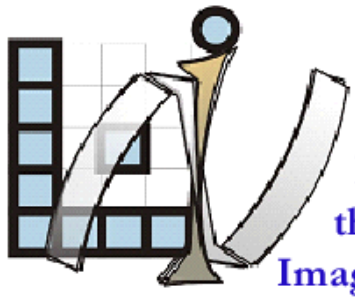
Segmentazione di palline colorate

Un esempio di applicazione di tecniche di classificazione alle immagini



Laboratory for the Analysis of Images and Vision

Problema: segmentazione

- Vogliamo **isolare** nell'immagine le palline di **colori diversi**

- Esempio semplice di applicazione a scopo di:
  - conteggio
  - controllo qualità
  - ...

(es. automazione industriale)

Approccio di risoluzione automatica

1. Quale algoritmo?
2. Su che dati?
3. Rappresentati come?

} scelte "collegate"

Dati e loro rappresentazione

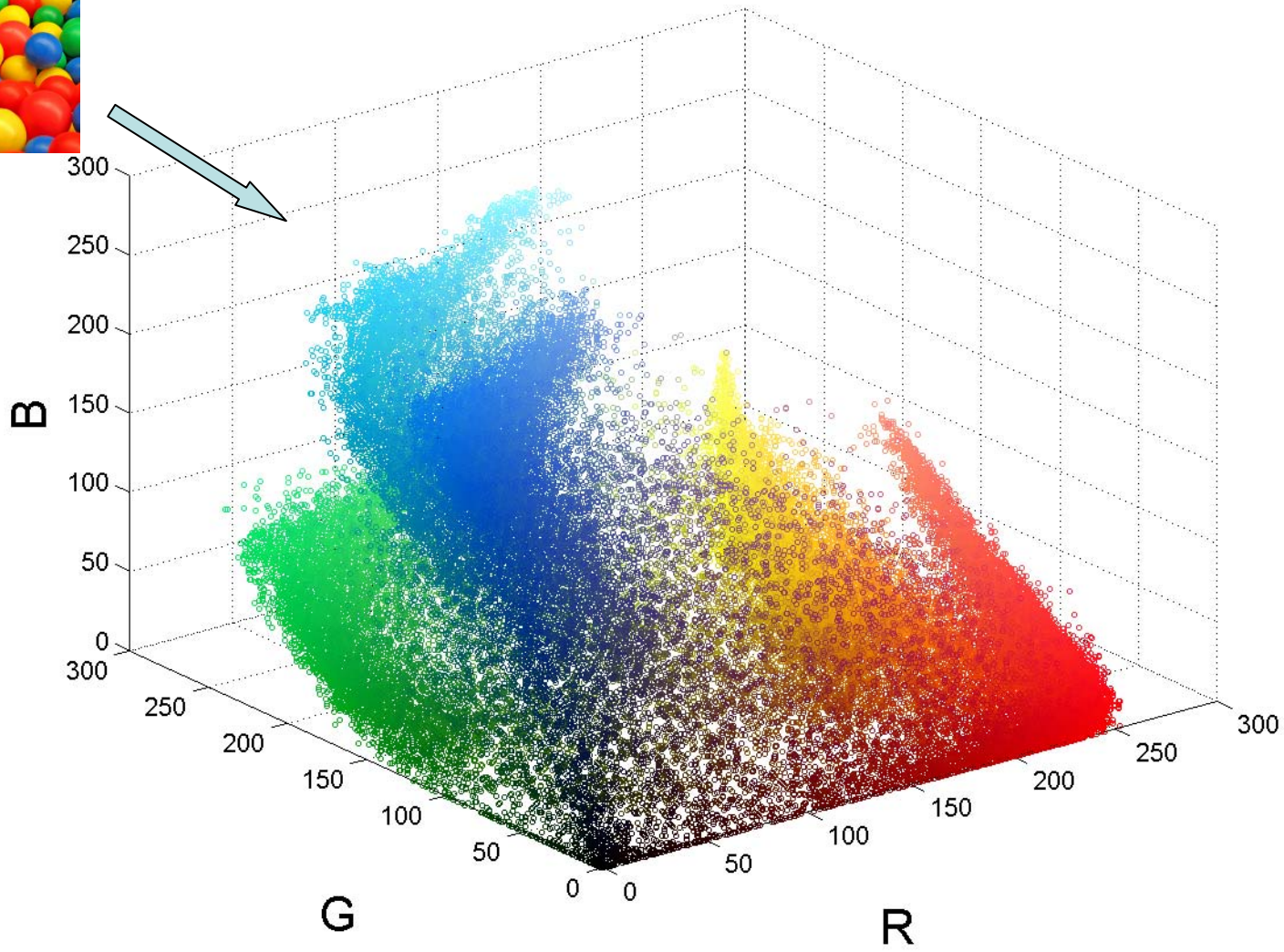
- caratteristica discriminante: **colore**

