Implementazione in Python di "Spatial Subspace Rotation"

Michele Maione

Sommario—In "A Novel Algorithm for Remote Photoplethysmography - Spatial Subspace Rotation" a cura di W. Wang, S. Stuijk e G. de Haan [1], gli autori propongono un algoritmo (2SR) per la fotopletismografia remota (rPPG).

L'algoritmo 2SR richiede il riconoscimento degli skin-pixels, e per lo scopo è stato utilizzato l'algoritmo (SkinColorFilter) proposto in "Adaptive skin segmentation via feature-based face detection" a cura di M.J. Taylor e T. Morris [2].

L'algoritmo SkinColorFilter riceve in ingresso un volto che deve essere estratto da un filmato. L'estrazione del volto è stata ottenuta mediante il classificatore a cascata Haar proposto da P. Viola e M. Jones in "Rapid Object Detection using a Boosted Cascade of Simple Features" [3].

Ambedue gli algoritmi sono stati implementati in classi Python: SSR.py, SkinColorFilter.py. Mentre per l'estrazione del volto è stata usata la libreria OpenCV.

1 Introduzione

La fotopletismografia remota consente di determinare processi fisiologici, come il flusso sanguigno, senza il contatto con la pelle. Utilizzando riprese video del viso si analizzano i cambiamenti nel colore della pelle del soggetto che non sono rilevabili dall'occhio umano.

Questa misurazione remota dei livelli di ossigeno nel sangue fornisce un'alternativa senza contatto alla fotopletismografia convenzionale.

Il sistema nervoso autonomo (ANS) è responsabile dell'attuazione delle interazioni psicosomatiche all'interno del corpo umano. Molte malattie legate allo stress sono il risultato della compromissione della funzione corporea regolata dall'ANS. Uno dei sistemi fisiologici più sensibili alle influenze regolatorie dell'ANS è il sistema cardiovascolare.

Importanza del problema: Tutti gli algoritmi rPPG esistenti sfruttano uno schema spazio-temporale comune per l'estrazione dell'impulso, che può essere generalizzato come "temporal combination of spatial color mean". Data una sequenza video contenente un soggetto, consiste di due passaggi:

1) Quantificando spazialmente i valori RGB dei pixel della pelle del soggetto in ogni singolo fotogramma (la media RGB);
2) Creando temporaneamente tracce RGB su più fotogrammi e combinandoli in un segnale.

Michele Maione - 931468 - michele.maione@studenti.unimi.it, Interazione naturale e Metodi di computazione affettiva, A/A 2018-2019, Universitá degli Studi di Milano, via Celoria 18, Milano, Italia.

La differenza fondamentale tra questi algoritmi è rappresentata dai diversi criteri utilizzati per combinare le tracce RGB in un segnale di impulso:

- "Blind Source Separation" (basato su PCA oppure su ICA), che utilizza differenti criteri per separare le tracce temporali RGB in indipendenti segnali per estrarre la pulsazione;
- CHROM, che combina linearmente i segnali di crominanza assumendo l'utilizzo di un colore di pelle standardizzato per bilanciare il bianco delle immagini;
- PBV, che utilizza la firma dei cambiamenti volumetrici del sangue in diverse lunghezze d'onda, per dinstinguere esplicitamente il cambiamento di colore indotto dalla pulsazione, tramite il rumore nelle misurazioni RGB;
- 2SR, che misura la rotazione temporale del sottospazio degli skin-pixels per l'estrazione della pulsazione.

L'algoritmo 2SR consiste di due passaggi:

1) Nel dominio spaziale, un sottospazio di skin-pixel è costruito nello spazio RGB;
2) Nel dominio temporale, l'angolo di rotazione dei sottospazi degli skin-pixels tra fotogrammi successivi viene misurato per l'estrazione dell'impulso.

Approccio seguito: L'algoritmo 2SR presuppone l'esistenza di sensori pixel della camera spazialmente ridondanti e una ben definita skin mask.

L'idea di base è di stimare un sottospazio degli skin-pixel e misurare la sua rotazione temporale per l'estrazione della pulsazione, che non richiede a priori la tonalità della pelle o degli impulsi, in contrasto con gli algoritmi esistenti.

L'estrazione del volto dal filmato è stata ottenuta mediante il classificatore a cascata Haar proposto da P. Viola e M. Jones in "Rapid Object Detection using a Boosted Cascade of Simple Features" [3].

Per il riconoscimento degli skin-pixels è stato utilizzato l'algoritmo SkinColorFilter proposto da M.J. Taylor e T. Morris. in "Adaptive skin segmentation via feature-based face detection" [2].

Ambedue gli algoritmi sono stati implementati in classi Python: SSR.py, SkinColorFilter.py. Mentre per l'estrazione del volto è stata usata la libreria OpenCV.

2 ANALISI DELLO STATO DELL'ARTE

I metodi remoti esistenti per ottenere la frequenza cardiaca da sequenze video possono essere classificati come: metodi basati sull'intensità del colore e metodi basati sul movimento. Attualmente, i metodi basati sull'intensità sono i più popolari.

2.1 Metodi basati sull'intensità

I metodi basati sull'intensità estraggono i segnali PPG catturati dalle fotocamere digitali. Il sangue assorbe la luce più dei tessuti circostanti e le variazioni volumetriche sanguigne influenzano la trasmissione della luce e la riflettanza. Ciò porta a sottili cambiamenti di colore sulla pelle umana, che sono invisibili agli occhi umani ma non alle telecamere.

