



UNIVERSITÀ DEGLI STUDI DI BRESCIA

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Triennale

Acquisizione del battito cardiaco mediante
tecniche fotopletismografiche e
strumentazione non dedicata

Laureando
Patrizio Giuliano Bertoni
Matricola 79021

Relatore
Ing. Nicola Adami
Anno Accademico 2011/2012

A Leo Stefanini

Indice

Indice	iii
1 Introduzione	1
1.1 Il battito cardiaco	1
1.2 Scopo della tesi	2
2 Elementi di teoria	3
2.1 Fotopletismografia	3
2.2 Sistemi dinamici LTI SISO	3
2.3 Processi stocastici	6
2.4 Elaborazione numerica dei segnali	8
2.5 Analisi delle componenti indipendenti	13
3 Acquisizione del battito cardiaco	15
3.1 Premessa	15
3.2 Acquisizione del segnale video	15
3.3 Individuazione del fenomeno	16
3.4 Rifinitura dei segnali	17
3.5 Analisi spettrale	25
4 Risultati sperimentali	27
4.1 Legenda e intepretazione	27
4.2 Un ottimo esempio	28
4.3 Un medio esempio	30

5 Conclusioni e sviluppi futuri	35
 Appendice A	
Supporto informatico	36
C/C++	36
OpenCV	39
Matlab	40
FFTW	41
Diagramma delle classi	42
 Appendice B	
Parametri di programma	43
 Appendice C	
Note sulla portabilità	45
Ambiente di sviluppo	45
 Bibliografia	47

Capitolo 1

Introduzione

1.1 Il battito cardiaco

L'informazione rappresentata dal numero di battiti cardiaci in un minuto (in seguito riferito, per brevità, con il termine battito cardiaco, heart rate o l'acronimo HR) è sempre stata utilizzata dall'essere umano come primitivo sapere sulla salute generale di una persona. Il metodo tradizionalmente usato per conoscere il battito cardiaco è l'esercitazione di una pressione con le dita sul polso del soggetto, sebbene qualsiasi parte del corpo abbia lo stesso valore di HR. Ci si affida in questo modo al solo proprio senso tattile.

L'auscultazione tramite stetoscopio in epoca moderna ha fornito alla scienza medica un sensore più affidabile per il problema in esame, sensore che però in certi casi non si dimostra ottimale al contesto nel quale si voglia effettuare la misura. Alcuni di questi appartengono nel settore clinico e riguardano persone soggette a fenomeni di aritmia o irregolarità. Altri casi, certo meno gravi, rientrano nell'ambito di attività di fitness o motorie, dove i sensori per la misurazione elencati si dimostrano veramente poco pratici e immediati.

L'interesse del laureando si è concentrato proprio su questo ambito, in particolare sull'utilizzo delle strumentazioni elettroniche come calcolatori, videocamere e smartphones, atte a realizzare un sistema di acquisizione del battito cardiaco per via fotopletismografica.

1.2 Scopo della tesi

Questo lavoro si propone di dimostrare come il laureando abbia saputo comprendere un fenomeno fisico e implementare un algoritmo che ne dia un modello sufficientemente fedele, avendo come traccia l'articolo *Advancements in noncontact, multiparameter physiological measurements using a webcam* [1] di Ming-Zher Poh.

Ipotesi cardine dell'elaborato è che l'acquisizione dei segnali dai quali si stimerà il battito cardiaco avvenga mediante una videocamera commerciale di bassa-media gamma, come può essere una webcam da personal computer che supporti risoluzioni come la 640x480 e la 1280x720 pixel, comunemente diffuse. Questo vincolo giustifica il didascalico "strumentazione non dedicata" nel titolo della tesi.

La congettura di Poh è di ottenere i suddetti segnali attraverso la registrazione, debitamente illuminata da incandescenza, della regione facciale del soggetto; il laureando, dopo aver cercato di seguirne l'esempio ed aver ottenuto un segnale troppo pieno di rumore per poter apprezzare i risultati, ha configurato il proprio software per acquisire tali dati da un dito del soggetto, retroilluminato e posto molto vicino all'obiettivo fotografico. Le misure a questo punto sono state considerate soddisfacenti e verranno mostrate nel capitolo "Risultati sperimentali".

Non è assolutamente tra gli obiettivi del laureando la smentita o la denigrazione nemmeno parziale del lavoro del Dottor Poh: si asserisce semplicemente di non essere riusciti a ottenerne un'identica versione apprezzabile e, salvo il punto ora discusso, si è data prova del suo corretto funzionamento.

Capitolo 2

Elementi di teoria

2.1 Fotopletismografia

La fotopletismografia (photoplethysmography in seguito riferita, per brevità, con l'acronimo inglese *PPG*) è una tecnica non invasiva per ricavare la pulsazione del volume sanguigno (Blood Volume Pulse, *BVP*). Durante il ciclo cardiaco, variazioni volumetriche dei vasi sanguigni modificano le lunghezze d'onda della luce rifratta dai tessuti, in modo che le consecutive variazioni temporali dell'intensità luminosa riflessa dall'epiderma possano fornire informazioni sui periodi del ciclo. L'inverso del periodo tra un battito e l'altro, moltiplicato per 60 secondi, è definito come *heart rate*.

Sebbene la PPG necessiti tradizionalmente di sorgenti luminose infrarosse, Verkruyse [2] ha mostrato che le misure degli impulsi sul viso umano sono ottenibili in normali condizioni ambientali, qualora siano sufficientemente irradiate dal sole, da incandescenza o addirittura da una potente sorgente LED.

2.2 Sistemi dinamici LTI SISO

Moltissimi sistemi fisici possono essere più o meno fedelmente modellizzati come sistemi *dinamici* (ovvero dove l'evoluzione dello stato dipende dallo stato stesso), *lineari* (dove vale il principio di sovrapposizione degli effetti), *tempo-invarianti* o *stazionari* (ovvero il cui comportamento in un lasso temporale finito è indipendente dal tempo d'inizio dello svolgimento del processo, a parità di tutte le altre condizioni). Infine è possibile definire

sistemi *a singola entrata* (input) e coerentemente *a singola uscita* (output). Questi sistemi d'ora in avanti saranno per brevità riferiti come sistemi LTI SISO, o semplicemente LTI.

2.2.1 Convoluzione

La convoluzione è un'operazione matematica commutativa tra due funzioni integrabili avente come prodotto una terza funzione che esprime la conseguenza dell'interazione tra i due operandi, in un periodo storico o rispetto a qualche altra dimensione significativa per il sistema [14]. Nel contesto dei sistemi LTI SISO, definito $x(t)$ il segnale in ingresso al sistema e $h(t)$ la risposta del medesimo all'impulso di Dirac $\delta(t)$, si ha che l'uscita $y(t)$ è la convoluzione tra $x(t)$ e $h(t)$. A seconda della tipologia dei segnali in esame, si parlerà di convoluzione lineare o circolare.

2.2.1.1 Lineare

La convoluzione lineare tra due funzioni che rispettano le ipotesi citate è scrivibile come

$$y(t) \triangleq x(t) * h(t) = \int_{\mathbb{R}} x(\tau)h(t - \tau) d\tau = \int_{\mathbb{R}} x(t - \tau)h(\tau) d\tau \quad (2.1)$$

2.2.1.2 Circolare

Data una funzione periodica di periodo T , la sua convoluzione con una funzione $h(t)$ è ancora una funzione periodica ed è scrivibile come

$$y(t) \triangleq x(t) \circledast h(t) = \int_{t_0}^{t_0+T} h_T(\tau)x(t - \tau) d\tau \quad (2.2)$$

essendo $h_T(t)$ la sommazione periodica di $h(t)$ definita come

$$h_T(t) \triangleq \sum_{k \in \mathbb{Z}} h(t - kT) \quad (2.3)$$

2.2.2 Risposta in Frequenza

L'analisi frequenziale è una teoria chiave nello studio dei sistemi LTI SISO. L'argomento è vastissimo e ne verrà ora presentata traccia dei concetti fondamentali. L'idea di 'risposta' coincide con il determinare, se esiste, una relazione chiusa tra ingresso e uscita di un sistema LTI. I vantaggi sono portentosi: equazioni integrodifferenziali ottengono una

rappresentazione algebrica in un altro dominio matematico, detto ‘delle trasformate’. La grande limitazione che essa impone è però di prescindere dallo stato attuale, determinante (per definizione) nel concetto di ‘sistema dinamico’. Se da un lato l’analisi frequenziale permette di semplificare sensibilmente i modelli matematici che si andranno ad applicare, dall’altro la perdita di informazione sullo stato non permette di conoscere l’evoluzione dei sistemi durante i transitori, ma solo quando si è raggiunta una condizione di asintotica stabilità detta *di regime*. I problemi reali che non possono ignorare l’informazione di stato afferiscono prevalentemente all’area dell’Automatica e del Controllo; nel Digital Signal Processing lo studio nel dominio frequenziale è per fortuna spesso applicabile.

2.2.3 Trasformata di Fourier

Il principale strumento del quale si avvale l’analisi in frequenza di un segnale temporale è la sua trasformata integrale di Fourier (riferita, per brevità, equivalentemente col termine *spettro del segnale*), che può esser vista come un caso degenere della trasformata integrale di Laplace. Sia $x(t)$ una funzione a decrescenza rapida definita su un sottoinsieme di \mathbb{R} , la sua F-trasformata $X(f)$ è descritta come:

$$X(f) \triangleq \int_{\mathbb{R}} x(t) e^{-j2\pi f t} dt \quad (2.4)$$

Essendo una funzione complessa, è possibile riferirvisi in termini di *spettro d’ampiezza* o *di modulo* e *spettro di fase*.

2.2.3.1 Trasformata discreta

Sono qui riassunte le considerazioni principali da compiere nel passaggio da analogico a numerico, senza oberare il lettore con troppe formalizzazioni matematiche. Il fatto fondamentale che avviene trae origine dalla trasformata del pettine di Dirac:

$$\delta_T(t) \longleftrightarrow \frac{1}{T} \delta_{T^{-1}}(f) \quad (2.5)$$

e da una possibile espressione del teorema della risposta in frequenza:

$$x_1(t) * x_2(t) \longleftrightarrow X_1(f)X_2(f) \quad (2.6)$$

La verità congiunta dei predicati implica che un segnale campionato, quale può essere un segnale numerico (una *sequenza*) visto nel dominio analogico, sia associato a uno spettro

periodico. La trasformata di Fourier *a tempo discreto* (DTFT) di una sequenza sarà quindi un segnale periodico.

Se inoltre si vede la sequenza $x(n)$, che si ipotizza periodica a supporto finito, come fosse un periodo fondamentale di essa stessa, per dualità si avrà uno spettro periodico e campionato. Sono queste le basi della trasformata *discreta* di Fourier (DFT o FFT, a seconda di dettagli implementativi).

