

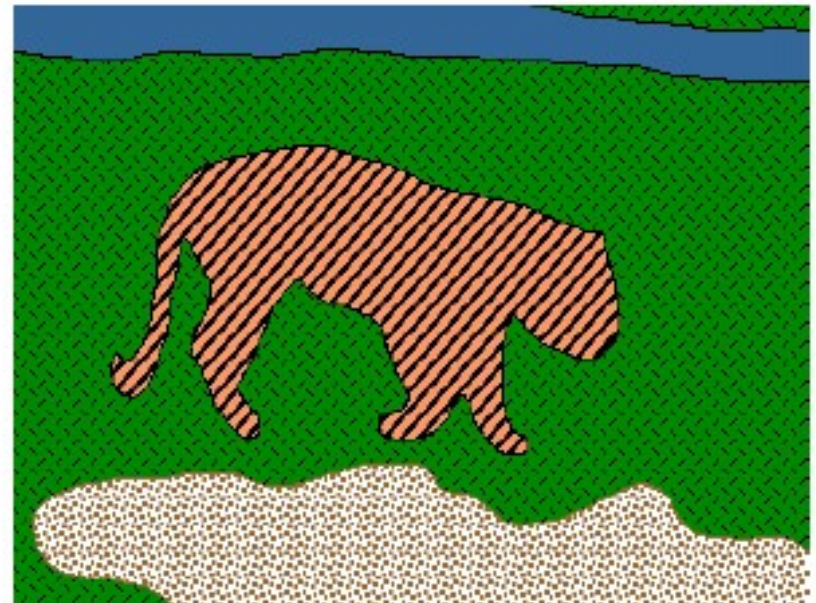
Corso di Visione Artificiale

Segmentazione

Samuel Rota Bulò

Segmentazione

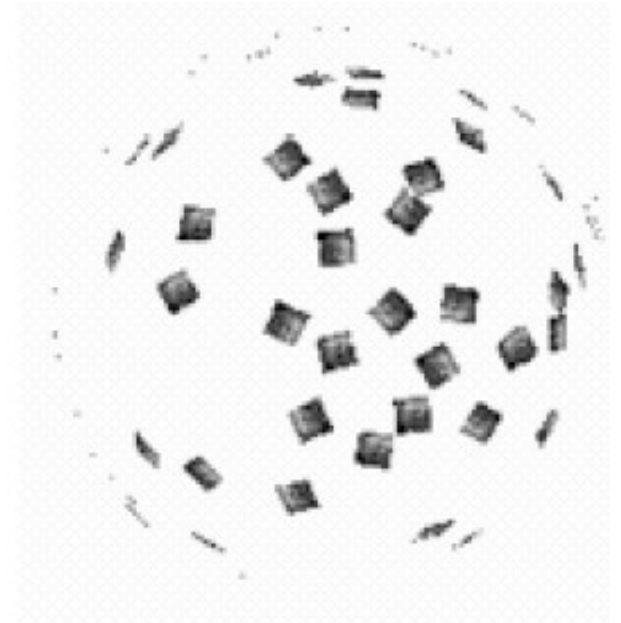
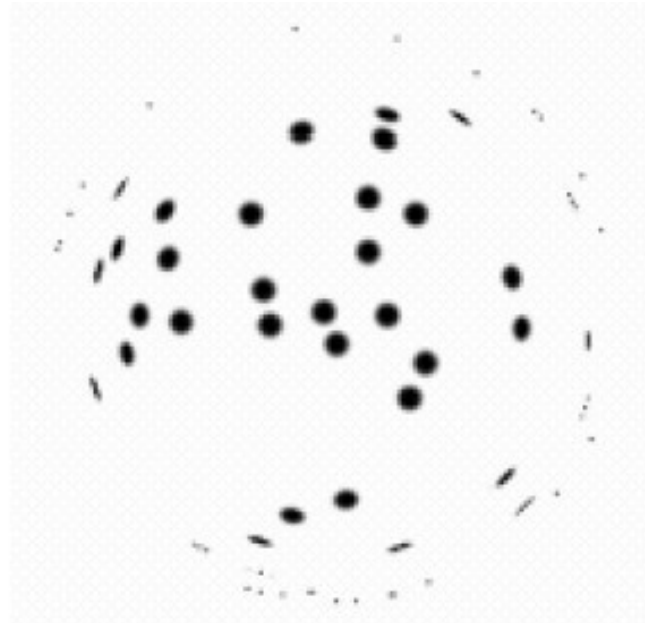
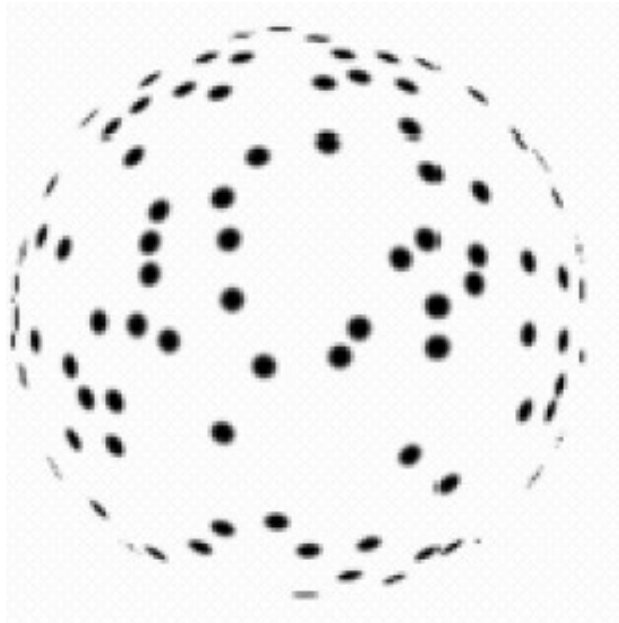
- La segmentazione ha l'obiettivo di partizionare un'immagine in regioni (insiemi di pixels) sulla base dell'intensità dei pixels e/o loro proprietà.
- Per riconoscere oggetti, anziché lavorare con troppi pixels, abbiamo bisogno di una rappresentazione più compatta.



La visione umana



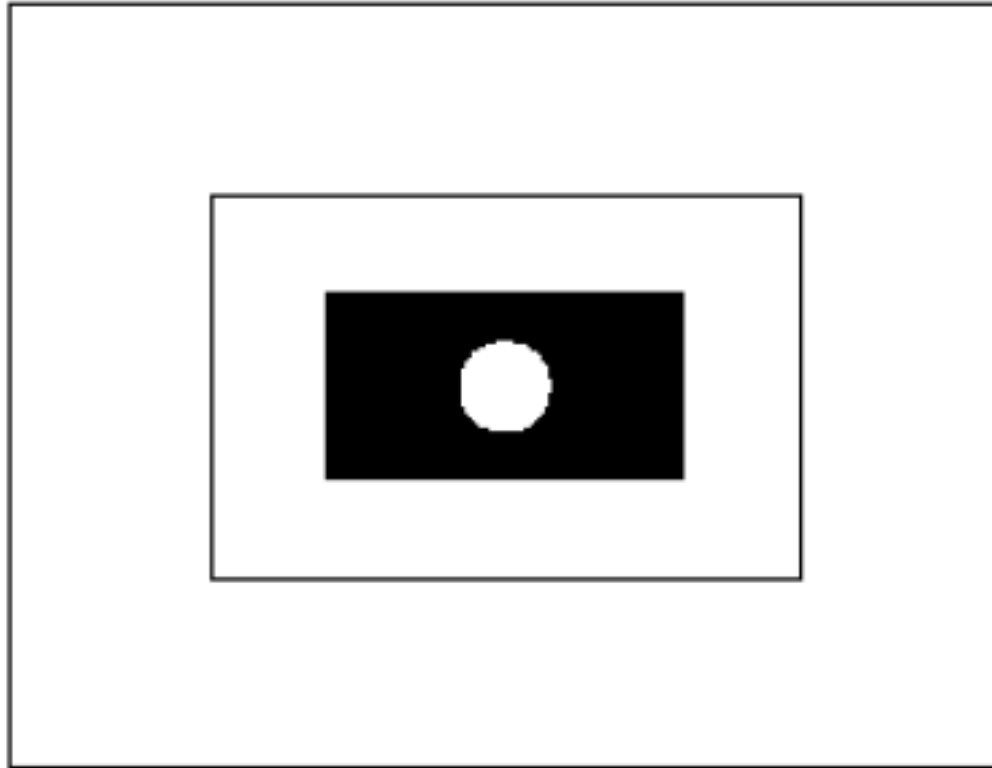
La visione umana



Idee base del raggruppamento

- **Separazione figura/sfondo**
 - raggruppamento può essere visto come allocare degli elementi alla figura (oggetto) e altri allo sfondo in cui la figura è immersa.
- **Scuola psicologica della Gestalt (anni '10-'30)**
 - rifiuta lo studio delle risposte a singoli stimoli (il tutto è diverso dalla somma delle singole parti)
 - enfatizza il raggruppamento come la chiave per capire la percezione visiva.

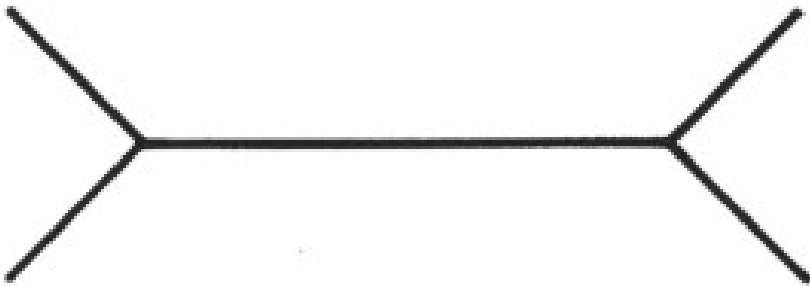
Figura/Sfondo



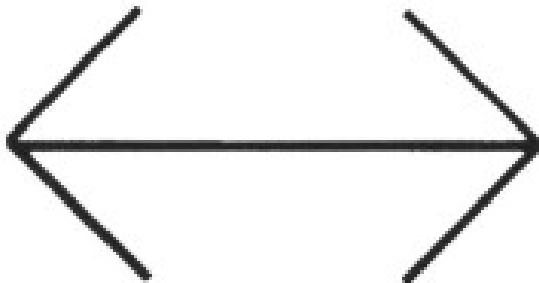
- E' il rettangolo con il buco al centro la figura? O il cerchio bianco? o il secondo rettangolo più grande ?
- La segmentazione può non essere unica e può dipendere dalla scala a cui si osservano le cose o dallo scopo che abbiamo.

Proprietà della Gestalt

- Gli elementi in una collezione di elementi hanno delle proprietà che derivano non dagli elementi di per sé ma da come questi si relazionano con gli altri.
- La Gestalt individua diversi fattori in base ai quali degli elementi possono essere raggruppati insieme.



- Effetto di Muller-Lyer: le linee centrali sembrano diverse perché il sistema visivo non le identifica come linee, ma come parti di un oggetto più complesso.
- Non riusciamo a guardare la figura ignorando le frecce in modo da vedere che le linee hanno effettivamente la stessa lunghezza.



Fattori della Gestalt

- **Prossimità:** elementi vicini.
- **Similarità:** elementi simili.
- **Destino comune:** elementi con moto coerente.
- **Regione comune:** elementi racchiusi all'interno di una stessa regione chiusa
- **Parallelismo:** elementi paralleli
- **Chiusura:** elementi che formano una curva chiusa
- **Simmetria:** elementi che portano a gruppi simmetrici
- **Continuità:** elementi che formano una curva continua
- **Congruenze familiari:** elementi che se raggruppati sembrano un oggetto familiare.

