**Elaborazione di immagine e video - 02FEQPC**

A.A. 2017/2018

# Esercitazione di laboratorio n. 1

**Obiettivi dell’esercitazione:**

* Applicazione di tecniche basate sull’istogramma dei livelli di grigio
* Riduzione del rumore tramite filtro a media e filtro mediano in presenza di diversi tipi di rumore
* Applicazione del filtraggio inverso tramite Wiener filter

1. **Aumento del contrasto e equalizzazione**

*Istogramma dei livelli di grigio*

**I**n Matlab, l’istogramma dei livelli di grigio dell’immagine I (matrice di livelli di grigio) può essere visualizzato con la funzione imhist(I);

*Image enhancement*

In Matlab la trasformazione dei livelli di grigio corrispondente ad un image enhancement lineare si può ottenere con la funzione Y = imadjust(I,[LOW\_IN; HIGH\_IN],[LOW\_OUT; HIGH\_OUT])[[1]](#footnote-1), dove I è l’immagine in ingresso (matrice di livelli di grigio) e Y è l’immagine in uscita. La funzione trasforma i livelli di grigio compresi tra LOW\_IN; HIGH\_IN in livelli di grigio compresi tra LOW\_OUT; HIGH\_OUT. I livelli di grigio minori di LOW\_IN vengono mappati (clipping) su LOW\_OUT. I livelli di grigio maggiori di HIGH\_IN vengono mappati su HIGH\_OUT. Per convenzione, i livelli di grigio sono sempre indicati tra 0 (nero) e 1 (bianco), indipendentemente dal formato dell’immagine in ingresso. Solitamente, i livelli LOW\_IN; HIGH\_IN sono scelti in modo tale che una percentuale predefinita dei pixel dell’immagine abbiano rispettivamente livello di grigio minore di LOW\_IN oppure maggiore di HIGH\_IN. I livelli LOW\_OUT; HIGH\_OUT corrispondono di default al minimo e massimo livello di grigio, ovvero 0 e 1. I livelli LOW\_IN; HIGH\_IN possono essere calcolati usando la funzione LOW\_HIGH = stretchlim(I, [LOW\_FRACT HIGH\_FRACT]). La funziona ritorna nel vettore LOW\_HIGH una coppia di valori tali che una frazione LOW\_FRACT dei pixel ha valore minore di LOW\_HIGH(1) mentre una frazione HIGH\_FRACT dei pixel ha valore minore di LOW\_HIGH(2). Per esempio, se si vuole un clipping dell’1% dei pixel sia in alto che in basso, si può usare:

LOW\_HIGH = stretchlim(I, [0.01 0.99]);

Y = imadjust(I,LOW\_HIGH);

*Equalizzazione*

In Matlab, l’equalizzazione dei livelli di grigio si può ottenere con la funzione Y = histeq(I,N). La trasformazione produce un’immagine con N livelli di grigio, dove la distribuzione dei livelli di grigio è approssimativamente uniforme.

*Esperimenti*

1. Si consideri l’immagine Goldhill512.tif (si utilizzi la funzione Matlab imread()per aprire l’immagine, si può utilizzare la funzione imshow() per visualizzare l’immagine. Si utilizzi il comando figure per aprire una nuova figura se si vogliono visualizzare più immagini). Si applichi un contrast enhancement lineare in modo tale che l’x% dei livelli di grigio in alto e in basso (totale 2x%) venga clippato a 255 e 0. Si provi con x = 0,1,5 e 10. Si visualizzi il risultato.
2. A partire dall’immagine Goldhill512.tif, si ottenga un’immagine equalizzata con 256 livelli di grigio. Si visualizzi il risultato. Si confronti l’istogramma dell’immagine ottenuta mediante equalizzazione con gli istogrammi delle immagini ottenute al passo precedente.
3. Si consideri l’immagine a colori Baboon.tif. Una volta aperta l’immagine a colori con Im = imread(nomefile), si ottengano le componenti R,G,B, come R = Im(:,:,1), G = Im(:,:,2), B = Im(:,:,3). Si applichi l’equalizzazione separatamente alle tre componenti. Si ricompongano le tre componenti equalizzate in una singola immagine equalizzata (si può usare il comando Imeq = cat(3,Req,Geq,Beq) ). Si visualizzi il risultato.
4. Si trasformi l’immagine a colori nello spazio HSI con HSI = rgb2hsi(Im). Si ricavino le componenti H,S,I come H = HSI(:,:,1), S = HSI(:,:,2), I = HSI(:,:,3)). Si equalizzi la sola componente I. Si ricompongano le tre componenti H,S e I equalizzata e si ritorni allo spazio RGB usando la funzione hsi2rgb(). Si visualizzi il risultato.

