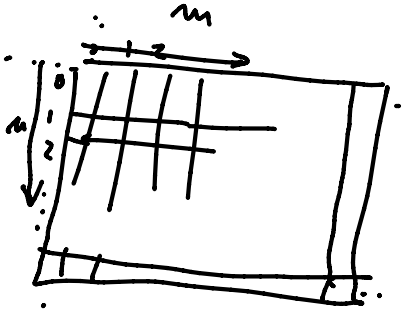
Nozioni Elementari sulla elaborazione delle Immagini 2

Corso: Signal Processing and Data Fusion
AA 2020-2021
Dipartimento di Ingegneria, Università della Campania
Laurea Magistrale in Ingegneria Elettronica
Prof. Francesco A. N. Palmieri

Le elaborazioni finora viste nelle immagini riguardavano modifiche pixel-per-pixel in maniera indipendente. Il passo successivo che possiamo fare è considerare un intorno di ogni pixel per definire delle operazioni di trasformazione delle caratteristiche spaziali di un'immagine.

Consideriamo per ora solo immagini monocromatiche (luminanza o B/W)



$f[n,m]$

Supponiamo per ora che la regione 2D non sia limitata

Date due regioni $f[n,m]$ e $h[n,m]$ la convoluzione lineare tra esse è definita come

$$(1)\quad y[n,m]=\sum_{\beta=-\infty}^{+\infty}\sum_{\alpha=-\infty}^{+\infty} f[\alpha,\beta]\,h[n-\alpha,m-\beta]=(f*h)[n,m]$$

cambiando variabili si può anche scrivere

$$(2)\quad y[n,m]=\sum_{\beta=-\infty}^{+\infty}\sum_{\alpha=-\infty}^{+\infty} h[\alpha,\beta]\,f[n-\alpha,m-\beta]$$

L'operazione di convoluzione 2D può essere visualizzata, come nel caso 1D, mediante operazioni di "flip-and-shift".

Guardando l'espressione (1) notiamo che

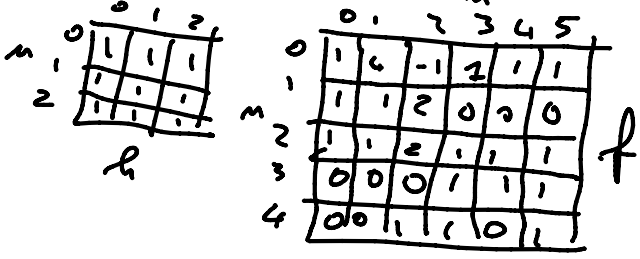
$$h[n-\alpha,m-\beta]=h[-(\alpha-n),-(\beta-m)]$$

$$h[n-\alpha, m-\beta] = h[-(\alpha-n), -(\beta-m)]$$

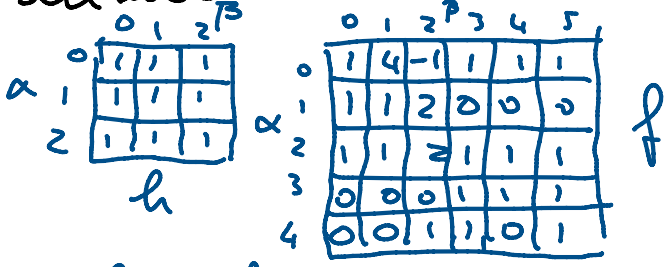
ovvero $h[n-\alpha, m-\beta]$ è la versione ribaltata di $h[\alpha,\beta] \to h[-\alpha,-\beta]$ (in entrambe le coordinate, e di una traslazione in 2D di n verticalmente e di m orizzontalmente

