

## Progetto di *Elaborazione e trasmissioni di immagini*

Anno accademico 2018/19

# REMOTE REHABILITATION

Algoritmo di match sagome-wireframe e estrazione parametri di match

Algoritmo scaricabile al seguente link:

<https://github.com/mattybass/progettoETI.git>

Per informazioni e note tecniche leggere il *readme*: che si trova in fondo alla pagina o al link

<https://github.com/mattybass/progettoETI/blob/master/README.md>

*Daniele Mattedi* – matricola **186778**

*Sergio Povoli* – matricola **185790**

*Riccardo Ricci* – matricola **181398**

## INDICE

• Consegna	2
• Introduzione	2
• Joint	2
• Struttura dati UML	3
• Schema di massima del progetto	4
• Lettura da file	5
• Dipendenza spaziale	6
• Dipendenza temporale	8
• Valutazione	13
• Relazione dell'output	15

## Consegna

Ricevute in ingresso sagoma animata e wireframe misurare parametri di scostamento, velocità, accelerazioni tra esercizio modello ed esercizio svolto dal paziente.

Generare un report con valutazioni medie.

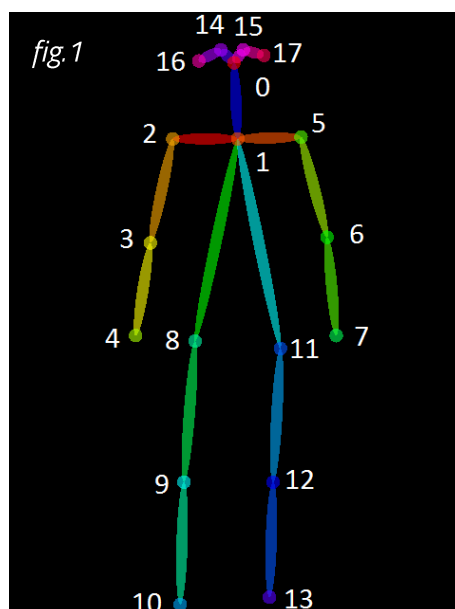
## Introduzione

La richiesta alla base del nostro progetto, ovvero la parte 3 del sistema di **remote re-habilitation**, è creare un *algoritmo di match sagome-wireframe* in grado di estrarre i parametri di match.

Ci basiamo sui movimenti di una sagoma ideale, ovvero di un modello generato in precedenza ed approvato da una figura preposta, che viene confrontata con quella del paziente, i cui dati sono estratti in tempo reale.

La nostra base di partenza sono dei parametri temporali e spaziali ricevuti in ingresso tramite un file di testo di tipo *json*. Quest'ultimo viene generato grazie al software *Openpose*, che elabora i video filmati da una telecamera 3-D grazie ad un sistema di triangolazione: contiene in particolare le coordinate X-Y-Z delle articolazioni e dei principali punti chiave del nostro corpo (joint) nel tempo (si può arrivare fino a 60 fps).

Il software che abbiamo sviluppato è stato scritto con linguaggio C++, per via della sua duttilità e versatilità. Si è rivelato fondamentale anche l'utilizzo di *Matlab* per l'analisi grafica della coerenza dei dati ottenuti e di quelli processati.



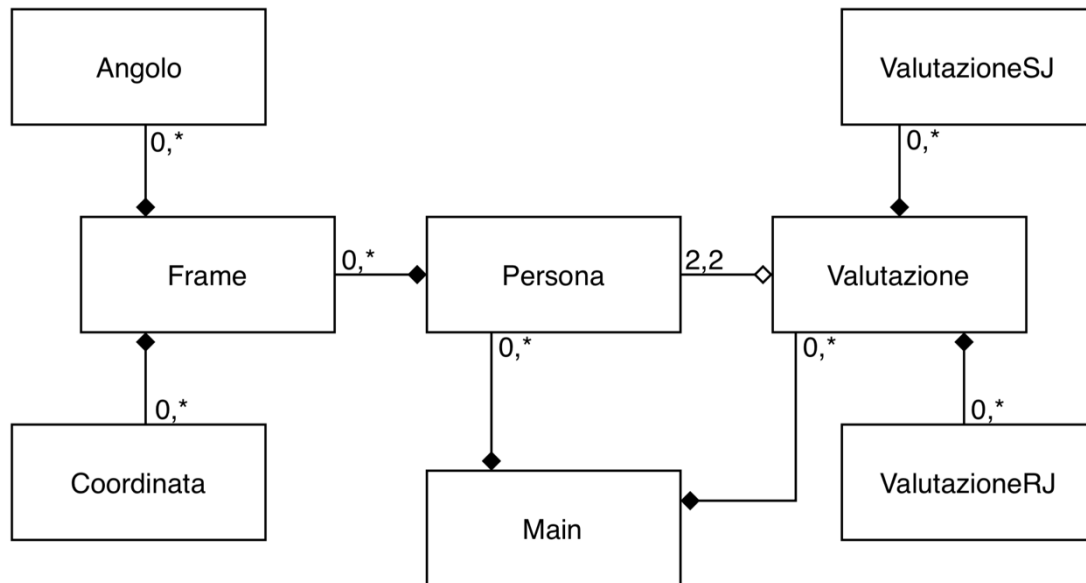
## Joint

Per joint, nel contesto del progetto si intende un'articolazione, con l'eccezione di poche posizioni che comunque mantengono un significato all'interno del corpo. Per le finalità del nostro progetto non abbiamo tenuto in considerazione le misure fornite sui **joints 14,15,16 e 17**, mentre ci siamo accorti che *openpose* non è riuscita ad estrarre le misure del **joint 0** (anch'esso comunque trascurabile).

JOINT	PARTE DEL CORPO	JOINT	PARTE DEL CORPO
0	Collo	9	Ginocchio dx
1	Sterno	10	Piede dx
2	Spalla dx	11	Anca sx
3	Gomito dx	12	Ginocchio sx
4	Mano dx	13	Piede sx
5	Spalla sx	14	Occhio dx
6	Gomito sx	15	Occhio sx
7	Mano sx	16	Orecchio dx
8	Anca dx	17	Orecchio sx

## Struttura dati e UML

fig.2



Nell'approccio della progettazione della struttura dati del nostro programma ci siamo basati sui concetti chiave della programmazione ad oggetti, riuscendo perciò a sfruttarne le innumerevoli caratteristiche e pregi.

