Lezione n.5 - Soluzioni

Laboratorio di Telecomunicazioni

L.Verdoliva

1 Rappresentazione delle immagini

Di seguito la funzione per rappresentare un'immagine in formato JPEG e memorizzarla in una matrice X:

```
function X = vedi(nomefile);
% carica un'immagine memorizzata in formato JPEG e la visualizza

fid = fopen(nomefile,'r');
if fid < 0,
    fprintf('Il file %s non esiste!',nomefile);
else
    X = imread(nomefile);
    figure; image(X); axis image; colormap(gray(256));
end
```

2 Elaborazione delle immagini

a) *Esempio di applicazione.*

```
X = double(imread('fax.jpg'));
figure(1); subplot(121);
image(X(1:80,30:150)); colormap(gray(256)); axis image;
title('Zoom immagine originale');
L = [2 3 4];
for i = 1:3,
    h = ones(1,L(i))/L(i);
    Y1 = conv2(h,X);
    h = h';
    Y = conv2(h,Y1);
    figure(1); subplot(122);
    image(Y(1:80,30:150)); colormap(gray(256)); axis image;
    title(sprintf('Zoom immagine filtrata con L=%d',L(i))); pause;
end
```

Potete osservare come elaborare l'immagine con un filtro media aritmetica tenda a sfocare il testo, e quindi a riempire i vuoti nei caratteri. Il testo in questa forma risulta così più adatto ad essere elaborato da un calcolatore che effettua riconoscimento automatico.

Inoltre notate come la cornice nera intorno all'immagine cresca all'aumentare della dimensione del filtro. Questo è un effetto dovuto al fatto che l'operazione di convoluzione prevede di ribaltare la risposta impulsiva e poi calcolare il prodotto con il segnale (in questo caso la riga o la colonna dell'immagine), che risulta essere nullo al di fuori dei bordi. Un modo per risolvere questo problema è utilizzare il comando `filter2` che fornisce in uscita un segnale delle stesse dimensioni di quello in ingresso (e non più lungo come prevede la convoluzione) perché prende solo la parte centrale del risultato, trascurando ciò che accade ai bordi.

- b) *Filtri che enfatizzano le discontinuità.* Questi filtri tendono ad evidenziare i bordi presenti in un'immagine. Notate poi come la somma dei coefficienti del filtro sia nulla, per cui su aree omogenee la risposta sarà nulla.

```
X = double(imread('lena.jpg'));
figure(1); image(X); colormap(gray(256)); axis image;
title('immagine originale');
h = [-2 -1 0 1 2];
Y1 = filter2(h,X);
h = h';
Y = filter2(h,Y1);
figure(2); imagesc(Y); colormap(gray(256)); axis image;
title('immagine filtrata');
```

- c) *Ridimensionamento.*

```
X = double(imread('lena.jpg'));
figure; image(X); colormap(gray(256)); axis image;
title('immagine originale');
[M,N] = size(X);
a = [2:2:8];
for i=1:4,
    Y = X(1:a(i):M,1:a(i):N);
    figure; image(Y); colormap(gray(256)); axis image;
    title(sprintf('immagine rimpicciolita con a=%d',a(i)));
end
```

In realtà per una visualizzazione più corretta bisognerebbe adattare la finestra alle dimensioni dell'immagine, usando il seguente comando:

```
figure; imshow(Y,[]);
```

3 Rappresentazione ed elaborazione di un segnale audio

a) *Decimazione.*

```
[x, fc] = wavread('Benigni_Troisi_progresso.wav');  
x = x(1:2:length(x));  
sound(x,fc);
```

b) *Generazione di echi.*

```
clear all; close all; clc;  
[x,fc,nbit] = wavread('audio');  
nx = [1:length(x)];  
sound(x,fc); pause;  
x = x';  
  
g = 0.8; M = 10000;  
ny = nx + M;  
y = g*x;  
y = somma(x,nx,y,ny);  
sound(y,fc);
```

In alternativa, riconoscendo che il sistema è LTI con risposta impulsiva $h(n) = \delta(n) + g\delta(n - M)$, è anche possibile usare il seguente codice:

```
g = 0.8; M = 10000;  
h = [1 zeros(1,M-1) g];  
y = conv(x,h);  
sound(y,fc);
```