

PROVA SCRITTA DI ELABORAZIONE DEI SEGNALI MULTIMEDIALI del 29.06.10
(Ingegneria delle Telecomunicazioni)

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

EX. 1 La maggior parte delle fotocamere digitali acquisisce una sola informazione di colore (rosso, verde, oppure blu) per ogni pixel, secondo il pattern mostrato nell'immagine Mosaic.bmp, e produce quindi, per ogni canale di colore, una matrice in cui solo alcuni dei valori sono non nulli. Gli altri valori sono poi ottenuti mediante interpolazione bilineare di quelli disponibili.

L'immagine Fiori_mosaic.bmp contiene appunto, nelle tre componenti, le matrici prima dell'interpolazione. Scrivete una funzione `function y = demosa1(x)` in cui estraete le tre componenti, effettuate l'interpolazione, ricomponete l'immagine e confrontate il risultato, sia visivamente che in termini di SNR, con l'immagine vera Fiori256.bmp.

Si può migliorare la qualità attraverso un'interpolazione adattativa. A tale scopo scrivete una nuova funzione `function y = demosa2(x)` in cui per ogni pixel da interpolare considerate i valori dei due pixel più vicini disponibili sulla stessa riga (W ed E) e sulla stessa colonna (N e S) e calcolate le differenze $|W - E|$ e $|N - S|$. Se $|W - E| > 3|N - S|$, usate solo N e S per l'interpolazione, se vale il contrario usate solo W ed E , altrimenti procedete normalmente.

EX. 2 Si vuole effettuare una pseudo-codifica a basso bit-rate dell'immagine Lena.bmp mediante DCT. A tale scopo, dopo aver effettuato la DCT per blocchi dell'immagine, si conservano solo alcuni coefficienti, annullando tutti gli altri, e si ricostruisce l'immagine pseudo-codificata. Confrontate sia visivamente che in termini di SNR le seguenti strategie:

1. blocchi 8×8 e solo il coefficiente DC conservato;
2. blocchi 16×16 e solo i 3 coefficienti a frequenza più bassa conservati (DC + 2 AC);
3. blocchi 32×32 e solo i 15 coefficienti a frequenza più bassa conservati (DC + 14 AC);
4. blocchi 32×32 e solo i 10 coefficienti con modulo più elevato conservati.

EX. 3 Nell'ipotesi di voler rilevare bordi spessi un pixel per individuare i contorni orizzontali, verticali e obliqui (nelle due direzioni) dell'immagine forme.jpg,

1. scrivete una funzione `function y = linedet1(x)` in cui definite opportunamente le maschere e mostrate le 4 mappe binarie dei bordi ottenuti, quindi effettuate l'elaborazione che vi permette di produrre un'unica mappa in cui sono presenti linee di qualsiasi tipo nell'immagine e memorizzate il risultato in `y`;
2. scrivete una funzione `function y = linedet2(x)`, che realizza lo stesso filtraggio nel dominio della frequenza e mostrate la risposta in frequenza dei filtri che riescono a rilevare i bordi nelle diverse direzioni;
3. infine, verificate che il risultato ottenuto al punto 1 è uguale a quello del punto 2, sia visivamente che contando il numero di pixel neri.