Diversi modelli ottici vengono applicati per estrarre l'intensità dei cambiamenti di colore causati dal battito. L'emoglobina e l'ossiemoglobina hanno entrambe: un'elevata capacità di assorbimento nell'intervallo del colore verde e una bassa capacità nell'intervallo del colore rosso. Ma tutti e tre i canali di colore contengono informazioni PPG.

Per classificare i metodi esistenti, dividiamo la procedura di rilevamento della frequenza cardiaca in tre fasi: elaborazione del video del volto, estrazione del segnale BVP (Blood Volume Pulse) della faccia e calcolo della frequenza cardiaca.

2.1.1 Elaborazione video del volto

L'elaborazione video del volto ha lo scopo di rilevare i volti, migliorare la robustezza rispetto al movimento, ridurre gli errori di quantizzazione e preparare i segnali in primo piano per un'ulteriore estrazione del segnale BVP. Esistono vari algoritmi per questa fase rispetto a quelle d'estrazione del segnale BVP e del calcolo della frequenza cardiaca.

2.1.2 Calcolo della frequenza cardiaca

La fase di calcolo della frequenza cardiaca mira a calcolare la frequenza cardiaca dal segnale cardiaco ottenuto nella fase precedente. In questa fase, i metodi possono essere raggruppati in analisi del dominio del tempo e analisi del dominio della frequenza:

- Per l'elaborazione nel dominio del tempo, viene applicato diffusamente il rilevamento del picco, per ottenere l'intervallo tra battiti (IBI), da cui viene calcolata la frequenza cardiaca;
- Nel dominio della frequenza, viene utilizzata principalmente la densità spettrale di potenza, in cui la frequenza dominante viene presa come frequenza cardiaca. Il calcolo della frequenza cardiaca può diventare complesso per applicazioni che includono funzioni di gestione del buffer per presentare i risultati della frequenza cardiaca dopo un certo periodo di tempo.

2.2 Metodi principali

2.2.1 CVPR14 - Remote Heart Rate Measurement From Face Videos Under Realistic Situations [4]

L'algoritmo per recuperare il segnale di impulso può essere suddiviso in diversi passaggi:

- Estrazione di segnali rilevanti dalla sequenza video (vale a dire colore della pelle e colore di sfondo);
- Correzione per l'illuminazione globale;
- Eliminazione del movimento non rigido;
- Filtering.

2.2.2 CHROM - Robust Pulse Rate From Chrominance-Based rPPG [5]

L'algoritmo per recuperare il segnale di impulso può essere suddiviso in diversi passaggi:

- Estrazione dei segnali del colore della pelle dalla sequenza video;
- Proiezione del colore medio della pelle nello spazio, definito, di crominanza;
- Filtro passa banda nello spazio di crominanza;
- Costruire il segnale della pulsazione.

3 MODELLO TEORICO

L'idea principale dell'algoritmo è di misurare la rotazione temporale del sottospazio degli skin-pixels per estrarre la pulsazione. I due passi da eseguire sono:

- Nel dominio spaziale, un sottospazio di skin-pixels è costruito nello spazio RGB;
- Nel dominio temporale, l'angolo di rotazione del sottospazio tra fotogrammi consecutivi è misurato per estrarre la pulsazione.

3.1 Costruizione del sottospazio di skin-pixels

Data la matrice $V_{N\times 3}$, degli N skin-pixels vettorizzata sui canali RGB, allora la matrice simmetrica di correlazione spaziale RGB, $C_{3\times 3}$, è data da:

$$C = \frac{V^\top \cdot V}{N} \tag{1}$$

per ogni matrice simmetrica reale $N \times N$, gli autovalori $\Lambda_{3\times 3}$ sono reali e gli autovettori $U_{3\times 3}$ possono essere scelti in modo tale che siano ortogonali tra loro:

$$C = U \cdot \Lambda \cdot U^\top \tag{2}$$

per la (2) possiamo quindi scomporre C in questo modo:

$$C = \lambda_1 \cdot u_1 \cdot u_1^\top + \lambda_2 \cdot u_2 \cdot u_2^\top + \lambda_3 \cdot u_3 \cdot u_3^\top \tag{3}$$

per cui il principale autovettore u_1 è lo skin-vector dominante nel cluster degli skin-pixels, u_2 e u_3 sono direzioni di variazione e sono ortogonali a u_1.

3.2 Misurazione dell'angolo di rotazione del sottospazio tra fotogrammi

Nel dominio temporale, la pulsazione sanguigna provoca variazioni nei canali RGB e, quindi, modifica il sottospazio degli skin-pixels. Poiché il sottospazio è costruito da skin-pixels spazialmente ridondanti senza normalizzazione temporale, non possiamo usare direttamente la traslazione subspaziale per misurare l'impulso, perché le variazioni pulsatili senza normalizzazione temporale sono proporzionali all'intensità della luminanza e, quindi, in una relazione moltiplicativa. Modelliamo la relazione temporale tra due sottospazi come rotazione e scalatura istantanea:

1) La rotazione tra autovettori (cambiamento di direzione) è correlata ai diversi contributi PPG nei canali RGB;
2) Il cambiamento degli autovalori (variazione di energia) è correlato alla pulsatilità della pelle misurata.