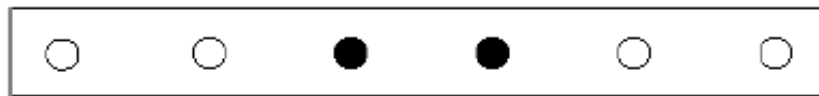
Fattori della Gestalt



Not grouped



Proximity



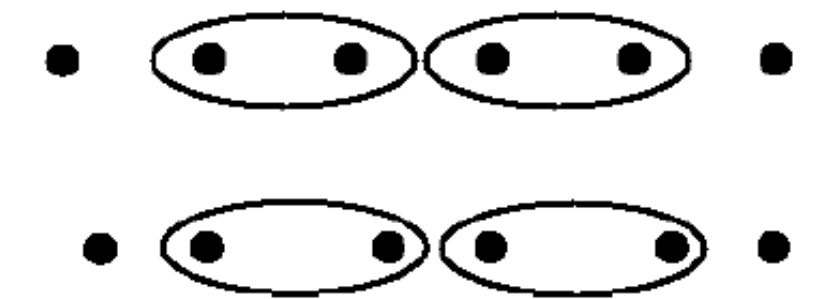
Similarity



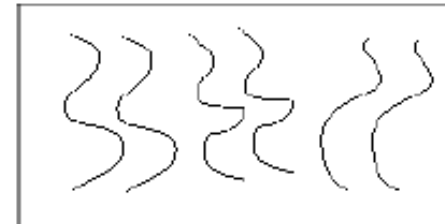
Similarity



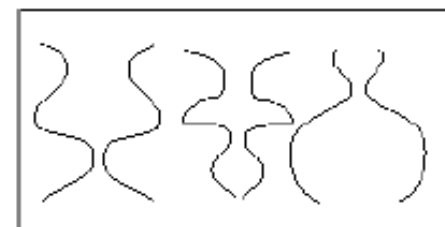
Common Fate



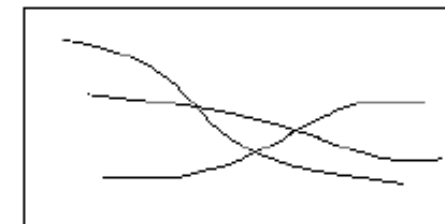
Common Region



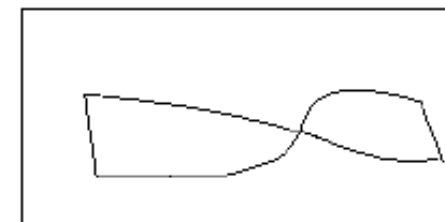
Parallelism



Symmetry



Continuity

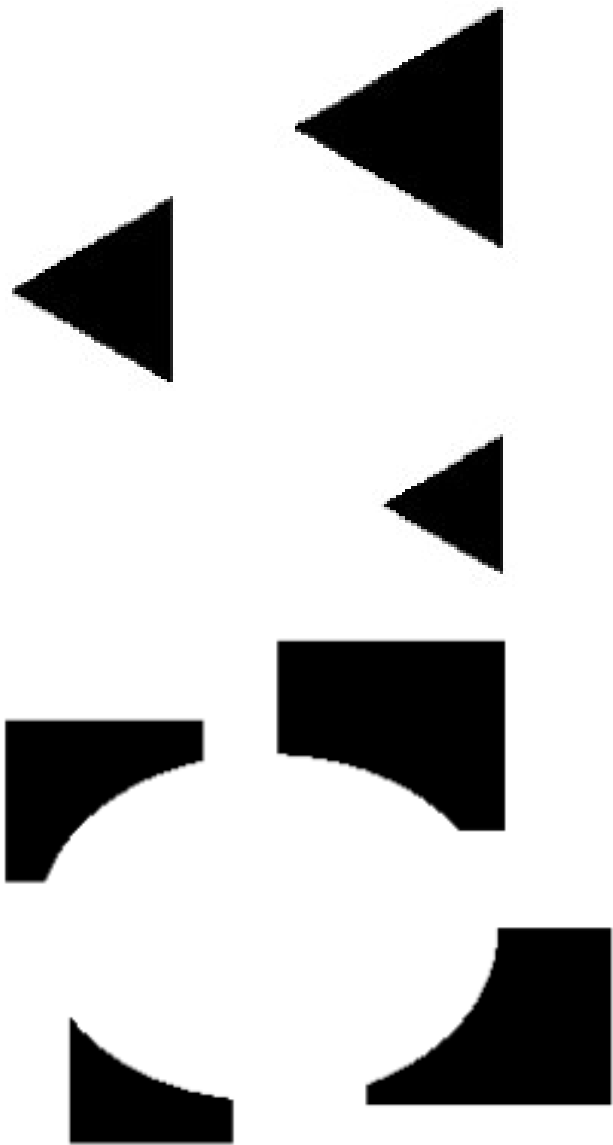


Closure

Occlusione



Contorni illusori



A



B



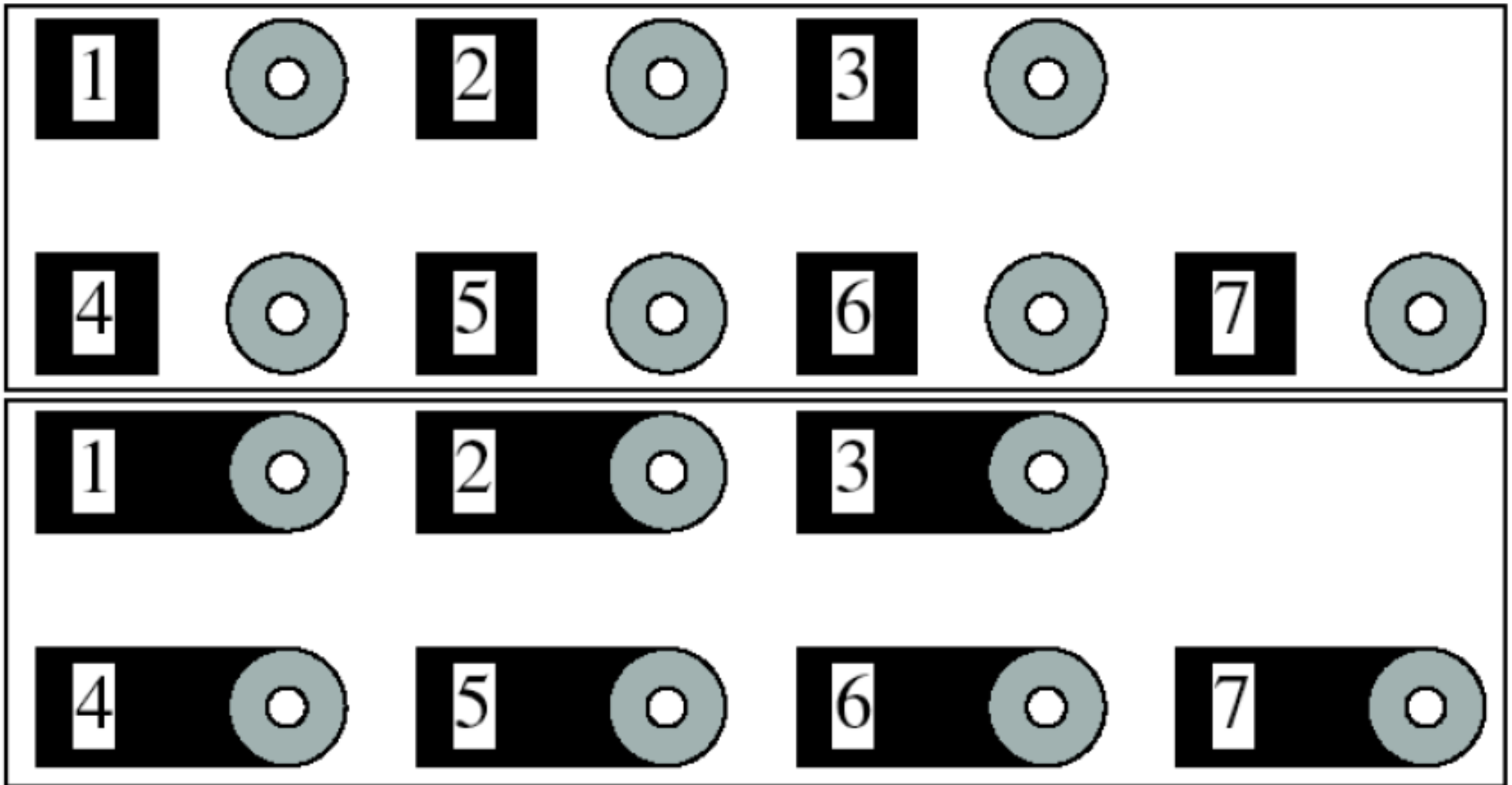
C



D



Altro esempio

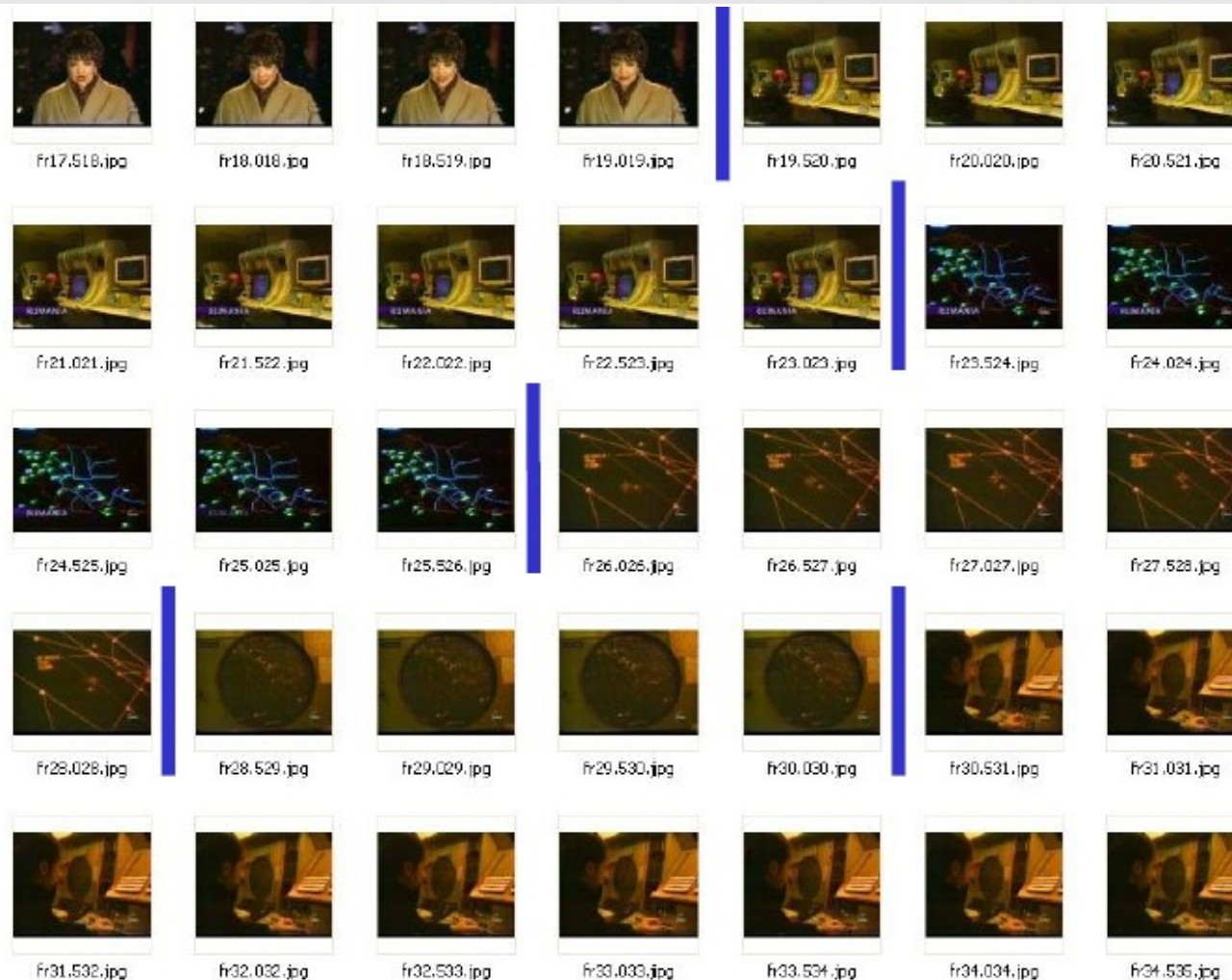


Applicazioni della segmentazione

- **Riassumere un video:** spezzarlo in sequenze di frame comuni e prendere un frame da ciascun segmento come rappresentativo.
- **Ricerca di linee/cerchi:** segmentare l'immagine in linee e cerchi a partire da bordi/punti salienti.
- **Cercare persone:** cerca parti del corpo umano e assemblarli – cercare parti di colore uniforme/texture.
- **Cercare edifici in immagini satellitari:** cercare poligoni a partire dai bordi/punti salienti dell'immagine.
- **Cercare all'interno di una collezione di immagini:** data un'immagine di query, cercare immagini simili – si segmentano le immagini e cercano parti comuni.

Bordi sequenze video

- Analisi differenziale tra frames.
- Bordi in corrispondenza di differenze elevate (come nel caso di bordi di immagini).



Rimozione di sfondo

- In alcune applicazioni, dove conosciamo l'aspetto dello sfondo (per es: immagini con videocamera ferma), possiamo identificare facilmente le figure.
- Applicazioni industriali, interazione uomo/macchina, tracciamento di macchine su una strada, video sorveglianza.
- **Approccio semplice:**
 1. stimare il valore dei pixels dello sfondo
 2. fare un test per ogni pixel e segnalare quelli in cui ci sono deviazioni consistenti dal valore di sfondo atteso.
 3. pulizia con filtri morfologici.
- Problema dell'approccio semplice è che lo sfondo non rimane necessariamente invariato col passare del tempo. La stima dello sfondo va aggiornata continuamente

Approccio con media variabile

Differenze tra
frame e ipotesi di
sfondo al tempo n

$B^{(0)}$: stima iniziale dello sfondo

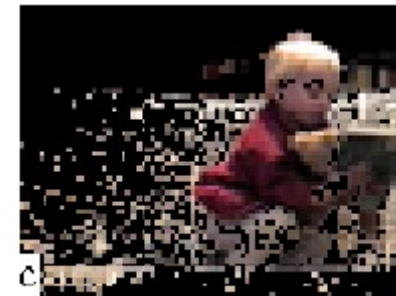
$$D^{(n)} = f(F^{(n)} - B^{(n)})$$

Aggiornamento
ipotesi di sfondo

$$B^{(n+1)} = \frac{w_a F^{(n)} + \sum_i w_i B^{(n-i)}}{w_c}$$



low thresh



high thresh



EM (later)