$\Rightarrow$ non consideriamo info spaziale di vicinanza tra pixel

$$I = \{\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_i, \ldots \mathbf{p}_{w \times h}\} \quad \text{dove} \quad \mathbf{p}_i = (p_i^R, p_i^G, p_i^B)$$

ripartiamo I in tante classi (esclusive) quanti sono i colori:

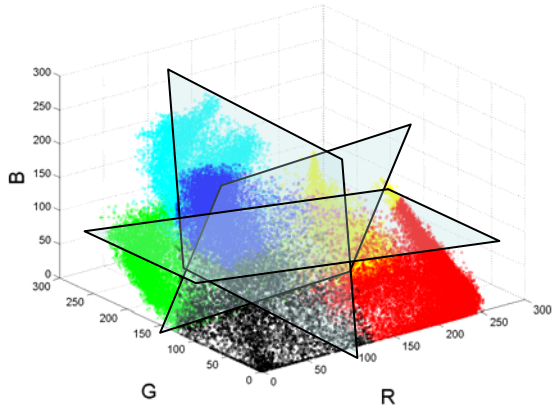
$$I = \bigcup_{l \in Colori} I_l \quad \text{t.c.} \quad \forall \mathbf{p}_i, \mathbf{p}_j,\ i \neq j,\ \mathbf{p}_i, \mathbf{p}_j \in I_l \Leftrightarrow$$ hanno lo stesso "colore"


B
G
R
300
250
200
150
100
50
0
300
250
200
150
100
50
0
0
50
100
150
200
250
300

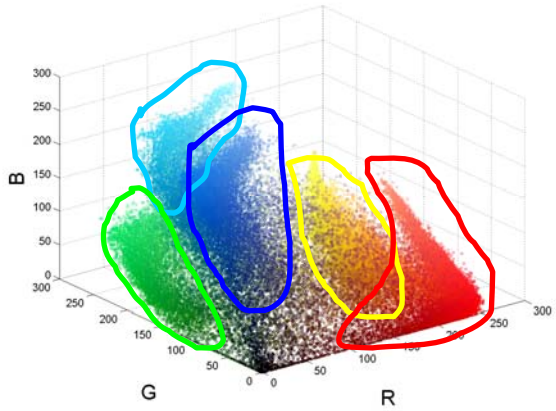
Tecnica di classificazione

- Due approcci alternativi:

1. Classificazione **supervisionata**:
   tariamo l'algoritmo su un'**immagine di campione già etichettata** per imparare a separare **analiticamente** le classi di colore per poi classificare *I*
   (es. usando il **percettrone** o sue evoluzioni)



2. Classificazione **non supervisionata**:
   i pixel di *I* vengono ripartiti in ***K* cluster** (classi) solo in funzione della loro mutua **distanza** nello spazio RGB



Algoritmo: *K*-means

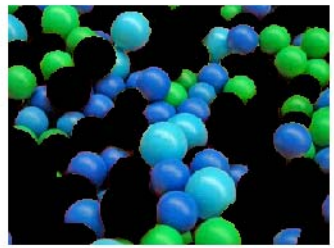
- Ipotesi: conosciamo **a priori** il numero di classi K
- Soluzione:
  - partizione di I
  - K "prototipi" di colore medio $\mathbf{c}_1, ..., \mathbf{c}_K$
- Criterio di ottimalità: $$\min \sum_{l=1}^{K} \sum_{\mathbf{p}_i \in I_l} distanza(\mathbf{p}_i, \mathbf{c}_l)$$
- Procedimento: randomizzato e iterativo

