

The virtual personal trainer

Scaramuzziono Giovanna¹* and Bosso Johan Andrey²*

¹email: giovanna.scaramuzzino1@stud.unifi.it, matricola:7022462

²email: johan.bosso@stud.unifi.it, matricola:7022631

*These authors contributed equally.

Introduzione. In ambito di attività fisica come possono essere esercizi di stretching o di riabilitazione può essere utile avere una stima della qualità dell'esercizio svolto. Lo scopo del seguente elaborato è quindi quello di realizzare un personal trainer virtuale avvalendosi di strumenti terzi per l'estrazione delle features da elaborare per il raggiungimento dell'obiettivo. Si vuole verificare la correttezza di un esercizio eseguito da un utente e proposto tramite video, tramite il confronto con un video modello dove un personal trainer esegue l'esercizio correttamente.

Strumenti terzi.

OpenPose. Al fine di estrarre una sintesi dello scheletro del soggetto è stata impiegata la libreria opensource Openpose[1] che sulla base di una rete neurale provvede ad individuare frame per frame lo scheletro di uno o più utenti presenti nel video input, restituendo in output le coordinate di 25 giunti che mappano il corpo, secondo una precisa indicizzazione. Openpose inizialmente individua quante persone ci sono nel frame in analisi tramite la ricerca di una porzione del corpo con minima varianza, come ad esempio la spalla che è facilmente individuabile nel corpo e non è soggetta a eccessive distorsioni in dipendenza della particolare posa in analisi. Successivamente procede a costruire lo scheletro del soggetto con un buon grado di affidabilità. In caso in cui vi sia occlusione di una parte del corpo del soggetto, Openpose, tramite politiche di interpolazione, provvede a stimare la collocazione dei giunti non visibili fintanto che tale stima riesce a soddisfare una soglia di affidabilità, in caso contrario i giunti mancanti non vengono restituiti.

DTW. Dovendo effettuare un confronto fra due video diversi (sequenze di pose che si sviluppano in un diverso numero di frame) è stato deciso di utilizzare l'algoritmo DTW[2](Dynamic Time Warping) al fine di evitare che tale dipendenza dai frame possa influenzare l'analisi ed il confronto. Il DTW prende in analisi due sequenze temporali e tramite concetti di similarità effettua un accoppiamento restituendo quindi due sequenze che associano ad ogni istanza del primo segnale una del secondo.[fig.1] Tramite poi l'estrazione di un cammino da questa sequenza si ottengono due sotto-campionamenti sincronizzati. Nonostante fossero già presenti online diverse implementazioni di tale algoritmo si è preferito scriverne una nostra versione per meglio adattarla ad i dati a nostra disposizione.

Implementazione. L'elaborato è stato svolto in Python versione 3.6. Le dipendenze installate per eseguire il codice im-

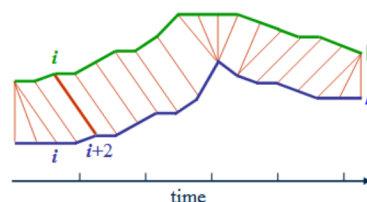


Fig. 1. Dynamic Time Warping

plementato sono:

- opencv[3]
- pyopenpose[1]
- scipy
- numpy
- math
- time

Tutte queste dipendenze possono essere installate tramite pip oppure tramite distribuzioni di terze parti come Anaconda. L'estrazione dei giunti ad opera di OpenPose è stata effettuata con il supporto di una scheda grafica GTX 1060 (Cuda), naturalmente tale componente non è indispensabile ma fortemente consigliato per elaborazioni in tempo utile. È stata implementata sia la parte di acquisizione del modello (personal trainer), all'interno del file Model_Acquisition.py, sia quella di confronto con il video utente per stabilire la correttezza dell'esercizio svolto, all'interno del file match.py. Entrambi i file devono essere avviati tramite riga di comando inserendo di seguito al nome del file .py, il path del video da esaminare ed il nome dell'esercizio.

Estrazione dei giunti e standardizzazione. Tramite la libreria pyopenpose (creata in fase di installazione del pacchetto OpenPose, con la specifica di utilizzo di questo su python) è stato possibile estrarre i 15 giunti di nostro interesse dello scheletro della persona all'interno del video. Tali informazioni sono state memorizzate frame per frame all'interno di una matrice. In caso in cui OpenPose non sia stato in grado di restituire un giunto è stato deciso di approssimare questo al suo corrispettivo nel frame precedente.

