

# Esperienze di programmazione

## Lezione 4

Gianna Del Corso <delcorso@di.unipi.it>

18 Marzo 2015

### 1 Immagini

Abbiamo già visto la scorsa volta che le immagini in MATLAB sono rappresentate mediante una matrice  $A$  di interi di dimensione  $p \times q$  associando all'elemento  $(i, j)$  di  $A$  un colore opportuno. Più precisamente se  $a_{ij} = k$  allora il colore del pixel  $(i, j)$  è quello definito dalla  $k$ -esima riga della matrice detta **colormap**. La colormap è una matrice  $nc \times 3$  con elemento tra 0 e 1 dove  $nc$  è il numero di colori che si vogliono usare e ciascuna riga rappresenta un colore attraverso la terna (r, g, b) che definisce la quantità di rosso, verde e blu.

Per creare una mappa di toni di grigio basta assegnare uguali quantità di rosso, verde e blu a ciascun colore.

```
>> map=[0:1/63:1]'*[1 1 1]
```

Cosa fa il comando precedente? Per cambiare palette e utilizzare i toni di grigio di `map` si scrive `colormap(map)`. Questo tipo di immagine si chiama immagine indicizzata.

Si posso anche avere “intensity image”. In questo caso non abbiamo associata una colormap ed i valori di  $A$  rappresentano intensità all'interno di un determinato range. L'immagine è rappresentata solo dalla matrice ed ogni elemento corrisponde ad un pixel. La matrice può essere di classe `double`, `uint8`, `unit16`. Al momento della visualizzazione, se uso `imagesc(A)`, viene utilizzata la colormap di default o quella corrente e associato il primo colore al pixel con intensità di colore inferiore.

Un altro formato è quello RGB anche detto “truecolor”. In questo caso l'immagine è una matrice  $m \times n \times 3$ . Ad ogni “slice” è associato l'intensità del colore rosso, verde e blu rispettivamente. L'intensità dei colori è memorizzata su 8 bit (quindi ogni pixel necessita di 24 bit).

Il comando `A=imread(FILENAME,FMT)` legge l'immagine specificata dal nome `FILENAME` nel formato `FMT`. I formati più conosciuti sono tutti supportati. Si provi a caricare un'immagine e a vedere se è un'immagine truecolor o indexed.

## 2 Manipolazioni geometriche di pixel

### 2.1 Istogramma dei livelli di grigio

L'istogramma dei livelli di grigio di un'immagine è una funzione che associa a ciascun livello il numero di pixel dell'immagine aventi quel livello di grigio. L'istogramma misura quindi la frequenza di occorrenza dei livelli di grigio dell'immagine.

*Esercizio 1.* Si prenda un'immagine con 255 toni di grigio come ad esempio l'immagine lenna. Si scriva una funzione `function h=istogramma(I)` che presa un'immagine `I` restituisce il suo istogramma calcolato nel seguente modo: Si definisce un vettore `h` di dimensione pari al numero dei livelli di grigio dell'immagine. Ciascun elemento del vettore rappresenta il contatore dei pixel aventi livello di grigio uguale all'indice dell'elemento. Quindi si scandisce l'immagine e si incrementa l'elemento del vettore avente indice pari al livello di grigio del pixel corrente.

L'analisi dell'istogramma fornisce generalmente utili informazioni sulle proprietà dell'immagine legate alle frequenze dei livelli di grigio.

*Esercizio 2.* Si scriva una funzione `function In= contrast_strach(I)` che implementi l'espansione della dinamica dei livelli di grigio in modo da migliorare la qualità di immagini caratterizzate da debole contrasto. L'operatore migliora il contrasto dell'immagine espandendo la dinamica dei livelli di grigio su un intervallo più ampio.

Detti  $G_m$  e  $G_M$  i livelli di grigio minimo e massimo dell'immagine iniziale, viene utilizzata una funzione di mapping lineare per espandere la dinamica nell'intervallo desiderato  $[G'_m, G'_M]$

$$In = (I - G_m) \left( \frac{G'_M - G'_m}{G_M - G_m} \right) + G'_m$$

Si provi la funzione su questa immagine, dopo aver fatto `I=imread('donna.png');` e `I=double(I(:, :, 1))`.

La trasformazione lineare considerata espande in modo uniforme la dinamica originale dell'immagine, producendo un effetto globale di miglioramento del contrasto. Talvolta però si ha l'esigenza di effettuare una trasformazione non uniforme, che agisca diversamente sui livelli di grigio inferiori e superiori.

Ad esempio, se l'immagine è sottoesposta i particolari interessanti sono poco evidenti e concentrati nelle zone scure; in tal caso può essere utile espandere la dinamica associata ai livelli scuri e comprimere quella dei livelli chiari. Viceversa, se l'immagine è sovraesposta i particolari interessanti sono poco evidenti e concentrati nelle zone chiare; in tal caso può essere utile espandere la dinamica associata ai livelli chiari e comprimere quella dei livelli scuri.

L'espansione non uniforme della dinamica dell'immagine può essere ottenuta mediante elevazione a potenza dei livelli originari. Si consideri la funzione  $y = x^r$ ,  $x \in [0, 1]$ ,  $r = 0.25, 0.5, 2, 4$ .

Se  $r < 1$  la funzione espande la dinamica dei valori inferiori e comprime quella dei valori superiori, se  $r > 1$  espande la dinamica dei valori superiori e comprime quella

dei valori inferiori. L'utilizzo della funzione  $y = x^r$  per la modifica della dinamica dei livelli di grigio richiede la normalizzazione degli assi rispetto all'intervallo  $[0, 255]$

$$y \rightarrow y/255, x \rightarrow x/255 : \frac{y}{255} = \left( \frac{x}{255} \right)^r \rightarrow 255^{1-r} x^r.$$

La relazione utilizzata è quindi:

$$P_o = 255^{1-r} P_i^r,$$

dove  $P_o$  rappresenta il nuovo valore del generico pixel e  $P_i$  è il valore nell'immagine originale.

*Esercizio 3.* Usando la vettorizzazione si scriva il comando per eseguire questa trasformazione. Si provi cosa si ottiene su questa immagine, dopo aver fatto `I=imread('uomo.png');` e `I=double(I(:, :, 1)).`