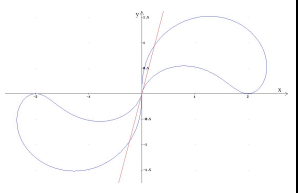
ISSN: 2037-6367

MatematicaMente

Pubblicazione mensile della sezione veronese della MATHESIS – Società Italiana di Scienze Matematiche e Fisiche – Fondata nel 1895 – Autorizzazione del Tribunale di Verona n. 1360 del 15 – 03 – 1999 – I diritti d'autore sono riservati. Direttore: Luciano Corso - Redazione: Alberto Burato, Elisabetta Capotosto, Carlo Marchiori, Giovanna Tessari – Via IV Novembre, 11/b – 37126 Verona – tel e fax (045) 8344785 – 338 6416432 – e-mail: lcorso@iol.it – Stampa in proprio – Numero 160 – Pubblicato il 30 – 06 – 2011



Compressione d'immagini

di Manuel Zambelli [1]

Il presente lavoro nasce durante il Progetto Lauree Scientifiche cui ho partecipato nell'anno in corso. Abbiamo trattato l'argomento dell'analisi armonica mediante la serie di Fourier di un segnale sonoro periodico.

Il professor Sisto Baldo, analista presso l'Università degli Studi di Verona, Facoltà di Scienze, ha introdotto l'argomento proponendo lo studio di semplici suoni, in particolare vocali, che dovevano essere simulati attraverso l'applicazione della trasformata discreta di Fourier. Ho pensato che lo stesso principio potesse essere applicato allo studio simulato d'immagini.

Con il software Mathematica 7.0 ho realizzato un programma per mezzo del quale, a partire da un'immagine che viene importata nel software, si sviluppa una procedura che comprime l'immagine e ne simula la costruzione attraverso la DFT (*Discrete Fourier Transform*). In questo breve articolo presento il lavoro da me fatto e i risultati che ho ottenuto.

Per quanto riguarda la teoria generale dell'analisi armonica di Fourier e delle sue trasformate in bibliografia sono indicati i testi fondamentali consultati [B.1, B.2].
Qui di seguito invece riporto quelle parti della teoria che ho usato al fine di risolvere il problema.

Data un'immagine in formato "*.png*" di $m \times n$ *pixel* adiacenti (dove m sta per il numero di righe di *pixel*, n per il numero di *pixel* in ogni riga (colonne) e dove ogni *pixel* è rappresentato come una terna di interi), quando essa viene importata in Mathematica è rappresentata come un vettore di vettori costituito nel seguente modo:

$$\begin{aligned}
&\{\quad \{\{a_{11}, b_{11}, c_{11}\}, \{a_{12}, b_{12}, c_{12}\}, \ldots, \{a_{1n}, b_{1n}, c_{1n}\}\},\\
&\quad \{\{a_{21}, b_{21}, c_{21}\}, \{a_{22}, b_{22}, c_{22}\}, \ldots, \{a_{2n}, b_{2n}, c_{2n}\}\},\\
&\quad \ldots\\
&\quad \ldots\\
&\quad \ldots\\
&\quad \{\{a_{m1}, b_{m1}, c_{m1}\}, \{a_{m2}, b_{m2}, c_{m2}\}, \ldots, \{a_{mn}, b_{mn}, c_{mn}\}\}\ \}
\end{aligned} \tag{1}$$

I numeri a_{ji}, b_{ji}, c_{ji} rappresentano l'intensità delle componenti cromatiche rossa, verde e blu del pixel di posto ji.
Questo può considerarsi il prototipo di qualsiasi immagine in formato "*.png*" importata in Mathematica.
Riarrangiamo il vettore così presentato in modo tale da formare la seguente nuova terna di vettori:

$$\begin{aligned}
A &= \{\{a_{11},a_{12}, \ldots,a_{1n}\}, \{a_{21},a_{22}, \ldots,a_{2n}\}, \ldots, \{a_{m1},a_{m2}, \ldots,a_{mn}\}\}\\
B &= \{\{b_{11},b_{12}, \ldots, b_{1n}\}, \{b_{21},b_{22}, \ldots,b_{2n}\}, \ldots, \{b_{m1},b_{m2}, \ldots,b_{mn}\}\}\\
C &= \{\{c_{11},c_{12}, \ldots, c_{1n}\}, \{c_{21},c_{22}, \ldots, c_{2n}\}, \ldots, \{c_{m1},c_{m2}, \ldots, c_{mn}\}\}
\end{aligned} \tag{2}$$

