

# Image Segmentation

# Image Segmentation

- Suddivisione di un immagine in sezioni
  - La suddivisione può avvenire secondo diversi criteri
  - La segmentazione è un esempio di applicazione dell'*image processing* volta ad estrarre qualche attributo, caratteristiche di regioni o pattern presenti nell'immagine

# Image Segmentation

- Esempi di applicazione
  - Identificazione di strutture di organizzazione dell'immagine
    - line, punti o altre forme geometriche
  - Classificazione di dettagli dell'immagine raffiguranti elementi
  - Classificazione di *pattern* presenti nell'immagine
  - Verranno presentati alcuni metodi per la segmentazione tra i più usati
    - In generale il tipo e la complessità della segmentazione dipende dagli specifici obiettivi applicativi

# Image Segmentation

- 1) Insieme di regioni  $R_1, R_2, \dots, R_n$
- 2) L'immagine è l'unione delle regioni su cui è suddivisa  $\cup_{i=1}^n R_i = R$
- 3) Le regioni di suddivisione non si intersecano  $R_i \cap R_j = \emptyset$
- 4) Una regione è definita da un *predicato* logico. Il test di questo predicato deve essere vero su tutti pixel di una regione
$$Q(R_i) = TRUE$$
- 5) Il medesimo predicato è falso su tutti i pixel che costituiscono l'unione di 2 regioni  $R_i$  e  $R_j$  se queste sono adiacenti
$$Q(R_i \cup R_j) = FALSE$$

# Image Segmentation

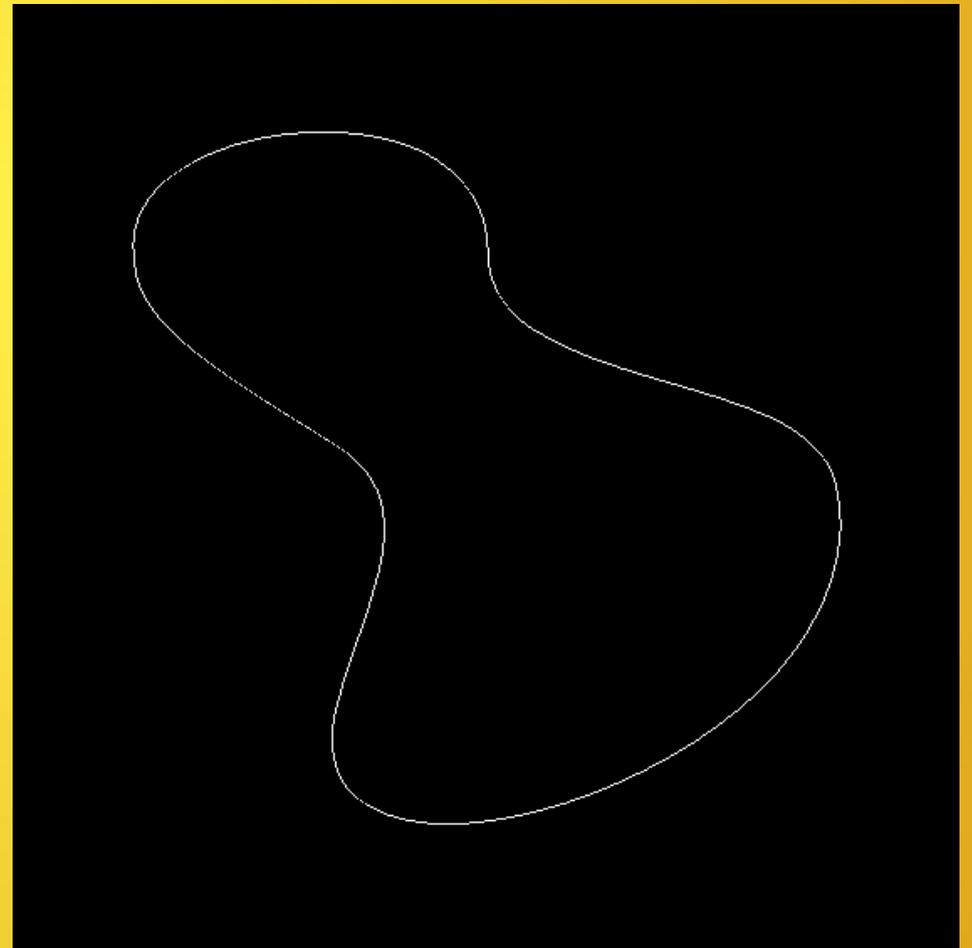
- Il risultato finale dell'operazione di segmentazione sarà un'immagine binaria, un'immagine-mappa che cataloga ogni singolo pixel in base alla condizione
- Nei casi specifici che vedremo questo equivale a chiedere se un pixel appartiene allo sfondo di un immagine (in generale una regione non di interesse) o appartiene ad oggetti contenuti in essa
- L'analisi della mappa binaria permette di raccogliere informazioni quantitative sulle regioni identificate