ESEMPIO

Si calcoli la convoluzione tra

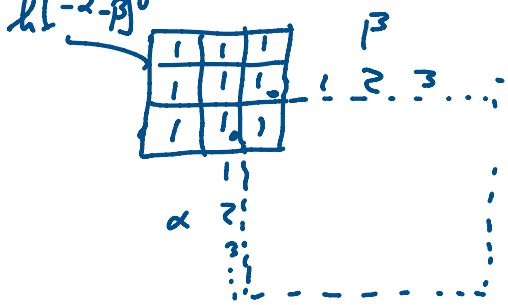


Traduciamo h e f nelle variabili all'interno della sommatoria

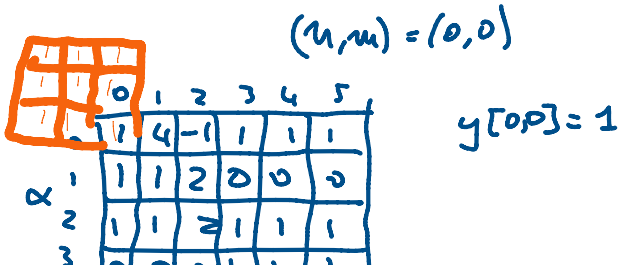


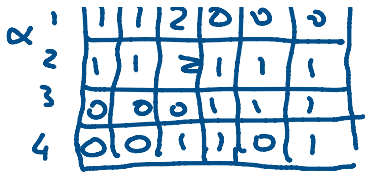
flip di h

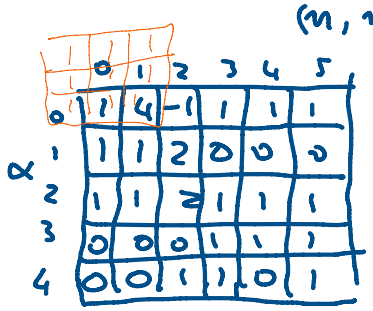
$h[-\alpha-\beta]$



ora dobbiamo moltiplicare per f e sommare per ogni valori di n, m.
Assumiamo che la due sequenze all'esterno dell'intervallo di definizione siano nulle.

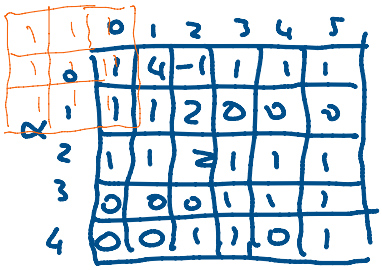






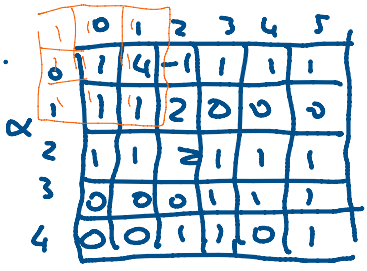
$(n, m) = (0,1)$

$y[0,1] = 4$



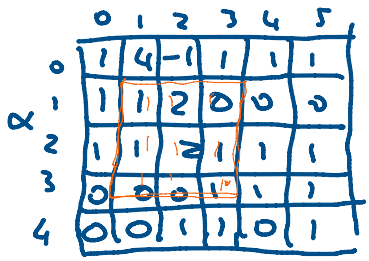
$(n,m) = (1,0)$

$y[1,0] = 2$



$(n,m) = (1,1)$

$y[1,1] = 7$

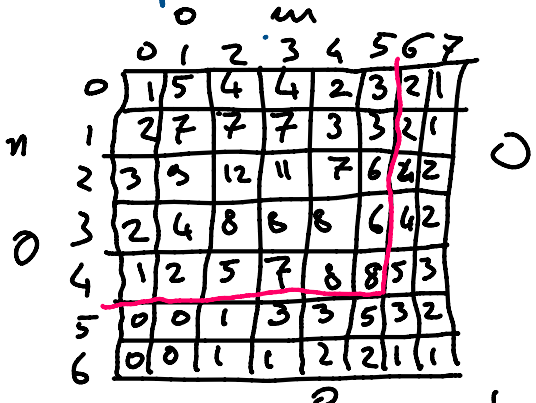


$(n,m) = (3,3)$

$y[3,3] = 8$

ecc.

fino a produrre il rivelatore

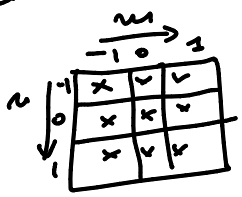


Ci si può ottenere con una formula da far scrivere sul foglio.

da far scivolare nel foglio.
Notare come il risultato si estende oltre gli intervalli [0,M-1] e [0,N-1].

La procedura è facilmente adattabile con una definizione diversa degli indici. Per esempio

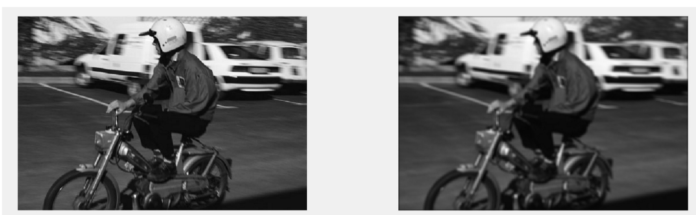
h[n,m]



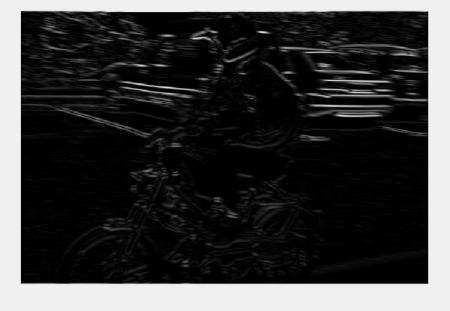
La convoluzione, che è una operazione abbastanza onerosa, può essere eseguita in MATLAB come segue

```
clear all
% Experiments of 2D filtering

% read an image in an uncompressed format
Ior = imread('moto.jpg');

Ic = im2double(Ior);
%reduce to B&W
c1=0.299;
c2=0.587;
c3=0.114;
I(:,:)=c1*Ic(:,:,1)+c2*Ic(:,:,2)+c3*Ic(:,:,3);
subplot(1,2,1)
imshow(I,[])


B=(1/25)*ones(5,5); %media mobile
%define mask
% B=    [1 0 -1 0;
%        1 0 -1 0;
%        1 0 -1 0;
%        1 0 -1 0;
%        1 0 -1 0;
%        1 0 -1 0;
%        1 0 -1 0]';
% B=(1/20)*diag([1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]);
% B=fspecial('sobel');
B = fspecial('laplacian')
% B = fspecial('unsharp',0.8)
Y=filter2(B,I,'same');
subplot(1,2,2)
imshow(abs(Y),[])
```

Il programma prevede diverse maschere con cui sperimentare la convoluzione .
La prima è una media mobile con una maschera 5x5.