Questa trasformazione (dove il numero di interi contenuti in ognuno dei vettori A, B, C è $m \cdot n$) permette di rendere più agevole l'accesso ai dati nel mio algoritmo, rendendo molto più chiaro il codice.
Il primo elemento (vettore) di A contiene la componente cromatica rossa di tutti i *pixel* della prima riga, il secondo elemento tutte le componenti rosse della seconda e così via fino all'm-esima riga. La stessa cosa vale anche per il vettore B (verde) e per il vettore C (blu). Per ricostruire l'immagine originale con la DFT basta allora considerare ogni elemento della lista A (e in modo analogo per le liste B, C) come campionamento di una funzione f_{Aj} 2π-periodica. Per l'approssimazione delle f_{Aj} possiamo utilizzare polinomi trigonometrici nella forma

$$\begin{cases} w_{Aj}(t) = \dfrac{p_{Aj0}}{2} + \sum_{k=1}^{N} p_{Ajk}\cos(k\cdot t) + q_{Ajk}\sin(k\cdot t) \\ j = 1,\ldots,m \end{cases} \tag{3}$$

dove p_{Ajk} sta a indicare il k-esimo coefficiente del coseno del j-esimo segnale 2π-periodico del vettore A.
Dobbiamo però determinare i coefficienti q_{Ajk} e p_{Ajk}.
Qui entra in gioco Fourier. Infatti, posta h la distanza tra ogni punto di campionamento (questa distanza è equivalente per ogni nostra f_{Aj}, essendo il numero di campionamenti equivalente per tutti e uguale a n), dove

$$h = \frac{2\pi}{n}, \tag{4}$$

i coefficienti q_{Ajk} e p_{Ajk} vengono calcolati come segue:

$$\begin{cases} p_{Ajk} = \dfrac{1}{\pi}\displaystyle\int_{-\pi}^{\pi} f_{Aj}(t)\cos(kt)\,dt \\ q_{Ajk} = \dfrac{1}{\pi}\displaystyle\int_{-\pi}^{\pi} f_{Aj}(t)\sin(kt)\,dt \end{cases} \qquad k \in \{0,1,\ldots,N\} \tag{5}$$

Le (5) danno i coefficienti di Fourier esatti delle funzioni f_{Aj}. Mathematica, essendo un software molto sofisticato e molto ben sviluppato, permette di calcolare integrali, se possibile, anche in modo esatto (simbolico).
Noi però non abbiamo a disposizione una funzione periodica definita in tutti i punti da passare a Mathematica, bensì una funzione campionata. Calcolandoci gli integrali delle (5) con il metodo dei rettangoli, dove il numero di rettangoli sui quali effettuare il calcolo dell'approssimazione numerica dell'integrale è n (essendo n il numero di campionamenti delle nostre f_{Aj}), la trasformata, nel nostro problema, diventa

$$\begin{cases} p_{Ajk} = \dfrac{1}{\pi}\cdot h\cdot\displaystyle\sum_{i=1}^{n} a_{ji}\cos\big(k\big(-\pi + h\cdot(i-1)\big)\big) \\ q_{Ajk} = \dfrac{1}{\pi}\cdot h\cdot\displaystyle\sum_{i=1}^{n} a_{ji}\sin\big(k\big(-\pi + h\cdot(i-1)\big)\big) \end{cases}, \quad k \in \{0,1,\ldots,N\} \tag{6}$$

detta anche *Discrete Fourier Transfom* (DFT).

Potremmo anche calcolarci i coefficienti con l'algoritmo della *Fast Fourier Transform*, che renderebbe molto più veloce tutto il processo, ma per semplicità ho deciso di implementare le (6).
Quanto grande è opportuno prendere N? O meglio, a quale coefficiente conviene fermarsi?
Empiricamente, ho notato che un N opportuno si ottiene da

$$6m\sqrt{n}, \tag{7}$$

ma il programma permette anche di poter scegliere un N arbitrario, per poter vedere come cambia l'immagine simulata al variare di N.
Il vettore formato dai coefficienti q_{Ajk} e p_{Ajk} costituisce la mia immagine compressa, che posso ricostruire tramite un algoritmo di decompressione. Infatti, avendo i coefficienti q_{Ajk} e p_{Ajk}, è immediato il calcolo dei $w_{Aj}(t)$ (vedi (3)) e, applicando questo procedimento anche ai vettori B e C, otteniamo questi 3 vettori:

$$\begin{aligned}
X &= \{\, w_{A1}(t), w_{A2}(t), \ldots\ldots, w_{Am}(t)\,\}\\
Y &= \{\, w_{B1}(t), w_{B2}(t), \ldots\ldots, w_{Bm}(t)\,\}\\
Z &= \{\, w_{C1}(t), w_{C2}(t), \ldots\ldots, w_{Cm}(t)\,\}
\end{aligned} \tag{8}$$