Poiché il sottospazio $U_{3\times3}$ è costituito da autovettori con norma unitaria, i suoi cambiamenti temporali possono solo causare una rotazione. Quindi, definiamo un passo temporale con la lunghezza l per analizzare la rotazione subspaziale. Considerando il sottospazio nel primo fotogramma di un passo U_τ come riferimento, la rotazione tra $U_{t,t\leq l}$ e U_τ è la matrice di rotazione $R_{3\times3}$:

$$R = {U_t}^\top \cdot U_\tau = \begin{pmatrix} {u_1^t}^\top \\ {u_2^t}^\top \\ {u_3^t}^\top \end{pmatrix} \cdot \begin{pmatrix} u_1^\tau & u_2^\tau & u_3^\tau \end{pmatrix}. \quad (4)$$

Poiché la norma euclidea degli autovettori è normalizzata a 1, gli elementi in R rappresentano essenzialmente gli angoli del coseno tra autovettori:

$$cos(\theta_{i,j}) = \frac{{u_i^t}^\top \cdot u_j^\tau}{\|u_i^t\| \cdot \|u_j^\tau\|}. \quad (5)$$

Quindi, R può essere riscritto come:

$$R = \begin{pmatrix} cos(\theta_{11}) & cos(\theta_{12}) & cos(\theta_{13}) \\ -cos(\theta_{12}) & cos(\theta_{22}) & cos(\theta_{23}) \\ -cos(\theta_{13}) & -cos(\theta_{23}) & cos(\theta_{33}) \end{pmatrix}. \quad (6)$$

A causa dell'ortogonalità degli autovettori in U, il coseno nelle voci non diagonali misura effettivamente le variazioni dell'angolo seno. Quindi, R può essere riscritto come:

$$R = \begin{pmatrix} cos(\theta_{11}) & sin(\theta_{12}) & sin(\theta_{13}) \\ -sin(\theta_{12}) & cos(\theta_{22}) & sin(\theta_{23}) \\ -sin(\theta_{13}) & -sin(\theta_{23}) & cos(\theta_{33}) \end{pmatrix}. \quad (7)$$

La pulsazione sanguigna cambia il tono della pelle, che è in effetti lo skin-vector dominante u_1 nello spazio RGB. Quindi, solo la rotazione temporale di u_1 è considerata. Tuttavia, come spiegato in precedenza, la rotazione tra u_1^t e u_1^τ non può essere utilizzata. Misuriamo solo la rotazione tra il vettore u_1^t e il piano ortonormale come:

$$R' = \left({u_1^t}^\top\right) \cdot \begin{pmatrix} u_2^\tau & u_3^\tau \end{pmatrix} = \left({u_1^t}^\top \cdot u_2^\tau \quad {u_1^t}^\top \cdot u_3^\tau\right). \quad (8)$$

Oltre alla rotazione subspaziale, anche gli autovalori corrispondenti alla varianza/energia degli autovettori sono influenzati dalla pulsazione sanguigna. Poiché λ_i, scomposta da C, è la potenza della varianza, ne deriviamo le variazioni di scala come:

$$S = \sqrt{\lambda_1^t} \cdot diag\left(\begin{pmatrix} \sqrt{\lambda_2^\tau} & 0 \\ 0 & \sqrt{\lambda_3^\tau} \end{pmatrix}^{-1}\right) = \begin{pmatrix} \sqrt{\frac{\lambda_1^t}{\lambda_2^\tau}} \\ \sqrt{\frac{\lambda_1^t}{\lambda_3^\tau}} \end{pmatrix}. \quad (9)$$

I segnali prodotti da S rappresentano il cambiamento della scala/energia del sottospazio ruotato, che è in realtà correlato alla pulsatilità della pelle misurata. Tuttavia, se i riflessi della pelle contengono anche le variazioni di intensità dello spettro (ad es. causate dal movimento), anche S potrebbe esserne influenzato. Quindi, non possiamo solo usare gli autovalori per ricavare l'impulso. Poiché S è stimato rispetto al sottospazio di riferimento, possiamo limitare i suoi cambiamenti alla direzione della rotazione subspaziale combinando (9) con (8):

$$SR = S^\top \odot R'. \quad (10)$$

Per ottenere un SR coerente nel tempo, SR deve essere analizzato nello stesso spazio. Quindi viene proiettato nello spazio RGB originale, cosicché il problema del segno arbitrario della decomposizione degli autovalori, dopo la retroproiezione è rimosso:

$$SR' = SR \cdot \begin{pmatrix} {u_2^\tau}^\top \\ {u_3^\tau}^\top \end{pmatrix}. \quad (11)$$

Deriviamo/amplifichiamo il segnale di impulso combinando le prime due tracce anti-fase, $\vec{p}_{l\times1}$, utilizzando $S\vec{R}'_i$, che è l'i-esima traccia di SR':

$$\vec{p} = S\vec{R}'_1 - \frac{\sigma(S\vec{R}'_1)}{\sigma(S\vec{R}'_2)} \cdot S\vec{R}'_2. \quad (12)$$

Il segnale di impulso a lungo termine $\vec{P}_{K\times1}$, viene stimato da passi successivi usando l'aggiunta per sovrapposizione:

$$\vec{P}^{t-l} = \vec{P}^{t-l} + (\vec{p} - \mu(\vec{p})). \quad (13)$$

4 Simulazione e esperimenti

Sul dataset COHFACE sono stati eseguiti i test dei 6 algoritmi per la fotopletismografia remota, e sono stati comparati i risultati.

