

PROVA SCRITTA DI ELABORAZIONE DEI SEGNALI MULTIMEDIALI del 20.07.10
(Ingegneria delle Telecomunicazioni)

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

EX. 1 Si consideri la maschera di Prewitt: $h_1(m, n)=[1, 0, -1; 1, 0, -1; 1, 0, -1]$, lavorando nello spazio RGB,

1. scrivete una funzione `function y = filtra(x,h1)` in cui realizzate il filtraggio dell'immagine `giove.tif` e determinate l'immagine $y(m, n) = |y_1(m, n)| + |y_2(m, n)|$, dove $y_1(m, n)$ è l'immagine filtrata mediante il filtro $h_1(m, n)$, mentre $y_2(m, n)$ è l'immagine filtrata mediante il filtro $h_2(m, n) = h_1'(m, n)$ e visualizzate il risultato;
2. scrivete una funzione `function y = filtrafreq(x,h1)`, che realizza lo stesso filtraggio nel dominio della frequenza e verificate che il risultato ottenuto y è uguale a quello del punto 2, calcolando l'MSE tra le due immagini;
3. scrivete una funzione `function z = enhanc(x,y)`, che realizza l'enhancement dell'immagine a colori x sfruttando l'immagine y e visualizzate l'immagine elaborata z .

EX. 2 Si vuole effettuare una codifica basata su DCT dell'immagine `peppers.y` (512×512 , uint8). A tale scopo, per ogni blocco 8×8 dell'immagine, si calcola la DCT, si vettorizzano i coefficienti, e li si inserisce in una matrice di dimensioni $(512 * 512/64) \times 64$, nelle cui colonne ci sono quindi i coefficienti omologhi di tutti i blocchi.

Calcolate la varianza di ogni coefficiente (quindi di ogni colonna della matrice) e selezionate i 12 coefficienti (a parte quello DC) con varianza maggiore. In fase di codifica, per ogni blocco copiate invariato il coefficiente DC, e quantizzate ad un bit, con il quantizzatore di Laplace (il cui codice è contenuto nel file `quant_laplace.m`), solo i 12 coefficienti selezionati. Valutate l'SNR risultante e mostrate l'immagine codificata.

Si può migliorare la codifica allocando opportunamente i 12 bit dei coefficienti AC. A tale scopo, a partire dalle varianze dei coefficienti, calcolate il numero di bit ottimo secondo la tecnica di Huang-Schultheiss (N.B. eliminate i coefficienti a cui competono risorse negative e ripetete la procedura finché sono tutte positive) e arrotondate infine i valori ottenuti avendo cura che la somma sia 12. Ripetete la codifica con l'allocazione ottima, calcolate SNR e mostrate l'immagine codificata.

EX. 3 Si vuole segmentare l'immagine `yeast.tif`; a tale scopo applicate l'algoritmo k-means su 3 classi e visualizzate la mappa ottenuta. Noterete come in questo modo sia possibile isolare le aree luminose dal background, tuttavia non risulti possibile individuare gli anelli grigio chiaro che circondano i cerchi bianchi. Per questo motivo si vuole segmentare l'immagine usando una soglia variabile basata sulle proprietà statistiche locali dell'immagine stessa. Scrivete la funzione `function mappa = thresholding_locale(x)` in cui:

1. si calcola l'immagine delle deviazioni standard locali, $\sigma_L(m, n)$, usando una finestra 3×3 ;
2. si ottiene la mappa binaria usando una soglia diversa per ogni posizione (m, n) come:

$$g(m, n) = \begin{cases} 1 & x(m, n) > a\sigma_L(m, n) \text{ AND } x(m, n) > b m_G \\ 0 & \text{altrimenti} \end{cases}$$

dove m_G è la media globale, e si pone $a = 30$ e $b = 1.5$.

Si confrontino le mappe ottenute mediante i due approcci. La mappa ottenuta nel secondo caso è binaria, attraverso un'ulteriore operazione riuscite a modificarla in modo tale che vi fornisca tre classi (una per il background, una per gli anelli e l'altra per le regioni a forma di chicco di riso)?