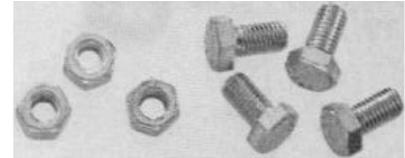


Esame del corso di Tecniche Avanzate per il Trattamento delle Immagini

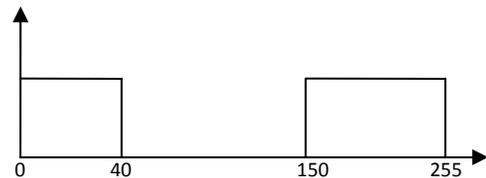


Data: 2 Marzo 2009

Es.1. [6 Pt] Indicare come, tramite l'impiego combinato della trasformata di Fourier 2D e di altre opportune trasformazioni, sia possibile ottenere delle *feature* invarianti a traslazione e rotazione per discriminare degli oggetti come quelli riportati in figura. Si vuole invece mantenere la capacità di distinguere scale diverse al fine di riconoscere dadi e bulloni di diverse dimensioni.

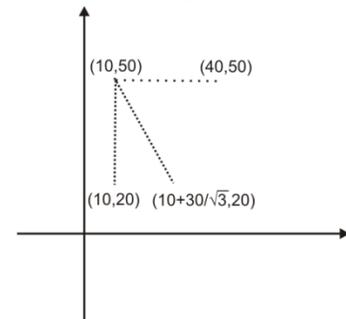


Es.2. [5 Pt] Un'immagine contiene al suo interno oggetti molto scuri e molto chiari; l'istogramma è riportato qui a fianco. Indicare una possibile funzione di trasferimento che consenta di utilizzare in modo opportuno tutta la dinamica disponibile migliorando la visibilità di entrambi gli oggetti. Indicare, inoltre, i passi necessari per applicare tale trasformazione.



Es.3. [8 Pt] Indicare la posizione dei tre massimi principali per la trasformata di Hough nell-immagine a fianco.

Es.4. [6 Pt] Sia data in ingresso un'immagine grayscale A e si voglia operare l'estrazione dei contorni di quest'immagine in tutte le direzioni. Si descriva l'operazione di estrazione dei contorni tramite due diversi algoritmi: a) operatore Laplaciano e b) operatore gradiente (scegliendo uno tra i vari tipi di operatori studiati). Si descrivano e si implementino mediante codice Matlab i passi necessari per estrarre i contorni con le due procedure.



Inoltre, si descriva una terza procedura (diversa dal Laplaciano e dal gradiente) per l'estrazione dei contorni. Si descrivano i passi necessari dell'algoritmo mediante codice Matlab.

Es.5. [7 Pt] Data un'immagine truecolor memorizzata in un file immagine.bmp ed avente i dati rappresentati ad 8bit per ogni piano di colore si implementino i seguenti punti mediante codice Matlab:

Leggere, caricare nel workspace e visualizzare l'immagine.

Convertire l'immagine di partenza (truecolor) ad una rappresentazione a 256 toni di grigio.

Calcolare e visualizzare l'istogramma dell'immagine a 256 toni di grigio.

Si equalizzi l'istogramma dell'immagine a 256 toni di grigio (utilizzando la funzione predefinita in Matlab), riducendo il numero dei bins da 256 a 64.

Poi si visualizzi l'istogramma equalizzato.

Calcolare la trasformata di Fourier mediante FFT dell'immagine ottenuta al passo precedente e visualizzare modulo, fase e spettro di potenza ottenuti.

Fare in modo che il valore di modulo della continua, risultante dalla FFT, appaia al centro dei rispettivi grafici (e, di conseguenza, i valori della altre frequenze spaziali).

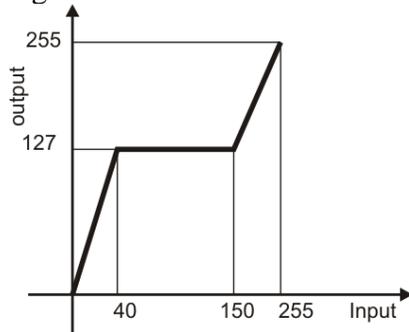
Elenco funzioni Matlab:

```
abs
angle
double
fft2
fftshift
getimage
histeq
im2bw
im2double
imag
imagesc
imfilter
imhist
imread
imshow
real
size
rgb2gray
wavedec2
waverec2
```

Soluzioni

Es. 1 Al fine di rendersi invarianti alla traslazione, una volta segmentati i diversi oggetti presenti, occorre applicare a ciascuno di essi una trasformata di Fourier Bidimensionale e prenderne il modulo. Una volta effettuata tale operazione, siccome la trasformata di Fourier 2D, a fronte di una rotazione dell'oggetto iniziale, ruota della stessa quantità è possibile effettuare una trasformata in coordinate polari del modulo della trasformata precedente. Una rotazione dell'oggetto viene così trasformata in una traslazione lungo l'asse degli angoli nelle coordinate polari. Una successiva ulteriore trasformata di Fourier 2D e la conseguente estrazione del modulo ci renderà invarianti anche a questa seconda traslazione, ottenendo così dei descrittori invarianti a traslazione e rotazione.

