

**PROVA SCRITTA DI ELABORAZIONE DI SEGNALI MULTIMEDIALI del 3.7.12**  
**(Ingegneria delle Telecomunicazioni)**

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

**EX. 1** L'immagine contenuta nel file lenarumorosa.y (512×512, int16) mostra un disturbo periodico (righe oblique) che la rende sgradevole. Dopo aver analizzato la trasformata di Fourier dell'immagine, progettate un filtro in grado di ridurre nel modo migliore possibile il rumore, mostrate l'immagine risultante e calcolate l'errore quadratico medio tra immagine filtrata e originale (contenuta nel file lena.y, uchar).

**EX. 2** Si vuole implementare il *ratio edge detector*, un rivelatore di bordi particolarmente utile per immagini affette da rumore moltiplicativo. Per ogni pixel si valutano i valori medi in due settori del vicinato e si cerca di capire se sono molto diversi fra loro. Ad esempio, per la maschera  $5 \times 5$  riportata sotto, utile per orli verticali,

$$\begin{bmatrix} A & A & 0 & B & B \\ A & A & 0 & B & B \\ A & A & 0 & B & B \\ A & A & 0 & B & B \\ A & A & 0 & B & B \end{bmatrix}, \quad (1)$$

si calcolano le medie  $V_A$  e  $V_B$  dei pixel contrassegnati con  $A$  e  $B$  rispettivamente, quindi si calcola la statistica

$$R = \max \left\{ \frac{V_A}{V_B}, \frac{V_B}{V_A} \right\},$$

Se  $R$  supera una data soglia  $T$  allora il pixel si classifica come orlo, verticale in questo caso. Scrivete una funzione dal prototipo `function map = ratio_detector(X,K,T)` che accetta in ingresso l'immagine  $X$ , la dimensione (dispari) della finestra  $K$  e la soglia  $T$  e fornisce in uscita la mappa degli orli, sia verticali che orizzontali. Nello script `ex2.m` applicate la funzione all'immagine `target_rumorosa.raw` ( $256 \times 256$ , double), determinate i parametri che vi sembrano migliori e visualizzate la mappa risultante.

**EX. 3** (6 CFU) Per ridurre il tempo di trasmissione dell'immagine  $X$  contenuta in `lena.y` ( $512 \times 512$ , uint8) nel file `ex1.m` effettuate una quantizzazione uniforme a 3 bit nel range  $[0 - 255]$ . Visualizzate il risultato  $Y$  e calcolate l'MSE rispetto all'originale. Per migliorare la qualità si può procedere come segue:

1. dividete l'immagine in blocchetti quadrati  $B$  di lato  $N$ ;
2. calcolate i valori minimo ( $B_{min}$ ) e massimo ( $B_{max}$ ) di ogni blocco;
3. adottate per ogni blocco un quantizzazione uniforme a  $K=2$  bit nel range  $[B_{min}-B_{max}]$ .

Usate il valore di  $N$  che garantisce lo stesso tasso di codifica per le due strategie (nella seconda, contegiate anche 16 bit a blocco per la codifica di  $B_{min}$  e  $B_{max}$ ). Visualizzate il risultato  $Z$  e calcolate l'MSE rispetto all'originale.

Infine, ripetete il secondo metodo scegliendo, fra tutte le coppie di valori  $[N$  (potenza di 2),  $K]$  che garantiscono un MSE inferiore a quello del metodo 1, quella col tasso di codifica più piccolo. Visualizzate il risultato  $W$  e calcolate l'MSE rispetto all'originale e il tasso di codifica.

**EX. 3** (9 CFU) Si consideri l'immagine `house.y` ( $512 \times 512$ , uint8), dopo aver aggiunto rumore additivo gaussiano con varianza 400, nel file `ex3.m` si scriva il codice per il progetto di un filtro di

denoising basato su trasformata wavelet. In particolare, il filtraggio deve realizzare un thresholding di tipo adattativo, quindi i passi da fare sono:

1. effettuare la trasformata wavelet (di Haar) diretta con 4 livelli di decomposizione;
2. soft-thresholding con soglia adattativa per ogni banda dettaglio (la soglia è quella universale  $T = \sigma\sqrt{2 \log K}$ , dove  $K$  è il numero di pixel della sottobanda);
3. effettuare la trasformata wavelet inversa.

Visualizzate il risultato dell'elaborazione e valutate il PSNR tra originale e filtrata.

N.B. Avete a disposizione i file `dwt.m`, `iwt.m` e `iwt2d.m`.