Ottenuta la matrice di tutti i giunti, tramite la lettura del file interest_point.txt (uno per ogni esercizio, contenente un elenco di giunti fondamentali per tali movimenti) si selezionano i giunti di interesse per l'esercizio in esame. Al

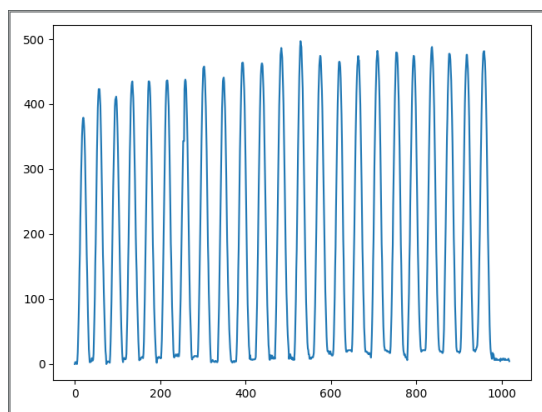


Fig. 2. Plot cicli

fine di creare uno spazio di confronto standard per ogni tipo di esercizio sono stati calcolati i punti a coordinata massima e minima per ogni frame, in modo tale da individuare il bounding-box che racchiude la persona, per poi poter eseguire le trasformazioni lineari: rotazione, traslazione e scala:

- **Rotazione:** tale trasformazione è stata effettuata in modo da vincolare due specifici giunti a giacere su una stessa retta. Ad esempio se i giunti 'neck' e 'sx_shoulder' sono fra quelli di nostro interesse si vincolano questi a giacere su una retta parallela a quella delle ascisse.
- **Traslazione:** scelto e traslato un giunto di interesse si provvede a traslare gli altri in modo tale da preservare la struttura dello scheletro.
- **Scala:** in seguito ad aver effettuato rotazione e traslazione, viene effettuato uno scaling dello scheletro in modo tale che la rappresentazione di questo risulti svincolata dalle caratteristiche della ripresa video.

Individuazione Cicli. Al fine di suddividere il video proposto in cicli è stato deciso di individuare nel giunto con massima variazione di distanza coseno, un robusto descrittore del moto. In ambito ciclico, dal grafico si può notare come la variazione di tale giunto descriva una funzione con punti di minimo e di massimo. Tagliando idealmente il grafico [fig.2] con una retta parallela a quella delle ascisse si vanno ad individuare i punti di intersezione con il grafico, per poi determinare i punti di inizio e fine ciclo come vertice di una parabola individuata dalle intersezioni. Si procede quindi alla creazione di una lista contenente vari dataframe (uno per ogni ciclo individuato)

Sincronizzazione Cicli. Per la sincronizzazione dei cicli, è stato utilizzato l'algoritmo DTW. Come ciclo principale di riferimento per la sincronizzazione è stato scelto il primo (poiché abbiamo sincronizzato tutti i cicli rispetto al tale ciclo) il quale sarà il nostro ciclo modello che verrà riutilizzato per la sincronizzazione con qualsiasi video utente che vuole confrontarsi con tale esercizio. Anche in tale caso, è stato preso in esame il giunto a massima varianza come



Fig. 3. Model Acquisition

parametro da fornire alla DTW. La sincronizzazione è stata salvata in un'unica matrice in modo tale da avere per ogni riga l'informazione di sincronizzazione nei vari cicli, ovvero la prima riga di tale matrice presenta informazioni di sincronizzazione relative al primo frame del ciclo modello e gli eventuali indici dei frame sugli altri cicli ad esso associati.

Distanza coseno. La distanza che è stata utilizzata per confrontare il video modello con quello dell'utente, è la distanza coseno. Tale scelta è stata effettuata per soddisfare criteri di robustezza: è stato notato, infatti, che la scelta di altre distanze come la distanza euclidea presentava criticità nel confronto di soggetti con corporature differenti. Utilizzando, invece la distanza coseno ci si pone in condizioni di indipendenza da tali parametri.

- **Model.csv:** contenente il dataframe relativo al primo ciclo
- **Variance.csv:** contenente le soglie
- **Medium.csv:** contenente le distanze coseno medie

Acquisizione del modello. L'acquisizione del modello prende in ingresso da linea di comando il path del video da analizzare e il nome dell'esercizio, tramite il quale individuare la directory in cui verranno collocati tutti i file utili per le valutazioni in fase di match.