4.1 Dataset

Il dataset utilizzato è il COHFACE della Idiap Research Institute. Esso contiene 160 sequenze video RGB, della durata di 100 secondi, di 40 soggetti (12 femmine e 28 maschi) con 4 differenti livelli di illuminazione e rotazione della testa. Le registrazioni dei volti sono sincronizzate con la frequenza cardiaca e quella respiratoria del soggetto esaminato.

Le sequenze video sono state registrate con un Logitech HD C525 a una risoluzione di 640x480 pixel e

un frame rate di 20Hz, mentre i segnali fisiologici sono stati acquisiti utilizzando i dispositivi della Tought Technologies insieme alla suite software BioGraph Infiniti v5.

Nella cartella `./samples/` ci sono le 40 cartelle contenenti i video dei soggetti.

4.2 Architettura del sistema

L'algoritmo è stato sviluppato per Python 3.6, utilizzando le seguenti librerie:

- numpy (numpy 1.16, http://www.numpy.org);
- cv2 (opencv-python 4.0, http://opencv.org);
- skimage (scikit-image 0.14, http://scikit-image.org).

È stato utilizzato un classificatore a cascata Haar, che necessita del file delle definizioni collocato in: `./data/haarcascade_frontalface_default.xml`. L'algoritmo può essere eseguito con il comando:

```
python3.6 virtualHeartRate.py
    $VideoFilePath $Method
    $RangeInSeconds
    $StepsInSeconds $OutFilePath
    $ShowVideo
```

passando al parametro "Method" il valore 0. Un esempio di esecuzione è il seguente:

```
python3.6 virtualHeartRate.py
    samples/1/1/data.avi 0 10 10
    outs/out_1_1.txt 0
```

4.3 Dettagli implementativi

L'algoritmo utilizzato per la fotopletismografia remota proposto da Wang, Stuijk e de Haan [1] è il seguente:

Algorithm 1 Spatial Subspace Rotation

Input: a video sequence containing K frames

$\vec{P} = zeros(1, K)$
for $k = 1, 2, \ldots, K$ **do**
  $C^k = \frac{V^{k\top} \cdot V^k}{N}$
  $[\Lambda^k, U^k] = eigs(C^k)$
  if $\tau = k - l + 1 > 0$ **then**
    for $t = \tau, \tau + 1, \ldots, k$ **do**
      $SR' = \sqrt{\frac{\lambda_0^t}{\lambda_1^\tau}} \cdot {u_0^t}^\top \cdot {u_1^\tau}^\top + \sqrt{\frac{\lambda_0^t}{\lambda_2^\tau}} \cdot {u_0^t}^\top \cdot {u_2^\tau}^\top$
      $\vec{SR'} \leftarrow$ concatenated by SR'
    $\vec{p} = \vec{SR'}_0 - \frac{\sigma(\vec{SR'}_0)}{\sigma(\vec{SR'}_1)} \cdot \vec{SR'}_1$
    $\vec{P}^{t-l} = \vec{P}^{t-l} + (\vec{p} - \mu(\vec{p}))$

Output: the pulse-signal $\vec{P}$

4.3.1 *L'algoritmo 2 è l'implementazione in Python dell'algoritmo 1. Esso necessita di un algoritmo per l'individuazione del volto in un frame (il classificatore a cascata Haar utilizzando OpenCV) e dell'algoritmo 6 per l'estrazione degli skin-pixels.*

Gli algoritmi per il calcolo della pulsazione sono stati tratti da [1]:

- calulatePulseSignal: Calcola il segnale della pulsazione P;
- buildP: Costruisce il segnale p da aggiungere alla pulsazione P;
- buildCorrelationMatrix: Costruisce la matrice di correlazione C dalla mappa dalla matrice degli skin-pixels V.

Per estrarre gli skin-pixels sono stati usati i seguenti algoritmi tratti da [2]:

- getSkinPixels: Estrae skin-pixels dai volti;
- RGMask: Calcola la maschera Red-Green;
- generateCircularMask: Genera la maschera circolare del volto;
- removeLuma: Rimuove i pixel con valori di luminanza estremi;
- EGP: Stima i parametri gaussiani dall'immagine;
- getSkinMask: Ottiene la skin-mask.

Algorithm 2 Calcola il segnale pulsazione P

function CALULATEPULSESIGNAL($images, fps$)
  $l \leftarrow fps$ ▷ passo temporale
  $K \leftarrow images.x$ ▷ numero di frames
  $P \leftarrow zeros((K))$
  $xml \leftarrow$ "$HaarCascadeFrontalFace.xml$"
  $HaarCC \leftarrow OpenCV.CascadeClassifier(xml)$
  for $k \leftarrow 0$ **to** K **do**
    $img \leftarrow images[k]$
    $faces \leftarrow HaarCC.detectMultiScale(img)$
    if $len(faces) = 1$ **then**
      $V \leftarrow getSkinPixels(faces, img, k = 0)$
      $C \leftarrow buildCorrelationMatrix(V)$ ▷ $C_{3\times3}$
      $\Lambda^k, U^k \leftarrow eigs(C)$
      if $k \geq l$ **then**
        $\tau \leftarrow k - l$
        $p \leftarrow buildP(\tau, k, l, U, \Lambda)$ ▷ $p_{l\times1}$
        $P^{t-l} \leftarrow P^{t-l} + p$
  return P