Altri esempi



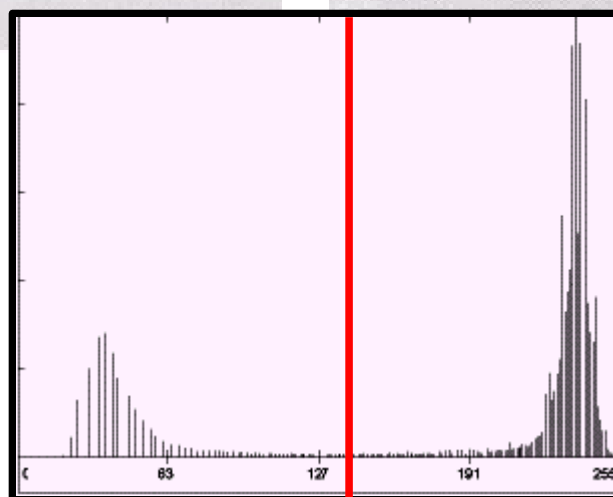
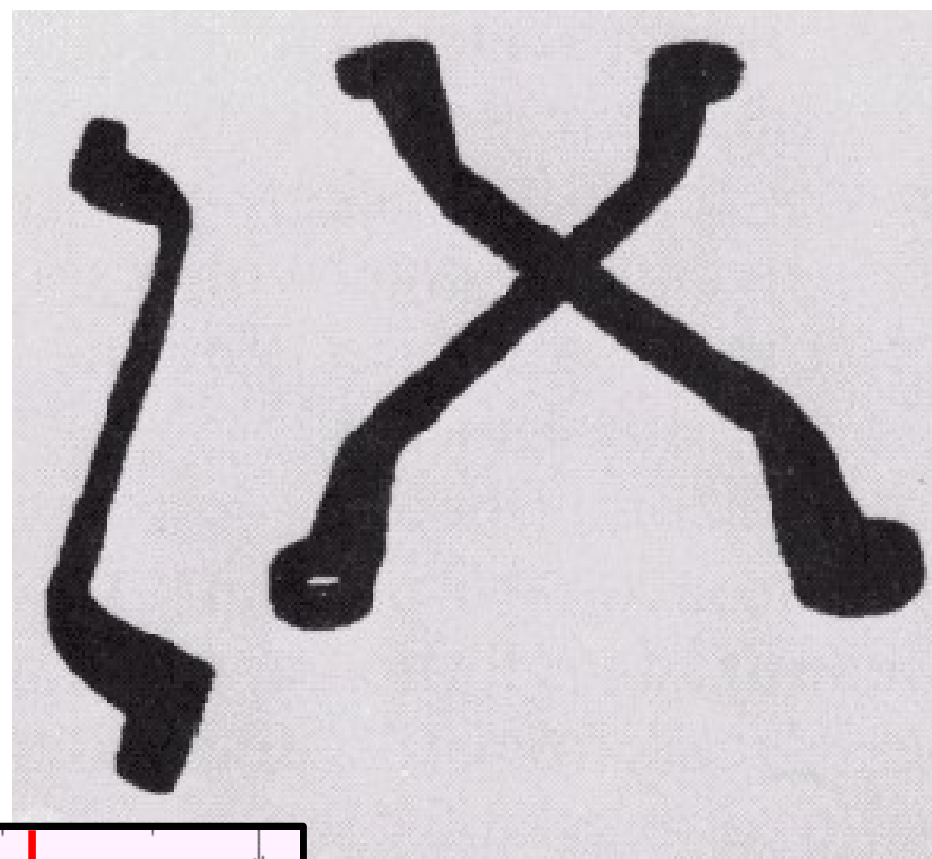
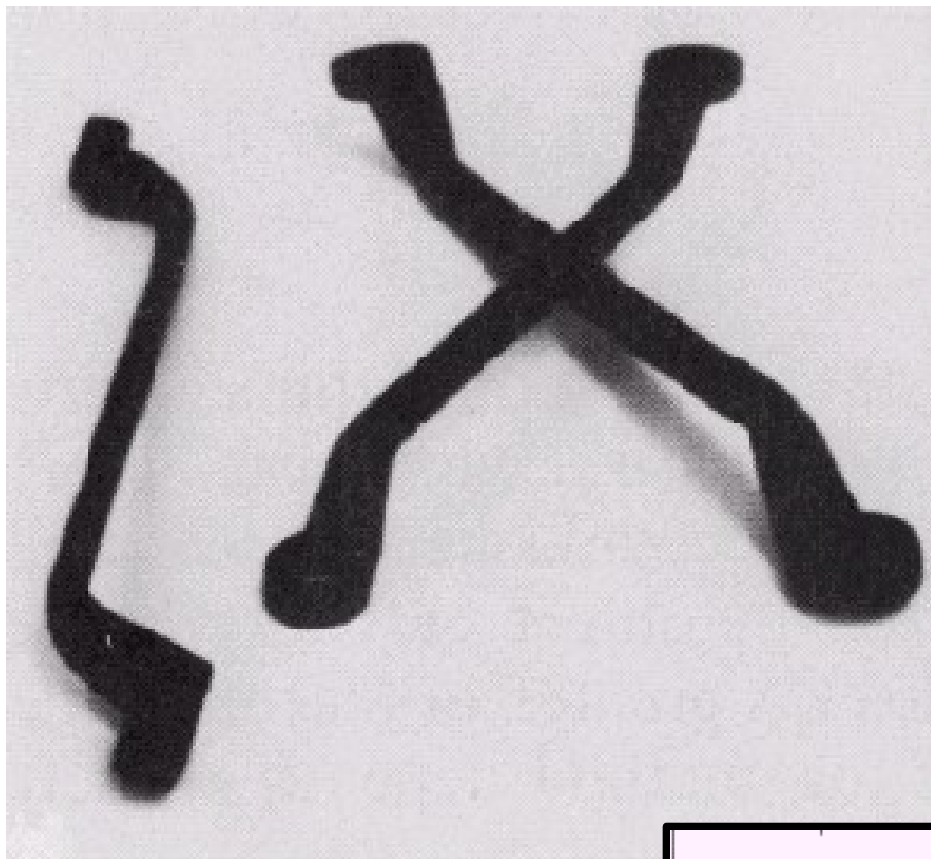
Sogliatura globale

- La sogliatura è un metodo semplice per segmentare immagini.
- Consideriamo il problema di segmentare in 2 parti un'immagine in cui abbiamo oggetti chiari su sfondo scuro.
- Un modo ovvio per farlo consiste nel trovare una soglia T e assegnare ogni pixel di intensità maggiore a T alla regione degli oggetti e quelli inferiori alla regione dello sfondo:

$$J[x, y] = \begin{cases} 1 & \text{se } I[x, y] > T \\ 0 & \text{se } I[x, y] \leq T \end{cases}$$

- Una soglia di questo tipo è detta **globale** perché viene applicata ad ogni pixel dell'immagine.

Sogliatura globale

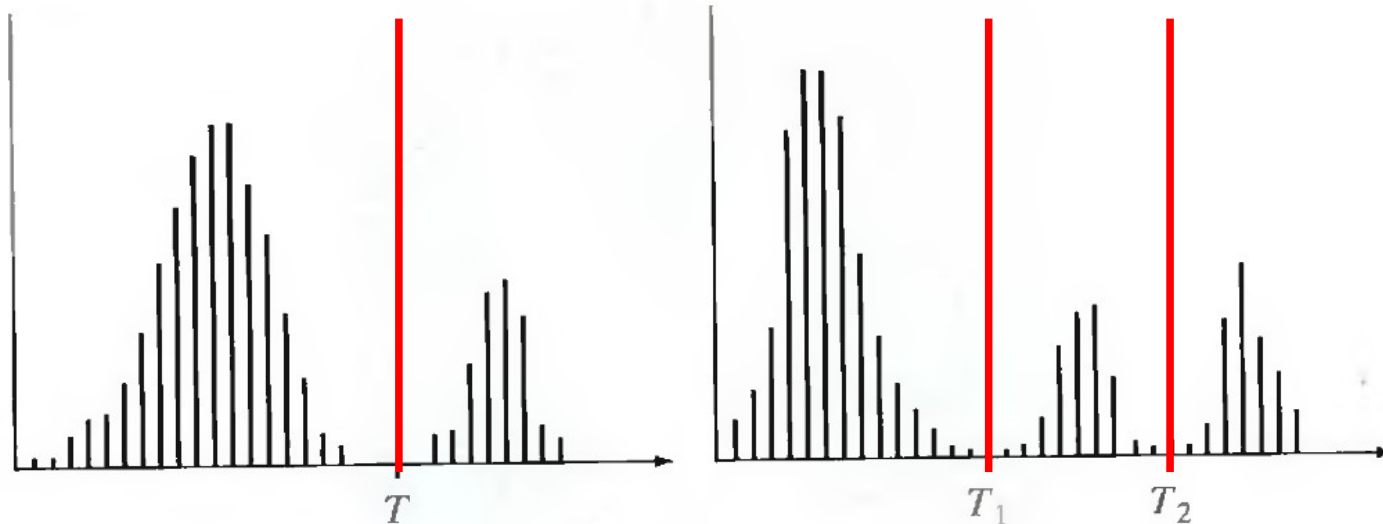


Sogliatura globale multipla

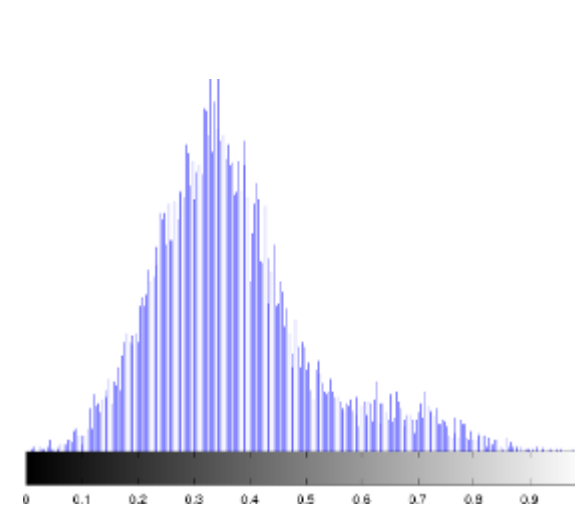
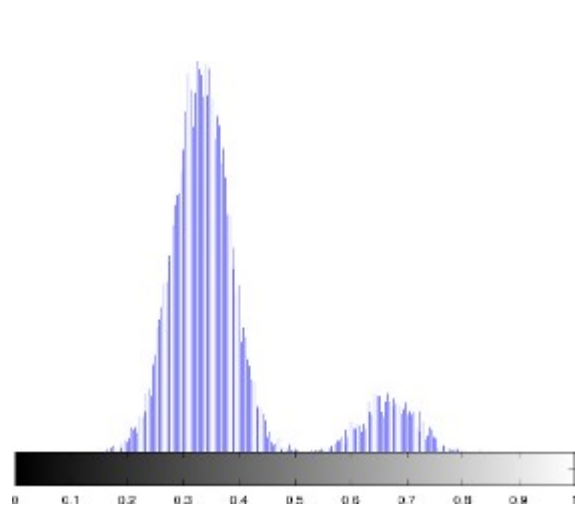
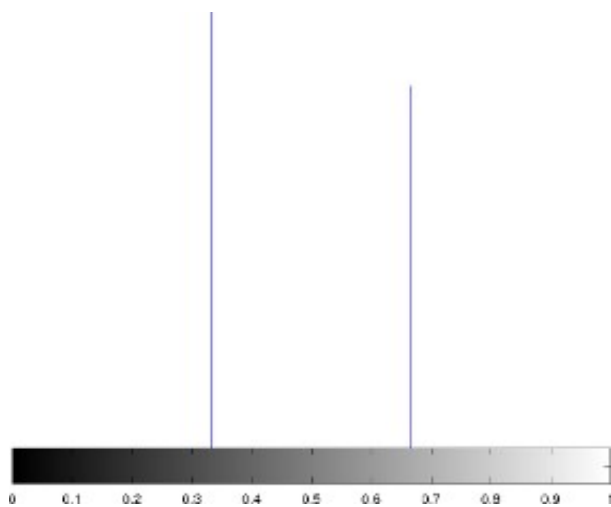
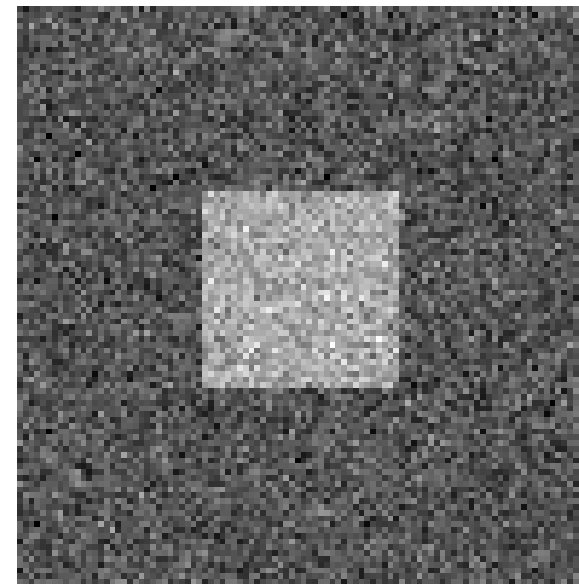
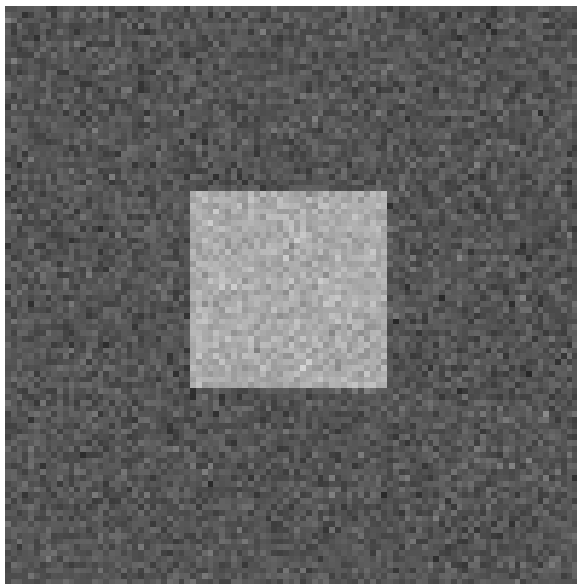
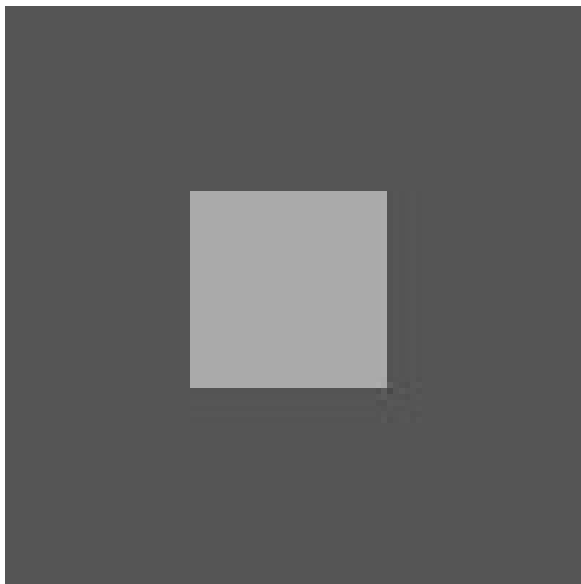
- In generale, potremmo avere più oggetti, di intensità diversa su sfondo scuro, da segmentare.
- In questo caso utilizzeremo più di una soglia.

$$J[x, y] = \begin{cases} a & \text{se } I[x, y] > T_2 \\ b & \text{se } T_1 < I[x, y] \leq T_2 \\ c & \text{se } I[x, y] \leq T_1 \end{cases}$$