Il risultato è una sfocatura dell'immagine.
L'esperimento successivo è con una maschera che opera una sorta di gradiente verticale e che quindi rivela le transizioni verticali

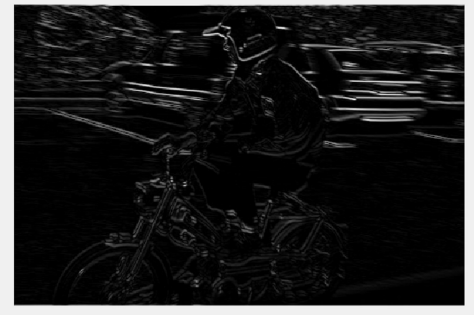


Ancora una maschera con elementi sulla diagonale che quindi sfoca i pixel in diagonale (effetto "mosso" in direzione diagonale)

Ancora una maschera Sobel

B = 1 2 1
0 0 0
-1 -2 -1

Che fa (simile all'esempio di sopra) una sorta di gradiente verticale



Ancora un esperimento con la maschera

B =0.1667 0.6667 0.1667
0.6667 -3.3333 0.6667
0.1667 0.6667 0.1667

Che è un approssimazione discreta del Laplaciano

E ancora con la maschera

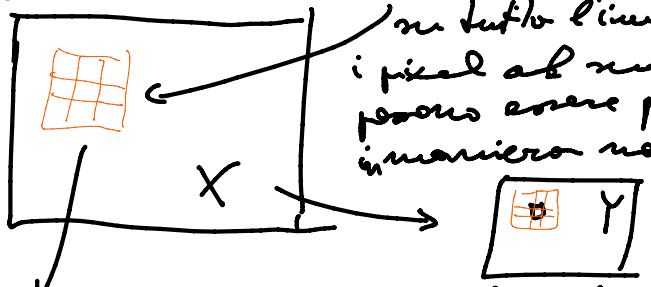
B =-0.4444 -0.1111 -0.4444
-0.1111 3.2222 -0.1111
-0.4444 -0.1111 -0.4444

Quindi le operazioni possibili con opportuna definizione della maschera di convoluzione possono produrre effetti più diversi. Altri esempi e discussioni sono contenuti nel libro di Gonzales, suggerito come lettura di riferimento.

Filtri non lineari:

Le operazioni viste sopra con la combinazione lineare di pixel possono essere generalizzate mediante delle funzioni non lineari:

definita una maschera da muovere su tutta l'immagine, i pixel al suo interno possono essere processati in maniera non lineare.



Es. 3×3

$$\begin{matrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{matrix} \Rightarrow$$

Assegna al pixel in Y corrispondente al pixel centrale $f(x_1, x_2 \dots x_9)$

Il filtro più famoso è il cosiddetto "filtro a mediana" in cui f estrae la mediana di $x_1, x_2 \dots x_9$.

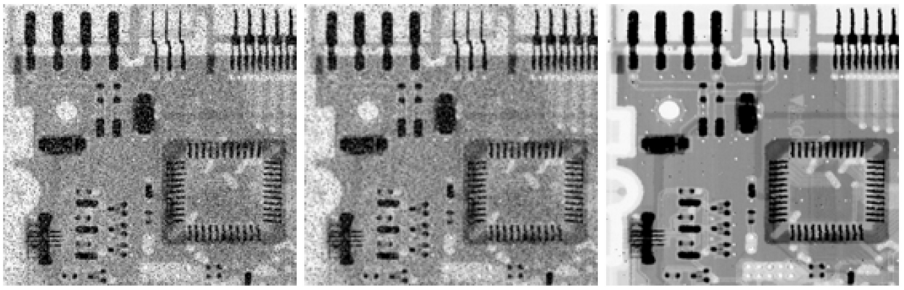
La mediana di un campione è il valore centrale del campione ordinato. Ad esempio

3 2 0 1 0 5 7 0 8 campione

0 0 0 1 (2) 3 5 7 8 campione ordinato

↓ mediana

Filtri a mediana sono eccellenti per rimuovere rumore impulsivo dalle immagini.



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Per esempi e approfondimenti sugli argomenti classici sulla elaborazione delle immagini, si rimanda lo studente all'ottimo libro
R. C. Gonzales and R. E. Woods, Digital Image Processing, 2nd edition, Prentice Hall, 2001.