$\Rightarrow$ Colore di $\mathbf{p}_i \in I_l$ è $\mathbf{c}_l$

Risultato: 2-means su RGB


originale

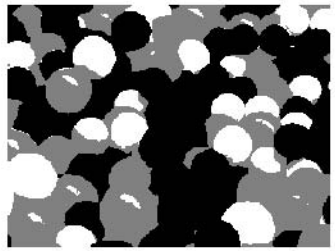

etichette

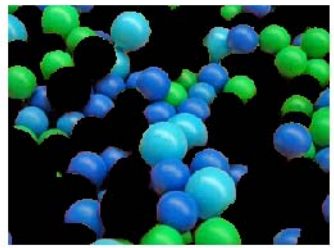

cluster 1


cluster 2

Risultato: 3-means su RGB


originale


etichette

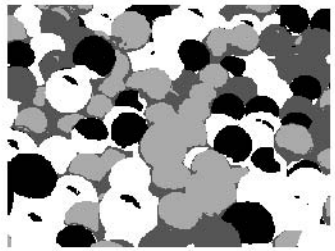

cluster 1


cluster 2

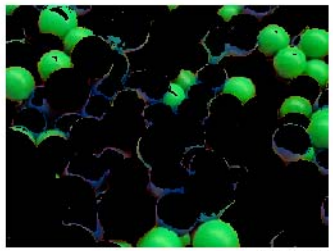

cluster 3

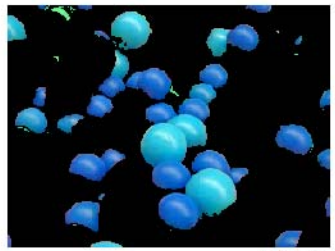
Risultato: 4-means su RGB