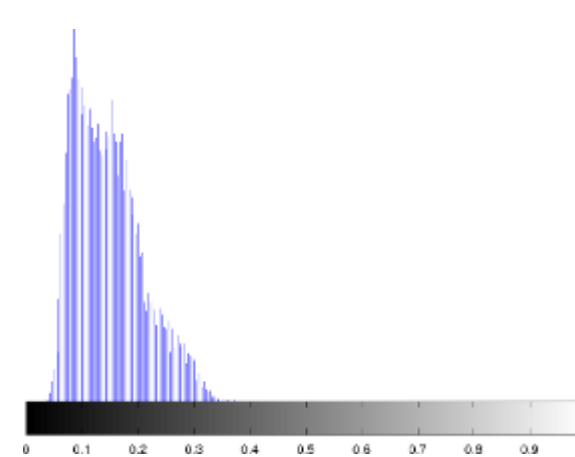
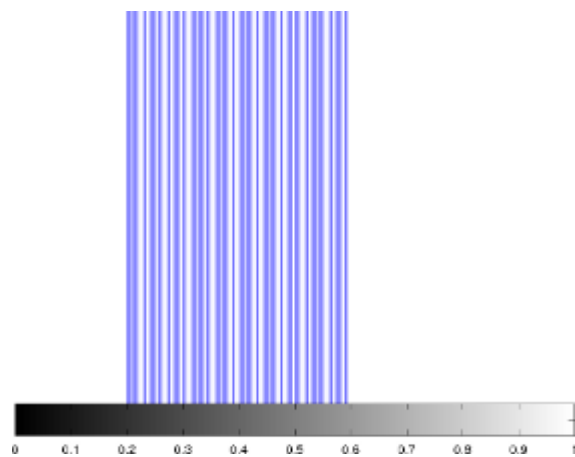
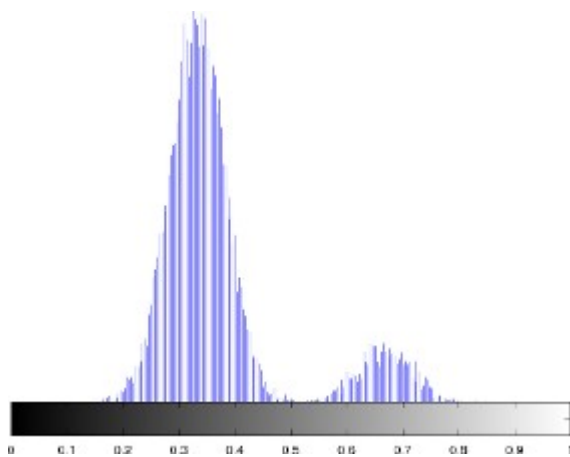
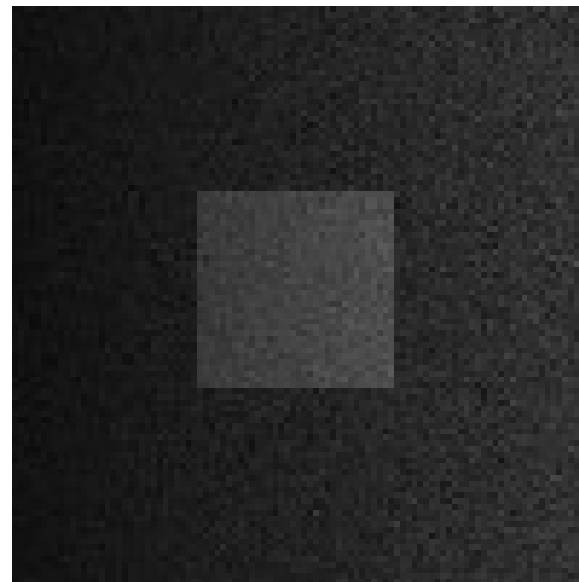
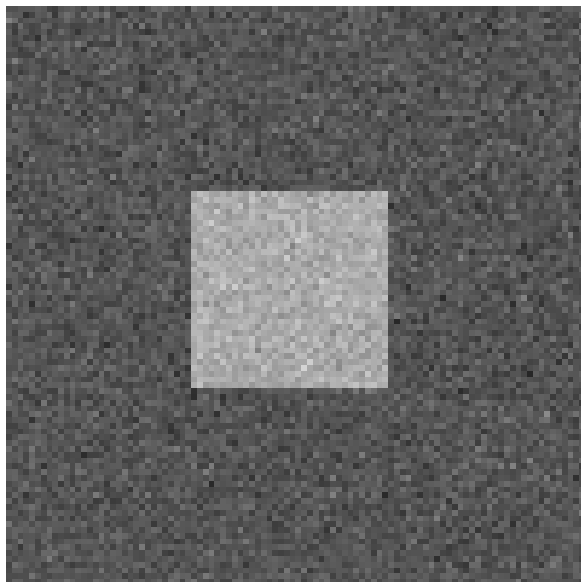
- Il successo del processo di sogliatura è direttamente legato alla profondità e lunghezza delle valli che separano le mode dell'istogramma.



Segmentazione e rumore



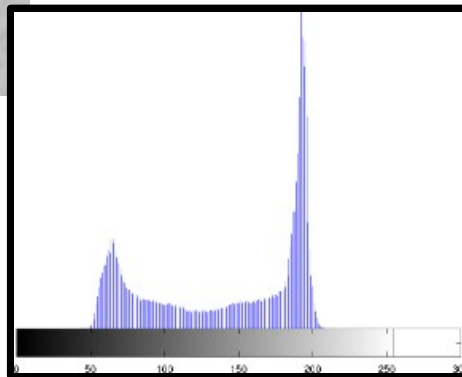
Segmentazione e illuminazione



Stima automatica della soglia

- Un algoritmo semplice di stima della soglia nel caso in cui abbiamo un problema di segmentazione in 2 regioni è il seguente:
 1. Scegliere una stima iniziale della soglia T
 2. Segmentare l'immagine utilizzando T . Otteniamo due gruppi di pixels: G_1 e G_2 .
 3. Calcolare le intensità medie m_1 e m_2 di G_1 e G_2 .
 4. Calcolare una nuova soglia come
$$T = \frac{1}{2}(m_1 + m_2)$$
 5. Ripetere 2-4 finché la soglia T in due iterazioni consecutive è inferiore ad una certa soglia.

Esempio



Metodo di Otsu (1979)

- La segmentazione può essere vista come un problema statistico di decisione, il cui obiettivo è di minimizzare l'errore commesso nell'assegnare pixels a 2 o più gruppi (dette classi).
- Il metodo di Otsu è un metodo ottimale nel senso che massimizza la varianza tra le classi.
- Utilizza esclusivamente computazioni basate su istogrammi.

Metodo di Otsu

- Sia I un'immagine a L livelli di grigio e X una v.a. che ritorna un pixel con distribuzione uniforme. La funzione di ripartizione di $I(X)$

$$F_k = F_{I(X)}(k) = \sum_{i=0}^k p_{I(X)}(i) = \sum_{i=0}^k \frac{n_i}{MN}$$

- Assumendo k la soglia abbiamo che le intensità di pixel appartenenti alla prima classe sono $0 \dots k$, mentre quelli della seconda sono $k+1 \dots L-1$
- Possiamo ora calcolare l'intensità media della prima e seconda classe assumendo k la soglia come

$$m_k = m_k^{(1)} = E[I(X) | I(X) \leq k] = \sum_{i=0}^k i \frac{p_{I(X)}(i)}{F_k}$$
$$m_k^{(2)} = E[I(X) | I(X) > k] = \sum_{i=k+1}^{L-1} i \frac{p_{I(X)}(i)}{1 - F_k}$$

Metodo di Otsu

- L'intensità media globale è data da:

$$m_G = E[I(X)] = m_{L-1} \quad m_G = F_k m_k^{(1)} + (1 - F_k) m_k^{(2)}$$

- La varianza globale è data da

$$\sigma_G^2 = Var[I(X)] = \sum_{i=0}^{L-1} (i - m_G)^2 p_{I(x)}(i)$$

- La varianza interclasse è data da

$$\begin{aligned} \sigma_B^2(k) &= F_k \left[m_k^{(1)} - m_G \right]^2 + (1 - F_k) \left[m_k^{(2)} - m_G \right]^2 \\ &= F_k (1 - F_k) \left[m_k^{(1)} - m_k^{(2)} \right]^2 \\ &= \frac{F_k}{1 - F_k} \left[m_G - m_k \right]^2 \end{aligned}$$

Metodo di Otsu

- Il valore di soglia ottimale è quello che massimizza la varianza interclasse

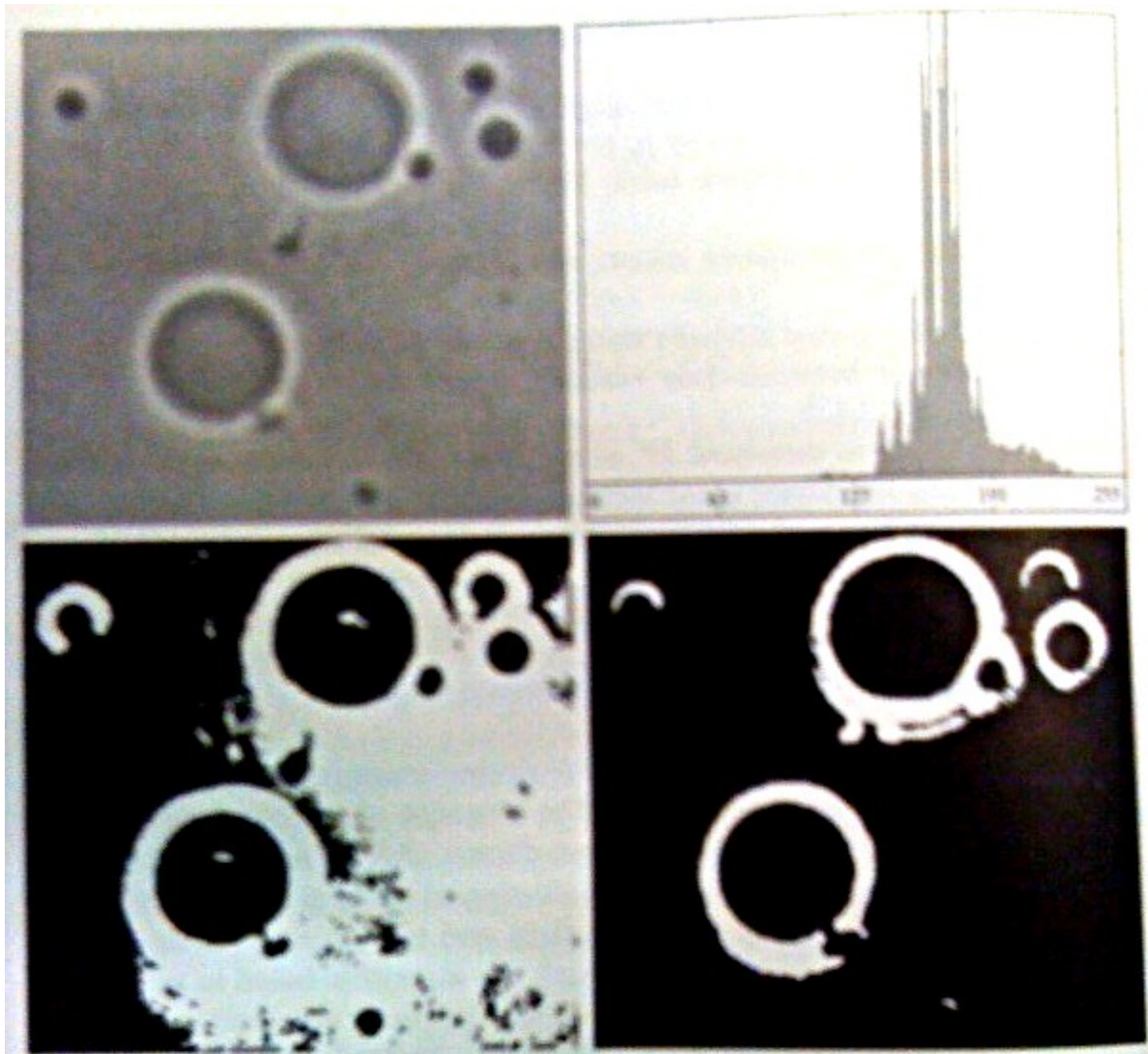
$$k^* = \arg \max_{k=0 \dots L-1} \sigma_B^2(k)$$

se il massimo non è unico si prende la media dei massimi trovati.

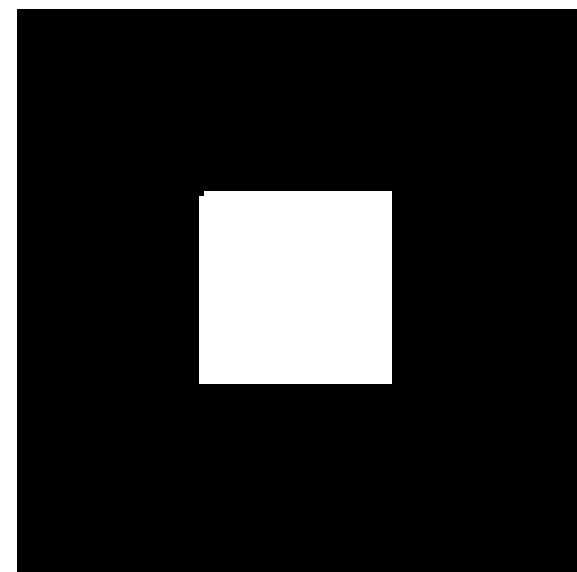
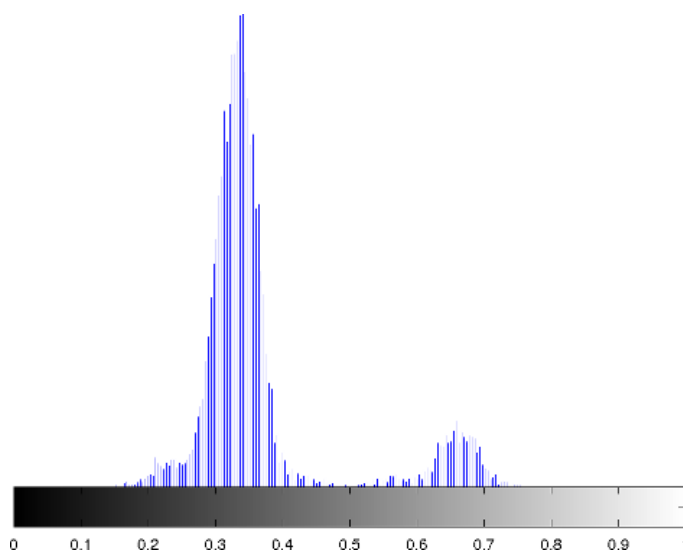
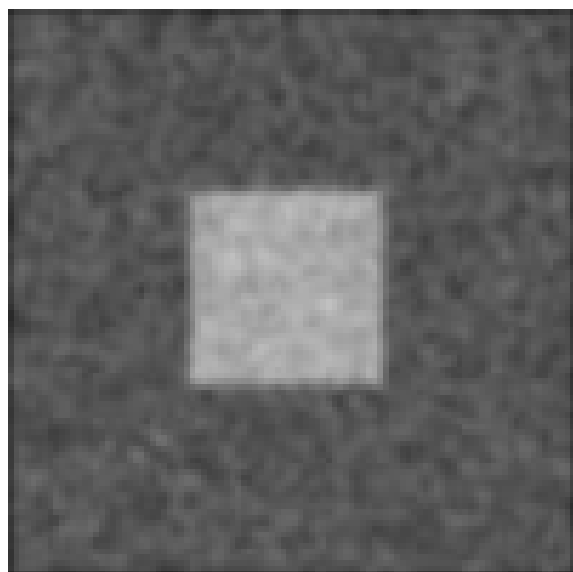
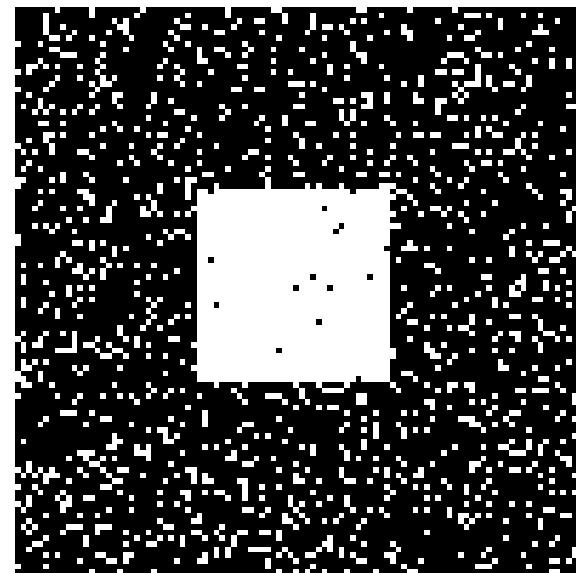
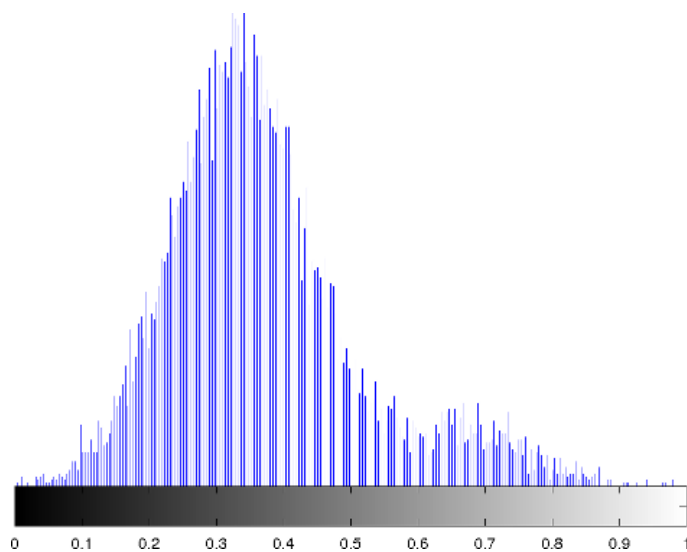
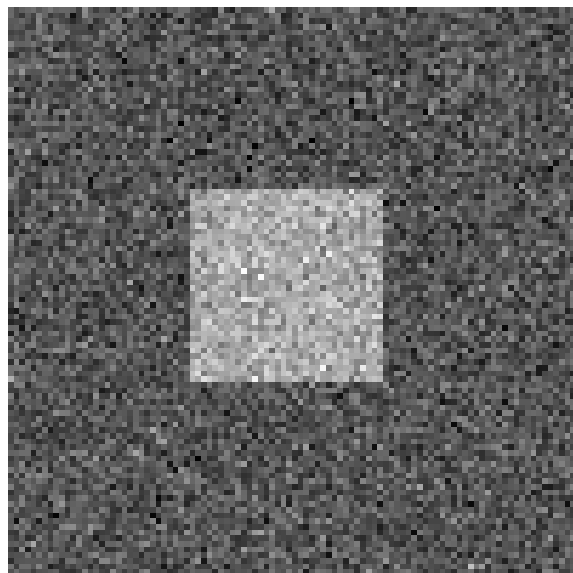
- Possiamo ottenere una misura di separabilità attraverso

