

**PROVA SCRITTA DI ELABORAZIONE DEI SEGNALI MULTIMEDIALI del 15.6.09**  
(Ingegneria delle Telecomunicazioni)

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

**EX. 1** Data l'immagine `lenarumorosa.y` (di dimensioni  $512 \times 512$ , `int16`) corrotta da rumore sinusoidale, si vuole rimuovere il rumore mediante un opportuno filtraggio.

1. Nello script `ex1.m` calcolate e visualizzate opportunamente la trasformata di Fourier dell'immagine e cercate di localizzare visivamente gli impulsi presenti (una coppia di impulsi corrisponde ad una sinusoidale);
2. scrivete una funzione `function y = rimuovi(x)` in cui progettate un filtro nel dominio di Fourier capace di eliminare il rumore sinusoidale e visualizzatene la risposta in frequenza; mostrate inoltre l'immagine filtrata e la sua trasformata di Fourier e valutate l'MSE con l'originale memorizzata nel file `lena.y` (di dimensioni  $512 \times 512$ , `uint8`).

**EX. 2** Si consideri l'immagine `lincoln.jpg`, che rappresenta una fotografia di Abraham Lincoln, in cui è presente un vistoso graffio. Si vuole realizzare un algoritmo che effettui l'enhancement *locale* dell'immagine, riguardante cioè la sola parte danneggiata, individuata dalla maschera binaria `lincmask.y` (di dimensioni  $430 \times 370$ , `uint8`). A tale scopo, si scriva il codice nello script `ex2.m` per elaborare in più passaggi l'immagine, dove in ogni passaggio:

1. si considerano tutti i pixel da ritoccare che confinano con almeno un pixel corretto (quindi solo quelli appartenenti al bordo della maschera), e per ognuno di essi:
2. si calcola il valore mediano dei soli pixel corretti presenti in una finestra  $5 \times 5$  centrata sul pixel in esame;
3. si sostituisce il pixel da ritoccare col valore mediano e si aggiorna la maschera.

Il programma termina quando tutti i pixel da ritoccare sono stati elaborati. Mostrate l'immagine originale e quella elaborata.

**EX. 3** Si vuole segmentare mediante thresholding l'immagine `rice.y` (di dimensioni  $256 \times 256$ , `uint8`). Per determinare la soglia ottima, scrivete la funzione di prototipo `function t = T_opt(x)` che realizza il clustering K-means di `x` con  $K=2$  e restituisce in `t` la semisomma dei due template.

In realtà il thresholding globale dà un risultato poco soddisfacente a causa dell'illuminazione non uniforme dell'immagine. Nello script `ex3.m` allora si effettui il thresholding *adattativo* dell'immagine, usando una diversa soglia per ogni gruppo disgiunto di  $L$  righe. Si usino i valori di  $L = [1, 2, 4, \dots, 256]$  (potenze di 2), e si valuti l'accuratezza della segmentazione contando il numero di pixel in cui l'immagine segmentata differisce da quella ideale contenuta in `rice_bw.y` (di dimensioni  $256 \times 256$ , `uint8`). Infine si mostrino l'immagine originale, la segmentazione globale, e la migliore segmentazione locale ottenuta.