Le classi che abbiamo creato possono essere suddivise in 2 grandi gruppi, il primo contiene tutti i dati, opportunamente gestiti e archiviati, ricevuti dal file JSON, il secondo invece si concentra sulla fase di valutazione contenendo tutte le metriche dalle quali costruiamo l'output.

### Elenco delle Classi di Dati:

<b>Coordinata</b>	Classe che contiene le coordinate (X-Y-Z) di un joint.
<b>Angolo</b>	Classe che contiene il valore <b>zenit</b> e <b>azimut</b> dell'angolo sotteso dall'articolazione considerata (vedere schema successivo sui nomi degli angoli rapportati alle varie articolazioni). Inoltre contiene altri due attributi, uno che specifica se si tratta di un massimo o un minimo locale, l'altro il numero di frame al quale quell'angolo si riferisce.
<b>Frame</b>	La classe Frame ha in composizione le due classi base di caratterizzazione spaziale, contiene due <i>map</i> ( <b>coordinatejoint</b> e <b>angolijoint</b> ) entrambi con chiave primaria il numero di angolo e come oggetto rispettivamente un oggetto coordinata e un oggetto angolo relativi all'articolazione considerata. Inoltre, un ulteriore attributo è il numero di frame corrispondente alle misurazioni di coordinata e angolo
<b>Persona</b>	La classe Persona, una delle due classi contenute nel main, è il livello più alto nella gestione dei dati estratti dal file json. Questa classe ha la caratteristica di contenere in maniera ordinata la lista di <b>Frame</b> rappresentanti il movimento della

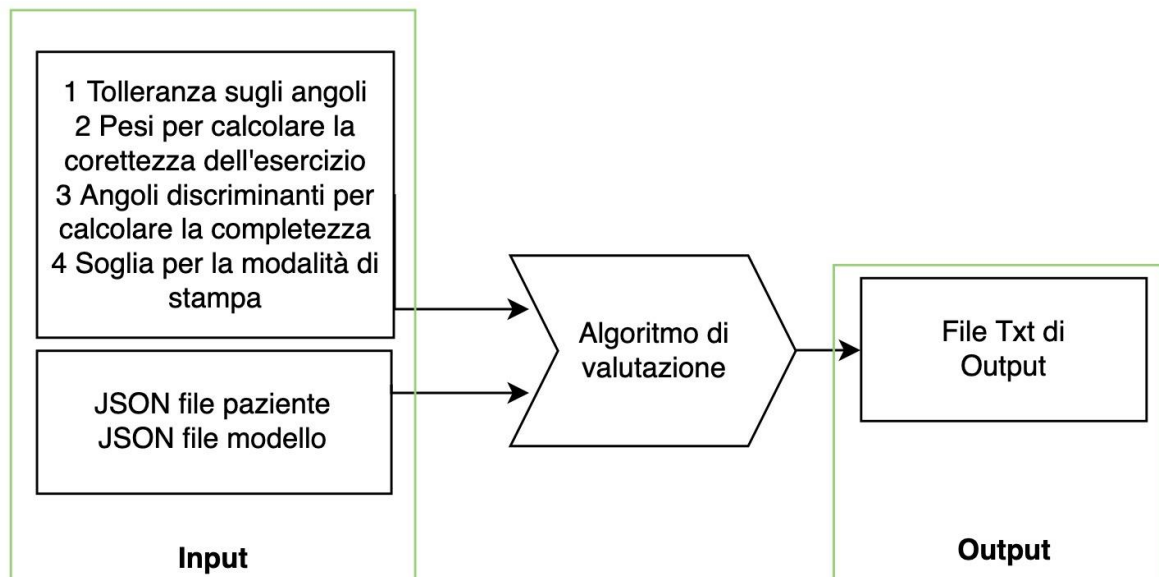
persona, e contenenti l'andamento di tutti gli angoli formati dalle varie articolazioni nel tempo.

### Elenco delle classi di Valutazione:

- Valutazione SJ** Classe che valuta i movimenti di una singola articolazione. Tramite questa classe andiamo a calcolare distaccamento spaziale e temporale tra i movimenti di una singola articolazione del paziente in relazione a quella del modello.
- Valutazione RJ** Classe che valuta i movimenti di tutte le altre articolazioni fissate le posizioni chiave dell'articolazione considerata. Questa classe ha perciò l'obiettivo di dare una valutazione più generale del movimento eseguito dal paziente, astruendo dal singolo joint.
- Valutazione** Classe che contiene, per ogni articolazione, un oggetto di tipo valutazione SJ e un oggetto di tipo valutazione RJ. Inoltre contiene i pesi che il medico fisioterapista assegna ad ogni articolazione (sia zenit che azimuth). Si occupa di dare in uscita la valutazione globale dell'esercizio.

### Schema di massima del progetto

fig.3



## Lettura da file

```
"frame_920": {  
  "person_0": {  
    "joint_2": {  
      "x": "-0.109600",  
      "y": "-0.365285",  
      "z": "2.382370"  
    },  
    "joint_1": {  
      "x": "0.068029",  
      "y": "-0.357171",  
      "z": "2.329452"  
    }  
  }  
}
```

Il file che *openpose* fornisce in output ha il seguente aspetto (contiene 17 joint per frame). Per leggere e caricare i dati nelle classi create si utilizzano le librerie **fstream** e **sstream**. Tutto il procedimento viene effettuato nel costruttore di *Persona*:

```
Persona(int _framesec, string nome_file, string percorso_file="json")
```