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$$

Metodo di Otsu

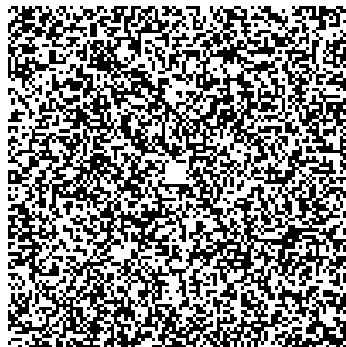
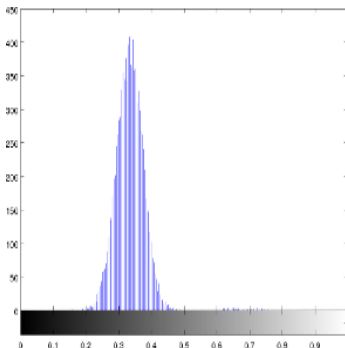
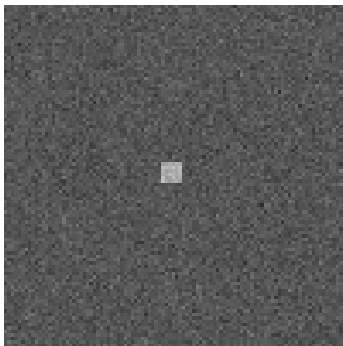


Smoothing per migliorare sogliaatura



Bordi per migliorare sogliatura

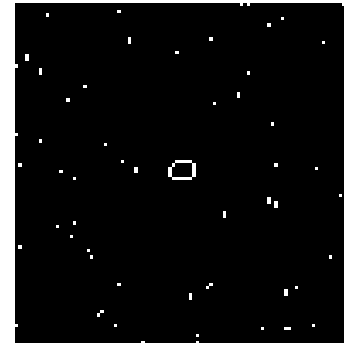
1. Utilizzare un algoritmo di rilevamento bordi (non deve essere necessariamente preciso).
2. Calcolare l'istogramma dell'immagine basandosi solo sulle intensità in corrispondenza di edges.
3. Usare un metodo di sogliatura automatica usando l'istogramma che ne risulta.



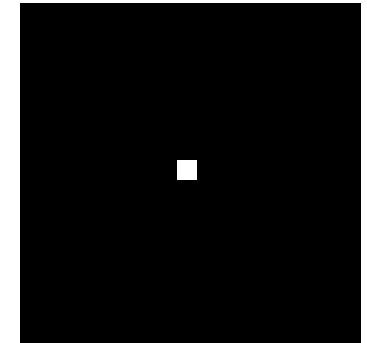
OTSU



OTSU+
smoothing



Filtro di
Sobel +
sogliatura



OTSU +
istogramma
basato su
bordi

Sogliatura multipla

- Il metodo di Otsu si generalizza al caso di classi multiple (più di 1 soglia).
- In pratica ha senso applicare questo approccio con al più 3 classi (2 soglie). In problemi in cui vi sono più classi, la sola informazione di intensità può essere insufficiente.
- Nel caso più generale si utilizzano algoritmi di riconoscimento di pattern.

Sogliatura adattiva

- La **sogliatura adattiva** consiste in soglie diverse applicate a diverse parti dell'immagine sulla base di una combinazione di informazioni locali e globali.
- Un metodo semplice di sogliatura consiste nel partizionare l'immagine ed applicare un metodo di sogliatura globale a ciascuna di queste parti.
- Un approccio più generale calcola per ogni pixel dell'immagine una soglia in funzione di proprietà calcolate localmente al pixel (intensità media/varianza nell'intorno del pixel) e globalmente (intensità media/varianza dell'immagine intera).

$$J[x, y] = \begin{cases} 1 & \text{se } I[x, y] \geq T_{xy} \\ 0 & \text{se } I[x, y] < T_{xy} \end{cases}$$

$$J[x, y] = \begin{cases} 1 & \text{se } Q(\text{parametri locali}) \\ 0 & \text{se } \neg Q(\text{parametri locali}) \end{cases}$$

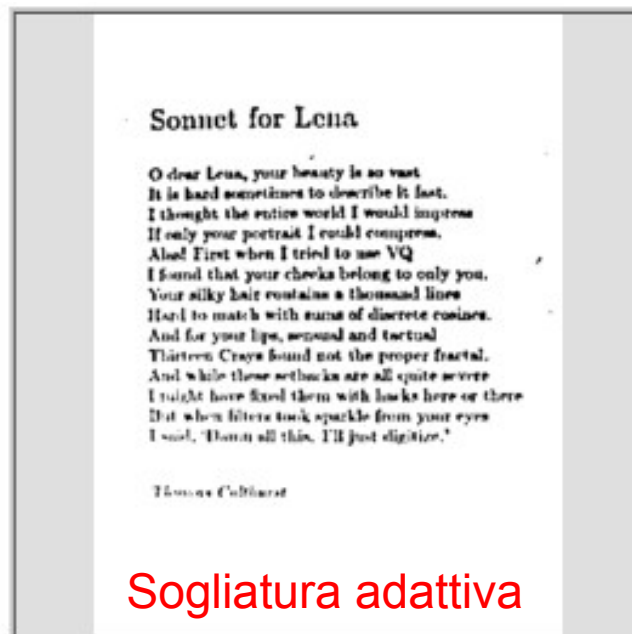
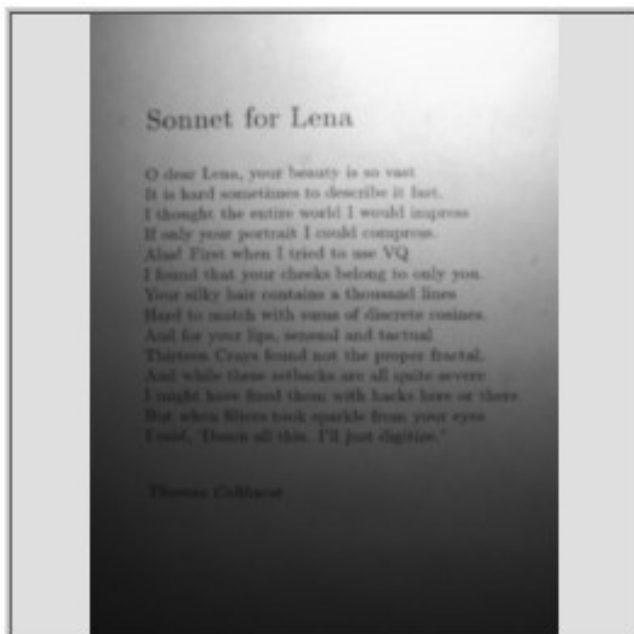
$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

oppure

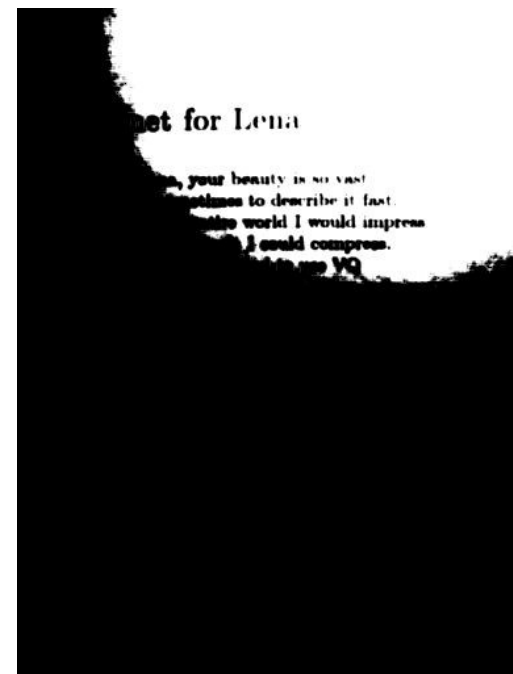
$$T_{xy} = a\sigma_{xy} + bm_G$$

Q : predicato

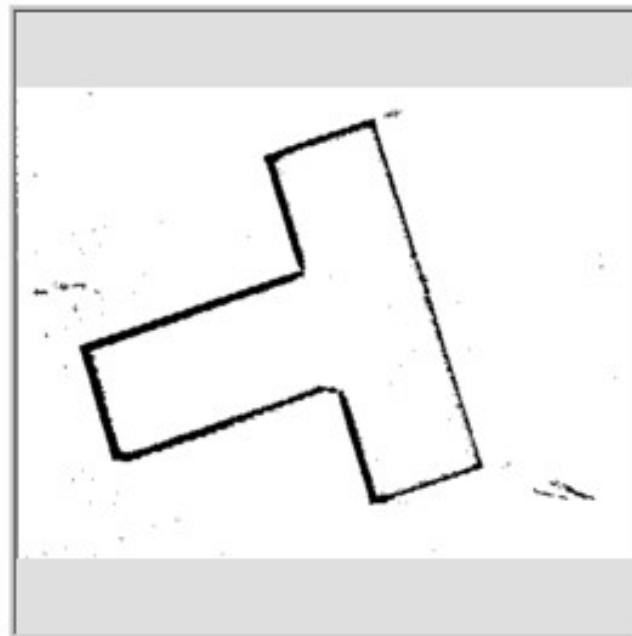
Sogliatura adattiva



Sogliatura adattiva



Sogliatura non
adattiva



Sogliatura multivariabile

- La **sogliatura multivariabile** consiste nel sogliare un'immagine basandosi su diverse variabili, e non solo l'intensità.
- Per esempio, se consideriamo immagini colorate abbiamo una variabile associata a ciascun canale di colore.
- Ad ogni pixel possiamo quindi associare un vettore \mathbf{z} n-dimensionale dove n è il numero di variabili in base alle quali vogliamo sogliare l'immagine.
- Un'immagine I diventa quindi un campo vettoriale dove $I[x,y]$ il vettore associato al pixel (x,y) .
- Possiamo sogliare un'immagine sulla base della distanza di ogni pixel da un vettore dato \mathbf{a} e un valore di soglia T .

$$J[x, y] = \begin{cases} 1 & \text{if } D(I[x, y], \mathbf{a}) < T \\ 0 & \text{altrimenti} \end{cases}$$

