



UNIVERSITÀ DEGLI STUDI
DI MILANO

Elaborazione delle Immagini

Raffaella Lanzarotti

Trasformazioni di intensità.

Contrast Stretching

- $J = \text{imadjust}(I)$:
 - mappa linearmente i valori di intensità da I a J t.c. l'1% dei l.d.g. chiari e scuri saturi ai valori limite
- $J = \text{imadjust}(I, [\text{low_in}; \text{high_in}], [\text{low_out}; \text{high_out}])$
 - valori in $[\text{low_in}, \text{high_in}]$ mappano in $[\text{low_out}, \text{high_out}]$.
 - Valori min di low_in mappano in low_out
 - Valori maggiori di high_in mappano in high_out .
 - Se $[\] \rightarrow [0,1]$

Trasformazioni di intensità.

Gamma Correction

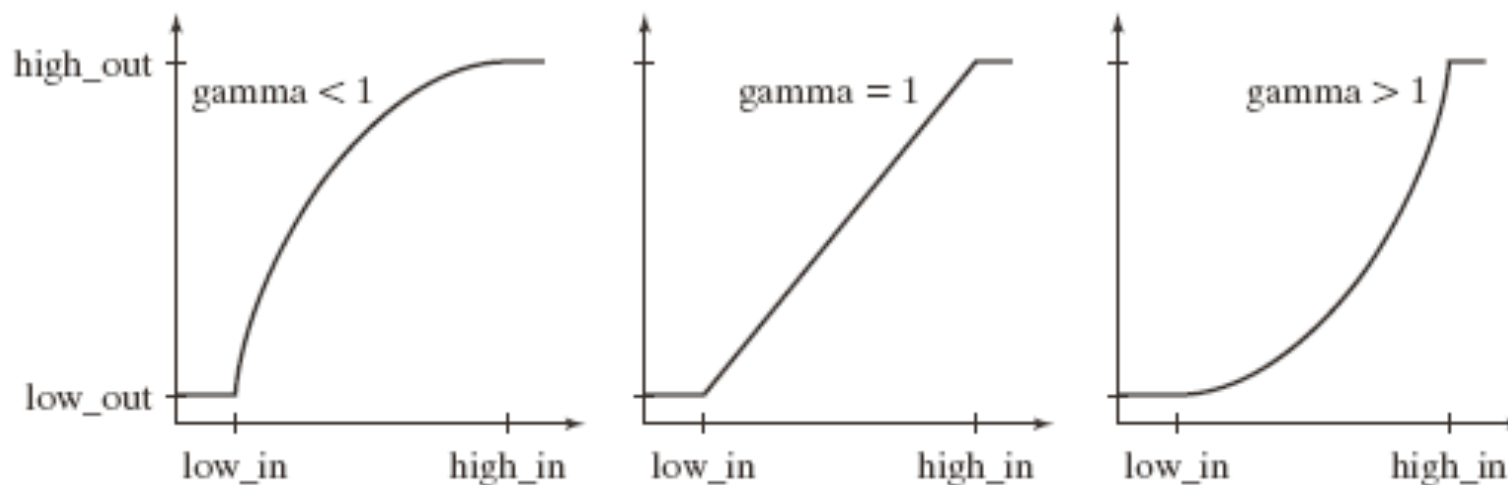
- `J= imadjust(I, [low_in;high_in], [low_out; high_out], gamma)`
 - gamma specifica la forma della curva di trasformazione:
 - Se gamma è $< 1 \rightarrow$ espande i chiari.
 - Se gamma è $> 1 \rightarrow$ espande gli scuri.
 - Se si omette gamma di default vale 1 (linear mapping).

Trasformazioni di intensità

a b c

FIGURE 3.2

The various mappings available in function `imadjust`.