2.3 Processi stocastici

2.3.1 Concetto di processo

Sia definito uno spazio di probabilità con $\Omega = \{\omega_i\}$ spazio campione (finito o meno) degli eventi ω_i , S una classe di eventi e $f(\cdot)$ una legge di probabilità definita su S . Sia individuato un set di funzioni del tempo $x_i(t)$ in numero pari ai risultati ω_i . Un processo stocastico $X(t, \omega)$ è una *corrispondenza* che associa a ciascun evento possibile una delle suddette funzioni, le quali prendono il nome di *realizzazioni* [6]. È possibile pensare un processo stocastico come una variabile casuale X che evolve nel tempo: come si nota in figura, il valore del processo in un generico istante t_c è il prodotto di un'estrazione da X .

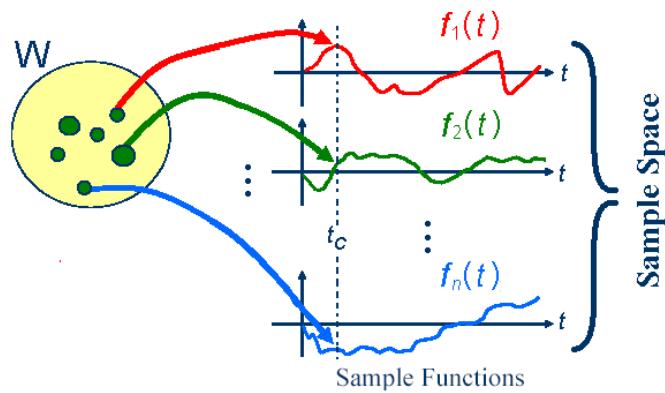


Figura 2.1: Tre realizzazioni casuali di un processo.

Si definisce la *stazionarietà* di un processo come l'invarianza nel tempo delle sue proprietà statistiche: si parla di stazionarietà *forte* quando in ogni istante possibile le densità

di probabilità sono le medesime; *debole di ordine* 2 quando a preservarsi sono solo i momenti statistici fino al secondo ordine, ovvero media e varianza. Si definisce *ergodico* un processo $X(t, \omega)$ tale per cui una sua singola realizzazione contiene sufficiente informazione per conoscere tutte le proprietà statistiche di $X(t, \omega)$. Le definizioni *forte* e *debole* ricalcano quelle viste per la stazionarietà. Si noti infine che l'ergodicità implica la stazionarietà.

2.3.2 Processi Gaussiani

I processi stocastici gaussiani hanno in ogni istante distribuzioni probabilistiche tipiche di una variabile aleatoria gaussiana. Sono spesso protagonisti nella scelta degli input per la modellizzazione e/o progettazione di sistemi di elaborazione, in quanto godono di proprietà e teoremi unici e significativi. Il teorema del limite centrale garantisce che la somma di un elevato numero di variabili casuali tende a disporsi come una variabile gaussiana, il che giustifica la possibile assunzione di gaussianità per molti processi. Un gaussiano se è stazionario lo è necessariamente in modo forte e debole, ed è completamente caratterizzato da media e varianza. Si scriverà quindi $X \sim N(\mu_X, \sigma_X^2)$.

2.3.3 Standardizzazione

La standardizzazione, detta equivalentemente normalizzazione, è una trasformazione lineare che mappa una variabile casuale X di media e varianza note (μ_X, σ_X^2) in un'altra variabile casuale Z avente media nulla e varianza unitaria. E' particolarmente utile per le variabili gaussiane, in quanto consente di riferirsi a dei dati tabulati per conoscere i valori della funzione di ripartizione evitando il calcolo di integrali non elementari [14]. Si definisce pertanto la standardizzazione come

$$Z = \frac{X - \mu_X}{\sigma_X} \quad (2.7)$$

2.3.4 Rumore Bianco

I processi bianchi sono caratterizzati da un diagramma di densità interspettrale di potenza costante sull'asse delle frequenze, il che implica la totale ricchezza armonica delle loro realizzazioni. Per le proprietà fondamentali delle trasformate questo significa che godono di funzioni di autocorrelazione impulsive, perciò in ogni istante di tempo diverso da quello

attuale il valore del processo è completamente scorrelato da quello che si ha per l’istante in esame.

2.4 Elaborazione numerica dei segnali

2.4.1 Campionamento e aliasing

Il passaggio dall’informazione analogica a quella digitale non è senza inconvenienti: se da un lato è intuitivo che un insieme finito di dati sia meno rappresentativo di uno non finito (sebbene questo non sia sempre vero), nella pratica avviene effettivamente un fenomeno di bias, riducibile a piacere ma non eliminabile. Dai concetti base dell’analisi frequenziale infatti viene postulato quanto segue:

- un qualsiasi segnale temporale a supporto finito, quali sono i segnali reali, ha necessariamente uno spettro a supporto infinito, e viceversa.
- un qualsiasi segnale temporale campionato è dotato di spettro periodico nel dominio delle frequenze (come si è visto nella sezione “Trasformata di Fourier”).

Il teorema fondamentale del campionamento (sampling) di Shannon-Nyquist fissa come frequenza minima di sampling $f_c \triangleq 2f_N$, essendo f_N la frequenza più alta contenuta nel segnale d’interesse. Noto quindi che il prodotto nei tempi tra un pettine di Dirac e un segnale $x(t)$ convolve in frequenza $X(f)$ con un pettine di periodo inverso, appare chiara l’opportunità che questa operazione possa non rispettare la non sovrapposizione (e in tal caso creare degli alias) dello spettro stesso.

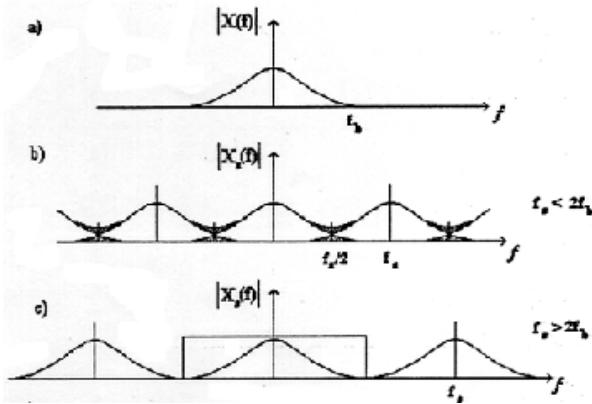


Figura 2.2: (a) uno spettro d'ampiezza generico; (b) un sampling insufficiente con generazione di aliases; (c) un sampling che rispetta la frequenza di Shannon-Nyquist.

2.4.2 Interpolazione

Nell'analisi numerica sono detti interpolanti quei metodi per individuare nuovi punti di uno spazio cartesiano sulla base di un set finito di punti conosciuti, sotto l'ipotesi che sia i primi che i secondi afferiscano a una stessa funzione $f(\mathbf{x})$ non nota a priori [14]. È possibile rilassare questa condizione molto stretta su $f(\mathbf{x})$ ponendo criteri di somiglianza: questa tecnica, sebbene non propriamente interpolante, è detta *interpolazione statistica* o meglio *curva di regressione*.

Applicazione principale dell'interpolazione è il raffinamento di dati di natura numerica, limitati per costruzione dal sampling (se provenienti da sensori) oppure dalla finitezza dei dispositivi di memoria principale/massa (se necessitano di storage), o chiaramente dalla combinazione delle due casistiche. I segnali più frequentemente interpolati sono serie storiche monodimensionali o immagini.

Svariati metodi sono stati all'uopo sviluppati: polinomiale, armonico, razionale... Per brevità verrà rapidamente esposto il metodo spline cubico, essendo il solo effettivamente adoperato durante l'acquisizione del battito cardiaco. Per l'interpolazione statistica invece verrà presentato il metodo dei minimi quadrati ordinario (abbreviato con OLS dall'inglese).

2.4.2.1 Spline cubiche

Siano $\{(x_i, y_i) \in \mathbb{R}^2 : f(x_i) = y_i \quad i = 0, \dots, n\}$ $n+1$ punti noti, si vuole trovare $s_3 : [x_0, x_n] \rightarrow \mathbb{R}$ che interpoli e che sia

- globalmente molto regolare: più che $C^2(x_0, x_n)$
- localmente cubica: $s_3(x)|_{x_k, x_{k+1}} \in \mathbb{P}^3 \quad k = 0, \dots, n-1$

Pertanto su ogni intervallino i -esimo dovrà essere

$$s_3^{(i)}(x) = a_1^{(i)}x^3 + a_2^{(i)}x^2 + a_3^{(i)}x^1 + a_4^{(i)} \quad (2.8)$$

La condizione sulla continuità invece impone

$$\begin{cases} s_3^{(i)}(x_i) = s_3^{(i-1)}(x_i) \\ (s_3^{(i)})'(x_i) = (s_3^{(i-1)})'(x_i) \\ (s_3^{(i)})''(x_i) = (s_3^{(i-1)})''(x_i) \end{cases} \quad (2.9)$$

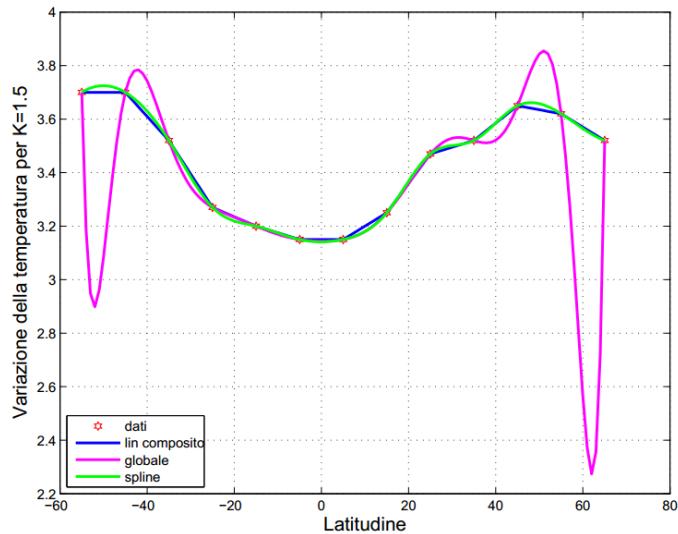


Figura 2.3: Tre interpolazioni diverse di una stessa serie. Si può notare come la spline cubica sia immune da gravi discrepanze in alcuni punti, delle quali altri metodi ne sono affetti.

2.4.2.2 Effetti frequenziali

L'interpolazione tramite spline di ordine dispari addolcisce il diagramma di densità interspetrale di potenza del segnale. La ragione di questo comportamento è che le trasformate di Fourier delle spline in questione divergono, introducendo un bias. Se si osservano nel dominio della trasformata Zeta (necessaria in ambito numerico) esse sono caratterizzate da poli esterni alla regione del cerchio unitario [9].