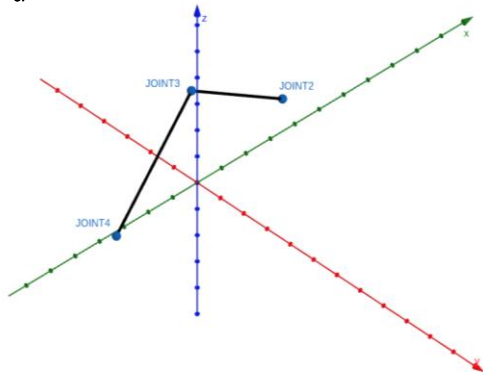
La lettura viene eseguita riga per riga (viene utilizzata la funzione **std::getline**). Nel momento in cui viene individuata la parola **frame** viene estratto il numero che lo caratterizza (nell'esempio sopra sarebbe 920) grazie a **std::string::find** (necessario per determinare la posizione dove termini il numero), **std::string::substr** (estrae la sottostringa) e **std::stringstream** (effettua la conversione da stringa ad un tipo numerico); richiamando i costruttori appositi si va a creare poi un oggetto frame. Lo stesso viene fatto per **joint**: in tal caso però vengono lette anche le righe successive per estrarre i parametri x, y e z.

Il sistema di riferimento utilizzato dalla *zedcam* non è quello canonico: infatti la profondità viene definita dalla **z**. Abbiamo preferito utilizzare un sistema di riferimento cartesiano in cui la **z** indica l'altezza, mentre la profondità è determinata dalla **y**.

## Dipendenza spaziale

Il primo grande problema è dover lavorare su soggetti diversi. Chiaramente non esiste alcun uomo

fig.4-a



vitruviano, un uomo con proporzioni perfette: ogni modello è caratterizzato da misure differenti.

Questo comporta che non ci possiamo basare solamente sulle coordinate dei joint per capire se il movimento viene effettuato nella maniera corretta. Osserviamo la *fig.4-a*: in una persona dotata di apertura di braccia superiore alla media **joint 4 e joint 7** (ma non solo) saranno più spostati verso il basso, a differenza di persone con arti di misura inferiore.

Per risolvere questo problema è stato deciso di confrontare gli angoli sottesi dai vari joint (ogni joint utile è intersezione di due segmenti).

Lavorando in 3-D si è reso necessario ottenere le misurazioni per due sistemi di riferimento differenti. Abbiamo scelto il piano X-Y ed il piano X-Z, chiamando gli angoli rispettivamente **azimut** e **zenit**.

fig.4-b: X-Y

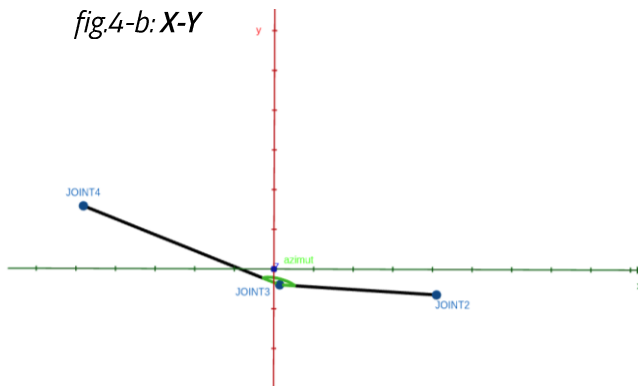
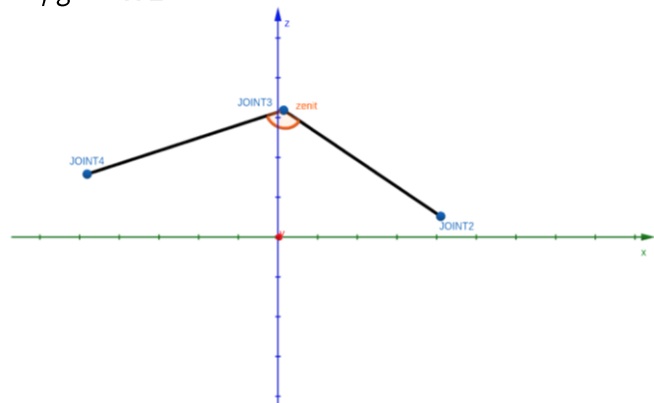


fig.4-c: X-Z



Questi due angoli vengono calcolati grazie a due funzioni presenti nella classe *Utility*:

```
double angolo_azimut(Coordinata& c1, Coordinata& c2, Coordinata& c3);
```

```
double angolo_zenit(Coordinata& c1, Coordinata& c2, Coordinata& c3);
```

ogni porzione di arto viene assunta come un segmento caratterizzato da un certo coefficiente angolare (pendenza) dal quale si giunge facilmente all'angolo formato da questo con l'orizzontale del nostro sistema di riferimento. I joint misurabili sono quindi quelli che congiungono due segmenti, dando la possibilità di calcolarne l'angolo sotteso, semplicemente come differenza tra i due precedentemente ricavati (formati con l'asse x). Questo compito viene svolto dalle seguenti funzioni

```
double angolo_XY(Coordinata &c1,Coordinata &c2);
```

```
double angolo_XZ(Coordinata &c1,Coordinata &c2);
```

la prima è utilizzata per il calcolo dell'angolo chiamato **azimut**, mentre la seconda per l'angolo **zenit**. Per calcolare il coefficiente angolare si è rivelato necessario utilizzare un sistema di riferimento antiorario, calcolando l'angolo come differenza rispetto all'asse x.

Partendo dalle coordinate di 3 joint abbiamo ottenuto così due angoli: **azimut**, ovvero rispetto a XY, e **zenit**, rispetto a XZ. Da ora in poi le coordinate non vengono più prese in considerazione: andremo ad analizzare la variazione degli angoli nel tempo.