Costruisco ora, nel modo seguente, la mia immagine decompressa:

$$
\begin{aligned}
&\{\quad \{\{\ w_{A1}(-\pi),\ w_{B1}(-\pi),\ w_{C1}(-\pi)\ \},\\
&\qquad \{\ w_{A1}(-\pi+h),\ w_{B1}(-\pi+h),\ w_{C1}(-\pi+h)\ \},\ \ldots\ldots,\\
&\qquad \{\ w_{A1}(-\pi+(n-1)\,h),\ w_{B1}(-\pi+(n-1)\,h),\ w_{C1}(-\pi+(n-1)\,h)\ \}\},\\
&\quad \{\{\ w_{A2}(-\pi),\ w_{B2}(-\pi),\ w_{C2}(-\pi)\ \},\\
&\qquad \{\ w_{A2}(-\pi+h),\ w_{B2}(-\pi+h),\ w_{C2}(-\pi+h)\ \},\ \ldots\ldots,\\
&\qquad \{\ w_{A2}(-\pi+(n-1)\,h),\ w_{B2}(-\pi+(n-1)\,h),\ w_{C2}(-\pi+(n-1)\,h)\ \}\},\\
&\quad :\\
&\quad :\\
&\quad \{\{\ w_{Am}(-\pi),\ w_{Bm}(-\pi),\ w_{Cm}(-\pi)\ \},\\
&\qquad \{\ w_{Am}(-\pi+h),\ w_{Bm}(-\pi+h),\ w_{Cm}(-\pi+h)\ \},\ \ldots\ldots,\\
&\qquad \{\ w_{Am}(-\pi+(n-1)\,h),\ w_{Bm}(-\pi+(n-1)\,h),\ w_{Cm}(-\pi+(n-1)\,h)\ \}\}\\
&\}
\end{aligned}
$$

Le immagini seguenti presentano i diversi output del programma al variare di N, con immagine in input costante.

