Elaborazione di Segnali Multimediali
a.a. 2017/2018

Elaborazioni nel dominio della frequenza

L.Verdoliva

In questa esercitazione elaboreremo le immagini nel dominio della frequenza, in particolare realizzeremo sia filtri di smoothing che di sharpening progettandoli direttamente nel dominio di Fourier.

1 Uso della DFT per il filtraggio lineare

Lo studio dei segnali e dei sistemi nel dominio della frequenza è uno strumento fondamentale per l'elaborazione dei segnali, infatti nel dominio della frequenza il prodotto tra la trasformata di Fourier dell'ingresso e quella della risposta impulsiva equivale alla convoluzione realizzata nel dominio spaziale. Il filtraggio nel dominio frequenziale risulta computazionalmente molto efficiente grazie proprio ad algoritmi veloci (FFT) per il calcolo della DFT. In realtà se si realizza il prodotto delle DFT dell'immagine di ingresso, $X(k,l)$, e della risposta impulsiva $H(k,l)$ si ottiene la convoluzione circolare delle immagini, che in generale non è uguale alla convoluzione lineare, operazione a cui siamo interessati per realizzare il filtraggio nel discreto. Tuttavia, si può dimostrare, che realizzando opportunamente uno zero padding dell'immagine di partenza la convoluzione circolare coincide con quella lineare. Quindi pur di effettuare il riempimento con zeri, la DFT può essere usata per realizzare il filtraggio lineare.

Cominciamo ad esplorare questi concetti nel caso monodimensionale per poi passare alle immagini. Proviamo allora a determinare attraverso la DFT e la IDFT la risposta del filtro con $x(n) = [1,2,3,4]$ e $h(n) = [1,1,1]$, entrambe causali. In questo caso bisogna calcolare la DFT almeno su $N = 4+3-1 = 6$ punti per garantire la correttezza del risultato:

```
x=[1 2 3 4];                % vettore di ingresso
h=[1 1 1];                  % risposta impulsiva
X=fft(x,6);                 % realizza la DFT su 6 punti
H=fft(h,6);
Y=X.*H;
y=ifft(Y,6)
```

Verificate che l'uscita è la stessa che fornisce il comando `y=conv(x,h)`, ripetete quindi l'operazione di filtraggio in frequenza calcolando la DFT su 4 punti, e verificando che il risultato è lo stesso di quello che otterreste con un'estensione periodica ai bordi.

Se si considera invece un'immagine, $x(m,n)$, di dimensioni $M \times N$ e un filtro $h(m,n)$ di dimensioni $A \times B$, è necessario calolare la DFT-2D su un numero di punti P e Q che soddisfino le condizioni $P \geq M + A - 1$ e $Q \geq N + B - 1$. Vediamo un esempio di implementazione nel dominio della frequenza del *filtro di Sobel* per la rivelazione dei bordi orizzontali. La risposta impulsiva del filtro è:

$$h(m,n) = \begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

Il codice Matlab per filtrare in frequenza un'immagine contenuta nella variabile `x` con tale filtro è il seguente:

```
h = [1 0 -1; 2 0 -2; 1 0 -1];
[M,N] = size(x);
[A,B] = size(h);
P=M+A-1;
Q=N+B-1;
X=fft2(x,P,Q);
H=fft2(h,P,Q);
Y=H.*X;
y=real(ifft2(Y));
figure; imshow(y,[]);
```

A rigore, il risultato della DFT inversa dovrebbe essere puramente reale, in quanto equivalente alla convoluzione nel dominio dello spazio di due segnali reali. In pratica, errori dovuti alla precisione limitata comportano che l'antitrasformata possieda una piccola parte immaginaria, che viene trascurata usando il comando `real`.

Questa procedura risulta computazionalmente più efficiente del calcolo diretto della convoluzione, solo se il filtro è molto lungo (almeno maggiore di 32). Verificate che il risultato ottenuto nel dominio della frequenza è praticamente uguale a quello che si può ottenere nel dominio spaziale (fate attenzione al fatto che la maschera non è simmetrica).