Gli angoli considerati sono i seguenti:

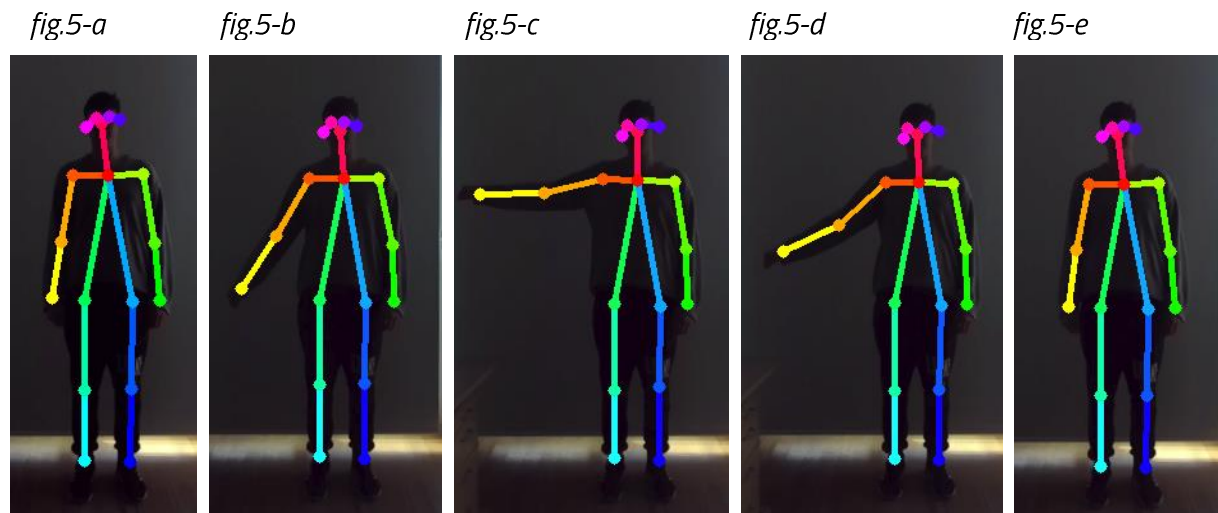
<i>JOINT</i>	<i>JOINT DI RIFERIMENTO PER IL CALCOLO DELL'ANGOLO</i>		
1	2	1	8
2	1	2	3
3	4	3	2
5	1	5	6
6	5	6	7
8	1	8	9
9	8	9	10
11	1	11	12
12	11	12	13



## Dipendenza temporale

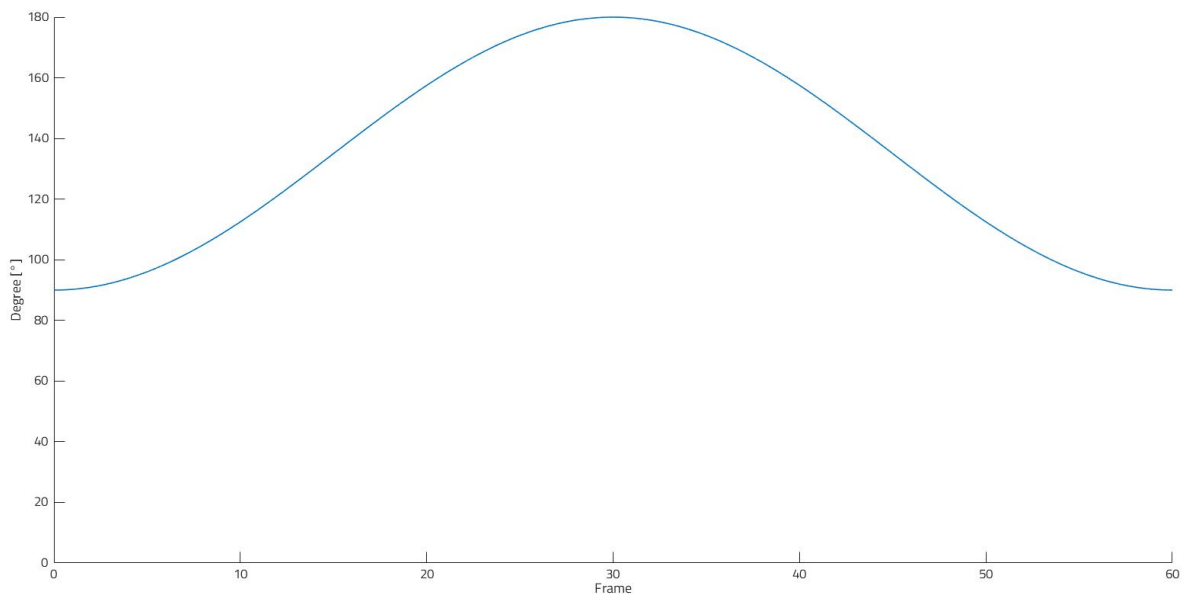
Risolto il problema della dipendenza spaziale, ci siamo subito scontrati con un altro ostacolo: la dipendenza temporale. Vogliamo avanzare una proposta completa e versatile: di conseguenza il nostro software deve considerare la variabile temporale al pari di tutti gli altri aspetti da valutare (un esercizio svolto alla giusta velocità riceve un punteggio più alto di un esercizio svolto più lentamente o più velocemente).

Consideriamo il seguente esercizio:



Focalizziamo la nostra attenzione sul **joint 2** (spalla dx) e sull'angolo che si forma tra la retta che collega gomito con spalla (**joint 3, joint 2**) e spalla con sterno (**joint 2, joint 1**). Il sistema di coordinate è quello X-Z: nella foto si può chiaramente individuare l'angolo **zenit**. Tale angolo avrà due minimi (*fig.5-a* e *fig.5-e*) ed un massimo (*fig.5-d*). In questo caso possiamo ipotizzare che l'andamento angolare del **joint 3** sia di tipo sinusoidale, con un grafico approssimabile al seguente.

*fig.6*



Nel seguente grafico si mostra l'andamento in gradi dell'angolo (asse y), all'aumentare del numero di **frame** (asse x).