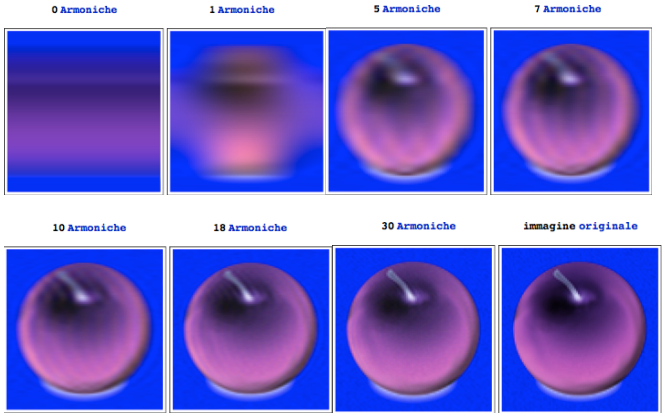


fig.1

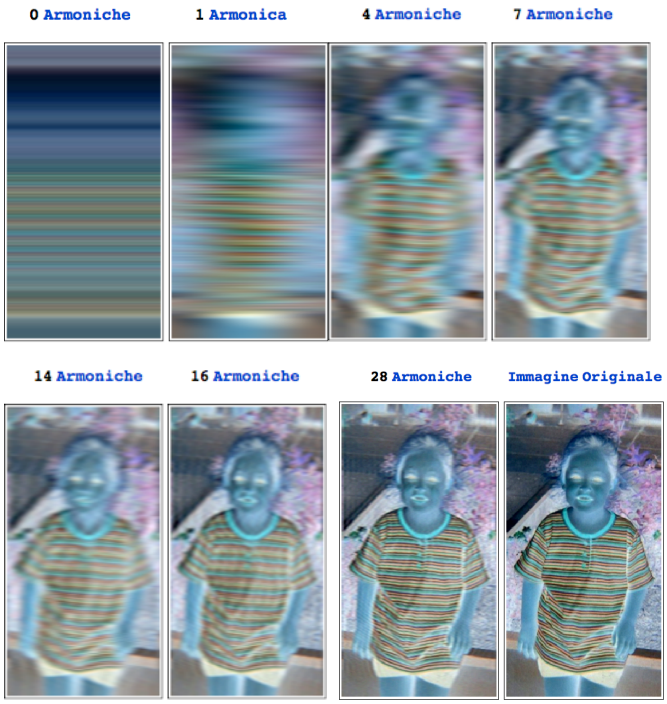


fig.2

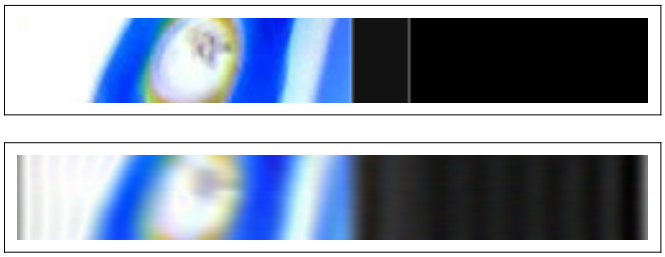


fig.3

Siccome per un'immagine reale non c'è ragione di supporre che $f_{Aj}(-\pi)=f_{Aj}(\pi)$, la funzione periodica f_{Aj} potrebbe risultare discontinua ai bordi dell'intervallo $[-\pi, \pi]$: se partiamo da una immagine che non abbia uno sfondo omogeneo, dobbiamo aspettarci degli artefatti di compressione vicino al bordo dell'immagine. Questo effettivamente è ciò che accade, come possiamo vedere nella fig. 3 (l'immagine in alto è l'originale, quella in basso la sua ricostruzione con il mio algoritmo). Nella parte sinistra dell'immagine simulata, dove dovremmo aspettarci solo il bianco, compare invece una forte componente di grigio. A destra, l'originale nero è mescolato con il bianco. Queste "stranezze" sono proprio gli artefatti di cui parlavamo prima.

Il mio algoritmo comprime separatamente ciascuna riga dell'immagine.

Gli algoritmi usati nel mondo reale (per esempio *".jpeg"*), invece, comprimono contemporaneamente tutta l'immagine usando una versione bidimensionale della DFT: in questo modo il file compresso risulta notevolmente più piccolo, ma ovviamente l'algoritmo è molto più complicato. Il vantaggio di questo algoritmo di compressione sta nel fatto che, scegliendo di fermarsi a $\sqrt{n}$ coefficienti, per memorizzare l'immagine ho bisogno di salvare $6m\sqrt{n}$ coefficienti di Fourier, al posto dei $3nm$ numeri interi presenti inizialmente nel vettore (1). Sostanzialmente, riesco a rimpiazzare il fattore n con un multiplo piccolo di $\sqrt{n}$, cosa molto conveniente per n grande!

Bibliografia: [B.1] Smirnov Vladimir Ivanovic, *Corso di Matematica Superiore*, Vol. II, p. 466 e ss., Editori Riuniti, Roma, 1977. [B.2] Mathematica 7.0 della Wolfram Research. [B.3] Baldo Sisto, *Appunti per il Laboratorio di Scienza del Suono: Parte Matematica*, Dipartimento di Matematica Università degli Studi di Trento, anno 2010.

Ringraziamenti: Ringrazio i professori Sisto Baldo (UNIVR), Alberto Burato e Luciano Corso (ITIS G. Marconi di Verona) per gli stimoli e gli utili consigli che mi hanno dato.

[1] Studente del 5° anno dell'ITIS G. Marconi – corso di informatica.
E-mail: manuelzambelli@gmail.com

Trasformata di Fourier discreta: applicazioni all'acustica...ed altro!

di Sisto Baldo [2]

L'articolo di Manuel Zambelli che compare in questo numero è un ... piacevole e inatteso prodotto di un'attività di laboratorio per gli studenti dell'ITIS Marconi di Verona, realizzata nel quadro del Piano Lauree Scientifiche, alla quale ho contribuito assieme agli insostituibili colleghi Burato, Corso e Visigalli.

Agli studenti del V anno che hanno partecipato su base volontaria al laboratorio, abbiamo proposto alcune attività incentrate sull'applicazione della Trasformata di Fourier Discreta (DFT) allo studio di fenomeni acustici (ricostruzione del timbro di strumenti musicali, compressione di un segnale periodico, individuazione delle formanti nelle vocali umane...).

In uno degli incontri con gli studenti, mi è capitato di accennare all'utilizzo della DFT per la compressione di immagini: lo studente Zambelli ne ha immediatamente approfittato per sperimentare un suo personale algoritmo, che impiega la DFT unidimensionale (che già stavamo usando per i suoni) per comprimere un'immagine riga per riga. Una soluzione certamente un po' ingenua rispetto agli algoritmi allo stato dell'arte... ma nondimeno semplice, istruttiva ed efficace!

Per chi fosse interessato ad una discussione elementare sulle applicazioni della DFT all'acustica, e a possibili spunti per un laboratorio di matematica, segnalo del materiale liberamente scaricabile dalla mia pagina web:

http://profs.sci.univr.it/~baldo/divulgazione/matfismus.html

[2] Docente di Analisi Matematica, Università degli Studi di Verona