originale


etichette


cluster 1

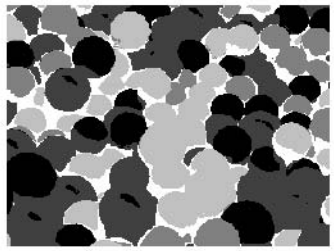

cluster 2


cluster 3

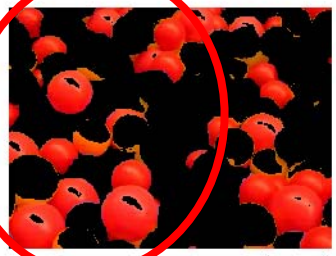

cluster 4

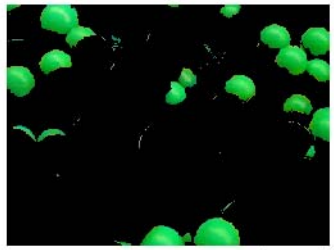
Risultato: 5-means su RGB

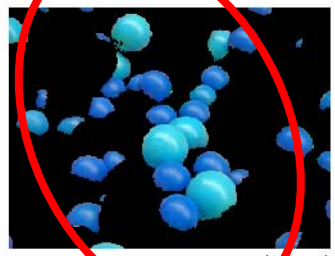

originale


etichette


cluster 1

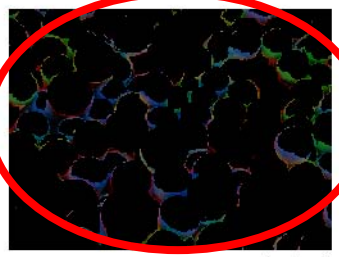

cluster 2


cluster 3


cluster 4

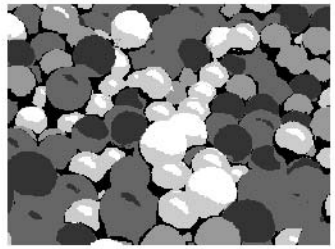
sovra-segmentazione