Algorithm 3 Costruisce il segnale p

function BUILDP(τ, k, l, U, Λ)
  $SR \leftarrow zeros((3, l))$
  for $t \leftarrow \tau$ **to** k **do**
    $A \leftarrow \sqrt{\frac{\Lambda_0^t}{\Lambda_1^\tau}} \cdot {U_0^t}^\top \otimes {U_1^\tau} \otimes {U_1^\tau}^\top$ ▷ $A_{3\times1}$
    $B \leftarrow \sqrt{\frac{\Lambda_0^t}{\Lambda_2^\tau}} \cdot {U_0^t}^\top \otimes {U_2^\tau} \otimes {U_2^\tau}^\top$ ▷ $B_{3\times1}$
    $SR^t \leftarrow A + B$
  $p \leftarrow SR_0 - \frac{\sigma(SR_0)}{\sigma(SR_1)} \cdot SR_1$
  $p' \leftarrow \mu(p)$
  return p' ▷ $p'_{l\times1}$

Algorithm 4 Costruisce la matrice di correlazione da V

function BUILDCORRELATIONMATRIX(V)
  $N \leftarrow V.y$ ▷ numero di skin-pixels
  $C \leftarrow \frac{V^\top \cdot V}{N}$ ▷ $V_{3\times N}$
  return C ▷ $C_{3\times 3}$

Algorithm 5 Ottiene autovalori e autovettori, ordinali

function EIGS(C)
  $\Lambda, U = eig(C)$ ▷ numpy.linalg.eig
  $idx = \Lambda.argsort()$ ▷ numpy.argsort
  $\Lambda' = \Lambda_{idx}$ ▷ ordina autovalori
  $U' = U_{idx}$ ▷ ordinati in base agli autovalori
  return Λ', U' ▷ $\Lambda'_{3\times 1}, U'_{3\times 3}$

Algorithm 6 Estrae skin-pixels dai volti

function GETSKINPIXELS($faces, img, doSkinInit$)
  for $(x, y, w, h) \leftarrow faces$ **do**
    $ROI \leftarrow img[y : y + h, x : x + w]$ ▷ $ROI_{w\times h\times 3}$
  if $doSkinInit$ **then**
    $M, C, CI, MU \leftarrow EGP(ROI)$
  $skinMask \leftarrow getSkinMask(ROI, CI, MU, 0.5)$
  $V \leftarrow ROI[skinMask]$
  return V ▷ $V_{w\cdot h\times 3}$

Algorithm 7 Calcola maschera Red-Green

function RGMASK(img)
  $R \leftarrow zeros((img.x, img.y))$
  $G \leftarrow zeros((img.x, img.y))$
  $channelSum \leftarrow img_0 + img_1 + img_2$
  $non0mask \leftarrow img_0 \vee img_1 \vee img_2$
  $R[non0mask] \leftarrow \frac{img[non0mask,0]}{channelSum[non0mask]}$
  $G[non0mask] \leftarrow \frac{img[non0mask,1]}{channelSum[non0mask]}$
  return R, G ▷ $R_{img.x\times img.y}, G_{img.x\times img.y}$

Algorithm 8 Maschera circolare del volto

function GENERATECIRCULARMASK($img, rRatio$)
  $w \leftarrow img.x$
  $h \leftarrow img.y$
  $xCenter \leftarrow \frac{w}{2}$
  $yCenter \leftarrow \frac{h}{2}$
  $X \leftarrow zeros((w, h))$
  $Y \leftarrow zeros((h, w))$
  $X_x \leftarrow range(0, w)$
  $Y_x \leftarrow range(0, h)$
  $Y \leftarrow Y^\top$
  $X \leftarrow X - xCenter$
  $Y \leftarrow Y - yCenter$ ▷ sposta il centro nell'origine
  $r \leftarrow rRatio \cdot h$
  $cMask \leftarrow X^2 + Y^2 < r^2$ ▷ matrice booleana
  return $cMask$ ▷ $cMask_{w\times h}$

Algorithm 9 Rimozione pixel con valori luma estremi

function REMOVELUMA($img, cMask$)
  $R \leftarrow 0.299 \cdot img[cMask, 0]$
  $G \leftarrow 0.587 \cdot img[cMask, 1]$
  $B \leftarrow 0.114 \cdot img[cMask, 2]$
  $luma \leftarrow R + G + B$
  $m \leftarrow \mu(luma)$ ▷ $numpy.mean$
  $s \leftarrow \sigma(luma)$ ▷ $numpy.std$
  $R \leftarrow 0.299 \cdot img_0$ ▷ $R_{img.x\times img.y}$
  $G \leftarrow 0.587 \cdot img_1$ ▷ $G_{img.x\times img.y}$
  $B \leftarrow 0.114 \cdot img_2$ ▷ $B_{img.x\times img.y}$
  $l \leftarrow R + G + B$ ▷ $l_{img.x\times img.y}$
  $lMask \leftarrow l > (m - 1.5 \cdot s) \wedge l < (m + 1.5 \cdot s)$
  return $lMask$ ▷ $lMask_{img.x\times img.y}$