2 Progetto dei filtri in frequenza

Proviamo adesso a progettare un filtro direttamente nel dominio della frequenza, e consideriamo un filtro passa-basso ideale, che ha come funzione di trasferimento:

$$H(\mu,\nu) = \begin{cases} 1 & D(\mu,\nu) \leq D_0 \\ 0 & \text{altrimenti} \end{cases}$$

dove $D(\mu,\nu) = \sqrt{\mu^2+\nu^2}$ e rappresenta la distanza dal punto (ν,μ) al centro del filtro, mentre D_0 rappresenta la frequenza di cut-off. Questo filtro ideale può essere simulato al calcolatore col seguente codice:

```
[M N] = size(x);          % dimensioni dell'immagine che si vuole elaborare
du = 1/M; dv = 1/N;
m = -1/2:du:1/2-du;
n = -1/2:dv:1/2-dv;
[l,k] = meshgrid(n,m);
D = sqrt(k.^2+l.^2);
D0 = 0.1;
H = (D <= D0);
figure; mesh(k,l,double(H));
```

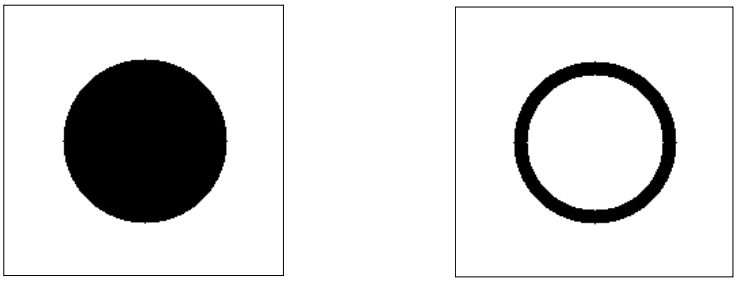

Figura 1: Esempio di filtro passa-alto ed elimina-banda (bianco è 1 e nero è 0).

A questo punto usiamo questo filtro per elaborare in frequenza l'immagine lena.jpg, visualizzando il risultato sia nel dominio frequenziale che in quello spaziale:

```
x = double(imread('lena.jpg'));
X = fft2(x,M,N);
X = fftshift(X);
figure; mesh(k,l,log(1+abs(X)));
Y=H.*X;
figure; mesh(k,l,log(1+abs(Y)));
Y=fftshift(Y);
y=real(ifft2(Y,M,N));
figure; imshow(y,[]);
```

Provate a cambiare la dimensione del raggio e visualizzate l'uscita. Noterete l'effetto di *ringing* che caratterizza un filtraggio ideale e che può essere sensibilmente ridotto se si usano filtri gaussiani con funzione di trasferimento:

$$H(\mu,\nu) = e^{-D^2(\mu,\nu)/2\sigma^2}$$

dove al crescere di σ aumenta la banda del filtro. Scrivete una funzione Matlab con il prototipo `function y = gaussLPF(x,sigma)` che implementa il filtraggio passa-basso gaussiano con un assegnato valore di σ. Quindi applicate questo filtro all'immagine `testo.tif`, che mostra un esempio di testo a bassa risoluzione che caratterizza di solito un fax o una fotocopia. Il testo in questa forma si presta poco ad essere elaborato da un calcolatore che effettua riconoscimento automatico, dato che sono presenti dei vuoti nei caratteri, un opportuno filtro passa-basso può migliorarne la visualizzazione. Questo filtro può anche essere usato per ringiovanire l'aspetto della donna raffigurata nell'immagine `volto.jpg`. Fate degli esperimenti, in entrambi i casi, al variare di σ.

Infine, implementate i filtri ideali passa-alto ed elimina-banda la cui risposta armonica visualizzata come immagine è mostrata in figura 1. Usateli per filtrare l'immagine lena.jpg e osservate il risultato al variare del raggio.

2.1 Filtraggio di rumore periodico

L'immagine contenuta nel file lenarumorosa.y (512×512, int16) mostra un disturbo periodico (righe verticali) che la rende sgradevole. Vogliamo progettare un filtro in grado di rimuovere solo la porzione dello spettro relativo alla sinusoide. A tal fine, valutiamo la trasformata di Fourier dell'immagine:

```
clear all; close all; clc;

fp = fopen('lenarumorosa.y','rb');
x = fread(fp, [512 512], 'int16');
x = x';
figure; imshow(x,[]); title('immagine rumorosa');
[M N] = size(x);

[n m] = meshgrid(1:N,1:M);
du = 1/M; dv = 1/N;
m = -1/2:du:1/2-du;
n = -1/2:dv:1/2-dv;
[l,k] = meshgrid(n,m);
X = fftshift(fft2(x));
figure; mesh(l,k,log(1+abs(X)));
title('Trasformata di Fourier immagine rumorosa');
```