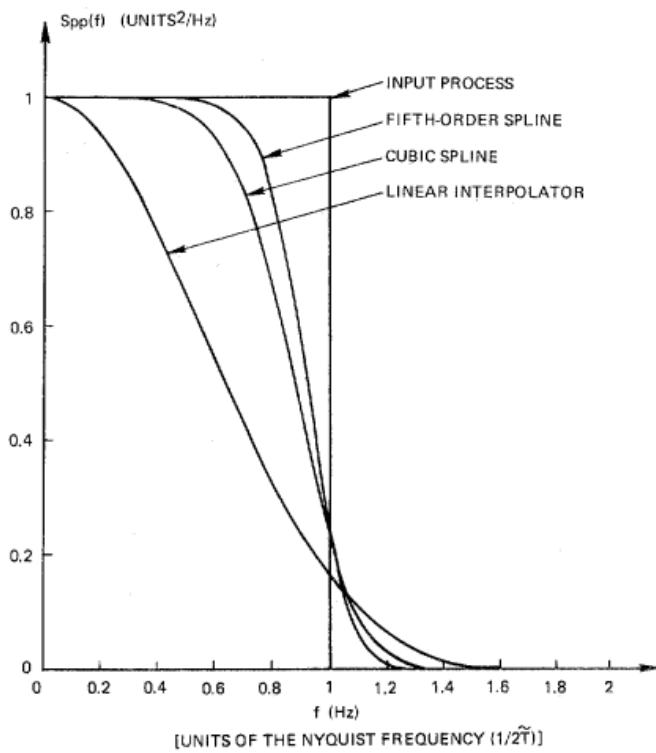


Figura 2.4: Densità interspettrali di potenza per processi ottenuti interpolando rumore bianco a banda limitata.

2.4.2.3 Minimi Quadrati Ordinari

Siano $\{(x_i, y_i) \in \mathbb{R}^2 : f(x_i) = y_i \quad i = 0, \dots, n\}$ $n+1$ punti noti, si vuole trovare $\tilde{f}(x) \in \mathbb{P}_m$, $m \ll n$ per cui

$$\sum_{i=0}^n [\tilde{f}(x_i) - y_i]^2 \leq \sum_{i=0}^n [p_m(x_i) - y_i]^2 \quad \forall p_m \in \mathbb{P}_m \quad (2.10)$$

Un eloquente utilizzo del metodo, fissato $m = 1$ si può osservare in figura

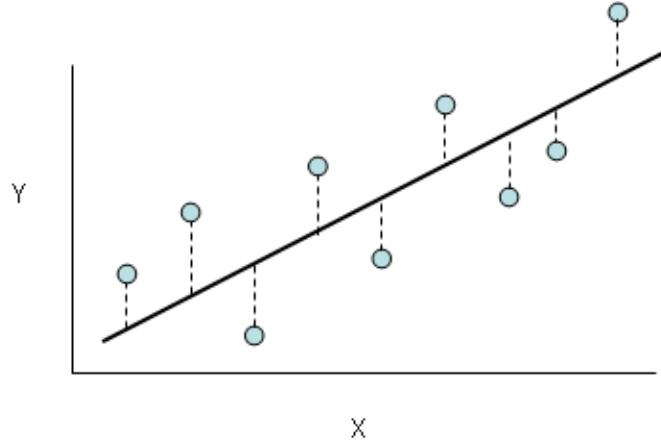


Figura 2.5: La retta di regressione di una serie temporale.

Graficamente l'obiettivo è minimizzare la sommatoria dei quadrati delle distanze tratteggiate. Sarà data ora un'altra rappresentazione del criterio di ottimalità proprio della regressione OLS. Fissato il grado a 1 si ha infatti $p_1(x) = a_1x + a_2$ che descrive la retta. Ora siano \mathbf{x} , \mathbf{y} i vettori delle x_i , y_i rispettivamente ordinate. Sia $\mathbf{u} = [1, \dots, 1] \in \mathbb{R}^{n+1}$ il vettore unitario. Il metodo OLS lineare impone

$$\begin{cases} (\mathbf{y} - a_1\mathbf{x} - a_2\mathbf{u})\mathbf{x} = 0 \\ (\mathbf{y} - a_1\mathbf{x} - a_2\mathbf{u})\mathbf{u} = 0 \end{cases} \quad (2.11)$$

conseguentemente $\mathbf{y} - a_1\mathbf{x} - a_2\mathbf{u}$ è ortogonale a \mathbf{x} e a \mathbf{u} , cioè al piano generato da questi vettori. In ultima analisi quindi (a_1, a_2) sono tali per cui $a_1\mathbf{x} + a_2\mathbf{u}$ è la proiezione di \mathbf{y} sul suddetto piano. Si osserva che il piano è il luogo delle ∞^2 soluzioni del sistema sovrardeterminato posto dal problema: anche in questo senso macroscopico quindi la soluzione offerta da OLS è quella che minimizza la funzione obiettivo [8].

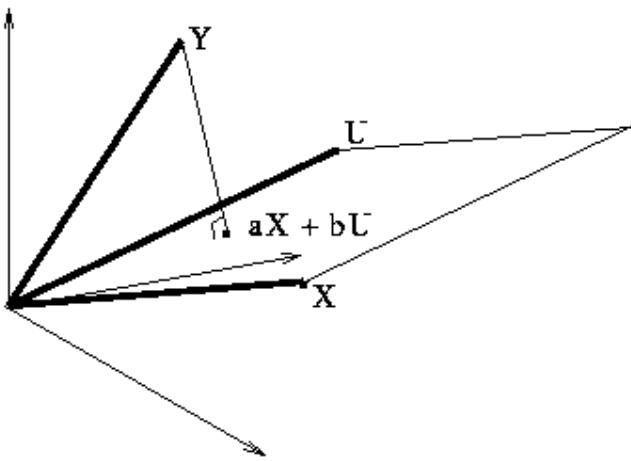


Figura 2.6: Ortogonalità dei minimi quadrati.

2.4.3 Differenze finite

Il metodo delle differenze finite è un'approssimazione numerica utilizzata nell'ambito della risoluzione delle equazioni differenziali ordinarie (ordinary differential equations, ODE). L'idea che sta alla base è di sostituire alla derivata, concetto prettamente infinitesimale, il rapporto incrementale, essendo il limite del medesimo per costruzione pari alla derivata stessa. Sia $y(t)$ una qualsiasi funzione derivabile, $h \in \mathbb{R}^+$ si scriverà

$$y'(t) \triangleq \lim_{h \rightarrow 0} \frac{y(t+h) - y(t)}{h} \simeq \frac{y(t+h) - y(t)}{h} \quad (2.12)$$

Il metodo delle differenze finite permette inoltre una snella analisi di convergenza, a costo di un errore d'approssimazione dell'ordine di h .

2.5 Analisi delle componenti indipendenti

2.5.1 Motion artifacts

Nella registrazione di un video della regione facciale i sensori RGB prelevano una composizione del segnale fotopletismografico riflesso, insieme ad altre sorgenti di fluttuazione luminosa d'origine artificiale quali il movimento delle palpebre o gli scostamenti involontari

del corpo, definiti *artifacts*. Siccome l'assorbività dell'emoglobina differisce tra gli intervalli visibile e invisibile dell'asse spettrale, ogni sensore di colore ha registrato una combinazione dei segnali originari con pesi leggermente differenti [1]. Ciò che si cerca è quindi di isolare questi artifacts confinandone la causa in un sottoinsieme dell'informazione totale, ripartendo quest'ultima in qualche modo tra BVP e tutto il resto non desiderato.

2.5.2 Ricerca di una base

Scopo dell'ICA (Independent Component Analysis) è determinare una base di N funzioni quanto più linearmente indipendenti possibile $s_i(t)$ $i = 1, \dots, N$ da un set di N funzioni $y_i(t)$ $i = 1, \dots, N$ osservate, supposto che

$$\mathbf{y}(t) = A \mathbf{s}(t) \quad (2.13)$$

essendo $A \in \mathbb{R}^{N \times N}$ la matrice di combinazione lineare delle basi. Il problema di stimare opportunamente A è un problema sovradeterminato che necessita di qualche criterio di ottimo per esser risolto. In questo caso, si cercheranno soluzioni (le basi) con la massima non-gaussianità possibile. Vista una generica $\mathbf{y}(t)$ come combinazione lineare delle $\mathbf{s}(t)$, per il teorema del limite centrale della statistica $\mathbf{y}(t)$ è necessariamente simile a una realizzazione di un processo gaussiano almeno quanto lo sia ciascuna delle sue basi. Per esempio, si pensi ad una combinazione lineare dei risultati di un numero elevato di dadi a sei facce. In questo contesto, avere basi non-gaussiane porta a una maggior indipendenza delle stesse. Nel caso limite in cui tutte le basi siano gaussiane, il risultato dell'ICA degenera e perde significato [4]. Si noti che l'applicazione dell'ICA produrrà le basi in un ordine indeterminato.

Capitolo 3

Acquisizione del battito cardiaco

3.1 Premessa

Il capitolo si ordina nativamente come la struttura procedurale dell'algoritmo per l'acquisizione. Molte delle fasi esposte sono concettualmente afferenti ai grafici del capitolo "risultati sperimentali", i quali faranno ad esse riferimento univoco. La scelta di una presentazione divisa è motivata dal principio di separazione tra metodi e effetti dei metodi (outputs), oltre al voler mostrare i suddetti grafici in un blocco unico piuttosto che sparsi, dal momento che è questo proprio il modo in cui essi vengono presentati all'utente del programma.

3.2 Acquisizione del segnale video

Il segnale video viene acquisito da una videocamera non professionale di fascia medio bassa, ed ha le seguenti caratteristiche:

- 24-bit RGB
- 3 canali monocromatici a 8 bits/canale
- numero di frames acquisiti N a discrezione
- frames per second (fps) a seconda delle prestazioni della macchina/camera
- risoluzione in pixels a discrezione.

Alternativamente allo streaming in tempo reale, è possibile fornire in input un video in formato AVI pre-registrato. Tutte le voci ‘a discrezione’ verranno specificate nell’appendice B (parametri di programma).

La bontà dell’algoritmo è stata provata attraverso la tecnica dei battiti al polso. Se ne discuterà nei risultati sperimentali.

Dopo un ciclo di acquisizione di N frames essi sono pronti per essere processati dall’algoritmo che cercherà di ricavarne il battito cardiaco. Al termine del calcolo si riceve la stampa dei risultati ottenuti ed è possibile procedere ad una nuova acquisizione che inizi logicamente da dove si era interrotta la precedente (il che chiaramente per una cattura da file non introduce discontinuità, mentre potrebbe farlo per un’acquisizione live, sebbene si sia cercato di svalutare numericamente questo effetto). Per ragioni di efficienza si è adottato un passo di tipo ‘overlap’ sui dati: ogni acquisizione dalla seconda in poi riguarda soltanto $OVERLAP < N$ (parametro a discrezione) frames nuovi, mentre riutilizza i restanti $N - OVERLAP$ frames precedenti per completare il blocco.

3.3 Individuazione del fenomeno

3.3.1 Scomposizione cromatica

Ciascun frame è pensabile come un array tridimensionale, dove due dimensioni mappano spazialmente un’intensità di colore, mentre la terza indicizza il canale di colore primario. Come specificato, il lavoro è basato su immagini a tre canali RGB (in italiano rosso, verde e blu) ove ogni pixel (ogni cella atomica dotata di due coordinate e un’intensità) abbia per ciascun canale un valore compreso tra 0 e $2^8 = 256$.