Crescita di regioni

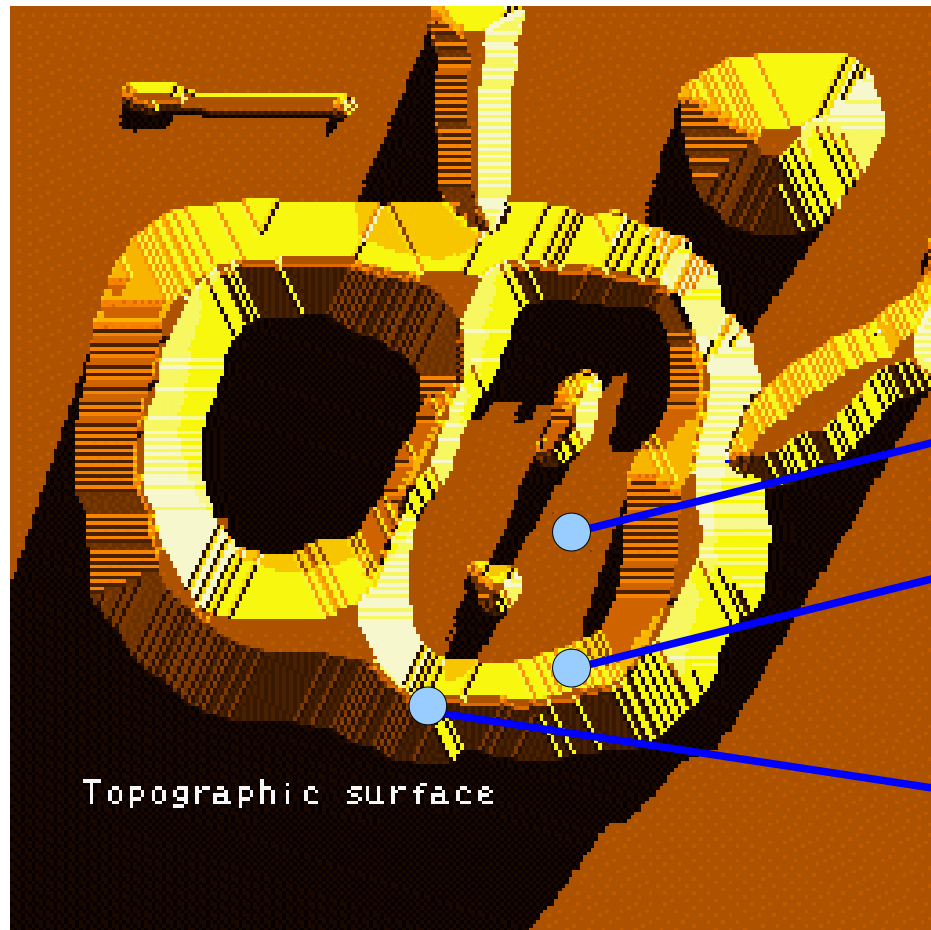
- La **crescita di regioni** è una procedura che raggruppa pixels o sottoregioni in regioni più grandi basandosi su criteri di crescita predefiniti.
- Il metodo parte da un insieme di “semi” e da questi le regioni crescono aggiungendo a ciascun seme pixels vicini che hanno proprietà simili al seme (es: colore simile).
- La scelta dei semi dipende dalla natura del problema. Se non ci sono informazioni a priori si utilizza un qualche algoritmo di clustering per trovare i primi semi
- La selezione del criterio di similarità non dipende solo dal problema ma anche dal tipo di immagine con cui si lavora.

Crescita di regioni

- Sia I un'immagine e S una matrice di semi della stessa dimensione di I con 1 in corrispondenza di un seme e 0 altrimenti. Sia inoltre Q un predicato da applicare ad ogni locazione (x,y) .
- Un algoritmo di crescita di regioni basato su 8-connettività tra i pixels è caratterizzato dai seguenti passi:
 1. Creare un'immagine I_Q con 1 in ogni locazione che soddisfa il predicato e 0 altrimenti.
 2. Aggiungere ad ogni seme in S tutti i pixels accesi e 8-connessi ad un pixel di seed.
 3. Ripetere 2 fino a stazionarietà.
 4. Etichettare ogni componente connessa con un valore differente.

Segmentazione spartiacque

- In questo metodo l'immagine viene vista come una superficie topografica in cui le intensità sono proporzionali all'altezza del terreno rispetto al livello del mare.
- Distinguiamo 3 tipi di punti: punti interni ad un bacino, punti lungo gli argini di un bacino, punti sulle cresta degli argini (linee spartiacque).



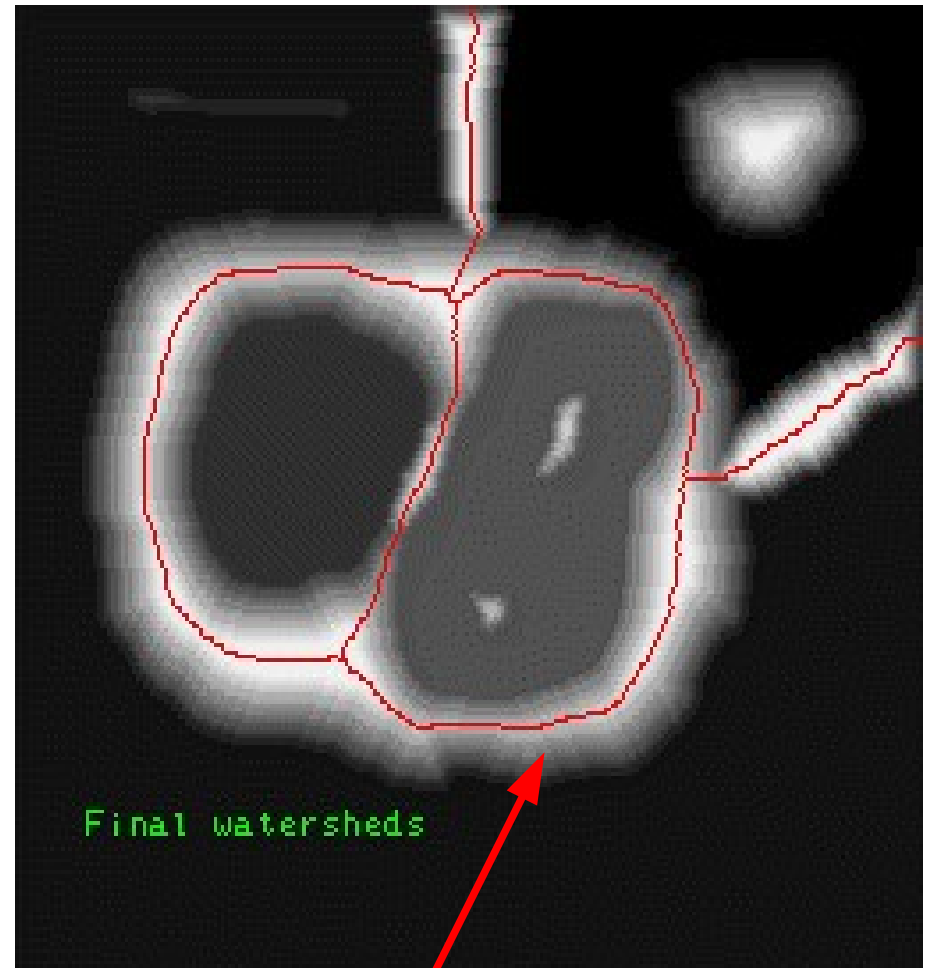
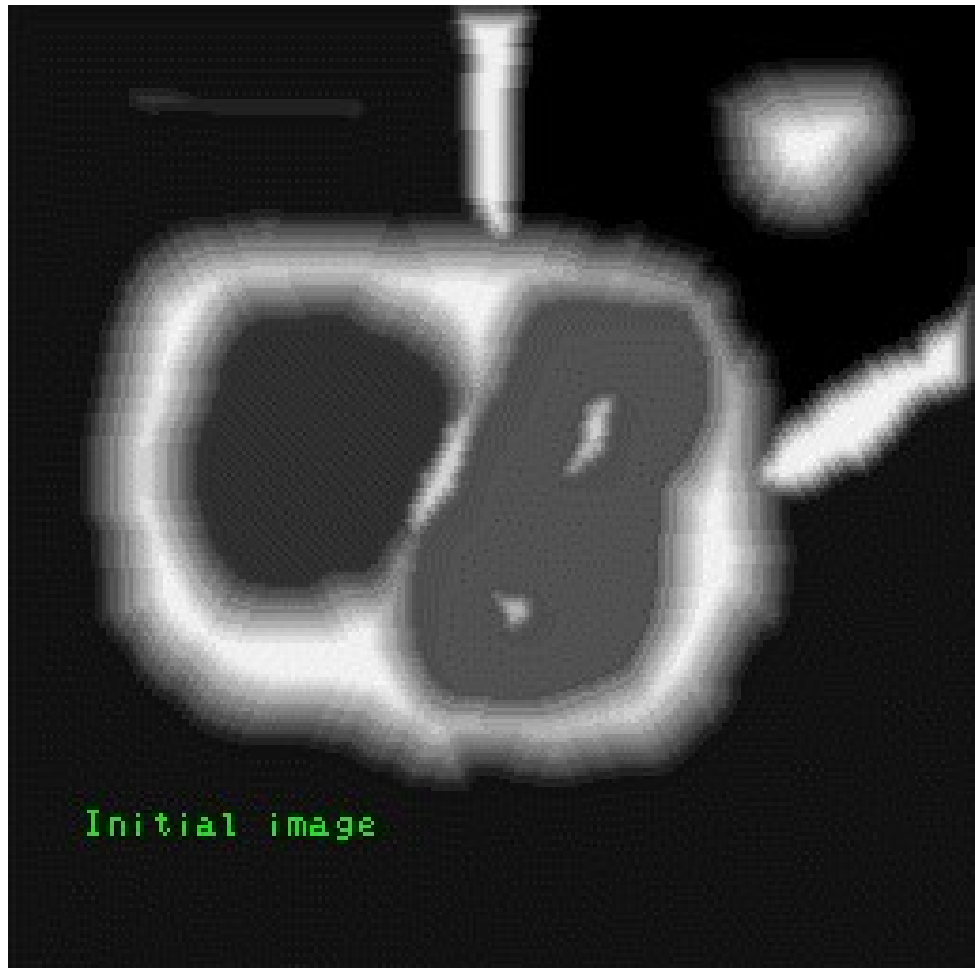
punto di
minimo
regionale

punto di
argine

punto
spartiacque

- L'obiettivo è quello di trovare le linee spartiacque.
- Immaginiamo di infiltrare acqua dai punti di minimo regionale ad una velocità uniforme.
- Quando due bacini stanno per fondersi, costruiamo una chiusa per mantenerli separati.
- Quando solo le chiusa sono visibili l'algoritmo è terminato e le chiusa rappresentano le linee spartiacque.

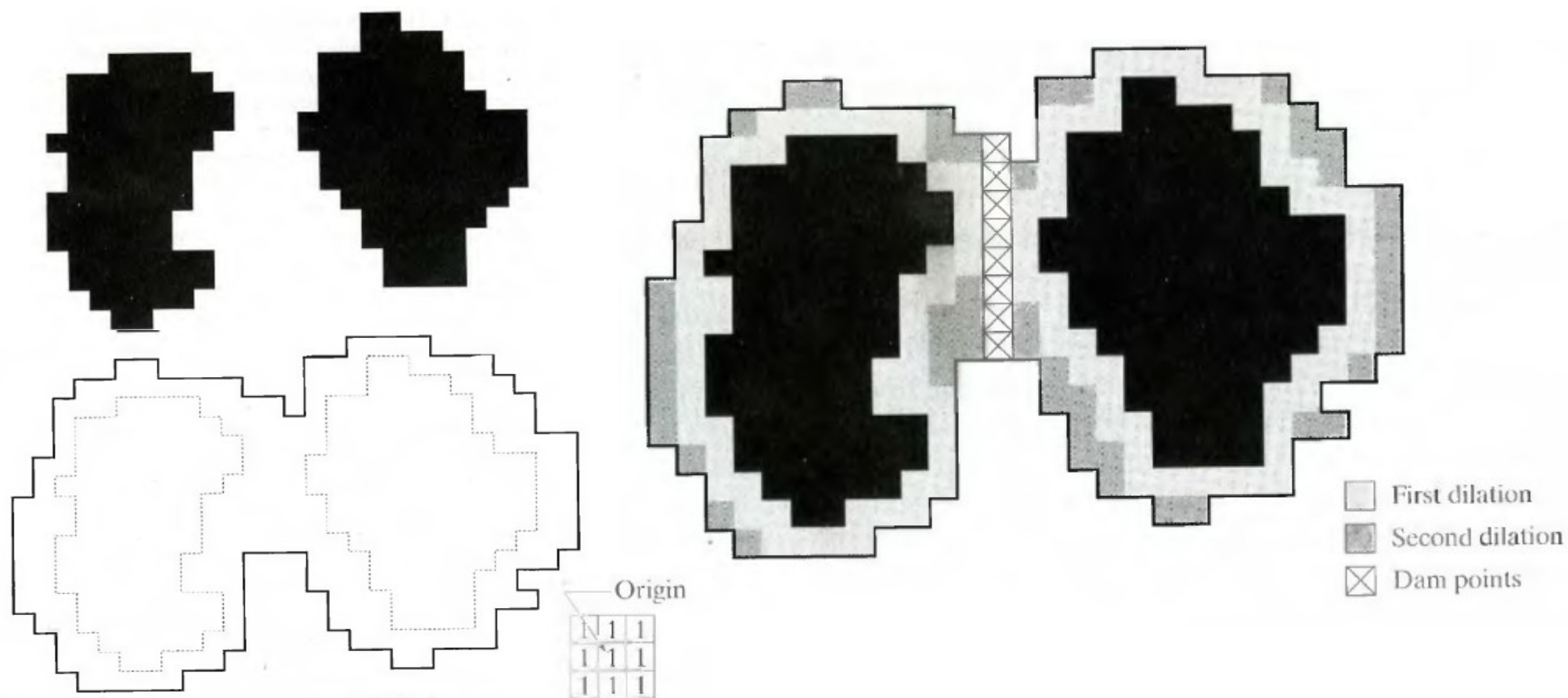
Segmentazione spartiacque



Le regioni che si ottengono hanno la proprietà di essere connesse

Costruzione di chiuse

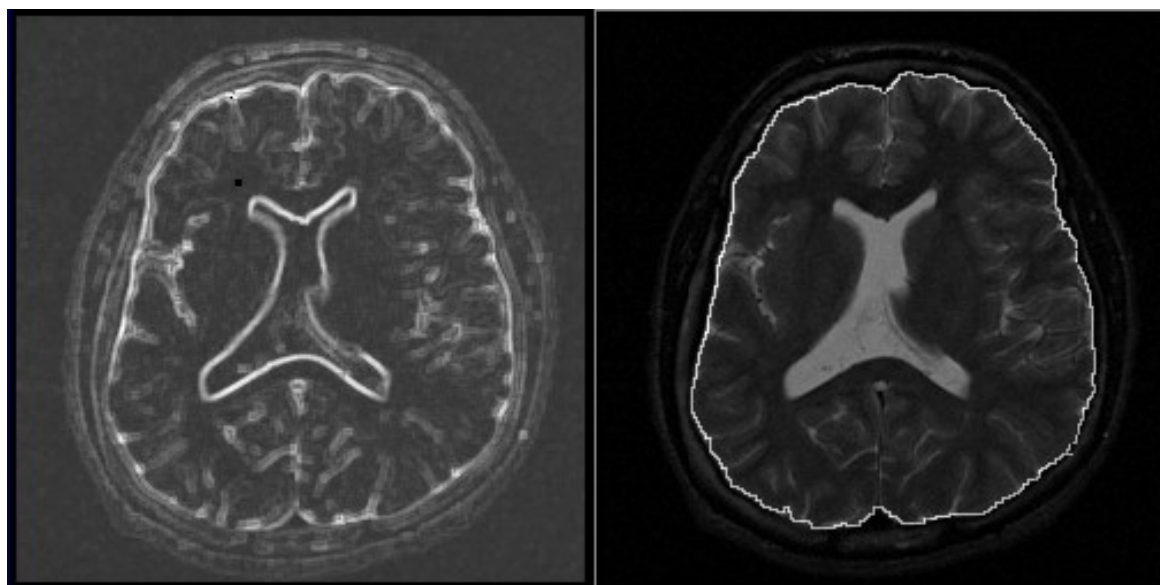
- La costruzione di chiuse può essere vista come un'operazione morfologica binaria basata sulla dilatazione.