*Problema*

Si consideri l’immagine challenge01.tif. Si applichi all’immagine una (o una combinazione) delle tecniche viste in precedenza per rendere visibili le scritte presenti sull’immagine.

1. **Riduzione del rumore e filtraggio inverso**

*Filtraggio lineare*

Per applicare un filtro lineare all’immagine I si può usare la funzione Y = imfilter(I,h,options,..), dove h contiene la maschera del filtro. Si consiglia di usare l’opzione ‘symmetric’, che comporta un’estensione simmetrica ai bordi. (NOTA: la funzione imfilter applica di default una correlazione tra maschera e immagine, ovvero la maschera non è ribaltata come avverrebbe con una convoluzione in due dimensioni. Prestare attenzione se si applicano filtri non simmetrici. Se si vuole una convoluzione 2D, usare il parametro opzionale ‘conv’.)

*Filtraggio mediano*

La funzione Y = medfilt2(I,[M N],option) applica un filtraggio mediano su una finestra di MxN pixel.si consiglia di usare l’opzione ‘symmetric’ per l’estensione ai bordi.

*Filtro di Wiener*

Il filtro di Wiener può essere applicato usando la funzione Y = deconvwnr(I,PSF,NSR). Il parametro PSF indica la risposta impulsiva del filtro che si vuole invertire. Il parametro NSR specifica il rapporto rumore/segnale per ogni frequenza spaziale. Se è uno scalare, lo stesso NSR è applicato per ogni frequenza spaziale, altrimenti deve avere le stesse dimensioni dell’immagine I.

*Simulazione di un pattern di rumore*

Una matrice contenente MxN valori di rumore a media nulla può essere ottenuta utilizzando una delle funzioniNoise = <distribution>\_noise(params,M,N), dove <distribution> può essere Gaussian, uniform, Laplacian, o impulsive. Si usi help <nome-funzione> per verificare quali parametri passare alla funzione a seconda della distribuzione. Una volta generato un pattern di rumore con le stesse dimensioni dell’immagine di test, un’immagine affetta da rumore additivo può essere simulata come I\_noisy = I + Noise.

*Misura rapporto segnale rumore (SNR)*

L’effetto di un filtraggio per ridurre il rumore può essere misurato per mezzo del rapporto tra la potenza dell’immagine originale e la potenza del rumore residuo dopo il filtraggio, calcolabile come SNR = 10log10(varianza immagine originale / varianza rumore residuo) . Data I immagine originale e Y immagine dopo il filtraggio, il rumore residuo si calcola come Res = Y – I. Se non applico nessun filtro Y = I\_noisy e Res = N. La varianza di un’immagine I (matrice) si può ottenere vettorizzando la matrice, ovvero come var(I(:)).

*Esperimenti*

1. Si considerino le immagini lenna512.mat e barbara.mat. Le immagini si possono caricare in Matlab tramite il comando load(‘<nome\_immagine>.mat’), senza indicare variabili in uscita, e saranno caricate nel workspace in variabili con lo stesso nome del file. Dato che le immagini sono in formato floating point ma hanno valori compresi tra 0 e 255, per visualizzare correttamente le immagini si può usare il comando imshow(I, [0 255]) . Per ogni immagine, si generino 4 immagini rumorose aggiungendo rispettivamente un rumore di tipo Gaussiano, uniforme, Laplaciano e impulsivo. Si consideri una deviazione standard di 12 per il rumore Gaussiano e Laplaciano, una ampiezza di 21 per il rumore uniforme (distribuzione uniforme tra -21 e 21) e due livelli pari a -120 e 120 con probabilità 0.005 ciascuno per il rumore impulsivo. Si visualizzino le immagini rumorose e si valuti il rapporto segnale rumore rispetto all’immagine originale per ciascuna immagine.
2. Si applichi un filtro a media 3x3 su ogni immagine rumorosa, si visualizzino le immagini filtrate e si valuti il SNR dopo il filtraggio.
3. Si applichi un filtro mediano 3x3 su ogni immagine rumorosa, si visualizzino le immagini filtrate e si valuti il SNR dopo il filtraggio.
4. Si consideri l’immagine boat.mat. Si generi un’immagine con motion blur corrispondente a un movimento di 31 pixel con un angolo di 17 gradi. La risposta impulsiva del filtro che modella il motion blur si può ottenere con la funzione PSF = fspecial(...) usando il parametro ‘motion’. Si applichi il filtro con la funzione imfilter() usando i parametri ‘conv’ e ‘circular’.
5. Si applichi un filtraggio inverso utilizzando la funzione Matlab deconvwnr() con parametro NSR pari a 0, corrispondente a inversione perfetta del filtro. Si valuti il SNR rispetto all’immagine originale.
6. Si aggiunga all’immagine sfocata del punto 2.d un rumore Gaussiano con deviazione standard pari a 2. Si applichi ora il filtro di Wiener provando queste scelte di parameteri:
   1. NSR = 0;
   2. NSR = varianza rumore / varianza immagine originale;
   3. NSR = varianza rumore ./ spettro immagine originale. Lo spettro di un’immagine si può stimare dalla sua DFT nel seguente modo: spettro = abs(fft2(I)).^2/numel(I). Il valore di NSR per ogni frequenza si può ottenere utilizzando l’operatore di divisione element-wise: NSR = varianza\_rumore ./ spettro. In questo caso NSR ha le stesse dimensioni dell’immagine;
   4. NSR = K, dove K varia tra 0.001 e 0.019. Si considerino 10 passi con incrementi lineari (0.001, 0.003, 0.005,…).