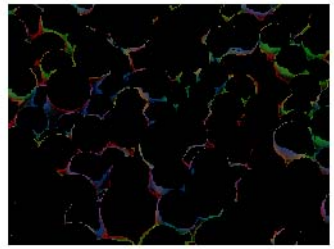
sotto-segmentazione


cluster 5

Risultato: 6-means su RGB

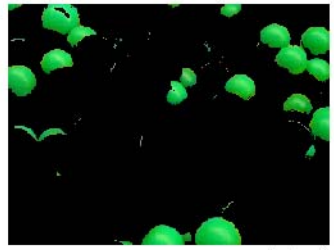

originale

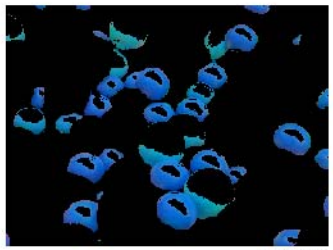

etichette

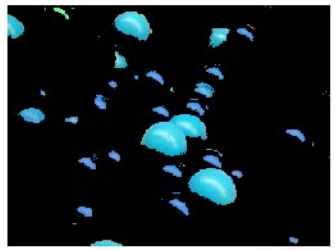

cluster 1


cluster 2

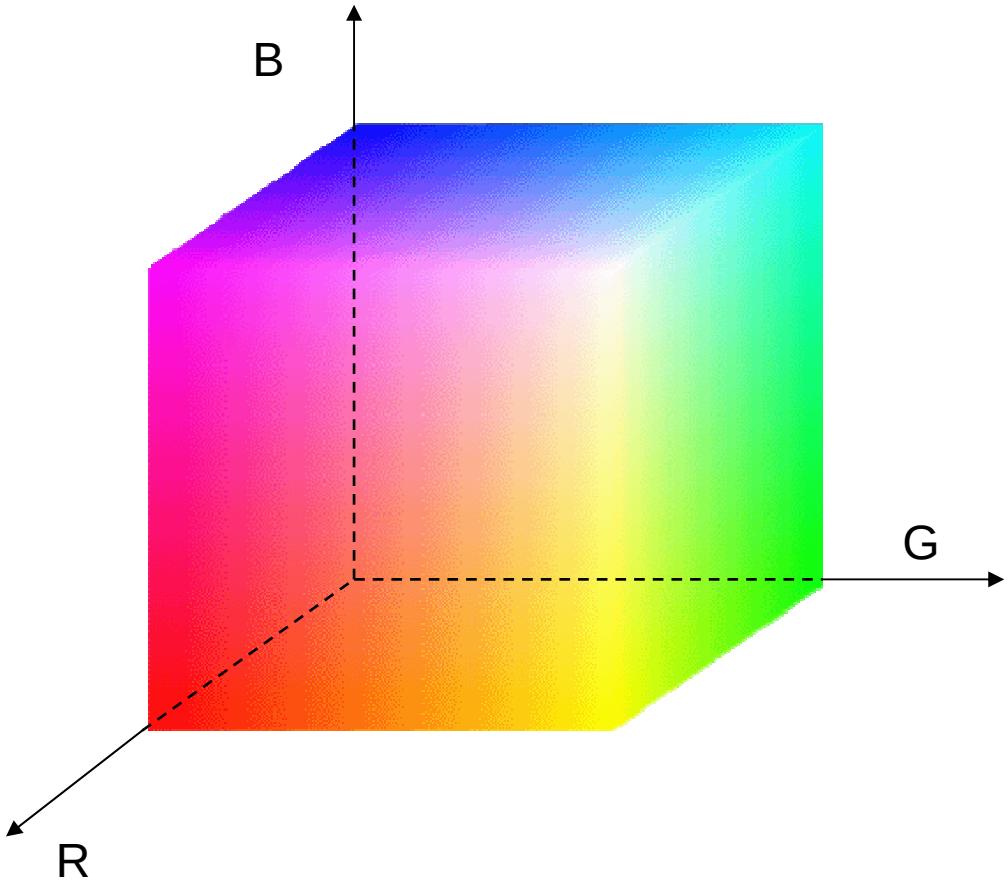

cluster 3

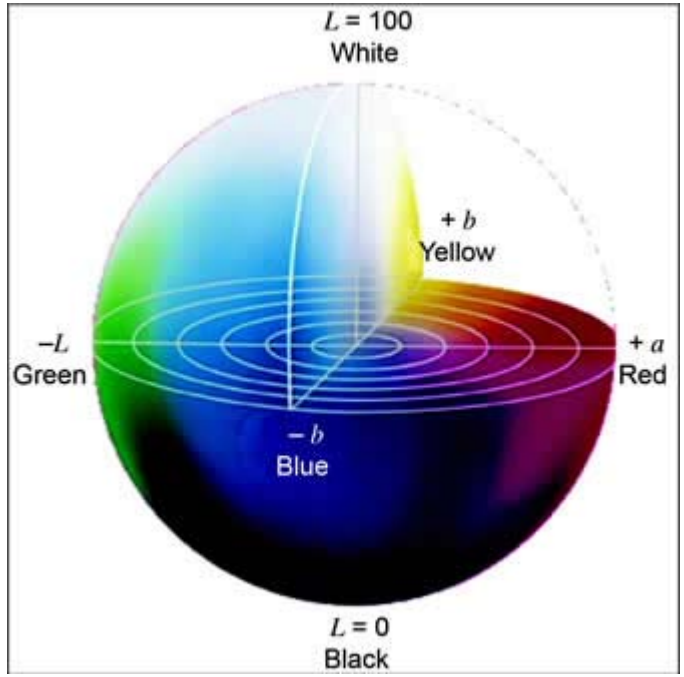

cluster 4


cluster 5


cluster 6

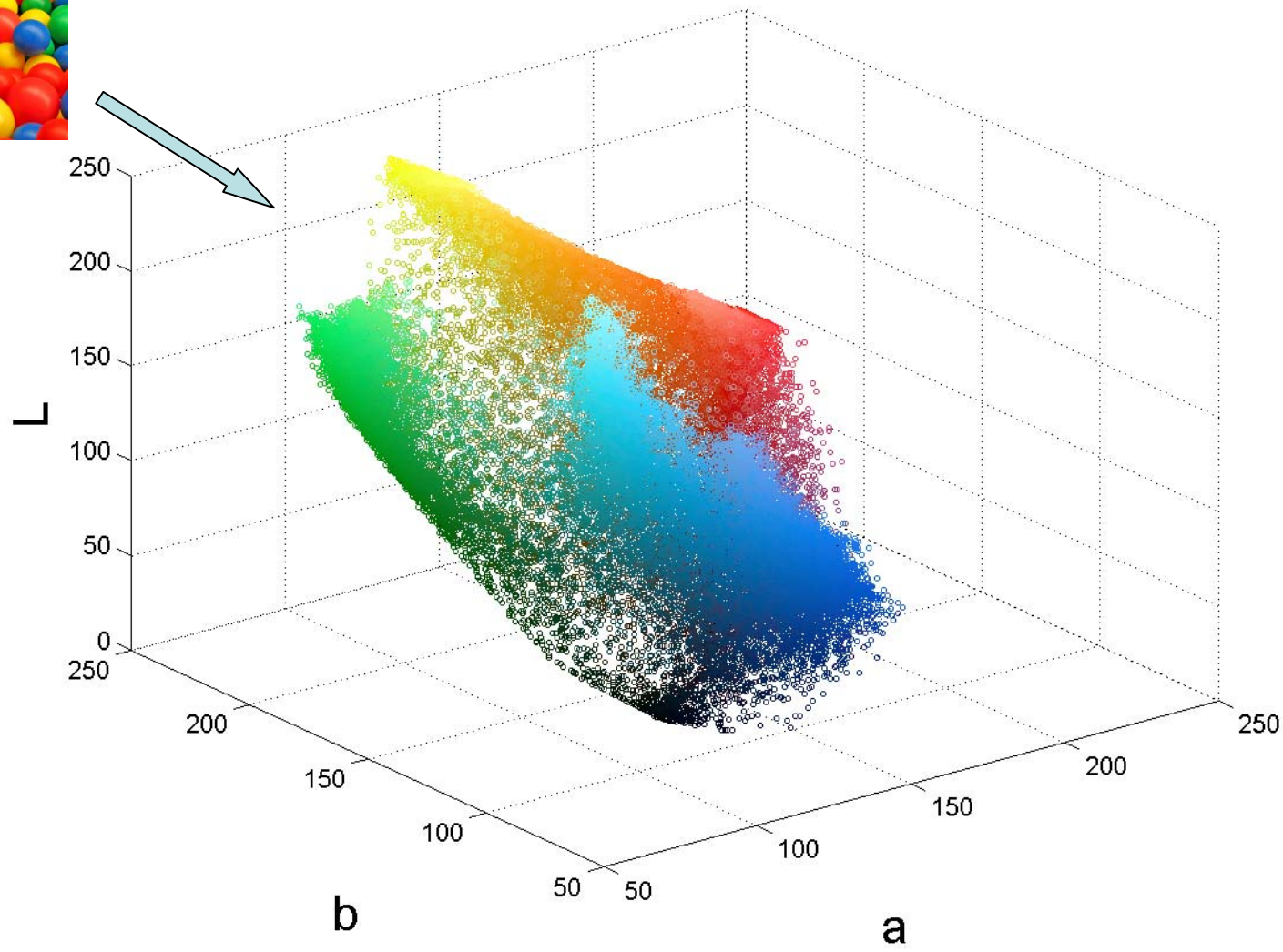
Cambio di rappresentazione





