

Elaborazione di Segnali Multimediali
a.a. 2017/2018

Elaborazioni nel dominio della frequenza Soluzioni

1 Progetto dei filtri in frequenza

1. *Testo a bassa risoluzione.* Usiamo il filtro passa-basso gaussiano per migliorare un testo a bassa risoluzione:

```
clear all; close all; clc;
x = double(imread('testo.tif'));
[M,N] = size(x);

P = M; Q = N;
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);
s = [0.05 0.15 0.25];
for i=1:3,
    H = exp(-D.^2/(2*s(i)^2));    % filtro gaussiano
    X = fftshift(fft2(x));
    Y = H.*X;
    y=real(ifft2(fftshift(Y),M,N));
    figure; imshow(y,[]);
    title(sprintf('Filtraggio gaussiano con sigma = %0.5g',s(i)))
end
```

2. *Filtraggio notch.* Questo problema richiede un po' di attenzione, infatti prima di procedere nel progetto del filtro è necessario analizzare attentamente sia l'immagine rumorosa che la sua trasformata di Fourier. In effetti dall'immagine si può notare come il fastidio si presenta con delle righe orizzontali, ovvero la sinusoide varia proprio lungo l'asse verticale così come si può osservare nel dominio di Fourier dai picchi di energia presenti proprio lungo l'asse verticale vicino all'origine. Un possibile filtro che rimuove il disturbo sinusoidale è il seguente:

```

% Definizione del filtro
B = 0.004;
H1 = (-B <= l) & (l <= B);
H2 = (-0.02 <= k) & (k <= 0.02);
H = 1-H1+H2.*H1;
figure; imshow(double(H), []);
title('Riposta in frequenza del filtro');
Y = X.*H;
figure; imshow(log(1+abs(Y)), []);
title('Trasformata di Fourier immagine filtrata');

y = real(ifft2(fftshift(Y)));
figure; imshow(y, []); title('Immagine filtrata');
noise = y - x;
figure; imshow(noise, []); title('Rumore eliminato');

```

Un'ultima considerazione. Nel progettare un filtro non esiste una regola fissa, il tipo di problema guida la scelta di filtro, e tale scelta non è detto che sia l'unica possibile.

3. *Pattern di Moiré.*

```

x = imread('car.tif');
[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; imshow(log(1+abs(X)), []);

% Definizione dei filtri
B = 0.023;
mu_1 = 0.15; nu_1 = 0.18;
mu_2 = 0.17; nu_2 = 0.16;
D = sqrt((k-mu_1).^2+(l-nu_1).^2);
H1 = (D <= B);
D = sqrt((k-mu_2).^2+(l+nu_2).^2);
H2 = (D <= B);
D = sqrt((k+mu_1).^2+(l+nu_1).^2);
H3 = (D <= B);
D = sqrt((k+mu_2).^2+(l-nu_2).^2);
H4 = (D <= B);

```

```

B = 0.015;
mu_1 = 0.32; mu_2 = 0.34;
D = sqrt((k-mu_1).^2+(1-nu_1).^2);
H5 = (D <= B);
D = sqrt((k-mu_2).^2+(1+nu_2).^2);
H6 = (D <= B);
D = sqrt((k+mu_1).^2+(1+nu_1).^2);
H7 = (D <= B);
D = sqrt((k+mu_2).^2+(1-nu_2).^2);
H8 = (D <= B);
H = 1-H1-H2-H3-H4-H5-H6-H7-H8;

Y = X.*H;
figure; imshow(log(1+abs(Y)), []);
title('Trasformata di Fourier immagine filtrata');

y = real(ifft2(fftshift(Y)));
figure; subplot(1,2,2); imshow(y, [min(y(:)) max(y(:))]);
title('Immagine filtrata');
subplot(1,2,1); imshow(x, [min(y(:)) max(y(:))]);
title('immagine originale');

```

4. Liveness Detection.

```

clear all; close all; clc;
x = imread('impronta1.tif');
y = imread('impronta2.tif');

[M,N]=size(x);
P=M; Q=N;
du=1/P; dv=1/Q;
m=-1/2:du:1/2-du; n= -1/2:dv:1/2-dv;
[l,k] = meshgrid(n,m);
X=fftshift(fft2(x,P,Q));
Z1=log(1+abs(X));
figure; imshow(Z1, []);
Y=fftshift(fft2(y,P,Q));
Z2=log(1+abs(Y));
figure; imshow(Z2, []);

D = sqrt(k.^2+l.^2);
D1 = 0.10;
H1 = (D <= D1);
D2 = 0.25;
H2 = (D <= D2);
H = H2-H1;

```

```
Zf1 = X.*H;
Zf1v=log(1+abs(Zf1));
figure; imshow(Zf1v, []);
E1 = sum(abs(Zf1(:)).^2)/(N*M);
E = sum(abs(X(:)).^2)/(N*M);
E1/E

Zf2 = Y.*H;
Zf2v=log(1+abs(Zf2));
figure; imshow(Zf2v, []);
E2 = sum(abs(Zf2(:)).^2)/(N*M);
E = sum(abs(X(:)).^2)/(N*M);
E2/E
```