Funzione correlata: stretchlim

- `LOW_HIGH = stretchlim(I)`
 - `LOW_HIGH`: vettore di 2 elem. che specificano i limiti inf. e sup.
Da usare nel contrast stratching per saturate all'1%
- `LOW_HIGH = stretchlim(I, TOL)`
 - `TOL`: vettore di 2 elem [`LOW_FRACT` `HIGH_FRACT`] che specifica il valore di saturazione inf. e sup. (es.: 0.02, 0.98)
 - se `TOL` è uno scalare è come se:
`LOW_FRACT = TOL`,
e `HIGH_FRACT = 1 - LOW_FRACT`,
 - Default: `TOL = [0.01 0.99]`
 - se `TOL = 0` → `LOW_HIGH = [min(I(:)); max(I(:))]`.

Esempio

- `J = imadjust(I, LOW_HIGH, []);`
- `J = imadjust(I, LOW_HIGH, [], 0.5);`
- `J = imadjust(I, LOW_HIGH, [], 1.5);`

Istogramma

- `h = imhist(I)`
 - Calcola l'istogramma di I con 256 bins
- `h = imhist(I, b)`
 - Calcola l'istogramma di I con b bins.
 - Es. $b=2$, e I è a l.d.g.
 - $h(1) = \text{\#pixel in } [1, 127]$
 - $h(2) = \text{\#pixel in } [128, 256]$

Istogramma normalizzato

- $p = \text{imhist}(I, b) / \text{numel}(I)$
 - Dove $\text{numel}(A)$ calcola il numero di elementi di una matrice A

Visualizzazione dell'istogramma

- `imhist(I);`
 - calcola «al volo» l'istogramma e lo visualizza
 - E' utile solo per ispezioni visive
- `h = imhist(I);`
- `plot(h);`
- `title('istogramma');`
- `xlabel('l.d.g.');`
- `ylabel('#pixels');`

Cumulative Distribution Function

- `CDF = cumsum(h) ;`
- `CDF = CDF / max(CDF) ;`
- `plot(CDF) ;`
- `title('CDF') ;`
- `xlabel('l.d.g.') ;`
- `ylabel('#pixels <= g') ;`

Esercizio 1

- Creare una funzione che aiuti l'utente nell'ispezione visiva:
 - ricevendo in input un'img a l.d.g., applica n (es 3 o 5) trasformate di intensità con gamma diversi (sia maggiori che minori di 1) e visualizzi i risultati e i rispettivi istogrammi in una figura unica avvalendosi del subplot. Per ogni trasformazione dovrà essere indicato nel titolo il gamma applicato.
- Cosa si può concludere?
- Esiste una trasformazione ottimale?
- Tutte le img trovano giovamento da almeno una delle trasformazioni fatte?