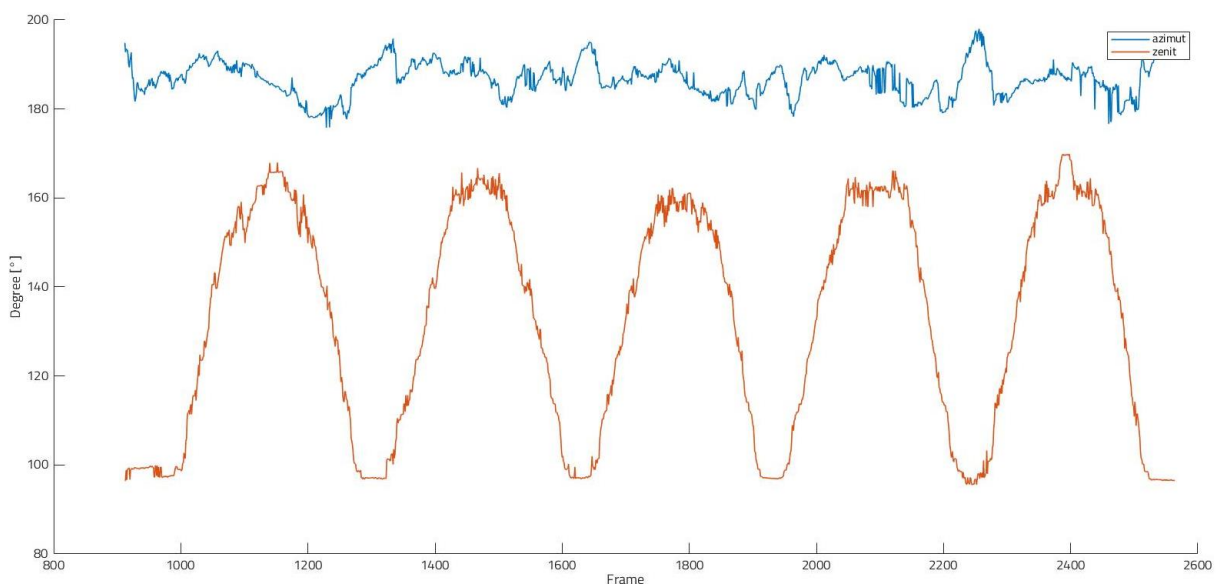
L'angolo varierà nel tempo da  $90^\circ$  a  $180^\circ$  circa: possiamo affermare che un movimento fisico sia proprio il passaggio di un angolo da un massimo o minimo o viceversa. Da ora in poi i massimi e minimi saranno chiamati **posizioni chiave**. Individuando queste posizioni possiamo confrontare due modelli: per la valutazione si potrà confrontare per esempio l'angolo di partenza e di arrivo del paziente rispetto al modello ideale, la velocità con cui è stato fatto il movimento e molti altri possibili parametri.

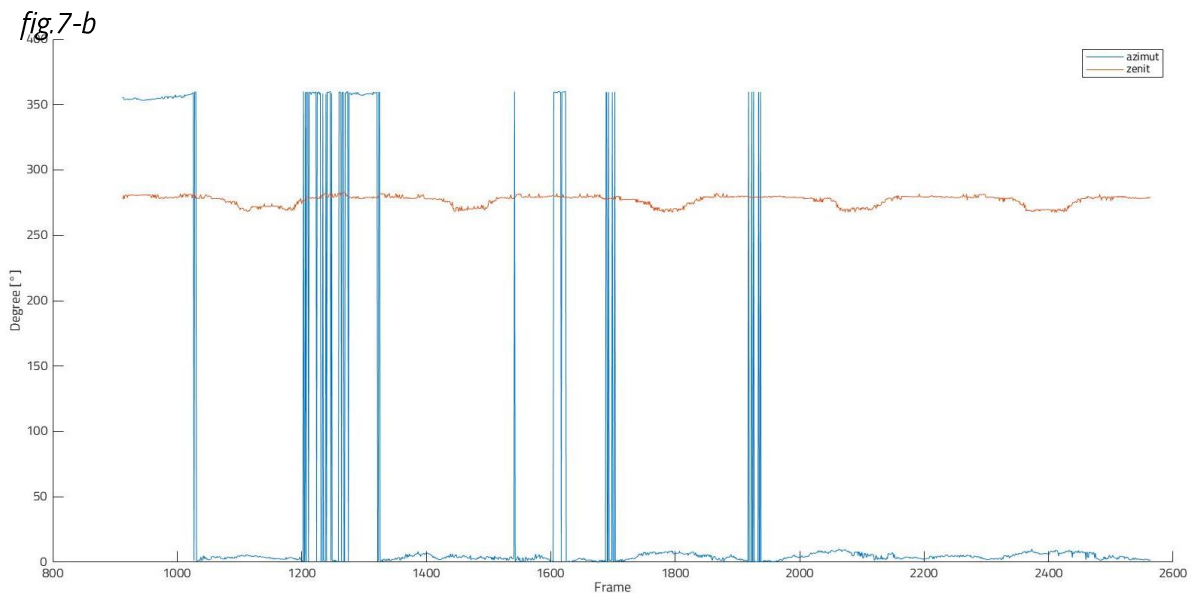
In *fig.6* si può notare che l'unità di misura sia frame e non secondi. I dati ricevuti in input sono frame by frame: conoscendo l'*FPS* (frequenza di cattura o riproduzione dei fotogrammi che compongono un filmato) a cui lavora lo strumento ottico utilizzato dal paziente, si riesce a convertire i frame in secondi e ottenere effettivamente la velocità con cui paziente effettua i movimenti.

L'obiettivo si sposta dunque sull'individuazione delle posizioni chiave. Nel caso di funzioni continue l'impegno per trovare massimi o minimi è trascurabile; i nostri dati però sono sperimentali, e di conseguenza sono caratterizzati da errori più o meno importanti: l'esperienza di utilizzo ci porta ad affermare che la telecamera e il software utilizzati (*Stereolabs ZED* e *OpenPose*) non siano completamente affidabili. Abbiamo notato scarsa accuratezza soprattutto nel caso della determinazione della profondità (azimut).

La non totale precisione degli strumenti utilizzati provoca due dirette conseguenze, osservabili in *fig.7-a* e *fig.7-b*.

