

Elaborazione di Segnali Multimediali

Elaborazioni nel dominio spaziale Soluzioni

1 Operazioni puntuali

1. *Negativo di un'immagine.*

Di seguito si presenta il codice per determinare, visualizzare e salvare su file il negativo di un'immagine. Fate attenzione al fatto che prima di salvare i dati in formato JPEG è necessario convertire il loro formato, rappresentandoli su 8 bit.

```
x = np.float32(io.imread('mammografia.jpg'))  
y = 255-x  
plt.imshow(y, clim=[0,255], cmap='gray')
```

2. *Full-Scale Histogram Stretch.*

```
def fshs(x):  
    """  
    effettua il Full-scale Histogram Stretch  
    """  
  
    m = np.min(x)  
    M = np.max(x)  
    y = 255*(x - m)/(M - m)  
    return y
```

3. Equalizzazione.

```
x = io.imread('granelli.jpg')
plt.figure(1); plt.imshow(x, clim=[0,255], cmap='gray');
plt.figure(2); plt.hist(x.flatten(), 256);

from skimage.exposure import equalize_hist
y = equalize_hist(x) # output e' nel range [0,1]
y = 255*y           # lo convertiamo nel range [0, 255]

plt.figure(3); plt.imshow(y, clim=[0,255], cmap='gray');
plt.figure(4); plt.hist(y.flatten(), 256);
```

4. Uso delle statistiche locali.

Utilizzando la funzione `scipy.ndimage.generic_filter` per il calcolo delle statistiche su blocchi $K \times K$ dell'immagine, il codice per l'enhancement è il seguente:

```
x = io.imread('filamento.jpg')
MED = ndi.generic_filter(x, np.mean, (3,3))
DEV = ndi.generic_filter(x, np.std, (3,3))
med = np.mean(x)
dev = np.std(x)
mask = (MED<=0.4*med) & (DEV<=0.4*dev) & (DEV>=0.02*dev)
y = x+3*x*mask

plt.figure(3); plt.imshow(y, clim=[0,255], cmap='gray');
```