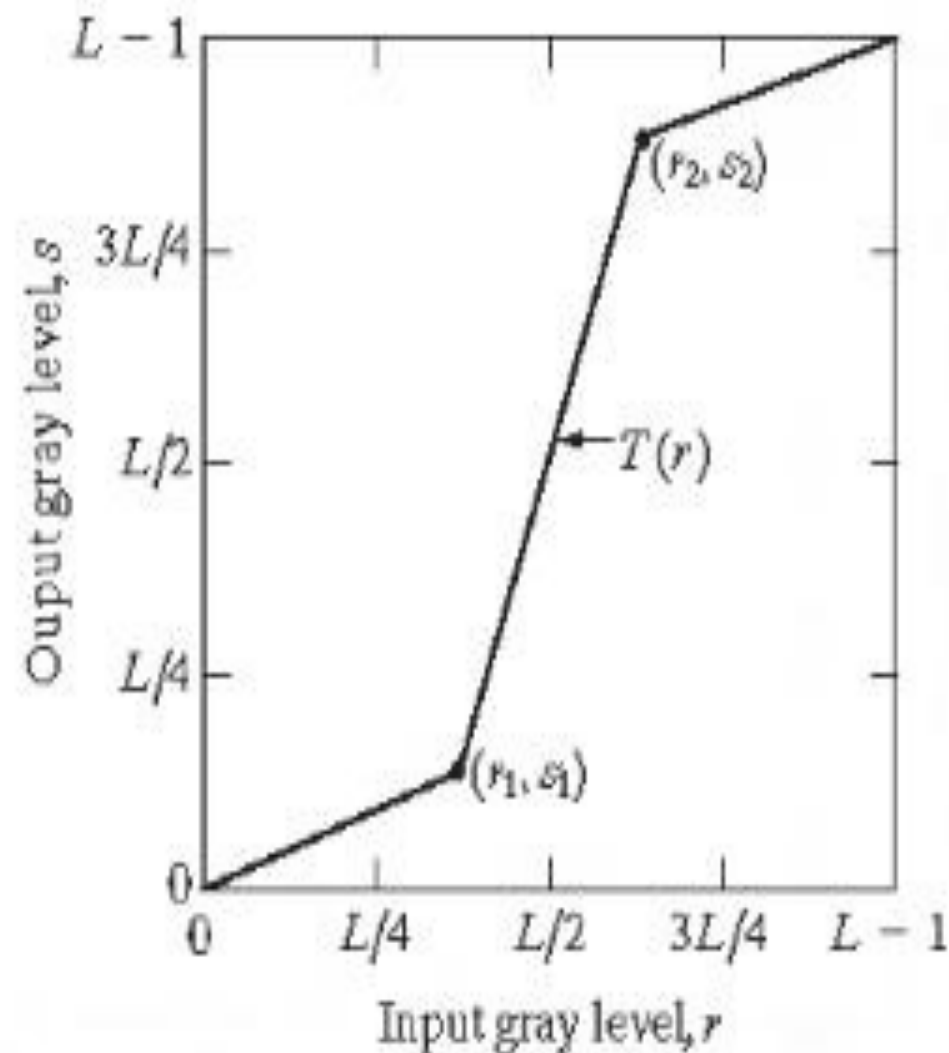
Esercizio 2

- Implementare la piecewise-linear transformation function

- Hint:

- Punti di controllo: $(\min(I(:)), 0)$, (r_1, s_1) , (r_2, s_2) , $(\max(I(:)), 255)$, con r_1, s_1, r_2, s_2 parametrici
- la formula delle rette passanti per due punti è:

$$y = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1) + y_1$$



Esercizio 3

- Implementare la trasformazione sigmoidale:

Let $a = 2$.

Let $x \in \{0, \dots, 255\}$

$$\sigma(x; a) = \frac{255}{1 + e^{-a(x-127)/32}}$$

o precostituendo la corrispondente LUT o applicandola direttamente all'img di input

Esercizio 4

Point Processes: Histogram Equalization

Task: remap image \mathbf{I} with $\min = m_{\mathbf{I}}$ and $\max = M_{\mathbf{I}}$ so that its histogram is as close to constant as possible and has $\min = m_{\mathbf{J}}$ and $\max = M_{\mathbf{J}}$.

Let $P_{\mathbf{I}}(\gamma+1)$ be the cumulative (probability) distribution function of \mathbf{I} .

Then \mathbf{J} has, as closely as possible, the correct histogram if

Using
intensity
extrema

$$\mathbf{J}(r, c) = (M_{\mathbf{J}} - m_{\mathbf{J}}) \frac{P_{\mathbf{I}}[\mathbf{I}(r, c) + 1] - P_{\mathbf{I}}(m_{\mathbf{I}} + 1)}{P_{\mathbf{I}}(M_{\mathbf{I}} + 1) - P_{\mathbf{I}}(m_{\mathbf{I}} + 1)} + m_{\mathbf{J}}.$$

Esercizio 5

- Implementare l'algoritmo di histogram matching per immagini a livelli di grigio

Esercizio 6

- Implementare l'algoritmo di histogram matching per immagini a colori
 - facendo riferimento alla CDF della luminanza dell'immagine target