*fig.7-a:*

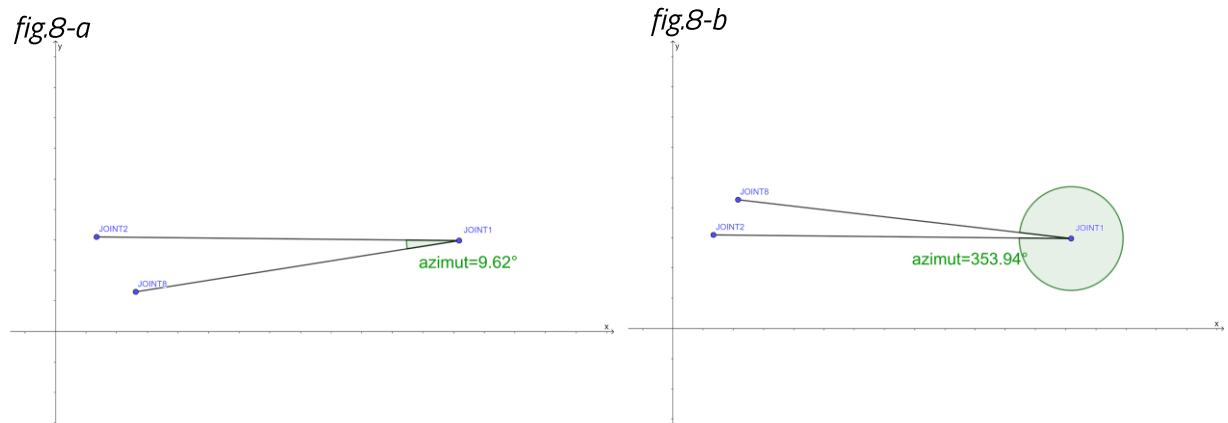




Si tratta dell'esercizio visibile nelle foto precedenti, ripetuto 5 volte: *fig.7-a* rappresenta i movimenti della spalla, mentre *fig.7-b* è relativa al **joint 1**. In rosso si può osservare l'andamento dell'angolo **zenit** (X-Z), mentre in blu è rappresentato l'**azimut**.

Entrambi i grafici sono caratterizzati da variazioni repentine ed andamenti talvolta in disaccordo con il movimento effettuato dal joint: la causa è da individuare in problematiche hardware e software precedenti alla conversione da coordinate ad angoli. Tale problema è molto evidente nel caso del sistema di coordinate X-Y e impedisce di lavorare direttamente su questi dati per individuare le posizioni chiave.

Osserviamo **azimut** in *fig.7-b*: si passa improvvisamente da un angolo di circa  $0^\circ$  ad uno che è di poco inferiore ai  $360^\circ$ . Nelle figure seguenti si può individuare facilmente la motivazione del comportamento degli angoli.



Come già detto i dati ricevuti sono caratterizzati da errore elevato nel caso della profondità. Inoltre in *fig.8-a* e *fig.8-b* si nota come il passaggio da un angolo positivo ad un angolo negativo possa avere effetti potenzialmente devastanti: nell'esempio si ha una variazione reale di poco più di  $15^\circ$ , mentre

la funzione che calcola la differenza individua ben  $345^\circ$  in più.

Per superare tale problema, visibile anche in *fig.7-b*, è stata implementata nella classe **Persona** la seguente funzione

```
void pulisci_errori_sequenzaAngolo(int _angolo);
```

si prendono in analisi gli angoli in successione temporale: nel caso in cui ci sia una differenza di angolo improbabile (impostata a 100) tra due frame contigui il secondo frame assume il valore dell'angolo precedente; in questo modo abbiamo corretto questi errori a sbalzo.

Ora il primo problema è stato eliminato e massimi o minimi dettati dal passaggio da un angolo positivo ad un angolo negativo è stato affrontato.

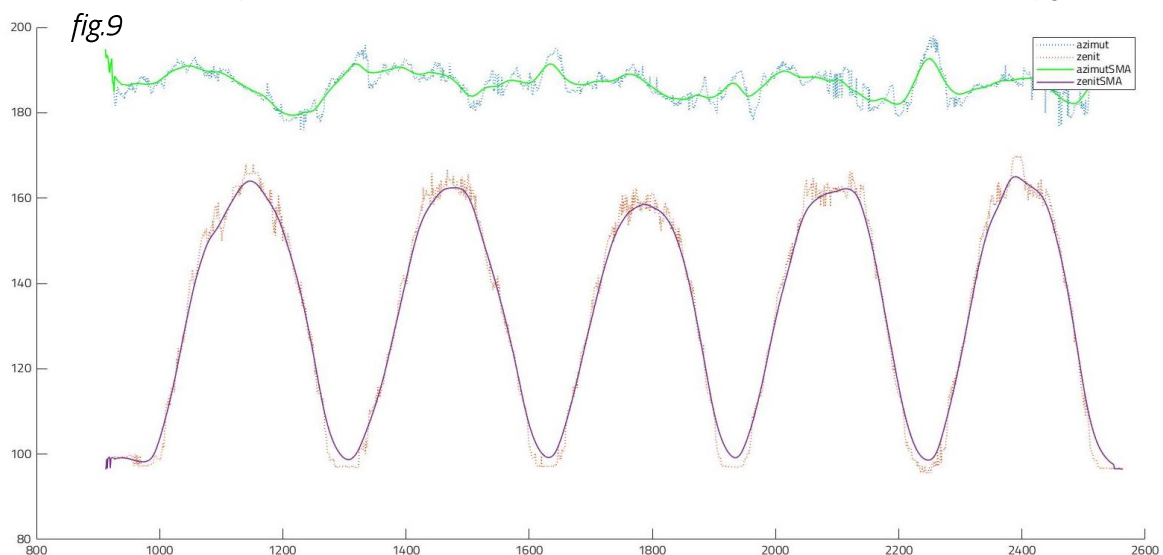
Analizziamo invece *fig.7-a*: l'obiettivo attuale è quello di individuare i punti chiave per un determinato joint ed un certo angolo (azimut o zenit). Il fatto che le misurazioni della zedcam siano leggermente irregolari, inizialmente, ci hanno creato problemi nella ricerca dei massimi e minimi. Si è reso necessario aggiungere uno strumento solitamente molto usato in analisi tecnica e finanza, la **media mobile**: dopo aver provato varie tipologie (semplice, esponenziale e adattiva) siamo giunti alla conclusione che la media mobile semplice (SMA) fosse più la più adatta per il nostro scopo.

```
void mediamobile_angolo(int _angolo, int _finestra);
```

Ipotizzando di essere posizionato su di un certo frame, il valore sperimentale dell'angolo viene sostituito con la media degli  $\_finestra/2$  istanti temporali precedenti, quello corrente e i  $\_finestra/2$  successivi. Questa funzione lavora su tutto l'intervallo temporale. Per i punti iniziali l'efficacia è ridotta in quanto la finestra è ridotta in relazione agli istanti cui si può retrocedere prima di incontrare il frame iniziale, stessa cosa si può dire dei punti finali.