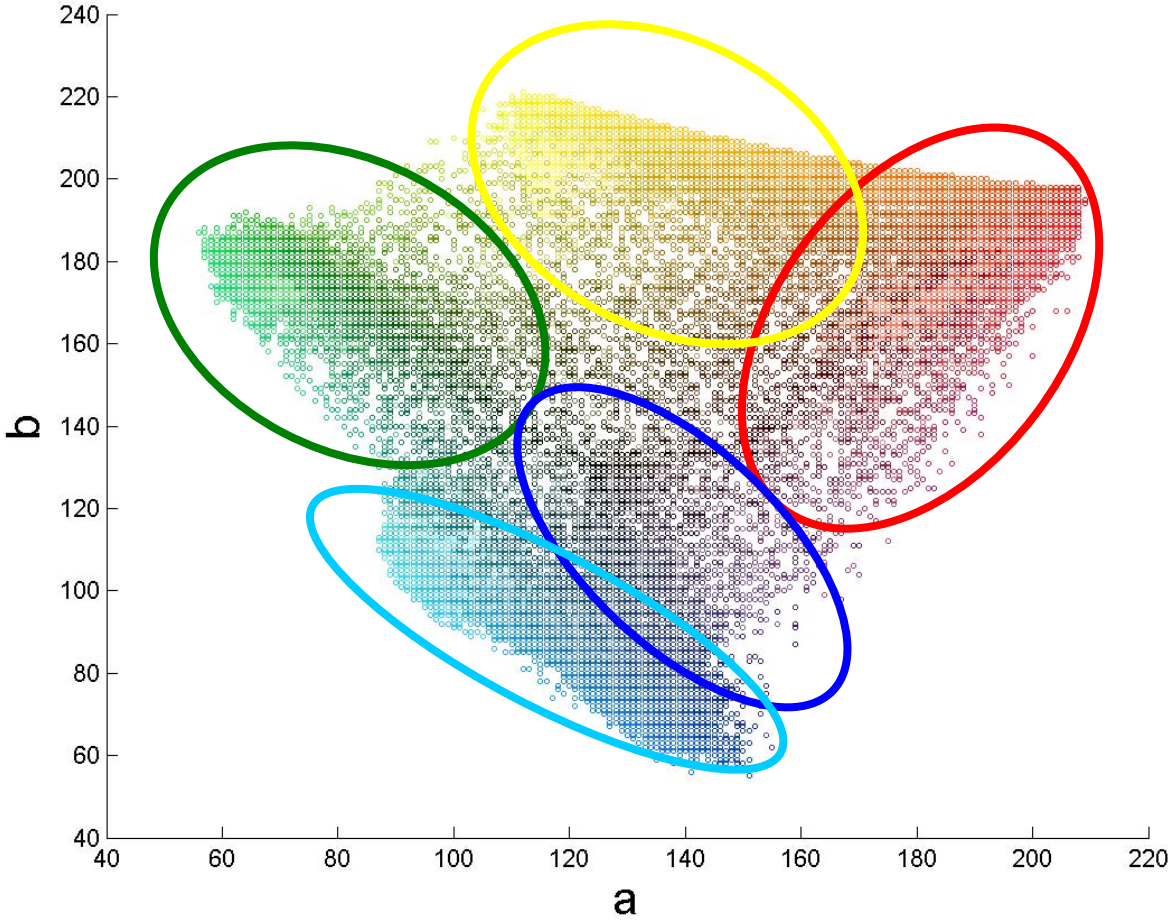
CIE Lab

- luminosità **separata** da cromaticità
- distanza tra colori basata sulla **percezione visiva umana**


L
250
200
150
100
50
0
250
200
150
100
50
b
50
100
150
200
250
a

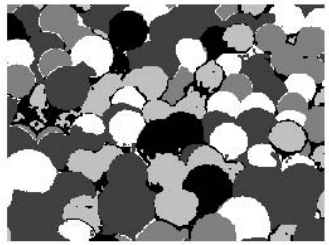
Selezione delle caratteristiche

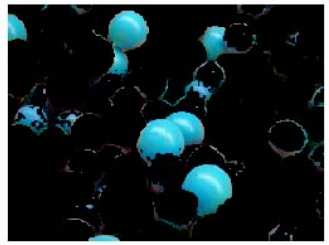
- Scartiamo l'informazione di luminosità

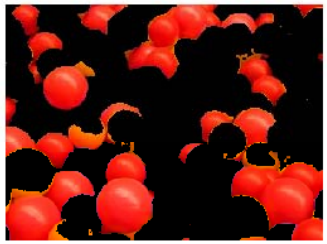


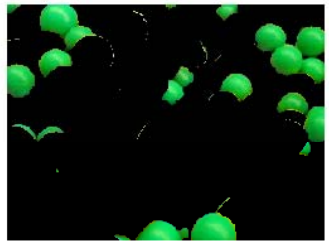
