

PROVA INTRACORSO DI ELABORAZIONE DI SEGNALI MULTIMEDIALI del 24.04.18
(Ingegneria delle Telecomunicazioni)
NON è consentito l'uso di materiale didattico e appunti propri.

EX. 1 Allo scopo di ridurre il contenuto di rumore nell'immagine, si vuole implementare un filtro direzionale che opera mediante finestra scorrevole di dimensioni 9×9 . Scrivete una funzione dal prototipo `function y = filtro_dir(x)`, e per ogni pixel dell'immagine realizzate le seguenti elaborazioni:

1. considerate le regioni laterali che includono il pixel centrale nelle otto direzioni (in figura ne sono mostrate due come esempio);
2. sostituite al pixel centrale la media della finestra che presenta varianza più piccola.

Nello script `ex1.m` effettuate un esperimento in cui aggiungete rumore gaussiano con deviazione standard 30 all'immagine `test.png` e filtratela con la strategia descritta sopra.

Dato che l'immagine è molto rumorosa, questo approccio potrebbe non funzionare bene. Verificare che il funzionamento migliora se le varianze si calcolano non sull'immagine rumorosa, ma su quella originale, anche se il filtraggio avviene sempre su quella rumorosa.

Infine, considerato che l'immagine originale non è disponibile, generate un'immagine pilota, `P`, filtrando con un filtro media aritmetica 3×3 quella rumorosa e ripetere la prova precedente usando l'immagine pilota per calcolare le varianze.

Per tutte e tre le prove, valutare l'MSE rispetto all'immagine originale.

EX. 2 L'immagine contenuta nel file `rumorosa.y` (380×572 , formato: single) presenta un disturbo periodico (righe verticali) che la rende sgradevole. Dopo aver analizzato la trasformata di Fourier dell'immagine, progettate un filtro nel dominio della frequenza in grado di ridurre nel modo migliore possibile il rumore (scrivete una funzione dal prototipo `function y = filtra_freq(x)`). Quindi, mostrate l'immagine risultante e calcolate il PSNR tra l'immagine filtrata e l'originale.

EX. 3 Si vogliono discriminare immagini di iridi vere da false, ottenute stampando l'acquisizione su carta fotografica di elevata qualità (usate il comando `rgb2gray` per trasformare l'immagine in scala di grigi). L'algoritmo da implementare per effettuare la classificazione dell'immagine $z(m, n)$ è descritto di seguito. Per ogni pixel dell'immagine:

1. calcolate $x(m, n) = 2z(m-1, n) - z(m-1, n-1) + 2z(m, n+1) - z(m+1, n+1) + 2z(m+1, n) - z(m+1, n-1) + 2z(m, n-1) - z(m-1, n+1) - 4z(m, n)$;
2. realizzate poi la seguente elaborazione: $y = \sum_{i=0}^7 u(x_i - x) 2^i$ dove x_i è l' i -esimo pixel del vicinato di x (vedi figura), mentre $u(\alpha) = 1$ quando $\alpha \geq 0$ e 0 altrimenti.

$$\begin{bmatrix} x_0 & x_1 & x_2 \\ x_7 & x & x_3 \\ x_6 & x_5 & x_4 \end{bmatrix}$$

Infine, valutate l'istogramma (non normalizzato) dell'immagine elaborata al punto precedente e la sua deviazione standard e classificate un'immagine come vera se la deviazione standard supera il valore 495. Applicare l'algoritmo alle due immagini `I1.png` e `I2.png` e fornire per ognuna il valore 1 se è vera, 0 se è falsa.