Algorithm 10 Stima parametri gaussiani

function EGP(img)
  $cMask \leftarrow generateCircularMask(img, 0.4)$
  $lMask \leftarrow removeLuma(img, cMask)$
  $mask \leftarrow lMask \wedge cMask$ ▷ $mask_{img.x\times img.y}$
  $R, G \leftarrow RGMask(img)$
  $MU \leftarrow [\mu(R[mask]), \mu(G[mask])]$ ▷ $MU_{2\times 1}$
  $R' = R[mask] - MU[0]$
  $G' = G[mask] - MU[1]$
  $S \leftarrow [R', G']$
  for $s \leftarrow S^\top$ **do** ▷ $s_{1\times 2}$
    $A \leftarrow s \otimes s$ ▷ $A_{2\times 2}$
  $C \leftarrow sum(A)$ ▷ $C_{2\times 2}$
  if $det(C) \neq 0$ **then**
    $CI \leftarrow C^{-1}$
  else
    $CI \leftarrow zerosLike(C)$
  return $mask, C, CI, MU$

Algorithm 11 Ottiene la skin-mask

function GETSKINMASK($img, CI, MU, threshold$)
  $R, G \leftarrow RGMask(img)$
  $R' \leftarrow R - MU[0]$ ▷ $R'_{img.x\times img.y}$
  $G' \leftarrow G - MU[1]$ ▷ $G'_{img.x\times img.y}$
  $V \leftarrow [R', G']$ ▷ $V_{img.x\times img.y\times 2}$
  $n \leftarrow img.x \cdot img.y$
  $V' \leftarrow V.reshape(n, 2)$ ▷ $V'_{n\times 2}$
  for $k \leftarrow V'$ **do**
    $probs \leftarrow k \cdot (CI \cdot k)$ ▷ $probs_{img.x\times img.y}$
  $skinMap \leftarrow exp(-0.5 \cdot probs)$
  return $skinMap > threshold$

5 RISULTATI OTTENUTI

Dalla comparazione dei 6 algoritmi (vedere Appendice) si evince che con riprese frontali con buona illuminazione artificiale (set 0) l'algoritmo 2SR ha risultati simili alla maggioranza degli altri approcci. Invece con le riprese frontali con illuminazione solare laterale (set 3) 2SR dà il risultato migliore rispetto agli altri algoritmi.

Gli algoritmi comparati sono i seguenti:

- Approccio 0: 2SR: Spatial Subspace Rotation;
- Approccio 1: JadeR: Blind separation of real signals with JADE;
- Approccio 2: BFF: Principal component analysis (PCA);
- Approccio 3: EVM: Eulerian Video Magnification (Eulerian vs. Lagrangian Processing);
- Approccio 4: rHRV: Remote heart rate variability for emotional state monitoring;
- Approccio 5: D&D: Detrending and demeaning + Butterworth bandpass filter.

6 COMMENTI CONCLUSIVI

Secondo i risultati riportati in [1], 2SR, data una ben definita skin mask per misurare il cluster della distribuzione degli skin-pixels, supera sia il popolare approccio ICA, sia i due algoritmi all'avanguardia CHROM e PBV, in condizioni di differenti tonalità della pelle, movimenti del corpo e complesse condizioni d'illuminazione.

Dai risultati pubblicati in "A Reproducible Study on Remote Heart Rate Measurement" [6], si evince che 2SR è fortemente dipendente dall'algoritmo utilizzato per il riconoscimento degli skin-pixels.

Utilizzando l'algoritmo 2SR in combinazione con SkinColorFilter [2] e il classificatore Haar [3], sul dataset COHFACE abbiamo ottenuto risultati simili agli altri algoritmi in condizioni di illuminazione controllata, mentre in condizioni di luce solare abbiamo ottenuto i risultati migliori.

RIFERIMENTI BIBLIOGRAFICI

[1] W. Wang, S. Stuijk, and G. de Haan, "A novel algorithm for remote photoplethysmography: Spatial subspace rotation," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 9, pp. 1974–1984, 2016.

[2] M. J. Taylor and T. Morris, "Adaptive skin segmentation via feature-based face detection," *Proc SPIE Photonics Europe*, 2014.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Accepted Conference on Computer Vision and Pattern Recognition 2001*. IEEE, 2001.

[4] X. Li, J. Chen, G. Zhao, and M. Pietikäinen, "Remote heart rate measurement from face videos under realistic situations," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