Risultato: 5-means su CIE Lab

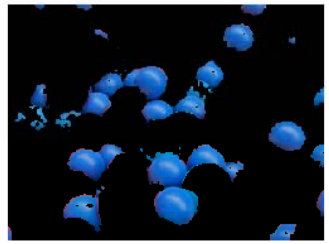

originale

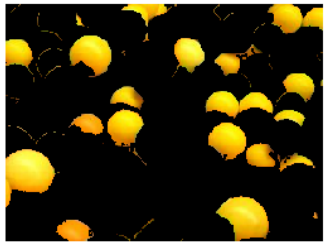

etichette


cluster 1


cluster 2

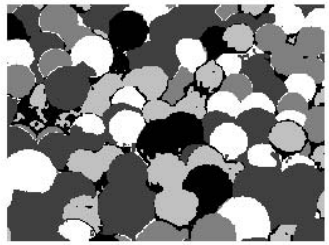

cluster 3

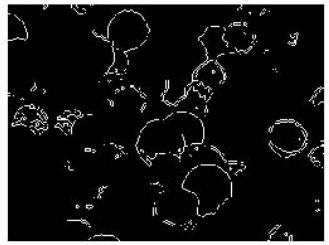

cluster 4


cluster 5

Risultato: 5-means su CIE Lab


originale


etichette


bordi del cluster 1


bordi del cluster 2


bordi del cluster 3


bordi del cluster 4


bordi del cluster 5