)	0.148	0.339	0.165
poligono addome (Hz)	0.435	0.122	/
poligono petto (Hz)	0.148	0.113	0.122
poligono torso (Hz)	0.104	0.374	0.165

## Soggetto 4

BPM: 81

Numero Respiri: 20

Respiri ECG: 0.173

Respiri Asse Z: 0.173

<b>Algoritmi</b>	<b>centro spalle</b>	<b>centro collo</b>	<b>poligono addome</b>
singolo punto (Hz)	0.487	0.261	0.252
poligono addome (Hz)	0.104	0.104	/
poligono petto (Hz)	0.157	0.261	0.270
poligono torso (Hz)	0.313	0.104	0.174

Nella tabella 3.4 sono riportati i valori del 'mean absolute value' per ogni algoritmo della prima colonna rispetto ai punti estratti presenti nella prima riga.

Algoritmi	centro spalle	centro collo	poligono addome
singolo punto (Hz)	0.131	0.053	0.146
poligono addome (Hz)	0.133	0.128	/
poligono petto (Hz)	0.139	0.110	0.173
poligono torso (Hz)	0.166	0.111	0.127

Tabella 3.4: MAE - movimento libero

Da quanto scritto nella tabella, i valori d'errore risultano essere elevati, portando ad avere delle predizioni di frequenze respiratorie totalmente diverse dalle frequenze reali.

Analizziamo i grafici per raccogliere qualche informazione in più e cercare di capire le cause dei risultati ottenuti. Consideriamo 2 coppie di grafici per effettuare un'analisi migliore. In Figura 3.23, 3.24 vengono mostrati i grafici dell'algoritmo 'singolo punto' rispetto al punto centrale del collo rispettivamente nel dominio della frequenza e del tempo per il soggetto 2. Il grafico in frequenza riporta la frequenza più alta pari a 0.183 Hz ed una frequenza con ampiezza leggermente minore che si aggira al valore 0.3 Hz (valore corretto della frequenza di respirazione). Questo vuol dire che il movimento non è stato eliminato correttamente. Oltre alle 2 frequenze principali ci sono altre frequenze con ampiezza minore, ma comunque rilevanti, che testimoniano la presenza del rumore, infatti il segnale rappresentato nel grafico nel dominio del tempo è totalmente coperto dal rumore.

Analogamente ai primi due grafici, la seconda coppia di grafici mostrati (3.25, 3.26) riporta lo stesso problema di rumorosità che copre totalmente il segnale nel dominio del tempo e non rende visibile la frequenza dei respiri nel grafico in frequenza.

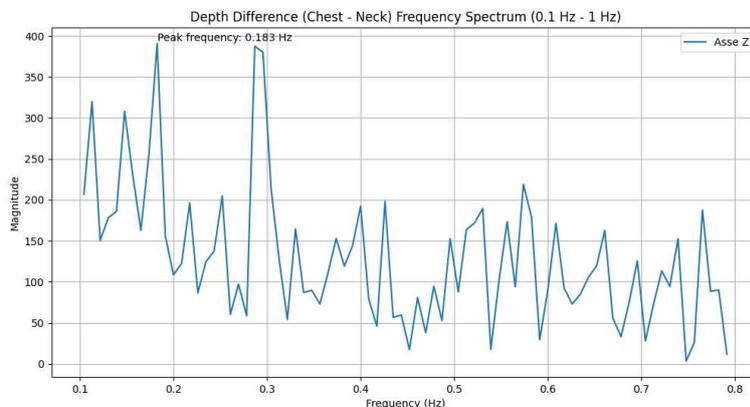


Figura 3.23: soggetto 2 - singolo punto rispetto a centro collo (frequenza)

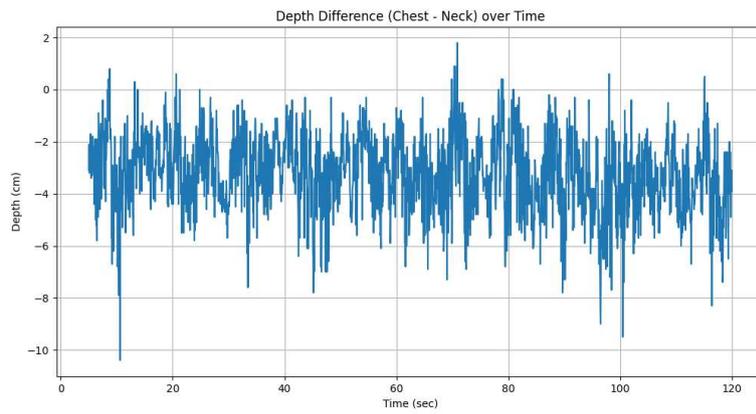


Figura 3.24: soggetto 2 - singolo punto rispetto a centro collo (tempo)