Per ciascuna prova, si visualizzino le immagini ottenute e si valuti il SNR rispetto all’immagine originale.

*Problema*

Si consideri l’immagine challenge02.mat. L’immagine è stata filtrata con un filtro Gaussiano di dimensioni 7x7 ed è presente un rumore additivo. Si applichi all’immagine una (o una combinazione) delle tecniche viste in precedenza in modo da ottenere un’immagine il più simile possibile a lenna512.

**Elaborazione di immagine e video - 02FEQPC**

A.A. 2017/2018

Componenti del gruppo:

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Esercitazione di laboratorio n. 1: SCHEDA DI VALUTAZIONE

# Quesiti.

1. **Aumento del contrasto e equalizzazione**

*Esperimenti*

* Si riportino i livelli di grigio (0,…,255) utilizzati per definire il contrast enhancement nei vari casi:

|  |  |  |
| --- | --- | --- |
| Clipping | LOW\_IN | HIGH\_IN |
| 0% |  |  |
| 1% |  |  |
| 5% |  |  |
| 10% |  |  |

* Si descrivano le differenze tra le immagini modificate tramite constrast enhancement e l’immagine equalizzata. In particolare, si considerino le differenze tra i rispettivi istogrammi.
* Si descrivano le differenze tra le immagini ottenute ai passi 1.c e 1.d. e si descrivano i motivi di tali differenze. In particolare, si confrontino gli istogrammi delle componenti H,S e I delle immagini equalizzate secondo le due procedure e si commentino le differenze.

*Problema*

Si descriva la soluzione adottata e la si giustifichi. Considerando le tecniche viste durante il corso, si potrebbe trovare una soluzione alternativa rispetto alla modifica dell’istogramma?

1. **Riduzione del rumore e filtraggio inverso**

*Esperimenti*

* Si riportino i valori di SNR ottenuti nei vari casi ai punti 2.b e 2.c:

Lenna512

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rumorosa | Media | Mediano |
| Gaussiano |  |  |  |
| uniforme |  |  |  |
| Laplaciano |  |  |  |
| impulsivo |  |  |  |

Barbara

|  |  |  |  |
| --- | --- | --- | --- |
|  | Rumorosa | Media | Mediano |
| Gaussiano |  |  |  |
| uniforme |  |  |  |
| Laplaciano |  |  |  |
| impulsivo |  |  |  |

* Si commentino le differenze di prestazioni dei vari filtri in funzione della statistica del rumore e dell’immagine.
* Si riportino i valori di SNR ottenuti nei casi 2.e a 2.f. (a,b,c)

|  |  |  |  |
| --- | --- | --- | --- |
| 2.e | 2.f-a | 2.f-b | 2.f-c |
|  |  |  |  |

* Qual è il caso migliore? Quale il peggiore? Si trovi una spiegazione al comportamento nei vari casi.
* Si riportino i valori di SNR per i diversi valori di K al punto 2.f-d

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.001 | 0.003 | 0.005 | 0.007 | 0.009 | 0.011 | 0.013 | 0.015 | 0.017 | 0.019 |
|  |  |  |  |  |  |  |  |  |  |

* Si descriva come cambia il comportamento del filtro al variare di K. Si confronti il valore di SNR nel caso migliore con i valori di SNR ottenuti nei casi precedenti e si commentino le differenze.

*Problema*

Si descriva la soluzione adottata e la si giustifichi.

1. Per ottenere una guida sull’uso delle funzioni si utilizzi il comando help <nome\_funzione> (guida breve) oppure il comando doc <nome\_funzione> (guida più dettagliata) [↑](#footnote-ref-1)