Es. 2 La soluzione non è univoca, si può scegliere di espandere gli oggetti scuri tra le intensità di 0 e 127 mentre gli oggetti chiari possono essere espansi tra 128 e 255. Per fare ciò si può applicare la seguente trasformazione:



Es. 3 La trasformata di Hough individuerà le tre rette principali tramite le loro distanze e dall'angolo formato dall'ortogonale rispetto alla retta, in particolare saranno:

$$\rho_1 = 10; \theta_1 = 0.$$

$$\rho_2 = 50; \theta_2 = 90^\circ.$$

$$\rho_3 = (50 + 10\sqrt{3})/2; \theta_3 = 30^\circ.$$

Es. 4 Le procedure richieste per l'estrazione dei contorni sono il filtraggio spaziale tramite operatore Laplaciano, l'estrazione dei contorni tramite operatore gradiente (in questa soluzione abbiamo scelto il gradiente di tipo Roberts) e l'estrazione dei contorni tramite scomposizione wavelet.

Con il metodo del Laplaciano, si filtra spazialmente l'immagine in input con l'operatore Laplaciano con isotropia per rotazioni di 45 gradi; poi, per evidenziare i contorni di un'immagine, si calcola la differenza tra l'immagine in input e l'immagine dopo l'applicazione del operatore Laplaciano.

Con il metodo dell'operatore gradiente di tipo Roberts, il gradiente viene calcolato separatamente per 45 e 135 gradi nell'immagine, per poi sommare i risultati dei due filtraggi e applicare una soglia per l'estrazione dei contorni.

Il metodo della scomposizione wavelet di un'immagine può essere un valido strumento per l'estrazione dei contorni di un'immagine. Si vedano di seguito i passi dell'algoritmo e il codice Matlab.

Codice Matlab per l'estrazione dei contorni tramite operatore Laplaciano:

- Si calcola l'operatore Laplaciano con isotropia per rotazioni di 45 gradi e si filtra spazialmente l'immagine in input A

```
h = [1 1 1;  
     1 -8 1;  
     1 1 1];
```

```
A_h = imfilter(A, h, 'symmetric', 'conv', 'same');
```

- Si calcola la differenza tra l'immagine in input e l'immagine dopo l'applicazione del operatore Laplaciano

```
A_edge_Laplacian = A - A_h;
```

Codice Matlab per l'estrazione dei contorni tramite operatore gradiente di Roberts:

- Si calcola il gradiente di Roberts separatamente per edge a 45 e 135 gradi, e si filtra spazialmente l'immagine in input A nelle due direzioni

```
h_x = [0 0 -1;
       0 1 0;
       0 0 0];
h_y = [-1 0 0;
       0 1 0;
       0 0 0];
A_h_x = imfilter(A, h_x, 'symmetric', 'conv', 'same');
A_h_y = imfilter(A, h_y, 'symmetric', 'conv', 'same');
```

- Si sommano i risultati dei due filtraggi e poi si applica un'operazione di soglia

```
A_h = A_h_x + A_h_y;
A_edge_gradient = im2bw(A_h, 0.3);
```

Codice Matlab per l'estrazione dei contorni tramite wavelet:

- Si calcolano i coefficienti della scomposizione wavelet dell'immagine in input ad un determinato livello, ad esempio 2.

```
livello = 2;
[C S] = wavedec2(double(A), livello, 'dbl');
```

- Si pongano a 0 i coefficienti della componente di approssimazione

```
C_no_a = C;
C_no_a(1, 1:(S(1,1)*S(1,2))) = 0;
```

- Si ricostruisca l'immagine a partire dai coefficienti della scomposizione modificati

```
A_edge_wavelet = waverec2(C_no_a, S, 'dbl');
```

Es. 5

```
img_rgb = imread('immagine.bmp');
figure;
imshow(img_rgb);
(oppure: figure;
        imshow('immagine.bmp');
        img_rgb = getimage);
```

- `img_gray = rgb2gray(img_rgb);`
- `h = imhist(img_gray);`
`figure; imhist(img_gray); title('Istogramma immagine grayscale');`
- `img_gray_equal = histeq(img_gray, 64);`
`figure; imhist(img_gray_equal); title('Istogramma equalizzato');`
- `img_gray_FFT = fft2(im2double(img_gray));`
`img_gray_FFT_magnitude = abs(img_gray_FFT);`
`img_gray_FFT_phase = angle(img_gray_FFT);`
`img_gray_FFT_power_spectrum = real(img_gray_FFT).^2+imag(img_gray_FFT).^2;`
`figure; imagesc(img_gray_FFT_magnitude); axis equal;`
`title('Modulo spettro immagine');`
`figure; imagesc(img_gray_FFT_phase); axis equal;`
`title('Fase spettro immagine');`
`figure; imagesc(img_gray_FFT_power_spectrum); axis equal;`
`title('Spettro di potenza immagine');`
- nel dominio delle frequenze:
`img_gray_FFT = fftshift(img_gray_FFT);`

nel dominio dei tempi opero prima lo shifting della continua e poi calcolo la FFT 2D:

```
for i = 1:size(img_gray, 1)
    for j = 1:size(img_gray, 2)
        img_gray_shift(i,j) = double(img_gray(i,j))*((-1)^(i+j));
```

```
    end  
end  
img_gray_FFT = fft2(im2double(img_gray_shift));
```