

Esercitazione di Laboratorio 19.4.2012 Elaborazione dell'Immagine

Nota. Il link di riferimento con i file comuni è la directory <http://www.mat.unimi.it/users/naldi/elabimm> (script file MATLAB, immagini di prova, documenti relativi alle attività di laboratorio si possono trovare in questa directory per tutta la durata del corso).

1 Primi passi

Utilizzare i comandi di base di lettura/scrittura visualizzazione delle immagini in MATLAB: `imread`, `imwrite`, `image`, `imagesc`.

Proviamo a “caricare” un’immagine:

```
I=imread('cameraman.tif');
```

Se la visualizziamo con `image(I)` cosa succede? Perché?

Guardare l’`help` per `colormap`. Provare a memorizzare l’attuale mappa dei colori `C=colormap`; che dimensioni ha?

Visualizzare l’istogramma (funzione MATLAB `hist`) dell’immagine: attenzione al numero di `bins` e a ricondurre la matrice ad un vettore.

Siano $h(i)$ i valori dell’istogramma dell’immagine, provare a trasformare l’immagine tramite equalizzazione dell’istogramma stesso, il valore equalizzato $y(x)$ del pixel con valore attuale x è definito come (qui 256 è il numero di livelli di grigio possibili),

$$y(x) = 255 \frac{\sum_{i=1}^{x+1} h(i)}{\sum_{i=1}^{256} h(i)}.$$

Attenzione: gli indici nelle sommatorie riportate sopra partono da 1 perché i vettori non hanno l’indice $i = 0$.

Dall’istogramma decidere una soglia e “binarizzare” l’immagine.

2 Esempi di rumore

Proviamo a verificare l’effetto di alcuni tipi di rumore sulla qualità di una immagine. Carichiamo e visualizziamo ancora l’immagine del cameraman

```
>> I=imread('cameraman.tif');  
>> figure,image(I),colormap gray(256);
```

utilizzare la funzione `rumori.m` per introdurre (artificialmente) del rumore e verificarne l’effetto sull’immagine. Le varie opzioni si possono vedere con `help rumori` (prima occorre cambiare il tipo di dato, per esempio `x=double(I)`). Alcune prove:

- Somma di rumore Gaussiano

```
>> y=rumori(x,'ag',12,.5);
>> figure,imagesc(y),colormap gray
```

- Somma di rumore distribuito uniformemente

```
>> y=rumori(x,'au',40,.5);
>> figure,imagesc(y),colormap gray
```

- Rumore *salt and pepper*

```
>> y=rumori(x,'sp',.1);
>> figure,imagesc(y),colormap gray
```

Esercizio. Completare la funzione `rumore` in modo tale che possa operare anche con l'aggiunta del rumore Gaussiano e uniforme moltiplicativo. Provare una delle opzioni e salvare l'immagine tramite la funzione `imwrite` (attenzione l'immagine da scrivere su file esterno deve essere in formato `uint8`).

3 Primi operatori locali

Esercizio.

Implementare il filtro *media* nel caso di segnale unidimensionale e bidimensionale (8-intorni) (sfruttare la funzione MATLAB `mean`). Utilizzare il filtro nel caso di rumore salt and pepper.

Provare varie condizioni al bordo (periodicità, estensione a zero dell'immagine, riflessione).

Esercizio.

Implementare il filtro *mediana* nel caso di segnale unidimensionale e bidimensionale (8-intorni) (sfruttare la funzione MATLAB `median`). Utilizzare il filtro nel caso di rumore salt and pepper.

Provare varie condizioni al bordo (periodicità, estensione a zero dell'immagine, riflessione).

Esercizio.

Estrarre una linea dall'immagine, per esempio `l=x(120,:)`; e rappresentarla graficamente. Implementare una funzione che valuti, attraverso un opportuno rapporto incrementale, la derivata discreta. Valutare il modulo della derivata discreta e rappresentare la posizione dei pixel a cui corrisponde un valore del modulo maggiore di una prefissata tolleranza `tolE`. Utilizzare condizioni al bordo periodiche oppure lasciare la possibilità di scelta di differenti condizioni al bordo.

Esercizio.

Il segnale appare “rumoroso”, far precedere l’operazione di derivazione con una operazione di regolarizzazione operando medie locali, ogni pixel p_i è sostituito dalla media (operazione locale)

$$p_i \rightarrow \frac{1}{|B(i)|} \sum_{j \in B(i)} p_j,$$

dove $B(i)$ è un intorno di p_i , $|B(i)|$ indica la cardinalità dello stesso intorno. Cosa succede iterando il procedimento di media locale?