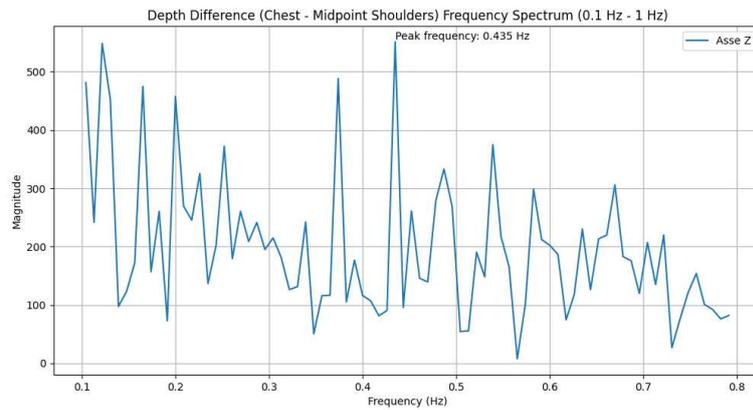


Figura 3.25: soggetto 3 - poligono addome rispetto a centro spalle (frequenza)

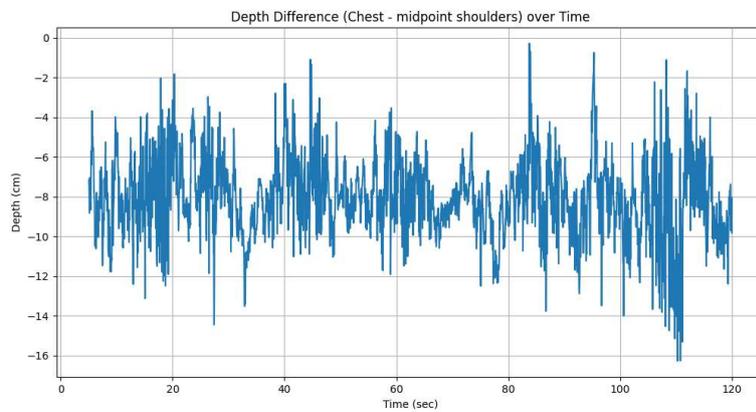


Figura 3.26: soggetto 3 - poligono addome rispetto a centro spalle (tempo)

Avendo raccolto tutti i dati e analizzato i grafici di principale interesse, possiamo affermare che, differentemente dalle condizioni statiche, in condizione dinamica con movimento del torso le frequenze risultano sufficientemente buone solo per l'algoritmo 'poligono addome' con un riferimento per l'estrazione del segnale respiratorio. Si ottiene una performance molto scarsa nel caso del movimento libero dei vari soggetti, per il quale nessuno degli algoritmi risulta affidabile per l'estrazione corretta del segnale respiratorio.



## Capitolo 4

# Sviluppi futuri

I risultati ottenuti dal capitolo 3 hanno definito le performance dei vari algoritmi in diverse condizioni. In condizioni statiche, sia in posizione seduto frontale sia in piedi con il corpo ruotato di circa  $45^\circ$ , gli algoritmi hanno raggiunto l'obiettivo prefissato, ovvero quello di leggere la frequenza respiratoria attraverso la telecamera Intel Realsense, senza fare uso di nessun sensore fisico. Nel caso dinamico si sono riscontrati vari problemi, soprattutto durante il movimento libero del soggetto. Le cause riguardano principalmente il movimento che non viene tolto completamente facendo la sottrazione tra un punto influenzato dal respiro ed un punto la cui distanza non dipende dalla respirazione, ed il rumore che rimane sovrapposto al segnale respiratorio estratto. Per questi motivi risulta complesso ricavare la frequenza respiratoria corretta nel grafico in frequenza. Per risolvere questi problemi in futuro, sarà necessario sviluppare algoritmi di filtraggio del segnale più avanzati, che possano separare efficacemente il movimento del corpo dal segnale respiratorio. L'integrazione di tecniche di machine learning, come reti neurali convoluzionali o modelli di deep learning, potrebbe migliorare significativamente la capacità di distinguere tra movimenti volontari e involontari. Inoltre, l'uso di modelli di previsione basati su dati storici potrebbe aiutare a migliorare la precisione del segnale respiratorio, riducendo il rumore e le distorsioni. Un'altra direzione promettente potrebbe essere l'utilizzo di più telecamere per ottenere una visione tridimensionale del soggetto, permettendo una migliore separazione dei movimenti del respiro da altri movimenti del corpo. Infine, lo sviluppo di un sistema di feedback in tempo reale che adatti gli algoritmi di rilevamento in base alle condizioni ambientali e al comportamento del soggetto potrebbe migliorare ulteriormente l'accuratezza e la robustezza del sistema. Questi sviluppi potrebbero portare a un sistema di monitoraggio della respirazione non invasivo e altamente affidabile, utile non solo in ambito medico ma anche in applicazioni sportive e lavorative.



