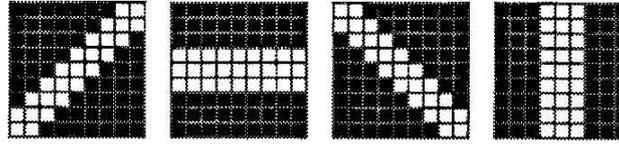


**PROVA SCRITTA DI ELABORAZIONE DEI SEGNALI MULTIMEDIALI del 18.7.13**  
**(Ingegneria delle Telecomunicazioni)**

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

**EX. 1** Allo scopo di migliorare il filtraggio di immagini rumorose si vogliono utilizzare filtri direzionali. In particolare, considerate le seguenti maschere  $9 \times 9$  per il filtraggio:



A questo punto aggiungete rumore gaussiano con deviazione standard 25 all'immagine zebre.y ( $321 \times 481$ , uint8). Quindi realizzate i seguenti passi per filtrare ogni pixel dell'immagine: calcolate la varianza locale usando le 4 maschere mostrate in figura considerando solo i pixel selezionati (zona bianca); quindi usate per il filtraggio del pixel la maschera corrispondente alla varianza minima con i pesi tutti uguali. Valutate il PSNR tra l'originale e la filtrata.

**EX. 2** Data l'immagine contenuta nella variabile  $\mathbf{z}$  del file quadrati.mat (usare il comando load per caricare i dati) si vuole applicare l'algoritmo di Canny per rivelare il bordo verticale e orizzontale presente. Settate sperimentalmente i valori della deviazione standard e della soglia richiesti dall'algoritmo allo scopo di ottenere una buona mappa di segmentazione e visualizzatela.

A questo punto per valutare la bontà della tecnica si può usare come indicatore di qualità la figura di merito (Figure of Merit, FOM) che richiede la conoscenza della mappa di segmentazione ideale (contenuta nella variabile  $\mathbf{I}$  del file mappa.mat). La figura di merito si definisce come:

$$\text{FOM} = \frac{1}{\max(N_d, N_r)} \sum_{k=1}^{N_d} \frac{1}{1 + \gamma d_k^2}$$

dove  $N_d$  è il numero di pixel riconosciuti come bordo dall'algoritmo di Canny,  $N_r$  è invece quello trovato nella mappa ideale,  $\gamma$  è una costante da porre pari a  $1/9$ . Infine  $d_i$  è la distanza euclidea tra la posizione dell' $i$ -esimo pixel rivelato come bordo nella mappa reale (R) e il pixel che rappresenta il bordo più vicino nella mappa ideale (I), cioè  $d_k = \sqrt{(m_{k,R} - m_{k,I})^2 + (n_{k,R} - n_{k,I})^2}$ . Scrivete una funzione dal prototipo `FOM_index = FOM(R,I)` per valutare FOM nel vostro caso.

**EX. 3** Si consideri l'immagine contraffatta auto.jpg, in cui è stata fatta un'operazione copy-and-paste inserendo l'automobile rossa in basso a destra. Dato che l'immagine dopo essere stata manipolata viene nuovamente salvata in formato JPEG, un modo automatico per rivelare la contraffazione è proprio scoprendo che l'immagine ha subito un processo di doppia compressione. In particolare, data un'immagine a colori  $x(m, n, k)$  si realizzano i seguenti passi:

1. si generano copie dell'immagine a diversa qualità  $x_Q(m, n, k)$  ( $Q = 1 : 10 : 100$ );
2. per ogni livello di qualità  $Q$  si valuta  $d_Q(m, n, k) = [x(m, n, k) - x_Q(m, n, k)]^2$ ;
3. si esegue un filtro media aritmetica lungo la terza dimensione (riducendo così il numero di bande da 3 a 1) seguito da un filtro media aritmetica lungo la direzione spaziale con finestra  $16 \times 16$  e si ottiene l'immagine  $y(m, n)$ .

In corrispondenza del fattore di qualità  $Q$  con cui è stata salvata la foto dopo la modifica, il risultato dell'elaborazione  $y(m, n)$  presenta nella regione alterata valori molto scuri (minimi detti *JPEG Ghosts*). Individuate il valore  $Q$  che fornisce tali minimi, infine applicate un'operazione di thresholding con soglia opportuna allo scopo di produrre una mappa binaria in cui il valore 1 è associato ai soli pixel contraffatti.