

## Elaborazione di Segnali Multimediali

# Elaborazioni nel dominio spaziale (1)

L.Verdoliva, D.Cozzolino

In questa lezione presenteremo le principali elaborazioni puntuali, cioè quelle operazioni che modificano singolarmente l'intensità di ogni pixel dell'immagine per migliorarne l'aspetto complessivo (*Image Enhancement*). Infine, studieremo le operazioni aritmetiche. Di seguito le istruzioni che ci permettono di importare i moduli Python che useremo in questa esercitazione:

```
# attiva la modalita interattiva di matplotlib
%matplotlib qt

import numpy as np          # importa Numpy
import matplotlib.pyplot as plt # importa Matplotlib
import scipy.ndimage as ndi  # importa Scipy per le immagini
import skimage.io as io      # importa il modulo Input/Output di SK-Image
```

## 1 Operazioni puntuali

Le elaborazioni puntuali si applicano ad ogni pixel dell'immagine senza considerare interazioni e dipendenze tra pixel vicini, e sono definite in funzione dell'intensità del pixel. Vedremo operazioni lineari e non lineari.

### 1.1 Offset additivo e cambiamento di scala

Cominciamo col valutare l'effetto di un offset additivo su un'immagine, realizziamo cioè la trasformazione:  $y = x + b$ , possiamo usare le immagini `vista_aerea.jpg` e `granelli.jpg`. La visualizzazione dell'istogramma può aiutarvi a scegliere il valore da assegnare all'offset  $b$ . In particolare, per la fotografia aerea un valore ragionevole è  $b = -50$ , quindi scrivendo  $y = x - 50$  si ottiene l'immagine elaborata. Se si vuole fare in modo che l'immagine in uscita abbia la stessa dinamica dell'ingresso, bisogna fare attenzione al fatto che i valori dell'immagine potrebbero diventare negativi. Si può scrivere allora il seguente codice:

```
mask = x > 0      # crea una matrice di vero (True) e falso (False)
x[mask] = 0       # si annullano tutti i valori minori di 0
```

In questo modo si gestiscono solo i valori negativi, per evitare anche di avere valori più grandi di 255:

```
x[x > 255] = 255
```

Visualizzate la maschera per vedere in quali punti eventualmente si ha un effetto di saturazione.

Notate come con questa operazione le immagini presentino globalmente un aspetto più o meno luminoso, tuttavia l'istogramma, che è stato traslato verso sinistra o destra, mostra come non tutti i valori siano stati usati. Provate ad effettuare un cambiamento di scala  $y = ax$  sulle due immagini proposte, determinando il valore appropriato di  $a$  in entrambi i casi e confrontando questa soluzione con quella precedente.

## 1.2 Negativi di un'immagine

Considerate l'immagine della mammografia `mammografia.jpg`, realizzatene il negativo ( $y = K - 1 - x$ ) con  $K = 256$ .

## 1.3 Full Scale Histogram Stretch

Il Full Scale Histogram Stretch – o Contrast Stretch – è una delle operazioni più utili nell'ambito dell'elaborazione e della visualizzazione di immagini digitali. Scrivete una funzione con il seguente prototipo: `def fshs(x)` che realizza questa operazione nell'ipotesi in cui  $K = 256$ . Applicate questa funzione ad immagini il cui istogramma non copre l'intera dinamica e visualizzate il risultato. Come già detto Python realizza automaticamente questa operazione attraverso il comando `plt.imshow(x, clim=None, cmap='gray')`, provate allora a confrontare il risultato ottenuto con quello che avreste visualizzando l'immagine originale con tale comando. Per calcolare massimo e minimo potete usare i comandi: `xmin = np.min(x)` e `xmax = np.max(x)`.

## 1.4 Trasformazione logaritmo e potenza

Applichiamo la trasformazione logaritmo  $y = \log(x + 1)$  in modo da migliorare il contrasto nelle zone scure per la trasformata di Fourier rappresentata nell'immagine `spettro.jpg`:

```
y = np.log(x+1)
plt.figure(); plt.imshow(y, clim=None, cmap='gray');
```

Provate invece ad applicare la trasformata potenza  $y = x^\gamma$  all'immagine di `vista_aerea.jpg`, con  $\gamma = 3$  e all'immagine di `spina_dorsale.jpg` (risonanza magnetica di una spina dorsale fratturata) con  $\gamma = 0.3$  (ricordate di realizzare sempre il contrast stretch dopo l'operazione non lineare).

```
y = x ** 3
z = fshs(y)
```

Realizzate l'enhancement di immagini troppo chiare o scure, con  $\gamma > 1$  e  $\gamma < 1$ , rispettivamente, visualizzando il risultato al variare di  $\gamma$ .

## 1.5 Equalizzazione dell'istogramma

L'equalizzazione dell'istogramma permette di migliorare il contrasto di un'immagine, quando sono stati usati tutti i valori appartenenti alla dinamica, ma la distribuzione appare fortemente sbilanciata. Caricate l'immagine `mart.jpg`, visualizzate il suo istogramma e poi effettuate l'equalizzazione dell'istogramma usando la funzione `skimage.exposure.equalize_hist` del Scikit-Image.

## 1.6 Uso delle statistiche locali

In questa sezione realizziamo l'enhancement solo su una parte dell'immagine `filamento.jpg`, allo scopo di evidenziare una struttura nascosta. Vogliamo effettuare un'elaborazione (amplificazione del valore) solo dei pixel appartenenti alle aree scure, che abbiano basso contrasto, ma non nelle regioni piatte. Questo si traduce in termini di condizioni su media e deviazione standard locale dell'immagine. Un pixel viene elaborato se:

1. la luminosità media attorno al pixel (media locale) è opportunamente minore della luminosità media dell'intera immagine;
2. la deviazione standard locale è relativamente bassa (indice di basso contrasto) e non è troppo piccola (per non enfatizzare le aree omogenee).

A questo punto solo se la media locale di un pixel è minore di quella globale opportunamente scalata ( $\mu_l \leq k_0 \mu$  per individuare zone scure) e se la deviazione standard locale soddisfa la disuguaglianza ( $k_1 \sigma \leq \sigma_l \leq k_2 \sigma$  per assicurare un basso contrasto), con  $k_0$ ,  $k_1$  e  $k_2$  costanti, il pixel va amplificato di un fattore  $E$ . Assumeremo  $k_0 = 0.4$ ,  $k_1 = 0.02$ ,  $k_2 = 0.4$  e  $E = 4$ . Per realizzare il codice che effettua tali operazioni bisogna determinare la matrice di test (maschera), che vale 1 laddove il pixel va elaborato, 0 altrimenti: Nell'ipotesi di aver già calcolato media, `med`, e deviazione standard, `std`, globali dell'immagine, di conoscere l'immagine delle medie locali, `MED`, e quella delle deviazioni standard, `DEV`, su blocchi  $3 \times 3$ , il comando che genera la maschera è:

```
mask = ((MED<=0.4*med)) & (DEV<=0.4*dev)) & (DEV>=0.02*dev)))
```

Scrivete uno script dal nome `localenhanc.m` che realizza le operazioni appena descritte e che richiama due funzioni per il calcolo delle medie e delle deviazioni standard, rispettivamente. Ricordate di visualizzare l'immagine originale e quella finale.