

PROVA SCRITTA DI ELABORAZIONE DEI SEGNALE MULTIMEDIALI del 27.7.09
(Ingegneria delle Telecomunicazioni)

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

EX. 1 Si vuole effettuare il filtraggio dell'immagine rumorosa `circuitorumoroso.y` (256×256 , `int16`) affetta da rumore gaussiano additivo. A tale scopo scrivete una funzione `function y = rimuovi1(x,k)` in cui si applica un filtro media aritmetica di dimensioni $k \times k$ sull'immagine `x` e si ottiene l'immagine `y`. Tracciate una curva in cui si mostra l'SNR tra l'immagine originale contenuta nel file `circuito.y` (256×256 , `uint8`) e quella filtrata al variare di $k = 3, 5, 7, 9$.

Allo scopo di migliorare il filtraggio si adotta una strategia di tipo adattativo. Detta σ_n^2 la varianza del rumore sull'intera immagine, μ_l la media locale e σ_l^2 la varianza locale di $x(m, n)$ calcolata su blocchi $k \times k$, bisogna effettuare la seguente trasformazione:

$$y(m, n) = x(m, n) - \frac{\sigma_n^2}{\sigma_l^2} [x(m, n) - \mu_l]$$

in questo modo, se $\sigma_l^2 > \sigma_n^2$ si restituisce un valore prossimo all'originale; invece se le varianze sono simili si effettua lo smoothing. Scrivete una funzione `function y = rimuovi2(x,k)` in cui realizzate la procedura descritta e stimate la varianza del rumore σ_n^2 nel dominio wavelet (usando le funzioni `fwt.m` e `fwt2d.m`) come $\hat{\sigma}_n = \frac{\text{median}(|w_{ij}|)}{0.6475}$ dove w_{ij} sono i coefficienti wavelet della banda HH (al primo livello di decomposizione). Tracciate nuovamente la curva dell'SNR al variare di k e mostrate le due immagini filtrate che nelle due strategie ottengono l'SNR più alto.

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

EX. 2 Si vuole realizzare il watermarking dell'immagine `lena.y` (512×512 , `uint8`) nel dominio trasformato usando la seguente strategia:

1. si effettua la DCT bidimensionale diretta di tutta l'immagine;
2. si conservano i $K(K+1)/2$ coefficienti DCT alle basse frequenze e si modificano i successivi L coefficienti DCT alle medie frequenze secondo la regola: $c(i, j) = c(i, j) + \alpha |c(i, j)| w(i, j)$, dove w è il watermark ed è costituito da valori pseudo-casuali ottenuti con il comando `randn`;
3. si effettua la DCT bidimensionale inversa di tutta l'immagine.

Scrivete uno script dal nome `ex2.m` in cui realizzate i passi descritti e mostrate sia l'immagine originale che quella marcata nell'ipotesi in cui $K = 180$, $L = 16350$ e $\alpha = 0.2$. Infine, quantizzate uniformemente l'immagine marcata usando la dinamica effettiva e 8 livelli di restituzione (posizionati al centro dell'intervallo) e mostrate l'immagine quantizzata.

EX. 3 Il file `test.yuv` contiene 16 fotogrammi di un file video, scrivete una funzione `function yuv = yuv_get(filename,k)` che preleva dal file in formato CIF (288×352) YUV 4:2:0 il fotogramma in posizione k e produce in uscita la matrice 3D YUV delle componenti del fotogramma (usate un'interpolazione lineare per le componenti U e V).

A questo punto modificate la funzione `me.m` e scrivete una nuova funzione `function mvf = yuv_me(cur, ref)` che preleva i fotogrammi in posizione `cur` e `ref`, ed effettua la stima del movimento, producendo il campo di vettori `mvf`. Il criterio di stima è la SAD (somma delle differenze in valore assoluto) calcolata su ogni componente e poi sommata, l'area di ricerca è di 8 pixel, la strategia di ricerca è la full-search e le dimensioni del blocco sono 16×16 . Utilizzate la funzione `displayMVF` per visualizzare i vettori di movimento per le frame 6 e 7 del file video.

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