

PROVA SCRITTA DI ELABORAZIONE DEI SEGNALI MULTIMEDIALI del 21.7.08
(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 color balancing dell'immagine `y` nello spazio MCYK per ridurre la quantità di blu e aumentare quella di verde. A tale scopo, scrivete una funzione: `function z = correct(y)` e visualizzate il risultato.

N.B. Il codice necessario per richiamare le funzioni va scritto in un unico script dal nome ex1.m.

EX. 2 Si vuole simulare la compressione di un'immagine con la trasformata Wavelet. Avendo a disposizione le funzioni `fwt2d.m`, `fwt.m`, `iwt2d.m` e `iwt.m`, scrivete uno script `ex2.m` in cui:

1. calcolate la wavelet diretta di `house.y` (dim. 512×512) con un livello di decomposizione;
2. quantizzate separatamente i coefficienti wavelet delle 4 sottobande scrivendo una funzione `function y = quantizeU(x,R)` per la quantizzazione uniforme della banda LL, e usando la funzione `quantizeL.m` per la quantizzazione non uniforme delle bande dettaglio. Il valore del tasso `R` da usare per le 4 bande va calcolato scrivendo il codice che implementa la regola di Huang-Schultheiss:

$$R_i = \bar{R} + \frac{1}{2} \log_2 \frac{\sigma_i^2}{\sigma_{GM}^2}$$

(Ponete a zero R_i qualora fosse negativo e, se non intero, arrotondatelo a quello più vicino);

3. calcolate la wavelet inversa.

Fate un esperimento ponendo $\bar{R} = 2$ bit/pixel e calcolate l'SNR tra immagine originale e compressa e il tasso effettivo utilizzato. Infine, confrontate l'SNR con quello che otterreste assegnando le stesse risorse ad ogni banda.

EX. 3 Si vuole realizzare edge detection dell'immagine 'luna.jpg' mediante i seguenti passi:

1. effettuare lo smoothing con un filtro gaussiano con deviazione standard σ iterativamente per 10 volte, scrivendo una funzione `function y = gaussfilt(x,sigma)`;

2. calcolare il gradiente con le maschere di Roberts: $a_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ $a_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

scrivendo una funzione `function y = gradiente(x)`;

3. generare la mappa dei contorni effettuando un'operazione di thresholding, in cui la soglia è posta pari al 10% del valore massimo e individuando poi i passaggi 0-1 e 1-0, scrivendo una funzione `function y = contorni(x)`;

Realizzate un esperimento in cui visualizzate la mappa dei contorni al variare di $\sigma = 1, 3, 5$.

N.B. Il codice necessario per richiamare le funzioni va scritto in un unico script dal nome ex3.m.