

Elaborazione delle immagini a colori

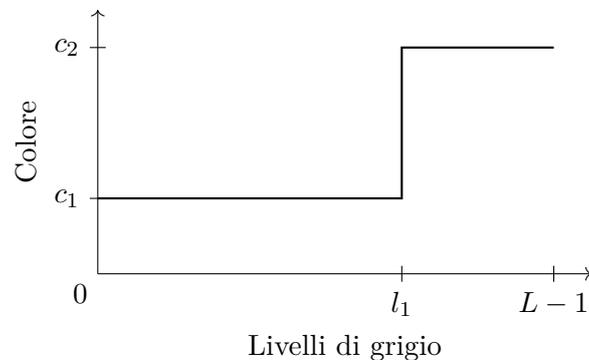
1 Pseudocolore

Quando si deve effettuare l'ispezione visuale di un'immagine a scala di grigi, può essere utile assegnare dei colori ai vari livelli di grigio (dato che l'occhio umano è in grado di distinguere simultaneamente numerosi colori, ma solo pochi grigi). L'immagine risultante si dice in **pseudocolore** (o falso colore), per differenziarla dalle immagini acquisite direttamente a colori (cioè misurando l'intensità della radiazione in più bande dello spettro elettromagnetico), che sono invece dette **full/true color**.

1.1 Intensity slicing

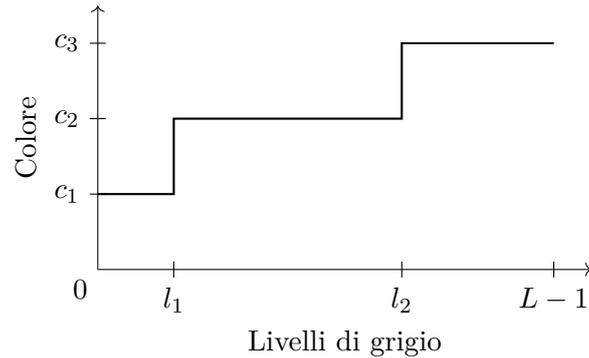
Una tecnica per assegnare i colori a un'immagine è l'**intensity slicing** (o **color coding**):

1. si definisce un valore di soglia per l'intervallo dei livelli di grigio;
2. si assegna un colore a tutti i valori sotto la soglia, e un altro a tutti i valori sopra la soglia.



Geometricamente, se si raffigura l'immagine nello spazio cartesiano tridimensionale, come una funzione di due variabili, $f(x, y)$, ciascun valore di soglia corrisponde a un piano perpendicolare all'asse delle intensità (e quindi parallelo al piano xy) che "taglia" l'immagine, associando un colore diverso a ciascun lato del piano.

Definendo due soglie, si possono assegnare tre colori, e così via.



In generale:

- Sia $[0, L - 1]$ la scala di grigi.
- Siano l_0 il nero e l_{L-1} il bianco.
- Si definiscono P piani perpendicolari all'asse delle intensità, ai livelli l_1, \dots, l_P , e con $0 < P < L - 1$.
- I P piani partizionano la scala di grigi in $P + 1$ intervalli, V_1, \dots, V_{P+1} .
- L'assegnamento di un colore a un pixel, in base al suo livello di grigio, è dato dalla relazione

$$f(x, y) = c_k \quad \text{se} \quad f(x, y) \in V_k$$

dove c_k è il colore assegnato al k -esimo intervallo, V_k .

2 Elaborazione di immagini full color

Per l'elaborazione di un'immagine a colori, supponendo che essa sia codificata in RGB, la finestra viaggiante diventa un "cubo viaggiante", poiché ogni pixel è definito in termini di tre componenti, situati sui piani R, G, e B.

In senso stretto, sarebbe necessario trattare i valori dei pixel come vettori a tre componenti. Solitamente, in pratica, si assume (semplificando) che i tre piani siano indipendenti, e si opera separatamente sui tre piani.