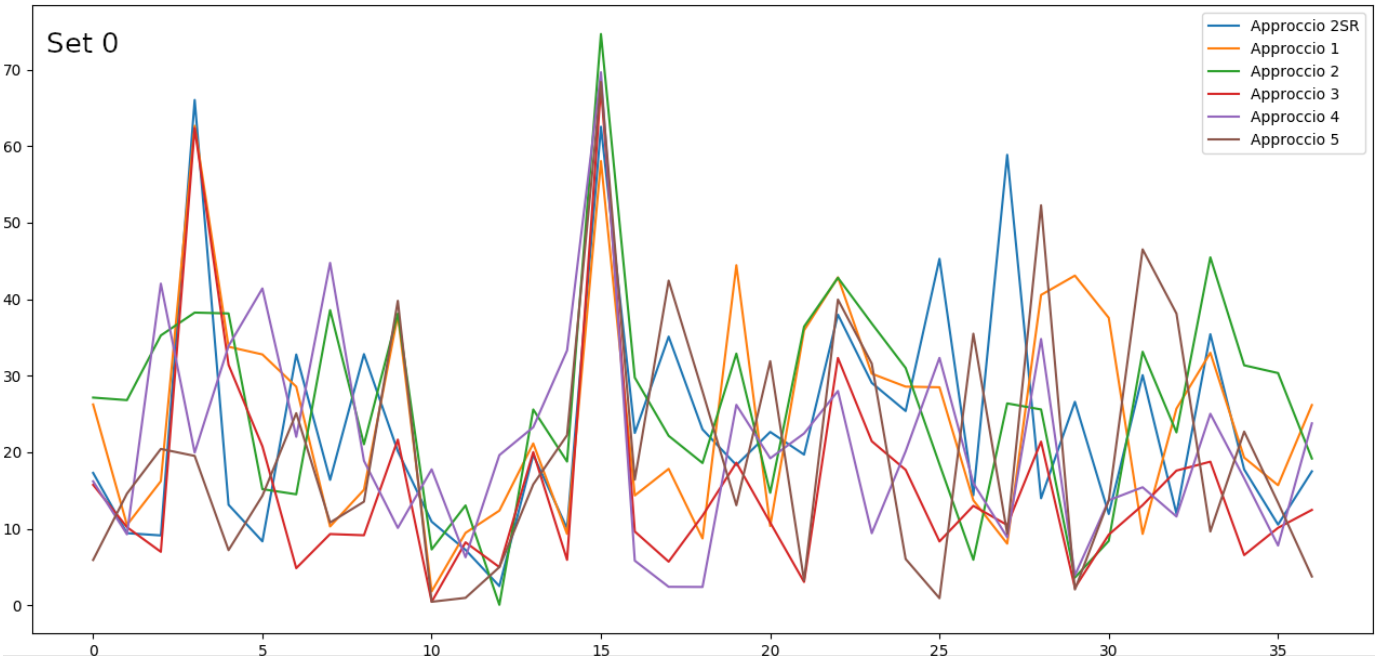
[5] G. de Haan and V. Jeanne, "Robust pulse rate from chrominance-based rppg," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2878–2886, 2013.

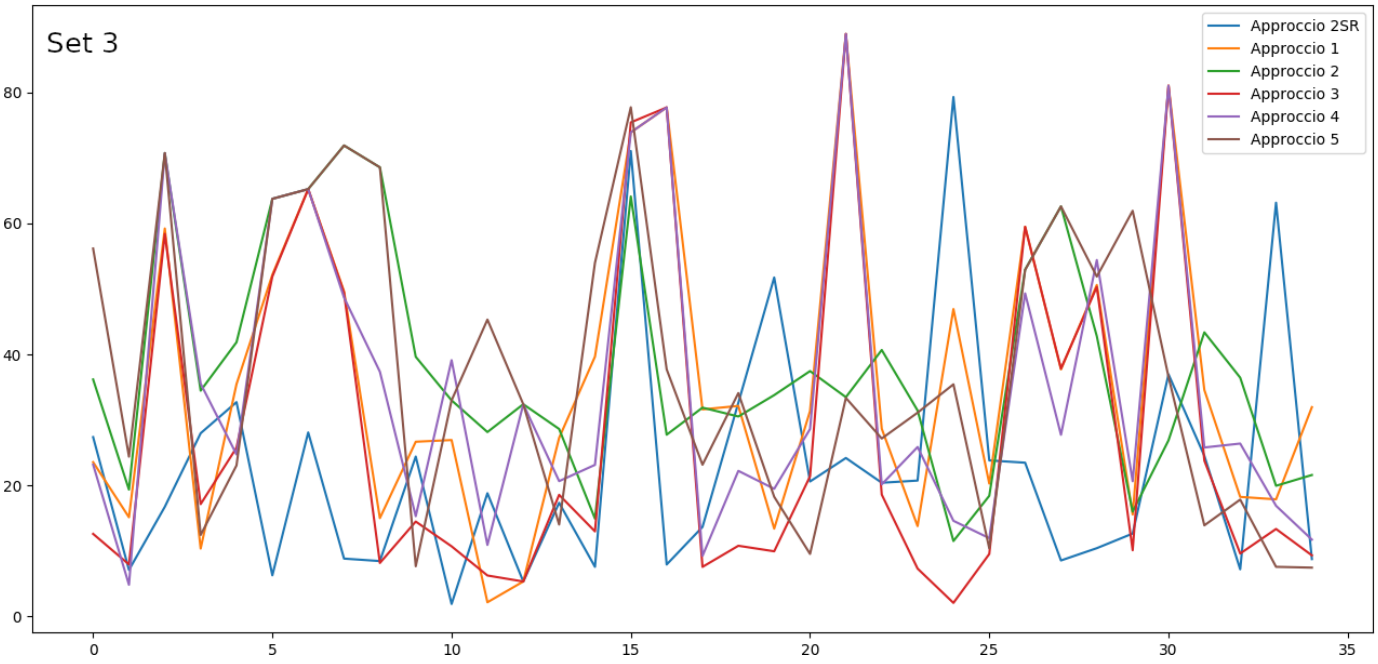
[6] G. Heusch, A. Anjos, and S. Marcel, "A reproducible study on remote heart rate measurement." Idiap Research Institute, 2017.

APPENDICE

Nei grafici e nelle tabelle sono riportati i risultati ottenuti per soggetto e la radice dell'errore quadratico medio.