Per ovviare a questo piccolo problema, che comunque non pregiudica l'efficacia del metodo, abbiamo deciso di applicarla due volte di seguito sui nostri dati usando una finestra di 50 frame.

È da osservare il fatto che questa media cambia il valore del punto centrale della finestra, che viene poi usato in misure successive. L'efficacia del metodo è quindi maggiore rispetto al caso in cui il punto modificato è il primo della finestra. Il risultato della media mobile è visibile nella *fig.9*:



Ora siamo in grado di applicare la seguente funzione

```
void maxminFind_angolo(int _angolo, int _finestra);
```

Per come abbiamo pensato di implementare la valutazione, la posizione iniziale e la posizione finale, sono sempre considerate come punti chiave, gli altri punti sono individuati dalla funzione riportata sopra.

Altro concetto importante di cui tenere conto è la tolleranza. Successivamente a **maxminFind\_angolo** viene infatti richiamata

```
void maxminClean_angolo(int _angolo, int_tolleranza_zenit, int tolleranza_azimut);
```

Come si può intuire dal nome, questa funzione elimina massimi e minimi in eccesso tenendo solo quelli che si discostano dalla media più di una certa soglia (la **tolleranza**). Tale parametro, in uno scenario di implementazione reale del sistema, sarebbe fornito dal medico fisioterapista, tenendo però anche conto della precisione dello strumento di acquisizione. Inoltre, ricordandoci del funzionamento della media mobile e del metodo di ricerca dei massimi e minimi (che funziona con una finestra in modo simile alla media) abbiamo deciso di eliminare i punti chiave trovati nei primi  $\_finestra/2$  frame, che osservando empiricamente i dati, erano in eccesso, decidendo, come detto prima di inserire il frame iniziale. Stessa cosa vale per  $i\_finestra/2$  frame finali, che sono stati eliminati a favore del punto finale.

Ora abbiamo tutti i dati necessari per valutare la qualità dell'esercizio.

## Valutazione

La valutazione di un esercizio viene effettuata confrontando i movimenti del paziente con quelli di un modello. Ciò avviene indipendentemente da quante volte venga svolto un esercizio: questo significa che il numero di punti chiave individuati non deve necessariamente combaciare.

Ovviamente il fatto che un esercizio viene svolto dal paziente più o meno volte di quelle indicate dal modello viene inserito nel report finale, e influisce ovviamente anche nel coefficiente di fedeltà totale (anche se però in maniera poco marcata).

In un esercizio ad ogni joint vengono assegnate **due valutazioni** (per **azimut** e per **zenit**): una è ottenuta focalizzandosi sulle posizioni chiave della singola articolazione, l'altra osservando il movimento delle altre in corrispondenza delle posizioni chiave dell'articolazione in esame (usando uno specifico metodo che redistribuisce i pesi sui joint restanti). In un secondo tempo i dati restituiti dalla doppia valutazione vengono pesati in base al ruolo che quell'articolazione riveste nell'esercizio. Abbiamo deciso di tenere conto della valutazione sul singolo joint per l'80%, mentre la valutazione sulla relazione tra joint per il 20%.

### Valutazione SJ

La valutazione sul singolo joint prevede di effettuare due confronti.

Il primo prende in esame le misure spaziali. Vengono comparate le ampiezze degli angoli in esame (**azimut** e **zenit**) in corrispondenza delle posizioni chiave del paziente con quelle del modello e viene fatta una media matematica degli scostamenti per tutte le posizioni chiave del paziente.

La seconda si basa invece su misure temporali (*velocità di esecuzione del movimento*). Ricordiamo che si deve conoscere la frame rate della camera: in questo modo si può risalire al tempo che il paziente e il modello impiegano per compiere un movimento (inteso come il susseguirsi di due punti chiave). Si può quindi calcolare la velocità media del paziente e del modello (media matematica), ricavando così una velocità relativa media, che quindi va a contribuire alla valutazione.

Oltre alla velocità relativa media si conserva anche le velocità di esecuzione dei singoli movimenti, per fornire poi in output un feedback dettagliato per ogni singolo movimento.

Nel caso in cui i punti chiave siano due significa che non sono stati individuati massimi e minimi; i punti sono semplicemente quello iniziale e quello finale. In questo caso (caso *stazionario*) la valutazione della velocità non viene effettuata mentre viene effettuata quella spaziali.

Il giudizio finale di **ValutazioneSJ** deriva in egual misura dai due confronti appena elencati: con i valori ottenuti da essi viene fatta una semplice media, ottenendo così per ogni joint un valore che indica la valutazione SJ globale.

### ValutazioneRJ

È più macchinosa della precedente, anche se concettualmente molto simile. È stata pensata per avere una misura più accurata per valutare se il paziente, durante l'esercizio, compia dei movimenti indesiderati con le altre parti del corpo per aiutarsi in un certo esercizio.

Questa misura tiene conto di tutti i joint meno quello che si sta considerando. Ad esempio, se il

procedimento viene eseguito per il **joint 2**, i confronti verranno effettuati per i joint **1,3,5,6,8,9,11,12**. A differenza di **ValutazioneSJ**, in questo caso l'analisi della velocità non ha senso, ci siamo limitati alla comparazione delle misure spaziali.