3.3.2 Creazione delle serie di differenze

Avendo scomposto in tre frames ogni singolo fotogramma in base al canale, occorre ora generare tre segnali temporali significativi per l’individuazione dei fenomeni sanguigni che porteranno all’obiettivo finale. Una soluzione semplice ed efficace è quella di costruire tre serie delle differenze finite $RR_R(t)$, $RR_G(t)$, $RR_B(t)$ dai rispettivi $r_R(t)$, $r_G(t)$, $r_B(t)$, ovvero gli andamenti temporali delle medie spaziali dell’intensità di colore: e.g. $r_B(t_k)$ rappresenta la media delle intensità di blu che si hanno per il k-esimo frame.

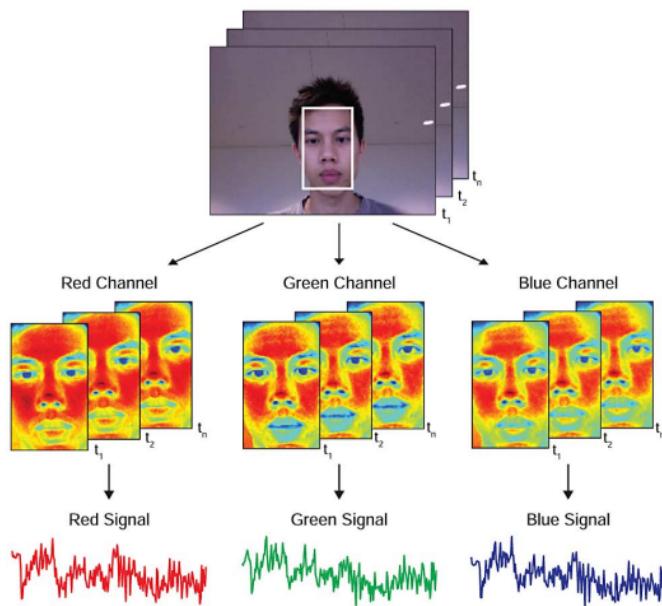


Figura 3.1: Il processo di scomposizione cromatica e di creazione delle tre serie di differenze.

3.4 Rifinitura dei segnali

3.4.1 Detrending

3.4.1.1 Fenomeno di tendenza

Si definisce trend una tendenza nei lunghi periodi predominante sulle altre e riscontrabile mediamente nelle realizzazioni di un processo stocastico. Questo andamento può essere di qualsiasi tipo, ma spesso delle modellizzazioni lineari o combinazioni lineari di esponenziali o armoniche sono sufficientemente fedeli al fenomeno. La prevalenza del trending lineare nel contesto del problema studiato (spesso associabile a fattori esogeni come la variazione di luminosità ambientale o al movimento relativo tra videocamera e area epidermica d'interesse) impedisce di focalizzarsi sulle lievi oscillazioni del battito cardiaco; in letteratura fenomeni simili sono classificati come distorsioni o artifacts.

Mika Tarvainen propone [3] un algoritmo di detrending basato su un unico parametro regolatore, studiato e implementato nel programma dal laureando. Il metodo è basato su

un approccio definito *smoothness priors*.

3.4.1.2 Ipotesi

Il fenomeno al quale si vuole applicare il detrending è un processo stocastico costituito da una componente stazionaria almeno in senso debole e da una componente lineare che si vuole stimare perché venga sottratta al processo stesso. Sia $z(t)$ una realizzazione qualsiasi, è possibile scrivere

$$z(t) = z_{stat}(t) + z_{trend}(t) \quad (3.1)$$

e modellizzare $z_{trend}(t)$ con un'osservazione lineare

$$z_{trend}(t) = H\theta + \mathbf{v} \quad (3.2)$$

essendo H matrice di osservazione, θ un vettore di parametri di regressione e \mathbf{v} l'errore commesso.

3.4.1.3 Formalizzazione

Il passo seguente è determinare $\tilde{\theta}$ tale che $H\tilde{\theta}$ sia una buona stima del trend. Viene per questo utilizzata una variante dell'approssimazione ai minimi quadrati, definendo

$$\tilde{\theta}_\lambda \triangleq \arg \min_{\theta} \{ \|H\theta - z\|^2 + \lambda^2 \|D_d(H\theta)\|^2 \} \quad (3.3)$$

ove λ è il parametro di regolazione e D_d l'approssimazione discreta del d-esimo operatore differenziale. Si noti che questa è una modifica dei minimi quadrati dove λ permette di polarizzare la soluzione a piacere verso valori di $\|D_d(H\tilde{\theta})\|$ infinitesimi.

Presa infine banalmente come osservazione la matrice identità, si dimostra che

$$\tilde{z}_{stat} = z - H\tilde{\theta}_\lambda = (I - (I + \lambda^2 D_2^T D_2)^{-1})z \quad (3.4)$$

che pertanto ha valenza di forma chiusa per l'algoritmo di detrending.

In figura 3.2 sono mostrati gli output dell'algoritmo per tre segnali di test, costituiti da componenti armoniche e lineari. Il parametro λ vale 10^9 . Valori alti sono comuni, poiché polarizzano verso la soluzione teoricamente migliore.

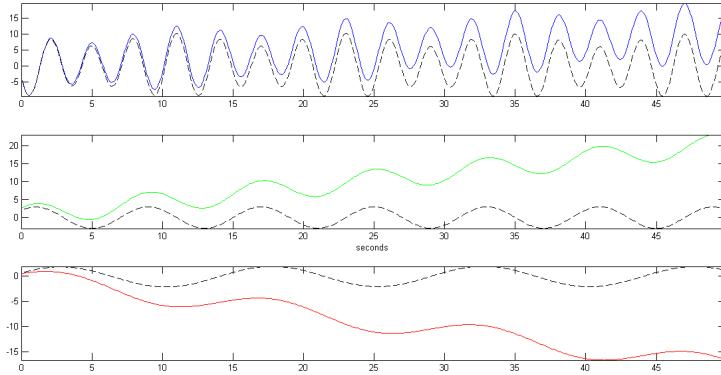


Figura 3.2: Output (linee tratteggiate) del detrending sui rispettivi input.

3.4.2 Standardizzazione

Le serie delle differenze ottenute hanno come contenuto informativo principale il loro spettro frequenziale, ma è ancora presto per iniziare un'analisi nel dominio delle trasformate. L'aver premesso questo però giustifica la normalizzazione delle tre serie, dal momento che l'informazione persa in questo passaggio (la media temporale e la norma) non inficiano la natura periodica delle stesse; in altri termini, non rimuovono contenuto frequenziale d'interesse.

Si noti che la standardizzazione è usata in modo improprio rispetto a quanto accennato nel capitolo “Elementi di teoria”: presentata come trasformazione lineare su variabile casuale, viene qui nella fattispecie applicata a delle realizzazioni concrete di un processo stocastico. Si consideri semplicemente $\mu_{x(t)}$ come la media temporale della realizzazione $x(t)$, e $\sigma_{x(t)}^2$ come la norma L2 della stessa.

3.4.3 Ricerca di una base di segnali

L'analisi delle componenti indipendenti (ICA) brevemente esposta nella teoria è il contesto entro il quale si va a cercare una base di segnali per le tre serie delle differenze associate ai canali. L'approccio JADE (Joint Approximate Diagonalization of Eigenmatrices, equivalentemente approssimazione della diagonalizzazione congiunta di automatici) consiste

nella ricerca di una base algebrica ortonormale che diagonalizzi quanto possibile ciascun elemento di un dato set di matrici quadrate (nel caso in esame, sarà un set di tre vettori rappresentanti le tre serie). L'ottimizzazione dei risultati ottenuti per via iterativa è basata su una variante dei minimi quadrati (si noti la varietà di utilizzi di questo metodo). Il metodo ipotizza che le serie temporali in input siano tutte a media nulla.

Sia $X \in \mathbb{R}^{n \times T}$ matrice nota e modellizzabile come $X = AS + N$ dove

- $A \in \mathbb{R}^{n \times n}$ è una matrice ignota e di rango pieno
- $S \in \mathbb{R}^{n \times T}$ ignota rappresentante i segnali sorgente e tale per cui
 - $\forall t$, le componenti di $S(:, t)$ sono statisticamente indipendenti
 - $\forall p$, le componenti di $S(p, :)$ sono realizzazioni a media nulla di segnali sorgenti
- $N \in \mathbb{R}^{n \times T}$ esprime l'incertezza come un rumore bianco e gaussiano.

Output della JADE sono le matrici A e S [7].

L'algoritmo si basa su una modifica del metodo di Jacobi dove il test d'arresto è sulla rotazione degli angoli Givens, per i quali si rimanda a [10]. Nelle figure 3.3, 3.4 e 3.5 è possibile vedere un esempio di applicazione dell'ICA.

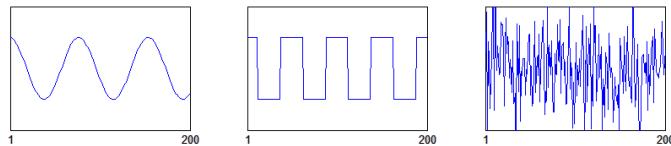


Figura 3.3: Una sorgente armonica, una tipo PAM e una di rumore.

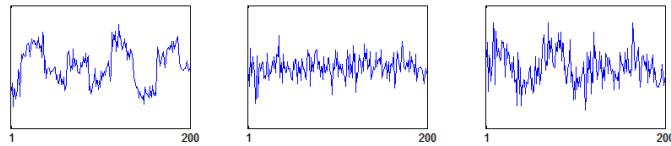


Figura 3.4: Combinazioni lineari casuali delle sorgenti in figura 3.3.

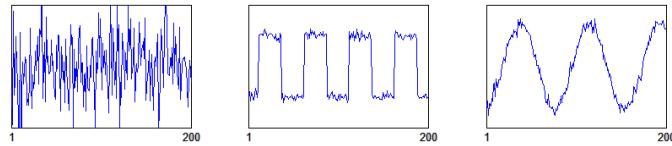


Figura 3.5: Output del JADE algorithm.

Come già detto, si nota che l'ordine delle sequenze è stato perso e non è più possibile associarvi biunivocamente i tre colori. E' ora quindi possibile determinare criteri per proseguire con una sola sequenza (ad esempio quella con la più promettente densità spettrale di potenza) o proseguire semplicemente con le tre sequenze, che d'ora in avanti non interagiscono più in alcun modo. Si è optato per tale scelta.

3.4.4 Filtraggio frequenziale

E' necessario ora applicare un rifinimento delle basi, per eliminare buona parte del rumore. Sebbene si sia rigorosamente seguito l'ordine proposto in [1], si ritiene altrettanto valido (se non migliore) effettuare questa operazione anteriormente alla fase ICA. La frequenza d'interesse è con buona certezza compresa tra i 0.9 e i 1.8 Hertz (corrispondenti al range 54-110 bpm), pertanto durante questa operazione è necessario assicurarsi che non siano tagliate le componenti armoniche sotto i 1.8 Hertz.

