Esercitazione di Laboratorio 10.5.2012
Elaborazione dell'Immagine

Nota. Il link di riferimento con i file comuni è la directory
`http://www.mat.unimi.it/users/naldi/elabimm`
(Utilizzare le immagini: *cameraman.tif*, *blood1.tif*, *tungsteno-test.tif*).

1 Convoluzioni e correlazioni, 1D

Dati due vettori S e H definiamo il prodotto di cross-correlazione $R = S \otimes H$ come,

$$R(n) = \sum_k H(k)S(n+k).$$

Si implementi attraverso un M-file tale prodotto: in ingresso si ricevano i vettori S, H e si restituisca il vettore R. Per le condizioni al bordo si consideri, a scelta, lo zero-padding (estendo a zero il segnale) oppure condizioni periodiche.
Si sperimenti il proprio M-file estraendo una riga dall'immagine test `cameraman` ed utilizzando le "maschere"

$$H1 = [1\ \ 1\ \ 1]/3, \quad H2 = [-1\ \ 1], \quad H3 = [1\ \ -2\ \ 1].$$

Si osservi se nella riga estratta sono presenti dei "probabili" bordi e si descriva l'effetto dei tre filtri sopra indicati.
Il prodotto di convoluzione tra S e H, $C = S \star H$ è invece definito come,

$$C(n) = \sum_k S(k)H(n-k).$$

Possiamo utilizzare la funzione scritta per il prodotto $\otimes$ per l'implementazione del prodotto di convoluzione? Come? Sperimentare il *Teorema di rappresentazione* con l'operartore rappresentato da $H2$ ed il segnale di prova

```
>> s=[1 5 0 10 15 2 20 3 2 1]
```

La funzione MATLAB `conv` calcola il prodotto di convoluzione tra due vettori `S` e `H`, `C=conv(S,H)`. In questo caso la convoluzione equivale al prodotto tra polinomi i cui coefficienti sono memorizzati nei due vettori dati (con le convenzioni per il grado e l'ordine dei coefficienti di MATLAB). Quindi la lunghezza `length(C)` del vettore `C` è uguale a `length(S)`+`length(H)`-1. I vettori S e H sono pensati prolungati a zero nel caso in cui gli indici escano dal dominio opportuno. Utilizzare la funzione MATLAB `conv` nell'esperimento numerico descritto sopra e confrontare il risultato con quello ottenuto con la propria funzione.

2 Maschere, bordi e correlazione, 2D

Nel caso bidimensionale il prodotto di cross-correlazione $R = S \otimes h$, con h maschera e S segnale originale (entrambe matrici), si scrive “componente per componente” come

$$R_{nm} = \sum_i \sum_j \; h_{ij} S_{n+i,m+j}.$$

Un metodo per la stima del gradiente nel pixel di posizione (x, y) all'interno di un'immagine S consiste nel calcolo della media di tre differenti incrementi (faremo un leggero abuso di notazioni):

$$\begin{aligned}
\frac{\partial S}{\partial x} &\approx \tfrac{1}{3}\Big[\tfrac{S(x+1,y)-S(x-1,y)}{2} + \tfrac{S(x+1,y-1)-S(x-1,y-1)}{2} + \\
&\quad + \tfrac{S(x+1,y+1)-S(x-1,y+1)}{2}\Big], \\
\frac{\partial S}{\partial y} &\approx \tfrac{1}{3}\Big[\tfrac{S(x,y+1)-S(x,y-1)}{2} + \tfrac{S(x-1,y+1)-S(x-1,y-1)}{2} + \\
&\quad + \tfrac{S(x+1,y+1)-S(x+1,y-1)}{2}\Big].
\end{aligned}$$

Implementare attraverso una opportuna scelta della maschera h il calcolo delle componenti approssimate del gradiente come sopra descritto con un prodotto di correlazione. Generare una nuova immagine in cui si visualizza il modulo del gradiente discreto così ottenuto. Ripetere il calcolo evitando le divisioni (cambia solo un fattore di scala) e confrontare le immagini del modulo del gradiente ottenute.
Utilizzare lo script per l'implementazione del prodotto di convoluzione (va modificata? come?),

