

**PROVA SCRITTA DI ELABORAZIONE DI SEGNALI MULTIMEDIALI del 16.6.14**  
**(Ingegneria delle Telecomunicazioni)**

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

**EX. 1** Data l'immagine a colori `test_color.jpg`, si vuole migliorare l'aspetto dell'immagine realizzando l'equalizzazione dell'istogramma nello spazio HSI. A tale scopo scrivete una funzione con il seguente prototipo: `function y = histeqcolor(x)` in cui effettuate l'equalizzazione dell'istogramma solo sulla componente di intensità I dell'immagine `x`; visualizzate l'immagine originale, quella elaborata e l'istogramma della componente I prima e dopo l'equalizzazione.

Si vuole realizzare poi il filtraggio dell'immagine `y` operando su ognuna delle componenti RGB con un filtro polinomiale con risposta in frequenza:  $H_{n,T}(\mu, \nu) = \max \left[ 1 - \left( \frac{\mu^2 + \nu^2}{T^2} \right)^{n/2}, 0 \right]$ , dove  $n > 0$ . A tale scopo, scrivete una funzione: `function z = pcolor(y,n,T)` e visualizzate l'immagine filtrata per  $n = 1$  e  $T = 2$ .

**EX. 2** Esistono diverse tecniche per cercare di scoprire se un'impronta digitale è autentica o contraffatta. Una delle più semplici opera nel dominio di Fourier, calcolando la frazione di energia contenuta alle medie frequenze, e dichiarando l'immagine autentica se tale frazione supera una certa soglia.

Scrivete una funzione `function EM = elabora(x, r1, r2)` che calcola la DFT-2D,  $X(\mu, \nu)$ , di un'immagine  $x(m, n)$ , valuta l'energia alle medie frequenze come  $E_M = \frac{1}{|\Omega|} \sum_{\Omega} |X(\mu, \nu)|^2$  dove  $\Omega = \{(\mu, \nu) : r_1 \leq \sqrt{\mu^2 + \nu^2} \leq r_2\}$  e  $|\Omega|$  è la cardinalità di  $\Omega$ , cioè il numero di punti che soddisfano questa condizione.

Applicate la funzione alle due immagini `impronta1.tif` e `impronta2.tif`, usando raggio interno  $r_1 = 0.10$  e raggio esterno  $r_2 = 0.25$  ed etichettate come vera quella che fornisce il valore maggiore di  $E_M/E$ , con  $E$  energia dell'immagine.

**EX. 3** Scrivete una funzione `function Y = coder(X)` che comprime l'immagine `X` attraverso le seguenti operazioni. L'immagine è divisa in blocchi  $8 \times 8$ , quindi per ogni blocco:

1. si effettua la DCT;
2. si conserva solo la continua (DC);
3. si effettua la codifica predittiva delle componenti DC, cioè si trasmette `DC_cur - DC_old` (eccetto per il primo blocco);
4. l'errore di predizione è quantizzato uniformemente con passo  $\Delta = 2$ ;

Scrivete quindi una funzione `function Xrec = decoder(Y)` che realizza le operazioni inverse e ricostruisce una versione distorta di `X` a partire da `Y`.

Nello script `ex3.m` usate le funzioni descritte sopra per codificare e decodificare l'immagine `Lena.y` (di dimensioni  $512 \times 512$  e formato `unsigned char`), valutate l'MSE tra immagine originale e decodificata e mostrate il risultato.