In seguito viene applicata la standardizzazione del video, la divisione del video in ciclo e la sincronizzazione di tali cicli utilizzando la DTW ??.

Il primo ciclo individuato, è stato utilizzato come modello e, rispetto a tale modello sono state calcolate delle soglie utili per confrontare la correttezza dell'esercizio dell'utente. Assumendo che l'esercizio svolto dal trainer sia da considerarsi sempre corretto, tali soglie, sono state calcolate per ciascun giunto e per ciascun frame del modello, prendendo i massimi discostamenti dalle distanze coseno medie.

Conclusa la procedura di acquisizione del nuovo modello si avrà quindi la generazione dei seguenti file:

.1. Confronto con un nuovo video. Il confronto di un nuovo video si può richiamare da linea di comando (Match.py) passando come argomenti la path del video e il tipo di esercizio.

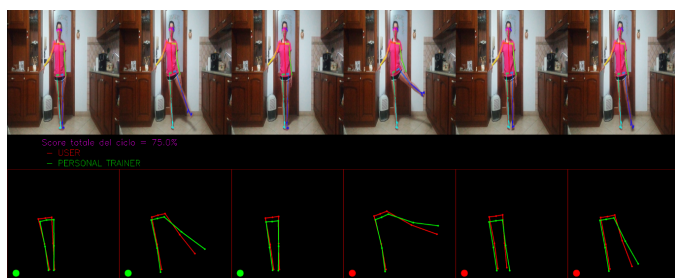


Fig. 4. Output confronto

Le informazioni estratte dal video vengono quindi processate come nel caso dell'acquisizione del modello, trasformazione lineare, individuazione cicli, suddivisione in cicli e sincronizzazione rispetto al primo ciclo del personal trainer. Si procede quindi ora a calcolare tutte le distanze coseno di interesse e si effettua il confronto con i dati statistici presenti nei file *Variaiance.csv* e *Medium.csv*. Tale confronto restituisce uno score di correttezza, in percentuale, per ogni ciclo e assegna un valore di inaccuratezza ad ogni frame sincronizzato.

Risultati. Effettuato il confronto vengono selezionati gli 8 peggiori frame, ovvero quelli a cui sono stati assegnati i valori di inaccuratezza più alti.

Tali frame vengono mostrati a video, come esempio di errori durante il singolo ciclo, in ordine crescente. In caso di alto score di correttezza potrebbe essere impossibile individuare frame errati, ed in tal caso vengono infatti mostrati alcuni frame corretti contrassegnati da un bollino verde.

Oltre ai frame utente vengono proposti a video schematizzazioni della posa utente e di quella del personal trainer.

- *Model.csv*: contenente il dataframe relativo al primo ciclo
- *Variance.csv*: contenente le soglie
- *Medium.csv*: contenente le distanze coseno medie

Conclusioni. In tale progetto è stata quindi realizzata una struttura di acquisizione e confronto utile ad un'analisi della correttezza delle pose. È da notare però che la figura umana, specialmente quando si parla di confronto fra due individui diversi, presenta diverse criticità, tra cui le più evidenti sono state risolte tramite la scelta della distanza coseno, ed altre invece non sono state trattate come le distanze coseno imposte dalla particolare conformazione fisica dei soggetti in esame. Si ipotizzi infatti di avere un modello dalla struttura longilinea ed un utente fortemente corpulento, in una posa rilassata come può essere quella di tenere le braccia lungo il tronco le angolazioni delle braccia vengono forzate dalla struttura del tronco, pertanto l'utente corpulento riscontrerà un errore in tale posa indipendente dal modo in cui esegue l'esercizio.

È possibile inoltre effettuare la definizione del modello e il confronto su campionamenti dei cicli per ridurre l'onere della computazione, tuttavia tale pratica potrebbe modificare sensibilmente i risultati poiché non si sta più considerando l'intero

set di dati ma una sua sotto porzione. Il risultato potrebbe essere inoltre influenzato da eventuali difetti di sincronizzazione della DTW, specialmente in caso si usi il campionamento dei video.

Riferimenti

- [1] *CMU-Perceptual-Computing-Lab*. URL: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>.
- [2] *Dynamic Time Warping (DTW)*. URL: <https://medium.com/data-driven-investor/dynamic-time-warping-dtw-d51d1a1e4afc>.
- [3] *OpenCV*. URL: <https://opencv.org/>.