Set 0						
Sogg.	2SR	JadeR	BFF	EVM	rHRV	D&D
1	17.30	26.25	27.15	15.75	16.20	**5.93**
2	9.43	10.50	26.82	10.21	**9.28**	14.60
3	9.14	16.23	35.27	**6.99**	42.04	20.44
4	66.04	62.66	38.25	62.37	19.97	**19.53**
5	13.14	33.79	38.14	31.43	33.89	**7.21**
6	**8.36**	32.79	15.18	20.78	41.41	14.32
7	32.78	28.57	14.51	**4.85**	22.02	25.11
8	16.39	10.31	38.58	**9.32**	44.75	10.82
9	32.83	15.10	21.05	**9.16**	18.89	13.54
10	20.14	37.91	38.14	21.66	**10.11**	39.79
12	10.93	1.81	7.30	**0.46**	17.77	**0.46**
13	7.25	9.47	13.05	8.23	6.28	**0.98**
14	2.51	12.38	**0.07**	4.99	19.63	4.99
15	19.67	21.15	25.59	20.02	23.27	**15.81**
16	10.06	9.33	18.78	**5.94**	33.31	22.24
17	62.54	**58.05**	74.66	67.74	69.66	68.39
18	22.52	14.36	29.76	9.65	**5.83**	16.43
19	35.12	17.84	22.15	5.71	**2.42**	42.44
20	22.99	8.74	18.60	11.71	**2.40**	27.95
21	18.32	44.44	32.90	18.64	26.20	**13.08**
22	22.66	**10.37**	14.71	10.81	19.21	31.91
23	19.70	35.93	36.42	**3.05**	22.46	3.25
24	37.99	42.86	42.81	32.33	**28.01**	39.96
25	29.07	30.27	36.83	21.44	**9.43**	31.59
26	25.39	28.58	31.02	17.70	20.13	**6.06**
28	45.28	28.50	18.44	8.35	32.34	**0.93**
29	14.38	13.72	**5.95**	13.00	15.97	35.49
30	58.87	**8.06**	26.39	10.50	8.90	9.39
31	**13.99**	40.56	25.60	21.39	34.80	52.28
32	26.59	43.09	3.63	2.29	3.93	**2.07**
33	11.94	37.57	**8.42**	9.24	13.73	13.72
34	30.08	**9.33**	33.14	13.09	15.43	46.52
35	11.98	25.71	22.60	17.61	**11.61**	38.13
36	35.43	33.00	45.47	18.77	25.03	**9.65**
37	17.76	19.31	31.37	**6.57**	16.59	22.70
38	10.55	15.69	30.35	10.13	**7.82**	13.42
39	17.51	26.17	19.18	12.47	23.78	**3.76**
AVG	23.42	24.88	26.17	**15.52**	20.93	20.13
Set 3						
Sogg.	2SR	JadeR	BFF	EVM	rHRV	D&D
1	27.37	23.54	36.18	**12.58**	23.17	56.18
2	7.07	15.12	19.35	7.91	**4.81**	24.38
3	**16.72**	59.24	70.76	58.42	70.76	70.76
4	27.96	**10.34**	34.46	17.13	35.53	12.42
5	32.71	35.46	41.86	25.75	24.72	**23.04**
6	**6.25**	52.17	63.77	51.90	63.77	63.77
7	**28.08**	65.25	65.25	65.25	65.25	65.25
8	**8.81**	49.63	71.92	49.41	48.69	71.92
9	8.43	14.99	68.58	**8.14**	37.35	68.58
10	24.40	26.66	39.64	14.47	15.29	**7.64**
12	**1.88**	26.92	32.98	10.68	39.12	32.98
13	18.79	**2.13**	28.14	6.23	10.89	45.32
14	**5.31**	**5.31**	32.39	**5.31**	32.39	32.39
15	17.33	27.25	28.61	18.55	20.64	**14.01**
16	**7.55**	39.66	14.91	12.96	23.13	53.99
17	71.07	73.92	**64.13**	75.44	73.93	77.76
18	**7.90**	77.73	27.73	77.73	77.73	37.72
19	13.64	31.60	31.87	**7.54**	9.23	23.15
21	32.61	32.13	30.52	**10.77**	22.21	34.09
22	51.75	13.38	33.78	**9.92**	19.47	18.25
23	20.58	31.35	37.45	21.38	28.61	**9.52**
25	**24.18**	88.95	33.47	88.95	88.95	33.35
26	20.39	28.65	40.68	**18.59**	20.14	27.15
28	20.73	13.77	31.43	**7.28**	25.85	31.10
29	79.32	46.94	11.49	**2.04**	14.58	35.43
30	23.81	20.26	18.40	**9.51**	11.93	10.37
31	**23.46**	59.50	52.96	59.50	49.32	52.91
32	**8.52**	37.71	62.61	37.83	27.72	62.61
33	**10.40**	50.62	42.78	50.31	54.42	51.89
34	12.63	15.54	15.99	**10.08**	20.62	61.94
35	36.96	81.05	**26.87**	81.05	81.05	36.59
36	24.51	34.57	43.37	23.77	25.78	**13.89**
37	**7.17**	18.23	36.45	9.58	26.38	17.82
38	63.17	17.86	19.93	13.34	16.87	**7.55**
39	8.73	31.94	21.58	9.30	11.71	**7.43**
AVG	**22.86**	35.98	38.07	28.25	34.91	36.95


Set 0
Approccio 2SR
Approccio 1
Approccio 2
Approccio 3
Approccio 4
Approccio 5
0
10
20
30
40
50
60
70
0
5
10
15
20
25
30
35


Set 3
Approccio 2SR
Approccio 1
Approccio 2
Approccio 3
Approccio 4
Approccio 5
0
20
40
60
80
0
5
10
15
20
25
30
35