

PROVA INTRACORSO DI ELABORAZIONE DI SEGNALI MULTIMEDIALI del 29.04.16
(Ingegneria delle Telecomunicazioni)
NON è consentito l'uso di materiale didattico e appunti propri.

EX. 1 Per ridurre il rumore in un'immagine si vuole realizzare un filtraggio adattativo che opera mediante finestra scorrevole. I passi da seguire per realizzare l'algoritmo sono i seguenti:

1. calcolate l'immagine delle varianze locali su una finestra di dimensioni $k \times k$;
2. definite una soglia T superata dal 30% delle varianze;
3. filtrate ogni blocco con un filtro media aritmetica di dimensioni $k \times k$ se la varianza locale è inferiore a T , altrimenti utilizzate un filtro di dimensioni $k - 2 \times k - 2$.

Scrivete una funzione `function y = adapt_filter(x,k)` che realizza tale filtraggio e applicatelo all'immagine `cicogna.jpg`, cui avete aggiunto rumore gaussiano bianco con $\sigma = 25$.

Calcolate il PSNR tra immagine originale e filtrata per $k = 3, 5, 7, 9$ e rappresentatelo graficamente confrontandolo con la soluzione non adattativa (filtro media aritmetica di dimensioni $k \times k$). Infine, visualizzate e confrontate l'immagine filtrata con $k = 5$ per le due strategie.

EX. 2 Si vuole scoprire quale macchina fotografica, fra due disponibili, ha scattato una data fotografia (identificazione di sorgente). A tale scopo, dall'immagine sotto test x^{large} , ritagliate la sezione in alto a sinistra, x , di $M \times N$ pixel (usate $M=N=512$) e calcolate su di essa quattro *feature* dell'immagine. Per ottenere le prime due feature, estraete da x il bit-plane B_2 associato al secondo bit meno significativo. Calcolate quindi

$$f_1 = \frac{\sum_{i=2}^M \sum_{j=1}^N |B_2(i-1, j) - B_2(i, j)|}{(M-1)(N)}, \quad f_2 = \frac{\sum_{i=1}^M \sum_{j=2}^N |B_2(i, j-1) - B_2(i, j)|}{(N-1)(M)}$$

Applicate poi un filtro gaussiano 3×3 con $\sigma = 0.5$ (usate il comando `fspecial`) all'immagine originale x per ottenere la sua versione filtrata y . Di entrambe calcolate la 2D-DFT, X e Y , e ricavate infine

$$f_3 = 10 \log_{10}(\text{MSE}(|X|, |Y|)), \quad f_4 = 10 \log_{10}[\text{MSE}(\angle X, \angle Y)]$$

Scrivete una funzione `function [camera] = test(x)` che estrae le quattro feature per l'immagine, calcola l'indicatore $f = 0.7 f_1 + 1.5 f_2 + 0.01 f_3 + 0.001 f_4$ e restituisce `camera=1` se $f > 1.5$ e `camera=2` altrimenti. Effettuare quindi il test alle immagini `1.png`, `2.png`, `3.png` e `4.png` e stabilite a quale fotocamera appartengono.

EX. 3 L'immagine y_0 (file `lenaB.png`) è un ritaglio ruotato dell'immagine originale x (file `lenaA.png`). Si vuole co-registrare le due immagini, cioè allineare l'immagine originale con quella ruotata senza conoscere l'angolo di rotazione. A tal fine, per tutti gli angoli θ da 0 a 360 gradi, con passo 5 gradi, effettuate le seguenti operazioni:

1. calcolate l'immagine y , versione ruotata di θ gradi di y_0 ;
2. calcolate R_{yx} , mutua correlazione tra y e x ;
(N.B. operate nel dominio della frequenza realizzando un opportuno zero-padding)
3. calcolate il valore massimo, $M(\theta)$, di R_{yx} .

Mostrate il grafico di M in funzione di θ ; l'angolo cercato θ_{opt} è quello per cui M è massimo. Ruotate l'immagine y_0 di θ_{opt} e verificate visivamente l'avvenuta co-registrazione.