

**PROVA SCRITTA DI ELABORAZIONE DI SEGNALI MULTIMEDIALI del 30.6.14**  
**(Ingegneria delle Telecomunicazioni)**

**Tempo: 2 ore e mezza. NON è consentito l'uso di materiale didattico e appunti propri.**

**EX. 1** Si vuole effettuare il filtraggio dell'immagine lena\_rumorosa.raw (256 × 256, float) affetta da rumore moltiplicativo. A tale scopo si vuole adottare una strategia di tipo adattativo che opera mediante finestra scorrevole sull'immagine  $x$ . L'algoritmo lavora in due passi:

1. si determina una prima stima usando una finestra  $k \times k$  con  $k = 5$  nel seguente modo:

$$y_1(m, n) = x(m, n)W_l + \mu_l(1 - W_l)$$

con  $W_l = (1 - C_u^2/C_x^2)/(1 + C_u^2)$  dove  $C_x = \sigma_l/\mu_l$  è il coefficiente di variazione locale (cioè relativo al blocco  $k \times k$ ) pari al rapporto tra la deviazione standard locale  $\sigma_l$  e la media locale  $\mu_l$ . Invece  $C_u$  è il coefficiente di variazione del rumore che in questo caso vale  $1/(2\sqrt{2})$ ;

2. a questo punto si ripete il filtraggio usando una finestra  $15 \times 15$  con la seguente formula:

$$y_2(m, n) = y_1(m, n)W_l + \mu_l(1 - W_l)$$

Per garantire che in uscita non ci siano valori negativi, usate un controllo nei due casi per cui quando il rapporto  $C_u^2/C_x^2$  diventa maggiore di 1, il valore  $y(m, n)$  coincide con la media locale  $\mu_l$ . Scrivete quindi una funzione: `function [y1, y2] = adap_filt(x)` che realizza i passi descritti e fornisce le due stime ( $y_1$  e  $y_2$ ). Infine, calcolate il PSNR tra l'immagine originale contenuta nel file lena.raw (256 × 256, uint8) e le due stime e mostrate anche i risultati a video.

**EX. 2** Realizzate l'edge detection dell'immagine test.bmp con l'approccio che vi sembra più appropriato, che sia quindi in grado di individuare i bordi degli oggetti (circolari e non) con il numero minore possibile di falsi allarmi. Mostrate la mappa binaria così ottenuta.

**EX. 3** Un modo per individuare i punti salienti (keypoint) di un'immagine è quello di adottare la seguente strategia:

1. si valuta la derivata nella direzione verticale,  $V(i, j)$ , orizzontale,  $H(i, j)$  e diagonale,  $D_1(i, j)$  e  $D_2(i, j)$ , dell'immagine  $x(i, j)$ :

$$\begin{aligned} V(i, j) &= x(i, j) - x(i - 1, j) & H(i, j) &= x(i, j) - x(i, j - 1) \\ D_1(i, j) &= x(i, j) - x(i - 1, j - 1) & D_2(i, j) &= x(i, j) - x(i - 1, j + 1) \end{aligned}$$

2. Per ognuna di queste quattro immagini si valutano i valori al quadrato  $Q(i, j)$  calcolati tramite finestra scorrevole su blocchi di  $5 \times 5$  pixel. Per esempio facendo riferimento alla derivata verticale:

$$Q_V(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 V^2(i + m, j + n)$$

3. Quindi si determina:  $Q_{min}(i, j) = \min \{Q_V(i, j), Q_H(i, j), Q_{D_1}(i, j), Q_{D_2}(i, j)\}$
4. A questo punto si calcola il valore massimo  $MQ_{min}(i, j)$  tramite finestra scorrevole su blocchi di  $3 \times 3$  pixel.
5. Infine i punti salienti sono quelli per cui:  $SP(i, j) = Q_{min}(i, j) > 500$  AND  $Q_{min}(i, j) = MQ_{min}(i, j)$ .

Applicate l'algoritmo all'immagine tetto.png e mostrate a video la mappa binaria  $SP(i, j)$ .