Notate che teoricamente la sinusoide dovrebbe essere rappresentata da due impulsi in frequenza. Nella pratica non abbiamo degli impulsi ideali, bensì due sinc. Questo è dovuto al fatto che il rumore sinusoidale aggiunto è limitato all'immagine, cioè risulta troncato nello spazio lungo le due direzioni attraverso una finestra rettangolare bidimensionale. Il prodotto tra la sinusoide e la finestra in frequenza è la convoluzione tra i due impulsi e la sinc bidimensionale (trasformata dell'impulso rettangolare), quindi due sinc localizzate proprio alle frequenze specificate dalla sinusoide. Un possibile filtro che rimuove la sinuoside è il seguente:

```
% Definizione del filtro
nu_0 = 0.2; B = 0.03;
D = sqrt(k.^2+(l-nu_0).^2);
H1 = (D <= B);
D = sqrt(k.^2+(l+nu_0).^2);
H2 = (D <= B);
H = 1-H1-H2;
figure; mesh(l,k,double(H));
title('Riposta in frequenza del filtro');

% Filtraggio
Y = X.*H;
figure; mesh(l,k,log(1+abs(Y)));
title('Trasformata di Fourier immagine filtrata');
y = real(ifft2(fftshift(Y)));
figure; imshow(y,[]); title('Immagine filtrata');

% Calcolo MSE
fp = fopen('lena.y','rb');
xo = fread(fp, [512 512], 'uchar');
xo = xo';
figure; imshow(xo,[]); title('immagine originale');

MSE = mean2((xo-y).^2)
```

In realtà, con questo filtro non si riescono a rimuovere perfettamente le righe verticale, a causa della presenza dei lobi laterali, che presentano valori non trascurabili. E' possibile progettare un filtro rettangolare molto stretto che (conservando i valori intorno all'origine che corrispondono al contenuto frequenziale rilevante del segnale) sia diretto proprio lungo l'asse orizzontale.

```
% Definizione del filtro
B = 0.004;
H1 = (-B <= k) & (k <= B);
H2 = (-0.02 <= l) & (l <= 0.02);
H = 1-H1+H2.*H1;
figure; imagesc(double(H)); colormap(gray(3)); axis off;
title('Riposta in frequenza del filtro');
Y = X.*H;
y = real(ifft2(fftshift(Y)));
figure; imshow(y,[]); title('Immagine filtrata');

MSE = mean2((xo-y).^2)
```

Notate come l'errore quadratico medio tra immagine filtrata e originale (contenuta nel file lena.y, uchar) diminuisca con usando questa seconda soluzione.

2.2 Esercizi proposti

1. *Filtraggio notch.* L'immagine anelli.tif mostra una parte degli anelli che circondano Saturno. Il rumore sinusoidale è dovuto ad un segnale AC sovrapposto a quello della fotocamera prima di digitalizzare l'immagine. Tale interferenza è semplice da rimuovere se si progetta un filtro notch in grado di cancellare il contributo del rumore. Calcolate quindi la trasformata di Fourier dell'immagine, analizzatela, individuate il contributo realtivo al segnale sinusoidale e cercate di eliminarlo con il filtraggio.

2. *Pattern di Moiré.* L'immagine car.tif è caratterizzata dal pattern di moiré, un artefatto piuttosto fastidioso che può essere generato da una scansione non appropriata di una fotografia stampata su di un giornale. Dopo aver osservato attentamente la trasformata di Fourier dell'immagine, scrivete il codice matlab in cui rimuovete questo disturbo attraverso un opportuno filtro ideale e mostrate l'immagine risultante.

3. *Liveness detection.* Esistono diverse tecniche per cercare di scoprire se un'impronta digitale è autentica o contraffatta. Una delle più semplici opera nel dominio di Fourier, calcolando la frazione di energia contenuta alle medie frequenze, e dichiarando l'immagine autentica se tale frazione supera una certa soglia.

   Scrivete una funzione `function EM = elabora(x, r1, r2)` che calcola la DFT-2D, $X(\mu, \nu)$, di un'immagine $x(m, n)$, valuta l'energia alle medie frequenze come $E_M = \frac{1}{|\Omega|} \sum_\Omega |X(\mu, \nu)|^2$ dove $\Omega = \{(\mu, \nu) : r_1 \leq \sqrt{\mu^2 + \nu^2} \leq r_2\}$ e $|\Omega|$ è la cardinalità di Ω, cioè il numero di punti che soddisfano questa condizione.

   Applicate la funzione alle due immagini impronta1.tif e impronta2.tif, usando raggio interno $r_1 = 0.10$ e raggio esterno $r_2 = 0.25$ ed etichettate come vera quella che fornisce il valore maggiore di E_M/E, con E energia dell'immagine.