# Capitolo 5

## Conclusione

In questa tesi è stato studiato il processo di estrazione del segnale respiratorio attraverso l'utilizzo della telecamera Intel Realsense D455, basandosi sulla variazione della distanza di una determinata area del corpo rispetto alla telecamera. Oltre a questa telecamera, è stato impiegato il sensore PolarH10 a contatto fisico, utile per estrarre il segnale respiratorio reale e confrontarlo con il segnale ricavato dalla telecamera, in modo da garantire l'accuratezza dei dati. Sono state effettuate delle acquisizioni di 120 secondi su soggetti maschi di età variabile tra i 20 e 30 anni in condizioni statiche e condizioni dinamiche. Per quanto riguarda le condizioni statiche sono state testate 2 posizioni: posizione da seduto frontale a distanza variabile e posizione in piedi ruotato di circa  $45^\circ$  rispetto alla telecamera. Le condizioni dinamiche invece riguardano la rotazione del torso e il movimento libero del soggetto nello spazio inquadrato dalla telecamera. Inizialmente è stato definito uno script per la lettura sincrona dei dati catturati con il PolarH10 e dei frame RGB-D registrati dalla telecamera Intel. Per il calcolo della distanza è stato utilizzato il modello di machine learning YOLO v8n-pose in grado di rilevare la sagoma della persona in ogni frame e quindi individuando anche le aree di maggior interesse. Per le condizioni statiche sono state considerate 4 aree del corpo: il punto centrale del petto (2.4.2), il poligono dell'addome (2.4.3), il poligono del petto (2.4.4) ed il poligono di tutto il torso (2.4.5). Calcolando la distanza di ogni area per ogni frame è stato prelevato il segnale respiratorio. Per le condizioni dinamiche sono state considerate 2 regioni di interesse in più: il punto centrale tra le due spalle (2.4.6) ed il punto centrale del collo (2.4.7). Questi punti, insieme all'area definita dal poligono dell'addome, daranno in uscita un segnale influenzato solo dal movimento che verrà poi sottratto al segnale di uno dei 4 algoritmi elencati in precedenza, in modo tale da ricavarne il segnale respiratorio. Per stabilire le performance abbiamo utilizzato il MAE (mean absolute error), calcolato per ogni algoritmo. In entrambi i casi delle condizioni statiche il MAE risulta basso per ogni algoritmo, con un picco massimo di 0.04 Hz in condizione in piedi ruotato, definendo quindi che il segnale respiratorio predetto risulta pari al segnale respiratorio reale. I risultati delle condizioni dinamiche vanno distinti. Riguardo al movimento rotazionale del torso, il MAE

risulta piuttosto alto, con un picco massimo del 0.107 Hz, tranne per gli algoritmi 'poligono addome' rispetto a 'centro spalle' e 'poligono petto' rispetto a 'poligono addome', per i quali vale rispettivamente 0.008 Hz e 0.005 Hz (valori molto bassi e quindi accettabili). Contrariamente si ha per il movimento libero del soggetto che riporta un MAE con un range tra 0.053 Hz - 0.173 Hz, causato dal forte rumore e dall'incapacità di estrarre correttamente il segnale respiratorio.



# Bibliografia

- [1] Anthony P Addison, Paul S Addison, Philip Smit, Dominique Jacquiel, and Ulf R Borg. Noncontact respiratory monitoring using depth sensing cameras: A review of current literature. *Sensors*, 21(4):1135, 2021.
- [2] Paul S Addison, André Antunes, Dean Montgomery, Philip Smit, and Ulf R Borg. Robust non-contact monitoring of respiratory rate using a depth camera. *Journal of Clinical Monitoring and Computing*, 37(4):1003–1010, 2023.
- [3] Jan Bayer and Jan Faigl. On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2019.
- [4] Peter H Charlton, Timothy Bonnici, Lionel Tarassenko, David A Clifton, Richard Beale, and Peter J Watkinson. An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram. *Physiological Measurement*, 37(4):610, mar 2016.
- [5] Andrew L Fielding, Ajay K Pandey, Yaqub Jonmohamadi, Riccardo Via, Damien C Weber, Antony J Lomax, and Giovanni Fattori. Preliminary study of the intel realsense d415 camera for monitoring respiratory like motion of an irregular surface. *IEEE Sensors Journal*, 21(13):14443–14453, 2020.
- [6] Zein Hajj-Ali, Kim Greenwood, JoAnn Harrold, and James R Green. Towards depth-based respiratory rate estimation with arbitrary camera placement. In *2022 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–6. IEEE, 2022.
- [7] Shan He, Zixiong Han, Cristóvão Iglesias, Varun Mehta, and Miodrag Bolic. A real-time respiration monitoring and classification system using a depth camera and radars. *Frontiers in Physiology*, 13:799621, 2022.
- [8] Wakana Imano, Kenichi Kameyama, Malene Hollingdal, Jens Refsgaard, Knud Larsen, Cecilie Topp, Sissel Højsted Kronborg, Josefine Dam Gade, and Birthe Dinesen. Non-contact respiratory measurement using a depth camera for elderly people. *Sensors*, 20(23):6901, 2020.