3.4.4.1 Low-pass media mobile

Il media mobile (MA) a N punti è uno dei classici filtri FIR più usati per abbattere il contenuto ad alta frequenza di un segnale. Si è scelto un MA a cinque punti: nelle figure è possibile vederne le caratteristiche.

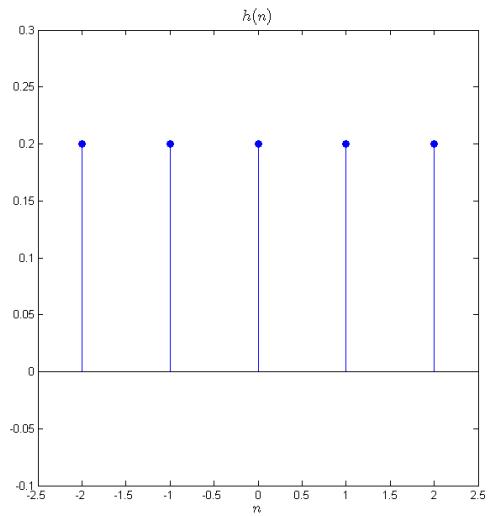


Figura 3.6: Risposta all'impulso del MA.

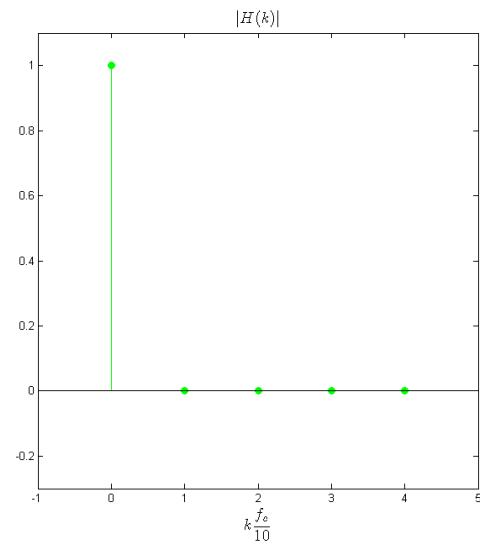


Figura 3.7: Modulo della risposta in frequenza del MA.

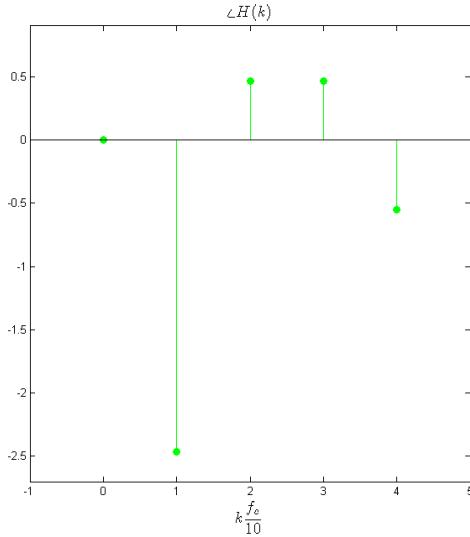


Figura 3.8: Fase della risposta in frequenza del MA.

Considerata la natura numerica delle sequenze, occorre trasformare linearmente la frequenza di taglio del media mobile numerico in una frequenza analogica. Supposto di aver rispettato la frequenza di Nyquist (si riveda la sezione di elaborazione numerica del capitolo 2) è possibile leggere dai grafici della risposta in frequenza del MA che tutte le frequenze analogiche tra 0 e $\frac{f_c}{2N}$ saranno schiacciate sul primo campione e perciò verranno preservate; le altre saranno annullate. Lo spettro di fase suggerisce un ritardo introdotto dal filtraggio, che non è qui rilevante. Si ricorda dalla teoria che il filtraggio in media mobile è raffinabile aggiungendo zeri al termine del rettangolo $h(n)$ e trasformando con più punti. In ultima analisi quindi, supponendo una $f_c = 20$ Hz, si preservano giustappunto le frequenze fino a 2 Hz.

3.4.4.2 Band-pass a finestra di Hamming

L'analisi prosegue addolcendo ulteriormente le sequenze nella ricerca delle sole frequenze d'interesse. Un filtro FIR comunemente usato come passabanda (BP) è la finestra di Hamming, per i suoi accettabili valori di ripple (oscillazioni in banda passante/oscura, rappresentazione numerica dei fenomeni di Gibbs). Si è scelta una finestra a 128 punti:

nelle figure è possibile vederne le caratteristiche.

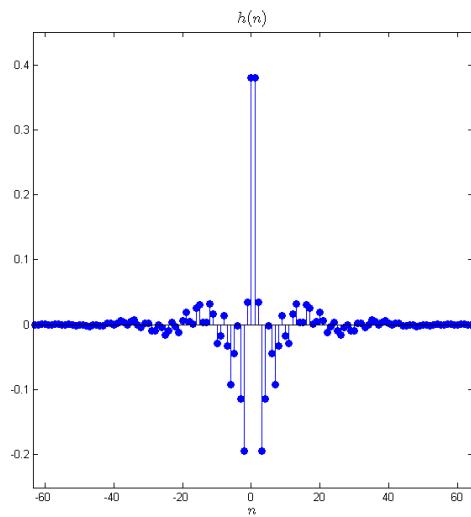


Figura 3.9: Risposta all'impulso del BP.

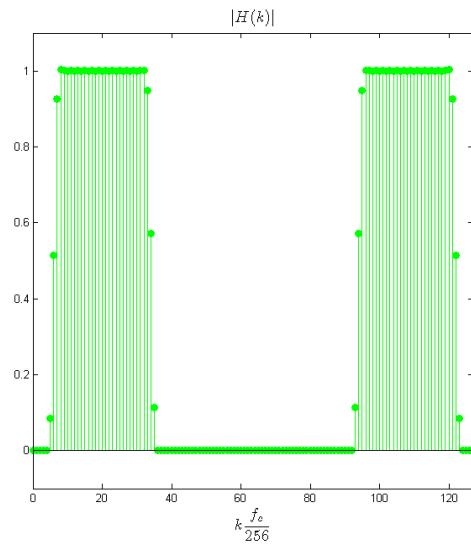


Figura 3.10: Modulo della risposta in frequenza del BP.

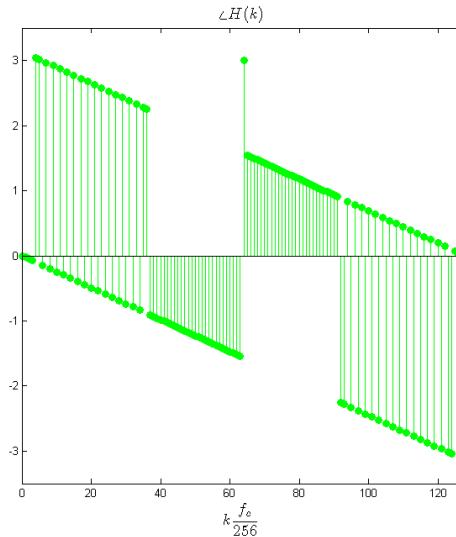


Figura 3.11: Fase della risposta in frequenza del BP.

3.4.5 Interpolazione con spline cubica

L'ultima fase per quanto concerne la rifinitura dei segnali è un'interpolazione di tipo spline cubico. Alternativamente a questa tecnica e come già emerso nelle considerazioni teoriche, è possibile aggiungere degli zeri al termine delle serie temporali e poi trasformarle secondo DFT. Scopo dell'interpolazione in [1] è l'ottenimento di un confronto diretto con l'output di un sensore da dito per il BVP, testato e approvato da enti di competenza, cioè della "strumentazione dedicata". La frequenza scelta per l'interpolazione, 256 Hz, è appunto la medesima del suddetto sensore. Sebbene nell'elaborato questo confronto non sia stato svolto, sarà fattibile (e adattabile ad altre frequenze) in futuri utilizzi.

3.5 Analisi spettrale

3.5.1 Trasformata FFT dei segnali

Ciascuna delle tre serie di N dati è finalmente trasformata tramite FFT a N punti. Considerata la natura a valori in \mathbb{R} delle serie, essendo interessati al solo modulo degli spettri, metà di ogni FFT rappresenta adeguatamente tutta l'informazione. Dalla Teoria dei

Segnali si ha infatti [5]

$$x^*(t) \longleftrightarrow X^*(-f) \quad (3.5)$$

e quindi in questo caso

$$X(f) = X^*(-f) \quad (3.6)$$

3.5.2 Ricerca dei picchi

Si ha ora tutta l'informazione necessaria ad individuare la traccia del battito cardiaco sulla registrazione della videocamera. I maggiori e più nefasti artifacts sono stati filtrati o rimossi, quindi ci si aspetta che la componente spettrale più grande in modulo corrisponda alla frequenza con cui il sangue circola nel corpo e alla quale per principio di causalità il cuore lavora. A livello operativo, quindi, è sufficiente individuare le ascisse di tre sequenze per le quali le ordinate sono massime globalmente. Le tre frequenze stimate per l'HR si ottengono banalmente moltiplicando tali ascisse per $256 \text{ Hz} \cdot 60 \text{ s}$ (la frequenza dei segnali interpolati per i secondi in un minuto).

Capitolo 4

Risultati sperimentali

4.1 Legenda e intepretazione

Vengono ora presentate due serie di outputs che l'utente riceve dopo un'acquisizione video. Entrambe sono state ottenute retroilluminando un dito con un forte LED e poggiandoci delicatamente la webcamera dall'altro lato. La risoluzione per entrambe è di 1280x720 pixel.

Il primo grafico, denominato *RR*, mostra le serie di differenze ottenute dai tre canali nelle modalità descritte nel capitolo 2. L'asse dei tempi (in questo e nei prossimi plots) è ottenuto a partire dal fps medio calcolato ad ogni ciclo di processing: senza un tracciamento del fps infatti non è possibile risalire all'effettiva velocità dello streaming e per causalità a quella della pulsazione cardiaca.

Il secondo grafico riporta in nero tratteggiato le serie del primo e sopra vi mostra l'effetto della standardizzazione.

Nel terzo grafico è possibile vedere in nero il prodotto dell'analisi delle componenti indipendenti ICA: si noti la perdita di ordinamento (cioè di colore) che subiscono le serie. In rosso invece è rappresentato l'effetto del filtraggio composito (MA + BP).

Il quarto grafico mostra in blu tratteggiato le interpolazioni sulle tre serie filtrate. L'asse temporale è sempre il medesimo, si può pensarlo come più fitto.

L'ultimo grafico disegna infine le prime metà dei moduli delle tre trasformate DFT delle serie interpolate. L'asse delle frequenze è determinato da $f_c = 256$ Hz ovvero la frequenza

raggiunta al punto quattro.