Per ogni posizione chiave del paziente, viene confrontata l'ampiezza degli angoli di tutte le altre articolazioni in relazione con il modello per poi sommare i vari contributi moltiplicati per i pesi dati dal fisioterapista, opportunamente ridistribuiti sottraendo al totale il peso dell'articolazione considerata. Il risultato di questa somma ci dà un valore che rappresenta la **ValutazioneRJ** globale di quell'articolazione.

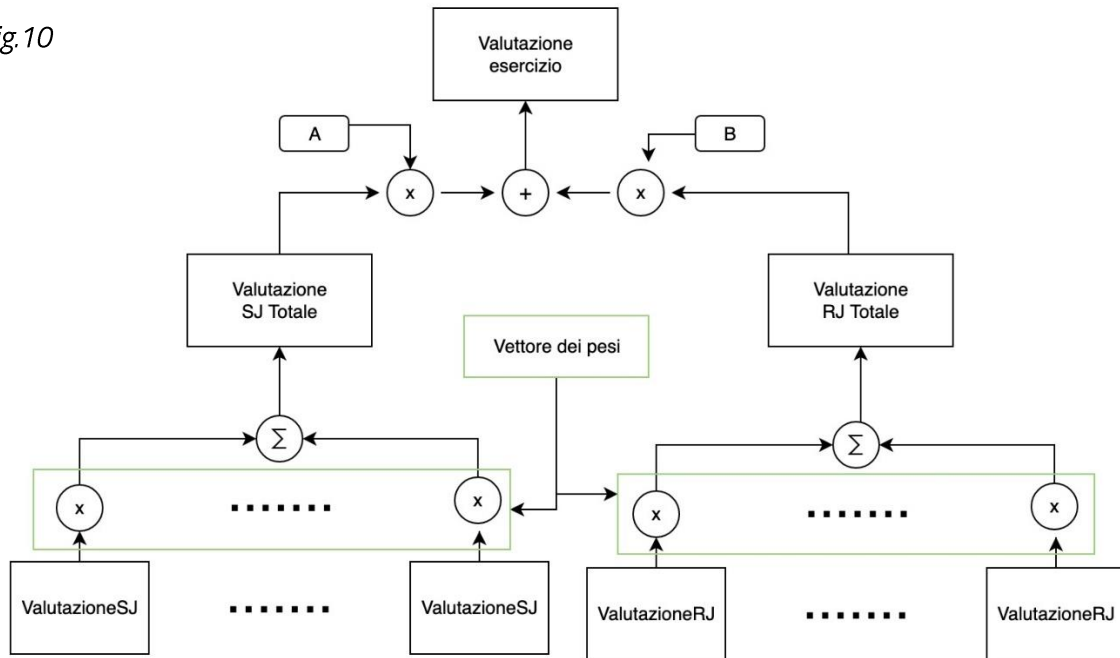
Di seguito presentiamo un esempio di redistribuzione dei pesi nel caso della **valutazione RJ**:

JOINT	PESO MEDICO	PESO RIDISTRIBUITO ( <i>ValutazioneRJ</i> – <i>Joint2</i> )
1	0.1	0.125
2	0.2	
3	0.3	0.375
5	0.1	0.125
6	0.1	0.125
8	0.1	0.125
9	0.1	0.125
11	0	0
12	0	0

La misura finale della valutazione (presente nella classe **Valutazione**) deriva da un'ultima pesatura dei contributi derivanti da **ValutazioneSJ** e da **ValutazioneRJ**, ottenendo un valore, compreso tra 0 e 100, che riassume l'accuratezza generale dell'esercizio svolto dal paziente.

Il programma calcola anche un valore che rappresenta la percentuale di completamento dell'esercizio. Sulla base delle articolazioni indicate dal fisioterapista il programma confronta i punti chiave raggiunti dal paziente con i punti chiave raggiunti dal modello, fornendo in uscita il valore in percentuale in base a quanti punti chiave sono stati raggiunti. Questa informazione è riscontrabile nella stampa in output, ed è stata implementato anche un ulteriore controllo che per ogni singola articolazione dice se il paziente ha eseguito il giusto numero di movimenti oppure in eccesso/difetto.

fig.10



## Relazione dell'output

I parametri descritti nel paragrafo precedente ora devono essere stampati per poter essere consultati dal medico fisioterapista.

Le informazioni vengono stampate su un semplice file di testo *.txt*

Per ogni articolazione ci sono due livelli di approfondimento: **relazione non approfondita dell'articolazione** e **relazione approfondita dell'articolazione**.

Il medico sceglie una soglia: se la somma dei pesi su zenit e azimuth di quell'articolazione supera tale valore le informazioni stampate saranno approfondite, altrimenti saranno non approfondite.

La relazione non approfondita contiene la misura di *discostamento medio* dell'angolo/articolazione rispetto al modello, la *percentuale di accuratezza media* e una frase che descrive la *velocità di esecuzione dell'esercizio* (sia per **zenit** che per **azimut**). Oltre a tali dati, che provengono da **ValutazioneSJ**, vengono inoltre stampate le informazioni metriche relative ai *movimenti congiunti delle altre articolazioni* (sotto forma di media discostamento da angolo modello e percentuale di accuratezza media).

La relazione approfondita viene completata aggiungendo alle indicazioni appena descritte una valutazione movimento per movimento sia dei parametri di **ValutazioneSJ** che di **ValutazioneRJ**.

Ciò significa che vi sarà una *valutazione approfondita degli scostamenti*, in cui per ogni punto chiave verrà valutata la correttezza dell'angolo raggiunto (corretto, sbagliato con errore contenuto, sbagliato) ed una *valutazione approfondita della velocità per ogni movimento* (movimento corretto, leggermente veloce o lento, veloce o lento). Per quanto riguarda le metriche relative ai movimenti congiunti delle altre articolazioni sarà stampata la media di discostamento dell'articolazione dal



modello, la percentuale e un feedback anche in questo caso movimento per movimento (i movimenti si riferiscono all'articolazione alla quale la relazione si riferisce. Inoltre per ogni articolazione, sia per zenit che per azimut viene aggiunta l'informazione sul numero di movimenti eseguiti dal paziente, cioè se sono giusti, in eccesso o in difetto.