- [9] Magnus Bruze Jenny Glimsjö Kajsa Källberg Thanisorn Sukakul Martin Mowitz Josefin Ulriksdotter, Cecilia Svedman. Contact dermatitis caused by glucose sensors, 2020.
- [10] Jochen Kempfle and Kristof Van Laerhoven. Breathing in-depth: A parametrization study on rgb-d respiration extraction methods. *Frontiers in Computer Science*, 3:757277, 2021.
- [11] Daniel G Kyrollos, Joshua B Tanner, Kim Greenwood, JoAnn Harrold, and James R Green. Noncontact neonatal respiration rate estimation using machine vision. In *2021 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2021.
- [12] Ivan Loddo. *Respiro patologico e le sue caratteristiche*, 2022.
- [13] Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinnasse, Hung Pham, Christopher Schölzel, and S. H. Annabel Chen. NeuroKit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, 53(4):1689–1696, feb 2021.
- [14] Mathias Minos-Stensrud, Ole Henrik Haakstad, Olav Sakseid, Baard Westby, and Alex Alcocer. Towards automated 3d reconstruction in sme factories and digital twin model generation. In *2018 18th international conference on control, automation and systems (ICCAS)*, pages 1777–1781. IEEE, 2018.
- [15] Mohsen Mozafari, Andrew Law, Samuel Béni Tchoudem Djouaka, James R Green, and Rafik A Goubran. Comparison of blind source separation techniques for respiration rate estimation from depth video. In *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–5. IEEE, 2022.
- [16] Manivannan Muniyandi and Rahul Soni. Breath rate variability (brv) - a novel measure to study the meditation effects. *International Journal of Yoga*, Accepted, 01 2017.
- [17] Jayashree V Patil and Preeti Bailke. Real time facial expression recognition using realsense camera and ann. In *2016 International Conference on Inventive Computation Technologies (ICICT)*, volume 2, pages 1–6. IEEE, 2016.
- [18] Bruce Rogers, Marcelle Schaffarczyk, and Thomas Gronwald. Estimation of respiratory frequency in women and men by kubios hrv software using the polar h10 or movesense medical ecg sensor during an exercise ramp. *Sensors*, 22(19):7156, 2022.
- [19] Surita Sarkar, Saptak Bhattacharjee, and Saurabh Pal. Extraction of respiration signal from ecg for respiratory rate estimation. 2015.

- [20] Vladimir Tadic, Akos Odry, Istvan Kecskes, Ervin Burkus, Zoltan Kiraly, and Peter Odry. Application of intel realsense cameras for depth image generation in robotics. *WSEAS Transac. Comput*, 18:2224–2872, 2019.
- [21] Paul van Gent, Haneen Farah, Nicole van Nes, and Bart van Arem. Heartpy: A novel heart rate algorithm for the analysis of noisy signals. *Transportation Research Part F: Traffic Psychology and Behaviour*, 66:368–378, 2019.
- [22] Yunlu Wang, Menghan Hu, Yuwen Zhou, Qingli Li, Nan Yao, Guangtao Zhai, Xiao-Ping Zhang, and Xiaokang Yang. Unobtrusive and automatic classification of multiple people’s abnormal respiratory patterns in real time using deep neural network and depth camera. *IEEE Internet of Things Journal*, 7(9):8559–8571, 2020.
- [23] Yun Xiao and Hai Liao. Lida-yolo: An unsupervised low-illumination object detection based on domain adaptation. *IET Image Processing*, 18:n/a–n/a, 12 2023.
- [24] Qi Zhan, Jingjing Hu, Zitong Yu, Xiaobai Li, and Wenjin Wang. Revisiting motion-based respiration measurement from videos. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 5909–5912. IEEE, 2020.