L'ascissa avente il maggior valore di ordinata (il quale non possiede alcun significato intrinseco) è il picco di frequenza da cercare. Esso, poi, moltiplicato per 60 restituirà l'HR finale.

4.2 Un ottimo esempio

Qui si può osservare contenuto informativo molto simile in tutti i tre canali. La distanza e la corretta illuminazione dell'epiderma sono fondamentali.

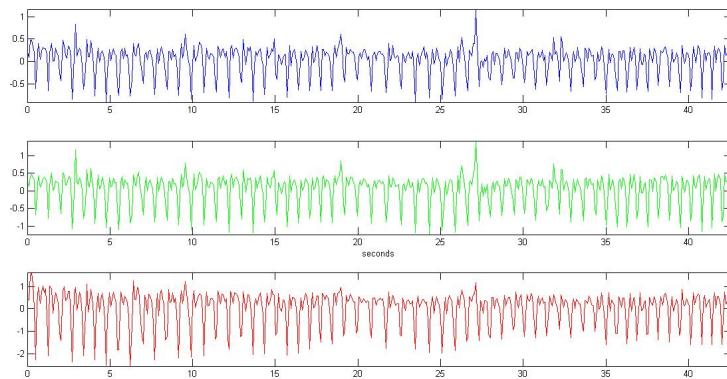


Figura 4.1: RR

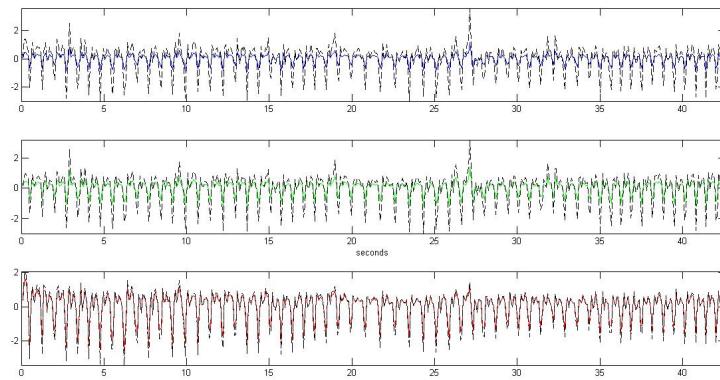


Figura 4.2: standardizzazione.

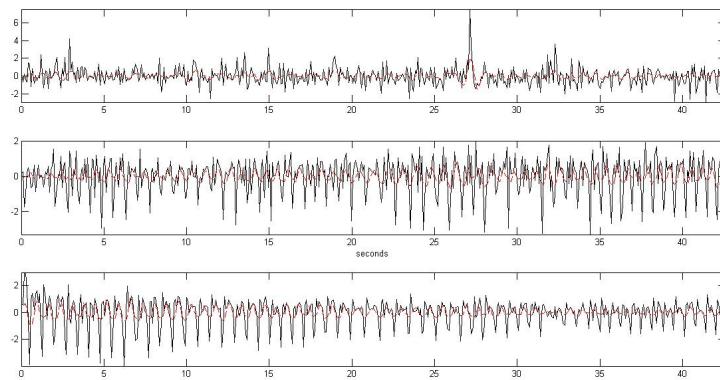


Figura 4.3: analisi ICA.

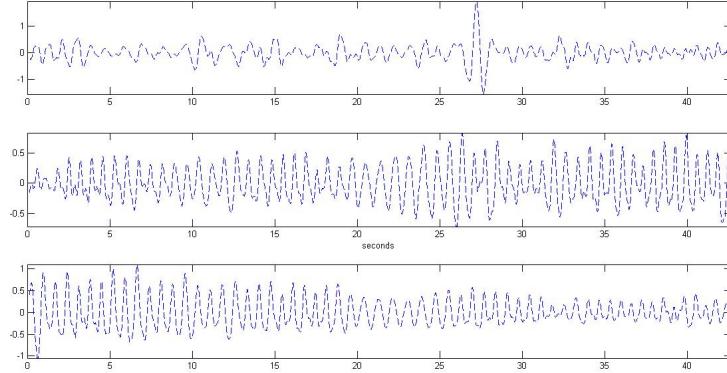


Figura 4.4: interpolazione.

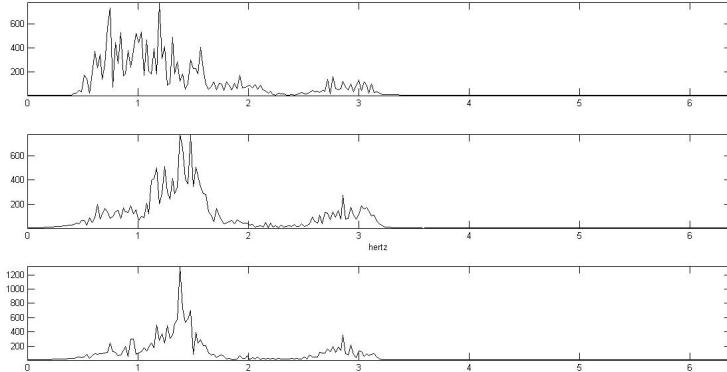


Figura 4.5: analisi spettrale.

I picchi, locati in $f = 1.2, 1.4$ Hz, forniscono rispettivamente HR = 72, 84 bpm.

4.3 Un medio esempio

In questo caso soltanto il canale rosso contiene a prima vista qualcosa di regolare correlabile all'heart rate. L'analisi procede e trova valori plausibili per quel canale. Si noti che, sebbene l'ICA abbia eliminato la corrispondenza canale-serie, soltanto una base ottenuta

ha contenuti frequenziali molto simili a quelli che si avevano per il rosso. Implementazioni future potrebbero fare un pre-processing in frequenza per scartare le acquisizioni che non porteranno a valori significativi.

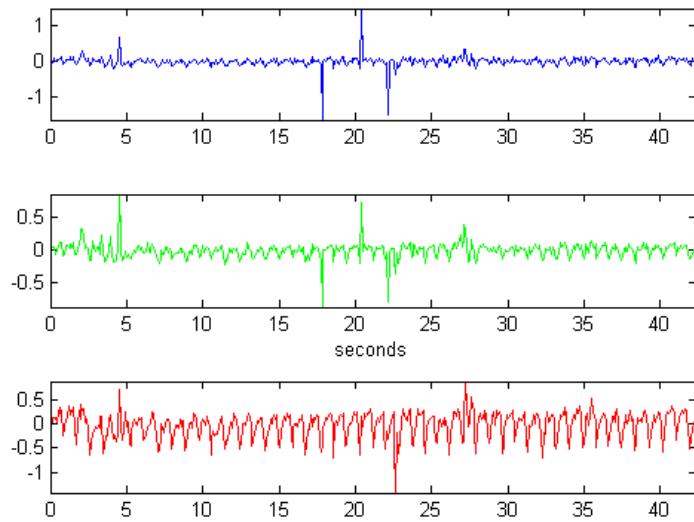


Figura 4.6: RR

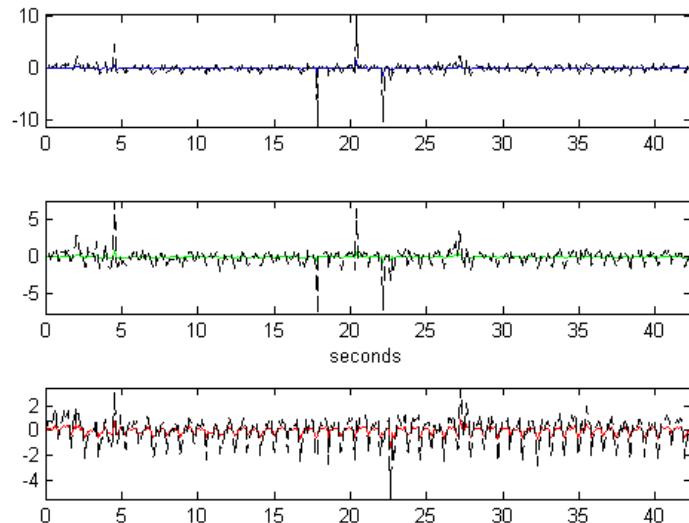


Figura 4.7: standardizzazione.

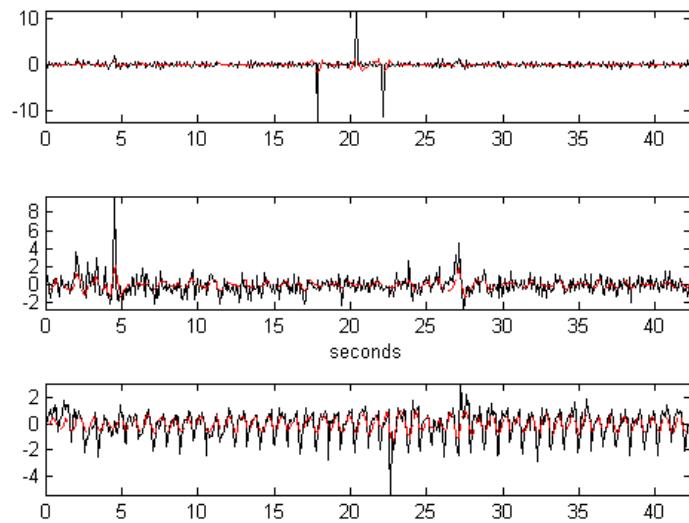


Figura 4.8: analisi ICA.

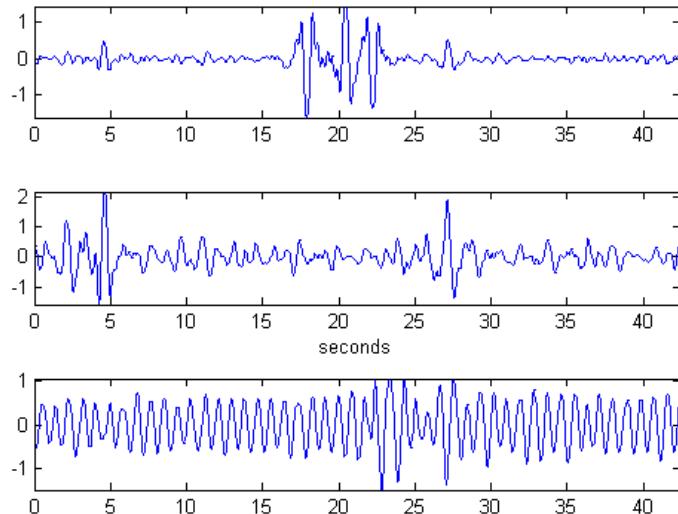


Figura 4.9: interpolazione.