$$C = S \star h, \quad C_{nm} = \sum_i \sum_j \; h_{ij} S_{n-i,m-j}.$$

Nota. Per una implementazione semplice dei prodotti $\star$ e $\otimes$ occorre prevedere dei cicli, in quello più interno si seleziona la sottomatrice opportuna di S ed, utilizzando la possibilità di operatore “moltiplicazione punto” `.*`, si calcola la sommatoria per la componente di C. [Osservazione: occorre “leggere” il filtro inversamente quando si opera con la convoluzione, si vedano le funzioni MATLAB `flipud` e `fliplr`]
Esercizio. Utilizzando il prodotto di correlazione come prodotto scalare creare una maschera h adatta per la ricerca di bordi “diagonali”, ovvero attraverso le variazioni rispetto alla diagonale principale e all'anti-diagonale in un $8-$intorno di un pixel.
Scrivere la maschera opporune per una operazione di smoothing utilizzando una opportuna media aritmetica nell'$8-$intorno di un pixel di posizione (x, y),

$$\frac{\sum_{i=-1}^{1} \sum_{j=-1}^{1} \; S(x+i, y+j)}{9},$$

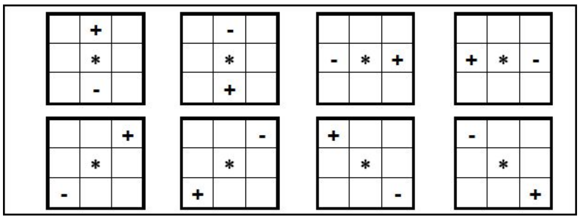

Figure 1: Disposizioni per la ricerca di uno zero crossing.

oppure utilizzando una media pesata in cui i pesi variano in accordo con la funzione Gaussiana

$$g(z,w) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}};$$

dove $d = \sqrt{(z-x)^2 + (w-y)^2}$ è la distanza da un pixel dell'intorno al pixel centrale (x, y) e σ un parametro opportuno. Estendere l'intorno e variare il parametro σ. Utilizzare il filtro ottenuto per un'immagine con rumore di tipo *salt and pepper*: cosa accade? Ripetere il calcolo del modulo del gradiente discreto facendolo precedere dall'operazione di smoothing.

Esercizio. Calcolare il valore del Laplaciano discreto

$$\Delta S = \frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2},$$

sempre attraverso un opportuno prodotto di correlazione (prima valutare un rapporto incrementale per la discretizzazione delle derivate seconde). Sia L la matrice che contiene i valori del Laplaciano discreto: scorrere ogni riga e trovare i punti *zero crossing*, ripetere l'operazione per le colonne. Un pixel sarà uno zero crossing se attraverso il pixel accade una delle situazioni riportate in Figura 1. Indicando con $\{a, -b\}$ i valori corrispondenti al passaggio da uno zero crossing, indichiamo con $|a+b|$ la pendenza dello zero crossing, applicare una operazione di soglia per selezionare solo gli zero crossing con pendenza maggiore della soglia prefissata. Rappresentare i punti selezionati.

Nota. In MATLAB è disponibile la funzione `conv2` per la convoluzione bidimensionale, utilizzo principale

```
C = conv2( X,m,'shape')
```

dove `shape` può assumere i valori `full`, valore di default, per ritornare il risultato completo della convoluzione bidimensionale; `same` per avere come risultato una matrice delle stesse dimensioni di `X`; `valid` per avere la parte della convoluzione che non richiede valori posti a zero per estensione di `X` (dimensioni quindi uguali a `size(C)=[ma-mb+1,na-nb+1]` dove `size(X)=[ma , na]`, `size(m)=[mb,nb]`).

Esercizio. Utilizzare la funzione `conv2` utilizzando l'immagine del cameraman e il filtro media,

$$m = \frac{1}{N^2} \begin{pmatrix} 1 & 1 & ... & 1 \\ 1 & 1 & ... & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & ... & 1 \end{pmatrix},$$

dove m è una matrice quadrata $N \times N$, per differenti valori di N.

Esercizio. Stimare la norma del gradiente tramite la convoluzione con gli operatori tipo Sobel

```
Mx = [-1 0 1;-2 0 2;-1 0 1]; My = [1 2 1; 0 0 0; -1 -2 -1];
```

per la valutazione delle componenti del gradiente discreto.

Scrivere una funzione per il calcolo di un filtro `h` di tipo Gaussiano, ovvero che corrisponda ad una certa distribuzione Gaussiana di assegnata varianza centrata nella posizione centrale della matrice, che deve quindi avere ordine dispari. Si consideri la distribuzione normalizzata, massimo valore uguale ad uno.