Algoritmo di Meyer

1. Calcolare il gradiente dell'immagine G in cui l'intensità di un pixel è data dalla lunghezza del gradiente e creare un'immagine ausiliaria C di uguale dimensione, inizializzata a 0.
2. Dato un'insieme di markers (estratti automaticamente o forniti da un utente), dare un'etichetta (un valore intero) diversa ad ogni componente connessa di questo insieme. Scrivere in C i markers etichettati con il valore della componente connessa a cui appartengono.
3. Inserire i vicini a 0 di ogni pixel in C in una coda a priorità, con livello di priorità basato sull'intensità del pixel in G .
4. Se la coda non è vuota, estrarre il pixel p con priorità più alta, altrimenti termina.
5. Se tutti i vicini di p in C non nulli hanno la stessa etichetta, assegnare questo valore anche a p , altrimenti assegnare un'etichetta speciale che rappresenta un pixel sulla linea spartiacque.
6. Inserire in coda tutti i vicini di p non etichettati e tornare a 4.
7. Alla fine in C sono segnati tutti i pixels spartiacque, inoltre troviamo tutte le regioni etichettate in modo diverso.

Esempi: segmentazione interattiva



Segmentazione con alg. clustering

- La segmentazione è un problema di clustering in cui gli oggetti sono elementi legati ad un'immagine (pixels, linee, textons, ...)



Clustering agglomerativo e divisivo

- **Clustering agglomerativo:**

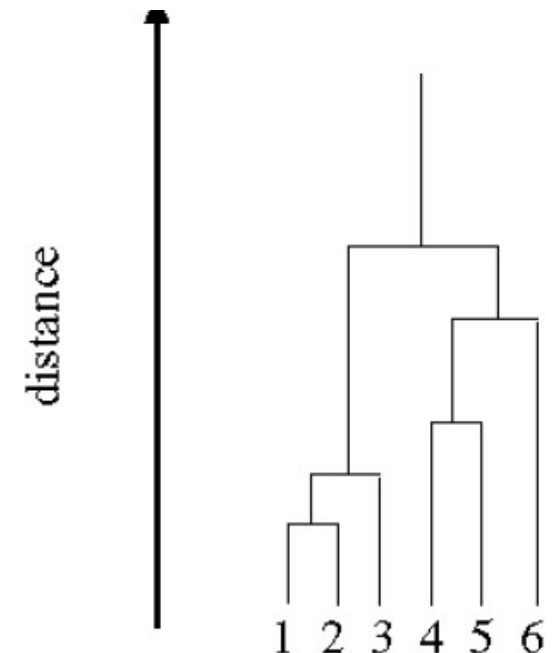
1. Ogni punti è un cluster separato
2. Fondere ripetutamente i due clusters con distanza minore fermando quando si è raggiunto un numero desiderato di cluster.

- **Clustering divisivo:**

1. Tutta l'immagine è un cluster
2. Spezzare ripetutamente in due il cluster che è meno coerente.

- Domande a cui rispondere:

- quando fermarsi?
- che distanza tra clusters? o misura di coerenza?



Clustering con K-means

- Consideriamo il caso di elementi immersi in uno spazio vettoriale (ogni elemento è un vettore).
- Dato un cluster di elementi, possiamo calcolare un centroide per il cluster prendendo la media dei vettori che lo compongono.
- Sia K un numero desiderato di clusters.
- Il problema del clustering per K-means è quello di trovare un'assegnamento di oggetti a K clusters in modo tale da minimizzare la somma delle distanze di ciascun oggetto dal centroide del proprio cluster (costo di assegnamento).

$$\mathcal{E}(\{C_i\}, \{\mathbf{x}_i\}) = \sum_{i=1}^K \sum_{j \in C_i} d(\mathbf{x}_j, \mathbf{c}_i)$$

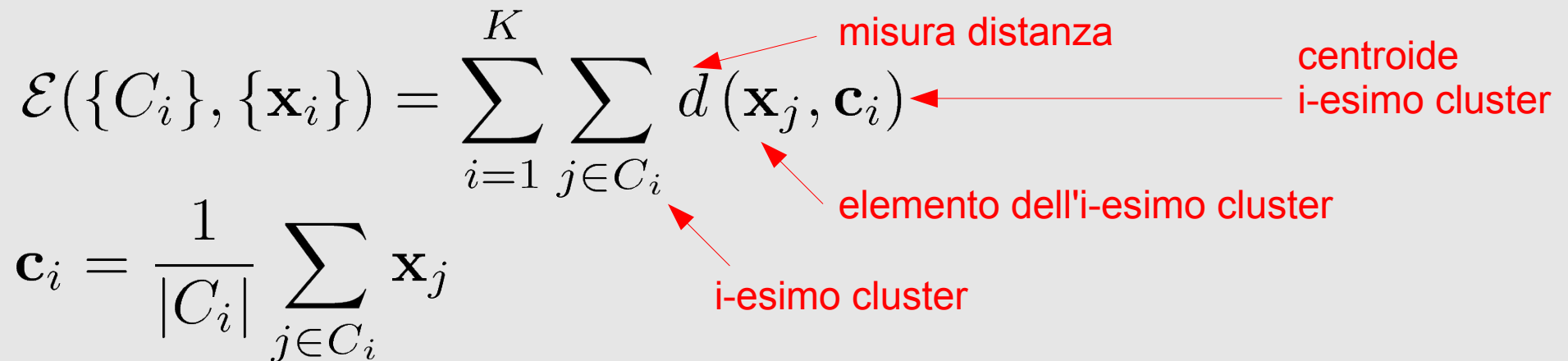
misura distanza

centroide
i-esimo cluster

$$\mathbf{c}_i = \frac{1}{|C_i|} \sum_{j \in C_i} \mathbf{x}_j$$

elemento dell'i-esimo cluster

i-esimo cluster



Algoritmo K-means

- Inizializziamo K clusters in modo random e iteriamo ripetutamente un processo a 2 fasi finché l'assegnamento non rimane invariato:

1. aggiornamento centroidi

$$\mathbf{c}_i^{(n)} = \frac{1}{|C_i^{(n)}|} \sum_{j \in C_i^{(n)}} \mathbf{x}_j$$

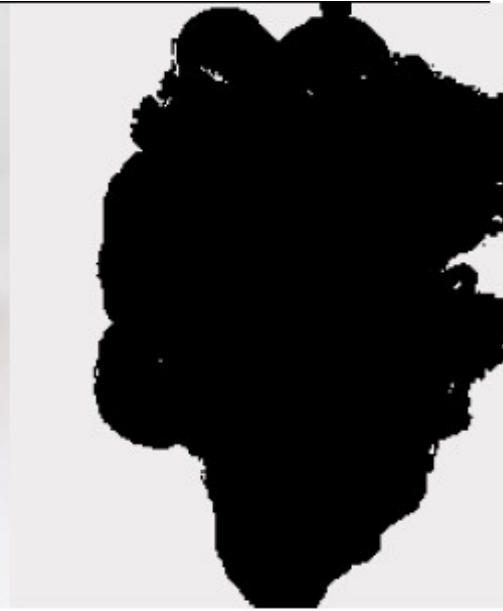
2. aggiornamento assegnamento

$$C_i^{(n+1)} = \left\{ j : i \in \arg \min_{k=1 \dots K} d(\mathbf{x}_j, \mathbf{c}_k^{(n)}) \right\}$$

- Questo processo minimizza localmente la funzione di costo

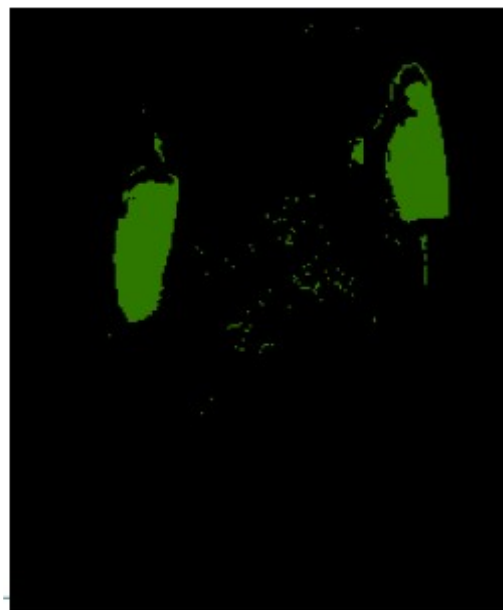
$$\mathcal{E}(\{C_i\}, \{\mathbf{x}_i\}) = \sum_{i=1}^K \sum_{j \in C_i} d(\mathbf{x}_j, \mathbf{c}_i)$$

Algoritmo K-means



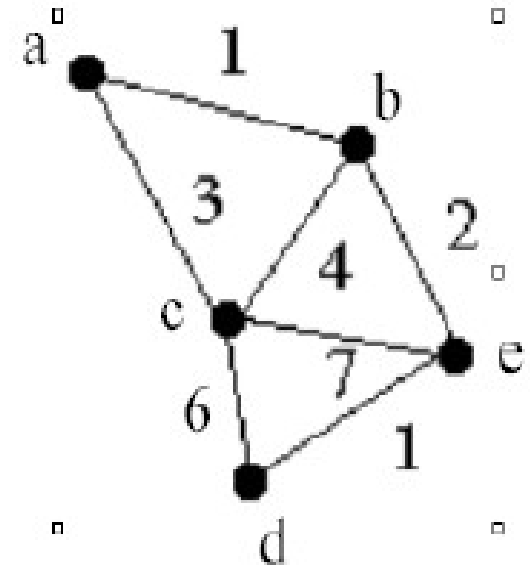
K=11

L'informazione di solo colore non è sufficiente per ottenere segmenti salienti.



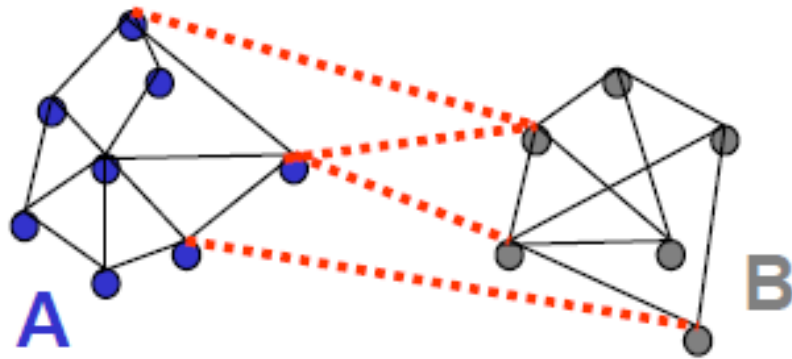
Clustering basato su teoria dei grafi

- Astrarre gli elementi dell'immagine come i nodi di un grafo e le similarità tra elementi come archi pesati.



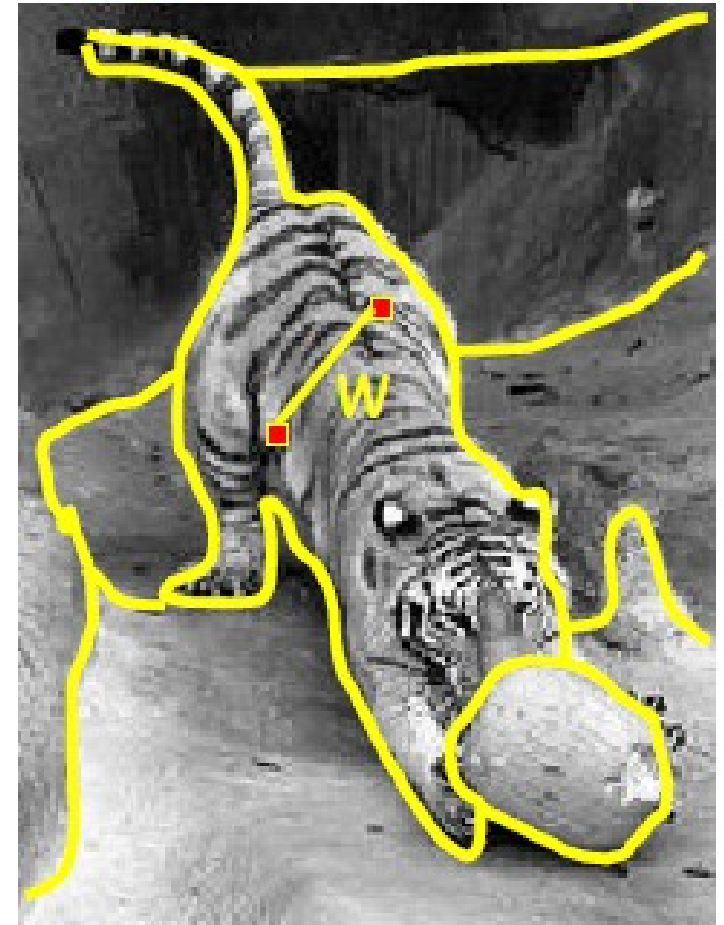
Segmentazione con tagli del grafo

- Possiamo segmentare un'immagine tagliando opportunamente il grafo che la astrae.



costo del taglio

$$cut(A, B) = \sum_{i \in A} \sum_{j \in B} w_{ij}$$



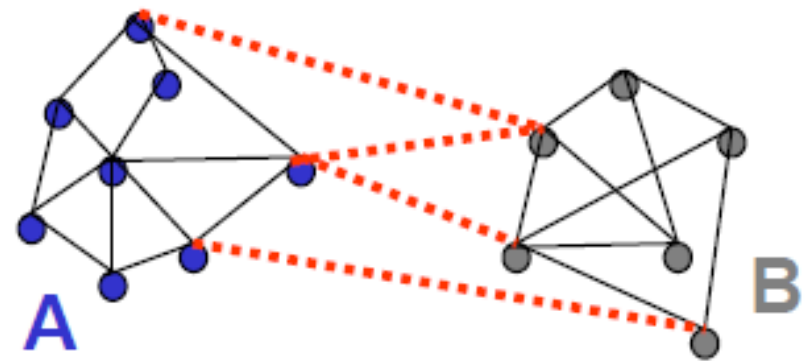
Segmentazione con taglio minimo

- La segmentazione con taglio minimo calcola ripetutamente il taglio di costo minimo sul grafo
- Sia $G=(V,E,w)$ un grafo pesato sugli archi.