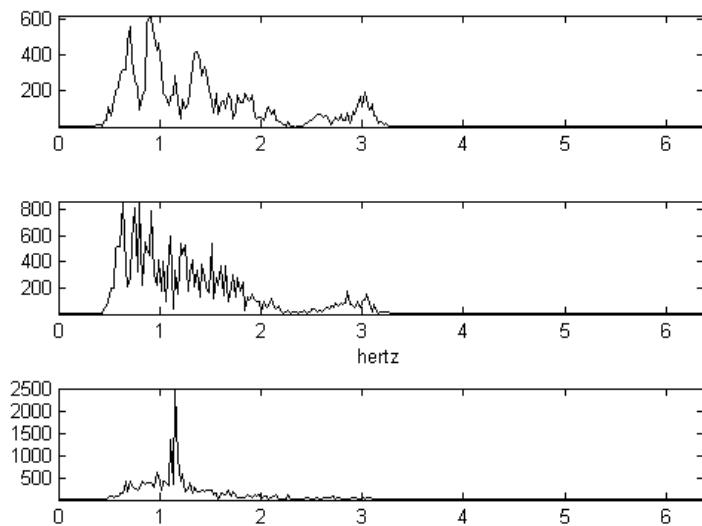


Figura 4.10: analisi spettrale.

L'unico picco che si ritiene fedele è locato in $f = 1.1 \text{ Hz}$ e pertanto $\text{HR} = 66 \text{ bpm}$.

Capitolo 5

Conclusioni e sviluppi futuri

L’elaborato ha dimostrato la plausibilità di strumenti non dedicati alla fotopletismografia per un’acquisizione accettabile del battito cardiaco. Le implicazioni di questo lavoro hanno un’utilità notevole qualora si progetti un porting del software prodotto verso sistemi mobili e smartphones, i quali hanno (vi sono esempi concreti nei repository di applicazioni dei più noti sistemi operativi mobili) sufficiente capacità computazionale per il compito. Applicazioni in grado di essere utilizzate in ogni momento e con comunissime intensità luminose di ambiente potrebbero esser di grande aiuto per soccorsi d’emergenza, intensa attività cardiaca nel fitness o per normali check-up di salute quotidiani.

Scopo del lavoro non è stato quello di produrre codice “state-of-Art”: raggiunta una sufficiente bontà dell’output, qualsiasi aspetto di efficienza computazionale viene in secondo piano, sebbene si sia cercato di governarlo mediante i parametri (vedere appendice B). Il sorgente pertanto *non* è assolutamente da prendere a esempio su *come* implementare efficacemente quanto teorizzato; sviluppi futuri possono considerare l’idea di una riscrittura pressoché totale del codice per una maggiore aderenza ai patterns e agli stili di buona programmazione dei quali si possono trovare ampi riferimenti in letteratura.

Appendice A

Supporto informatico

C/C++

Classificazione del linguaggio

Il C è stato sviluppato nel 1972 da Dennis Ritchie e i suoi collaboratori. Lo scopo dei progettisti era di sintetizzare in un unico linguaggio le caratteristiche di:

- essere ad *alto livello*, ovvero di offrire costrutti per programmi portabili, in grado di non dipendere dalla macchina usata per la progettazione (caratteristica garantita dalla standardizzazione *ANSI C*)
- permettere uno controllo a *basso livello*, cioè la raggiungibilità del singolo bit di memoria nello spazio dedicato al programma dalla macchina, per la massima ottimizzazione delle prestazioni.

I due presupposti, che a prima vista possono sembrare contradditori, sono stati rispettati dando origine al linguaggio probabilmente più diffuso nella storia, sebbene in ambiti specifici come quello scientifico o dei sistemi informativi esso sia stato surclassato da linguaggi dedicati. I settori dove esso predomina e dominerà per molto tempo sono quello dei sistemi operativi, del controllo dei processi industriali e dei sistemi *real time response*; essi hanno in comune una strettissima tolleranza su faults più o meno gravi del software/hardware, e la stabilità del C è buon attributo per l'affidabilità di questi sistemi. Con la diffusione del paradigma di programmazione OOP (object oriented programming), si ritiene che i

linguaggi procedurali non siano i più efficienti o intuitivi per alcune grandi famiglie di problemi dove gli enti principali della computazione possono esser pensati in termini di classi e oggetti. Il C++ viene pensato in questo ambito e basato su due punti imprescindibili:

- offrire al programmatore costrutti di OOP più semplici possibile, conservando un'alta efficienza del codice
- mantenere la massima compatibilità col linguaggio C, per non rendere di classe *legacy* il software prodotto antecedentemente e per permettere di gestire sottoproblemi prettamente procedurali nel loro modo nativo.

La vastità del C++ ha storicamente svantaggiato la sua standardizzazione, diversamente dal C: molto spesso i developers preferiscono concentrarsi su una specifica piattaforma o sistema per sfruttarla al massimo, sacrificando la portabilità. Nell'appendice C saranno elencate le conseguenze di questo problema nell'ambito dell'elaborato.

Allocazione dinamica

Un oggetto (un dato aggregato, una variabile) può essere allocato in memoria

- *staticamente* in una posizione fissa per tutta la durata dell'esecuzione
- *automaticamente* nello stack a seguito delle chiamate e dei ritorni di funzione
- *dinamicamente* mediante variabili dette *puntatori*.

I primi due metodi hanno forti limitazioni: nell'allocazione statica è necessario conoscere, al momento di compilazione, le dimensioni delle variabili e non è possibile eliminare dalla memoria oggetti non più necessari con evidente spreco di spazio. Nell'allocazione automatica invece i dati non possono persistere per più chiamate di funzioni [14].

Allocando dinamicamente in uno spazio di memoria chiamato *heap* è possibile sopprimere a tutte queste lacune: il programmatore ha il comando totale dei dati, accedendo a variabili aventi come valore un *indirizzo* di un'altra variabile: è questa una definizione di puntatore.

Nell'elaborato le primitive di allocazione dinamica sono servite principalmente a memorizzare temporaneamente i frames in attesa di gestire un blocco di processing, oltre alle strutture dati intermedie durante l'elaborazione delle serie temporali.

Struttura del programma

Il software prodotto nell'elaborato è concettualmente divisibile in due classi, sebbene al loro interno esse siano fortemente imperative. La classe esterna, dove risiede il *main* dell'applicativo, si incarica di inizializzare gli oggetti che verranno usati generalmente da ogni ciclo di acquisizione e processing: tra essi si annoverano per esempio il responsabile della comunicazione col device di acquisizione video, il puntatore all'*Engine Matlab* ® (che verrà a breve presentato), la finestra grafica ove la cattura del device viene istantaneamente mostrata, e un buffer per la memorizzazione dei frames, in attesa che vengano inviati al processing vero e proprio.

La classe interna possiede una coda FIFO di N frames entro i quali deve individuare una traccia del battito cardiaco. Al termine di ogni periodo di acquisizione, questa classe viene istanziata in un oggetto al quale, tramite una chiamata esterna del main ad un metodo *public*, verrà assegnato il compito specificato, nei modi e nelle tecniche concettualmente sintetizzati nel capitolo 3. Al termine del calcolo i risultati verranno presentati sia mediante forma grafica che su *standard output*.

Compilazione precondizionale

Una grande opportunità messa a disposizione da C/C++ è di effettuare compilazioni molto diverse dello stesso programma, includendo/escludendo o modificando delle sue sezioni al variare di pre-istruzioni dette *macros*. Sia in fase di debug che in fase di test o di release, quindi nella pressoché totalità del tempo dedicato allo sviluppo del software, può essere infatti molto utile concentrarsi su specifiche sezioni dell'esecuzione, oppure si può volerne tralasciare altre, per motivi di sicurezza o di non pertinenza, oppure ancora si può voler variare strutturalmente il behaviour del software a fronte degli stessi input: le possibilità sono numerosissime.

Si può obiettare che tutto questo sia realizzabile mediante i soli costrutti *sequenziale*, *condizionale* e *iterativo* offerti dal linguaggio C. Questo è formalmente garantito dal teorema di Bohn-Jacopini, ma la compilazione precondizionale, poichè composta per definizione da costrutti condizionali *if-then-else*, rispetta appunto il teorema. Infine essa consente di pilotare il codice in modo esterno, pulito, visibile, alleggerendo inoltre le dimensioni del

codice macchina prodotto, il quale nulla potrà sapere di ciò che è avvenuto prima della sua compilazione.

Per un'idea di come questo insieme di opportunità sia stato utilizzato si rimanda all'appendice B: Parametri.

OpenCV

Un po' di storia

Il mondo dell'opensource ha prodotto moduli e librerie che a volte si sono affermati come standards nell'Information Technology e nell'industria, dimostrando l'effettiva capacità di questa corrente di pensiero di generare software utilizzabile, adattabile ed estremamente efficiente. Un caso concreto è la *Open Computer Vision library*, inizialmente patrocinata da IBM e poi presa in gestione da terze parti. L'ambiente dove si propone OpenCV è quello del riconoscimento e della gestione dell'informazione fotografica e video da parte delle macchine.

Inizialmente sfruttata nell'elaborato per gestire il riconoscimento facciale tramite tecniche HAAR come accennato nell'introduzione, dopo l'eliminazione di questa feature OpenCV è rimasta parte integrante del codice, per l'utilizzo delle due classi che saranno ora presentate.

Classe Mat

Un oggetto Mat è l'istanza di una matrice di numeri dello stesso tipo (in senso informatico). Come detto nel capitolo 3 alla sezione “Individuazione del fenomeno”, un'immagine è un vettore di tre matrici bidimensionali, mappanti ciascuna una diversa intensità cromatica fondamentale in ogni punto spaziale. E' ragionevole quindi che Mat sia stata sviluppata principalmente sotto la prospettiva di rappresentare immagini, ma conserva validità generale: se ne è fatto largo uso anche nella gestione delle serie temporali, concettualmente vettori monodimensionali. La scrittura matriciale come un vettore di tre vettori (uno per canale) ha alleggerito notevolmente la gestione del codice.

Classe CvCapture

Rappresenta quanto più astrattamente possibile un'acquisizione video. Due costruttori distinti permettono poi di aprire una cattura da camera (si è sperimentata una grande compatibilità) o da file in formato AVI. E' possibile poi nei due casi effettuare operazioni diverse, come il focus sulla videocamera, la gestione della compressione del file, *etc.* Il programma necessita di una cattura video per non terminare in fase di inizializzazione.

Matlab

Classificazione del linguaggio

Matlab è un ambiente di sviluppo prodotto da *Math Works* per problemi di calcolo numerico e analisi dei dati. Comprende un ambiente grafico, un linguaggio (riferito indifferentemente come "Matlab"), un interprete ad esso correlato e uno strumento CAD per la progettazione e simulazione di sistemi (Simulink). Milioni di ingegneri e ricercatori utilizzano Matlab per i più disparati argomenti, come la biochimica, l'analisi statistica, il calcolo simbolico, l'analisi finanziaria e il digital signal processing [11].

Il nome è la sigla di MATRIX LABoratory e bene fissa la struttura dati fondamentale del linguaggio: la matrice numerica. Ogni variabile Matlab è una matrice, al limite una 1x1, e tutte le operazioni sulle matrici godono di efficienza, leggibilità e scrivibilità molto elevate.