2.1 Operatori puntuali

Gli operatori puntuali lavorano sui valori delle componenti di un singolo pixel:

$$s_i = T_i(r_1, \dots, r_n) \quad \text{con} \quad i = 1, \dots, n$$

- n è il numero delle componenti colore (RGB ha $n = 3$, CMYK ha $n = 4$, ecc.);

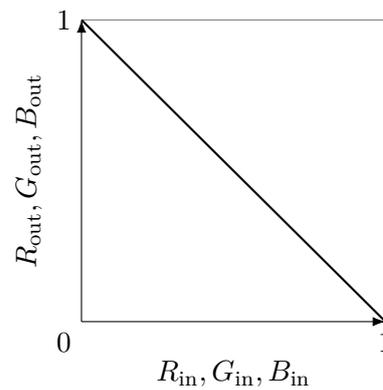
- r_i e s_i denotano le componenti colore di un pixel, rispettivamente dell'immagine originale e di quella risultante;
- la trasformazione T_i riceve in input tutte le componenti colore del pixel, e produce in output il nuovo valore dell' i -esima componente;
- l'insieme delle trasformazioni T_1, \dots, T_n definisce l'operatore puntuale.

Il modo di effettuare una stessa operazione cambia a seconda della codifica. Ad esempio, per scurire un'immagine:

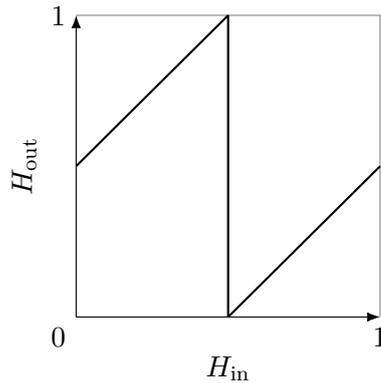
- in RGB, è necessario diminuire i valori di tutti e tre i piani;
- in CMY, si *aumentano* i valori dei tre piani (ciò corrisponde a un aumento della quantità di pigmento, quindi viene assorbita più luce);
- in HSI, bisogna diminuire solo i valori del piano intensità (I), lasciando tinta (H) e saturazione (S) inalterate, per evitare di alterare i colori.

Un altro esempio è la sostituzione di ciascun colore con il suo complementare:

- in RGB, si effettua il negativo dei tre piani;



- in HSI, è necessario:
 - “ruotare” la tinta di 180° (o di 0.5, se essa è rappresentata con valori nel range $[0, 1]$);



- fare il negativo dell'intensità;
- lasciare inalterata la saturazione.

Invece, agendo individualmente sui vari piani colore, si può effettuare un'operazione di **color balancing**.

2.2 Trasformazioni spaziali

Le trasformazioni spaziali, ad esempio di smoothing e sharpening, si applicano solitamente in modo separato sui tre piani dell'immagine RGB.

In HSI, invece, queste trasformazioni lavorano solo sul piano dell'intensità, perché non si vogliono cambiare i colori presenti nell'immagine.

I risultati dell'applicazione di una stessa trasformazione in RGB e HSI non sono, però, perfettamente identici (perché la conversione tra i due spazi colore può comportare qualche approssimazione, e perché in HSI si lavora, appunto, solo sull'intensità, invece che indipendentemente su ciascun piano colore).

2.2.1 Smoothing con filtro di media

Sia $c(x, y)$ un pixel di un'immagine RGB, con componenti $R(x, y)$, $G(x, y)$, e $B(x, y)$,

$$c(x, y) = \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix}$$

e sia $S_{x,y}$ un insieme di coordinate che definiscono un intorno di tale pixel. L'operazione di smoothing con filtro di media sull'intorno $S_{x,y}$ è definita come:

$$\bar{c}(x, y) = \frac{1}{k} \sum_{(s,t) \in S_{x,y}} c(s, t) = \begin{bmatrix} \frac{1}{k} \sum_{(s,t) \in S_{x,y}} R(s, t) \\ \frac{1}{k} \sum_{(s,t) \in S_{x,y}} G(s, t) \\ \frac{1}{k} \sum_{(s,t) \in S_{x,y}} B(s, t) \end{bmatrix}$$

In pratica, essa corrisponde a operazioni di smoothing separate sui tre piani RGB.

2.2.2 Operatore Laplaciano

Come lo smoothing, anche il Laplaciano di un'immagine RGB (utile, ad esempio, per lo sharpening) è calcolato lavorando separatamente sui tre piani:

$$\nabla^2 c(x, y) = \begin{bmatrix} \nabla^2 R(x, y) \\ \nabla^2 G(x, y) \\ \nabla^2 B(x, y) \end{bmatrix}$$