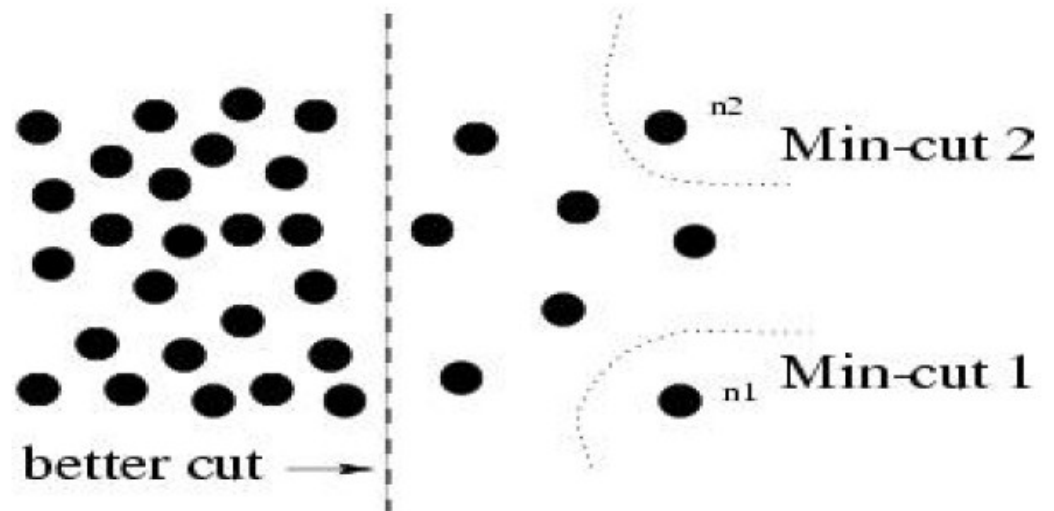
$$(V_A, V_B) = \arg \min_{V_A, V_B} \text{cut}(V_A, V_B)$$

$$V_A \cap V_B = \emptyset$$

$$V_A \cup V_B = V$$



- Riapplicare il procedimento ai sottografi individuati dal taglio. Finché non si ha raggiunto un numero di clusters desiderato.
- Problema, spesso segmenti piccoli !



Segmentazione con taglio normalizzato

- Un taglio minimo penalizza segmenti grandi.
- Si risolve il problema cercando il taglio minimo normalizzato con la dimensione dei segmenti.
- Sia $G=(V,E,w)$ un grafo pesato sugli archi.

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

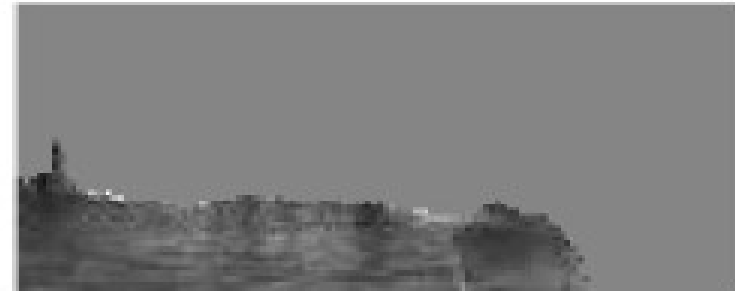
$$assoc(A, V) = \sum_{i \in A} \sum_{j \in V} w_{ij}$$

- Il problema è NP-Hard ma una buona approssimazione si ottiene prendendo l'autovettore \mathbf{x} con secondo autovalore (generalizzato) più piccolo soluzione di

$$(D - W)\mathbf{x} = \lambda D\mathbf{x} \quad D = (d_{ij}) = \begin{cases} \sum_j w_{ij} & i = j \\ 0 & i \neq j \end{cases}$$

- Estraiamo A e B sogliando opportunamente \mathbf{x} .

Segmentazione con taglio normalizzato



Trasformata di Hough

- Consideriamo il problema di trovare delle linee a partire da un insieme di punti ottenuto attraverso un algoritmo di rilevamento bordi.
- Una linea è tipicamente caratterizzata da 2 parametri nella forma esplicita:

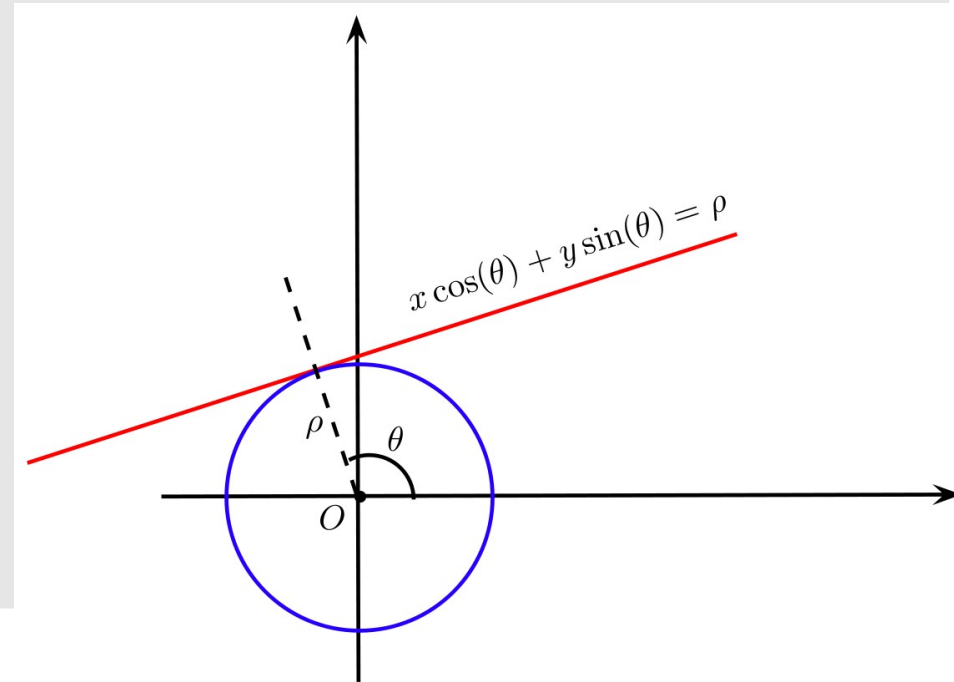
$$y = mx + q$$

- Un problema legato a questa parametrizzazione sono le linee verticali, in cui $m = \infty$. Per ovviare al problema passiamo ad una rappresentazione polare:

$$x \cos(\theta) + y \sin(\theta) = \rho$$

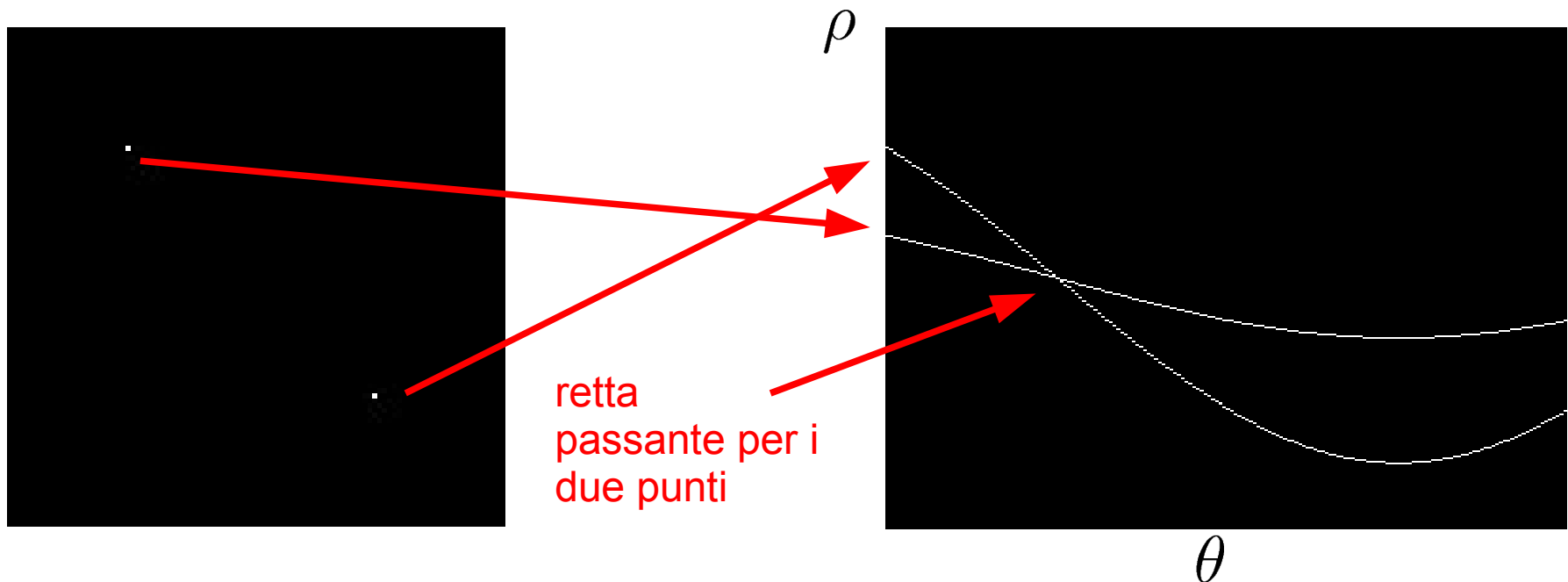
dove $\theta \in [0, 2\pi]$ e $\rho \geq 0$.

- Un'altra parametrizzazione polare valida è $\theta \in [-\pi/2, \pi/2]$ e ρ reale.



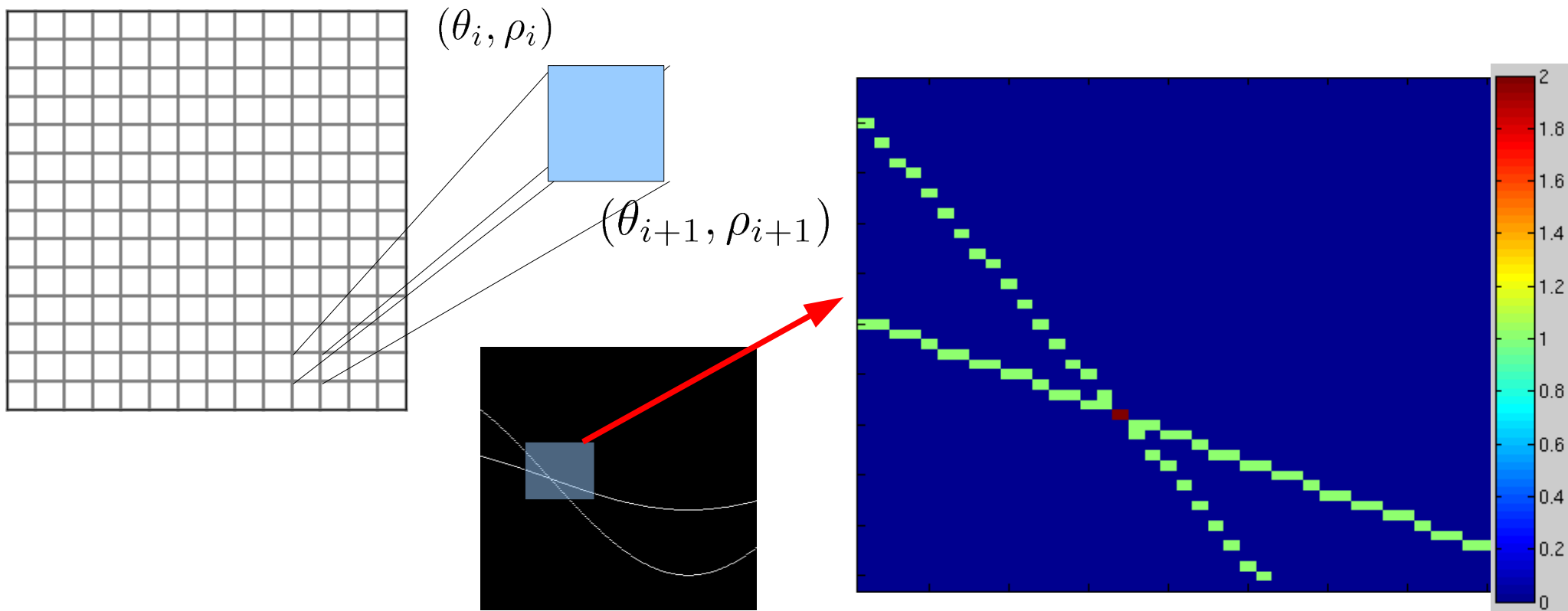
Trasformata di Hough

- La trasformata di Hough consiste in una trasformazione di punti nello spazio immagine xy in uno spazio dei parametri $\theta\rho$.
 - Ogni punto nello spazio xy corrisponde ad una sinusoidale nello spazio $\theta\rho$.
- $$\rho(\theta) = x_0 \cos(\theta) + y_0 \sin(\theta)$$
- Ogni linea nello spazio xy corrisponde ad un punto nello spazio $\theta\rho$.
 - Dati due punti, la retta che passa per i due punti è rappresentata nello spazio $\theta\rho$ dall'intersezione delle due sinusoidi prodotte dai due punti.



Trasformata di Hough

- Lo spazio $\theta\rho$ è continuo e deve essere trasformato in uno spazio discreto mediante opportuna quantizzazione dei valori di θ e ρ .
- Ogni cella dello spazio discretizzato è un accumulatore che conta il numero di sinusoidi che passano attraverso la cella.



Trasformata di Hough - algoritmo

1. Scegliere una discretizzazione del piano $\theta\rho$, creare il piano inizializzando ogni cella a 0.
2. Ottenere un insieme di punti dell'immagine attraverso un rilevatore di bordi.
3. Per ogni punto trovato
 - a) incrementare di 1 il valore delle celle attraversate dalla sinusoide associata al punto.
4. Cercare i valori di massimo tra le celle del piano $\theta\rho$ per individuare le potenziali linee.
5. Fare un'analisi più fine per determinare gli estremi della linea e spezzarle se presentano discontinuità troppo lunghe.

Esempio