Integrazione con programmi C++

Non essendo un linguaggio general-purpose ma offrendo una grande potenza in task scientifiche, nel tempo è emersa la convenienza di altro software ad appoggiarsi a un interprete Matlab, per sottoproblemi matematici particolarmente gravosi. Attualmente è supportato l'interfacciamento a run-time (si ricorda che Matlab è interpretato) con: Java, COM, .NET, Fortran e C/C++. In particolare per quest'ultimo, dopo aver incluso header e linkato librerie opportuni, nel sorgente C++ ogni operazione con Matlab verrà demandata a un oggetto Engine, visibile in figura 1. Il motore Matlab offre primitive per il parsing in/out di codice Matlab e per il load/store di variabili nel workspace, oltre alla possibilità di creare e gestire finestre grafiche totalmente indipendenti dal processo main.

Utilizzo nell'elaborato

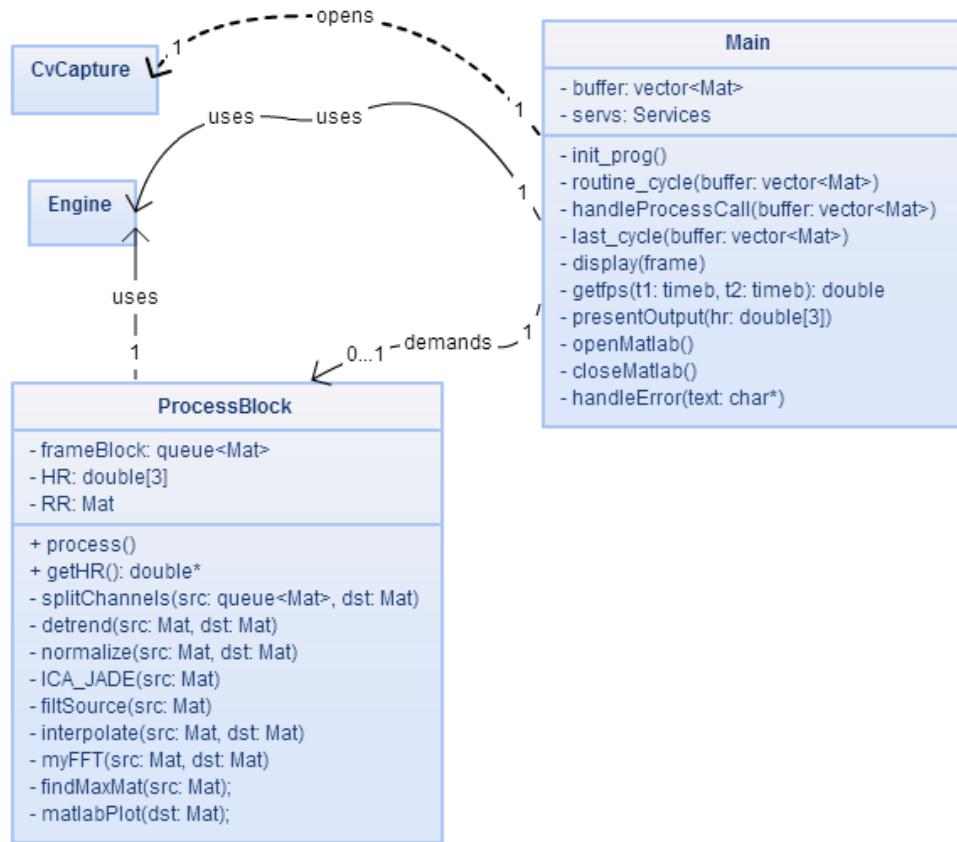
E' lecito obiettare ora che le matrici elaborate dal programma potevano essere gestite in Matlab piuttosto che con le strutture OpenCV. Così non è stato, poiché si sarebbe sacrificata notevolmente la scrivibilità del codice, a giudizio del programmatore. Inoltre OpenCV dispone di alternative (classe Mat) rispettabili. Il software scritto si è invece appoggiato a Matlab per due ragioni:

- Richiamare funzioni e script di terze parti affidabili, velocemente e senza effettuare un porting (che può sempre essere fonte di errori)
- graficare i dati ottenuti durante le diverse fasi del processing, cosa peraltro non possibile senza qualche tipo di pacchetto grafico.

FFTW

E' una libreria di routines C per computare la trasformata di Fourier discreta DFT in una o più dimensioni, di dimensione arbitraria, sia per sequenze reali che complesse, oltre ad implementare le trasformate seno/coseno per sequenze dispari/pari. La libreria è incentrata sulla portabilità e questo la spinge ad avere prestazioni costanti su benchmarks di macchine molto diverse [13]. Peculiare è l'implementazione adattiva: il programmatore non richiama una funzione *trasforma()*, ma definisce un *plan* che si incarichi di gestire ogni richiesta di trasformate. A run-time il plan cerca di ottimizzare le prestazioni di FFTW operando diverse scelte algoritmiche per le trasformate domandate, con lo scopo di assestarsi, dove l'uso della libreria è massiccio, su un ottimo specifico per la macchina ove gira.

Diagramma delle classi



[online diagramming & design]  creately.com

Figura 1: Diagramma UML delle classi del programma.

Appendice B

Parametri di programma

Qui è elencata una serie di parametri macro da specificare a tempo di compilazione. In riscritture future del codice è considerabile l'idea di portarne almeno una parte a run-time mediante allocazione dinamica della memoria; questa non è però rientrata negli obiettivi di first-release che il laureando si è posto.

- MATLAB: se definito, consente l'appoggio esterno a Matlab.
- CAMERA: se definito, inizia una cattura da webcam, altrimenti apre un file AVI.
- WINTIMESTAMP: se definito, utilizza funzioni per il calcolo del *cputime* testate solo in *Visual Studio*.
- DONOTCLOSEENGINE (richiede MATLAB): se definito, al termine del processo non chiama la chiusura del Matlab Engine per velocizzare le esecuzioni future.
- RES_1280_720 (richiede CAMERA): se definito, prova a imporre l'alta risoluzione alla webcam, altrimenti, o in caso di fallimento del tentativo, verrà usata quella di default del device. Fattore di onerosità nel calcolo e nella memoria.
- SRC_FACE: se definito, predispone l'acquisizione da regione facciale e il riconoscimento tramite HAAR. Deprecato, e probabilmente guasto.
- FRAMEBLOCK: intero positivo per il numero di frame contigui che formino un blocco di processing. Fattore di onerosità nel calcolo e nella memoria.

- OVERLAP: intero positivo minore di FRAMEBLOCK per l'acquisizione parziale di nuovi frames.
- DETREND: se definito, considera la fase di detrending.
- LAMBDA: (richiede DETREND): parametro regolatore dell'algoritmo di detrending.
- VERBOSE: flag per il livello di verbosità del programma nella comunicazione del proprio operato.
- HUNGRY: flag per la rigorosità delle deallocazioni in memoria. Più alto, più severo.
- AVGPOINTS: punti per il filtro MA.
- BPPOINTS: punti per il filtro BP.
- INTERPFREQ: frequenza alla cui interpolare.

Appendice C

Note sulla portabilità

Ambiente di sviluppo

Inizialmente si è optato per lavorare in un sistema operativo Linux x86, ma a causa di numerosi problemi nella compilazione di OpenCV il laureando si è spostato in ambiente Windows 7 x86. Il codice è stato sviluppato, testato e compilato in *Visual Studio 2010*. L'elemento che ha influenzato maggiormente la scelta è la possibilità del linking del proprio codice con numerose dll (dynamic linked library) esterne anzichè dover compilarne i sorgenti da zero. Questo chiaramente rende il codice non portabile fuori dall'ambiente Windows.

Headers propri di Visual Studio inoltre sono stati utilizzati per il calcolo dell'fps in acquisizione. Questo dato è fondamentale per il processing, perché ricordiamo esso riveste le serie di una veste temporale precisa. Qualora lo sviluppo del codice proseguia fuori da Windows, si dovrà ricorrere a diversi metodi di accesso ai tempi macchina.

L'interfacciamento con Matlab consta di un header da includere e tre librerie *lib* da collegare. Sebbene illustrato a prescindere dal sistema operativo [12], esso è stato testato solo in Windows e con una versione Matlab “portable”. Durante la compilazione è possibile però escludere Matlab, rinunciando ai grafici di output e a un algoritmo di terze parti [7] per l'ICA, al cui posto verrebbe compilato del codice analogo ma non testato completamente.

Ringraziamenti

Grazie a mia madre, semplicemente per aver portato tutto sulle spalle.

Grazie a mio padre, per avermi dato il dovere del lavoro.

Grazie a tutti gli Avi, che mi guardino dal Cielo.

Grazie a Erminia, per aver volato abbracciati nell'Iperurano.

Grazie alla prof.ssa Paola Gervasio, spesso si ignora l'effetto che si suscita negli altri.

Grazie alla prof.ssa Lidia Tiraboschi, per l'amore della Matematica.

Grazie alla prof.ssa Valentina Valota, per l'amore della Verità.

Bibliografia

- [1] Ming-Zher Poh, Daniel J.McDuff and Rosalind W.Picard, *Advancements in noncontact, multiparameter physiological measurements using a webcam* IEEE Transactions on biomedical engineering, vol.58, no.1,January 2011.
- [2] W. Verkruyse, L. O. Svaasand, and J. S. Nelson, *Remote plethysmographic imaging using ambient light*, Opt. Expr.,vol.16,pp.21434–21445, Dec. 2008
- [3] Mika P.Tarvainen, Perttu O.Ranta-aho and Pasi A.Karjalainen, *An advanced detrending method with application to HRV analysis*.
- [4] Ruye Wang, *Non-Gaussianity is Indipendence*, <http://fourier.eng.hmc.edu/e161/lectures/ica/node3.html>
- [5] R.Leonardi and P.Migliorati, *Esercizi di Teoria dei Segnali* Società Editrice Esculapio, 2011, III Ed.
- [6] M.Luise, G.M.Vitetta, *Teoria dei Segnali* McGraw-Hill, 2003, II Ed.
- [7] J.Cardoso, *Blind source separation and indipendent component analysis*, <http://perso.telecom-paristech.fr/~cardoso/guidesepsou.html>
- [8] Thomas Banchoff, *Least Squares Fit*, <http://www.geom.uiuc.edu/~banchoff/linalg/linalg.html>
- [9] Larry Horowitz, *The effects of spline interpolation on Power Spectral Density*, IEEE Transactions on acoustics, speech and signal processing, vol.

ASSP-22, no. 1, February 1974, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01162536>

- [10] Julien Dompierre, *Householder Reflections and Givens Rotation*, http://www.cs.laurentian.ca/jdompierre/html/CPSC5006E_F2010/cours/ch05_Householder_Givens.pdf
- [11] *Matlab: il linguaggio del calcolo tecnico*, <http://www.mathworks.it/products/matlab/>
- [12] *Interface Matlab code with C, C++ and Fortran*, <http://www.mathworks.it/support/compilers/interface.html>
- [13] *FFTW: Introduction*, <http://www.fftw.org>
- [14] Wikipedia.