# Image Segmentation

- Edge based segmentation
  - L'immagine è suddivisa sulla base dei bordi di oggetti
  - I bordi vengono identificati dallo studio delle variazioni di luminosità di un'immagine grayscale
  - I bordi sono i *confini* tra le regioni della segmentazione
  - Con questa tecnica le regioni vengono quindi divise e catalogate in base ad una *discontinuità* che le separa
  - I pixel sono catalogati in base alla loro posizione rispetto alle linee di discontinuità costruite

# Edge Based Segmentation

Esempio: identificazione della regione interna più chiara



# Edge Based Segmentation

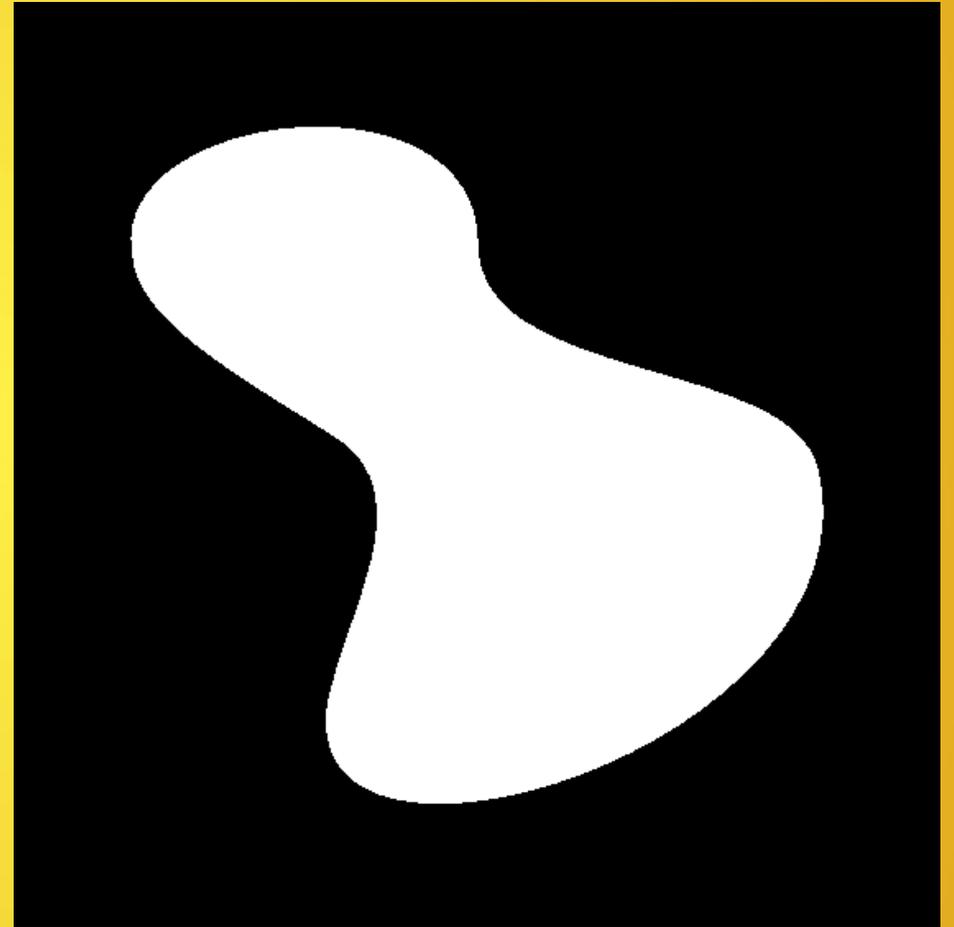
Una volta determinato il bordo della regione ad ogni pixel deve essere applicato un test (il predicato  $Q$  che deve stabilire l'appartenza ad una regione) che deve stabilire se appartiene all'interno o all'esterno

Si noti che la determinazione del confine, anche in questo caso apparentemente semplice, non è perfetta e ci sono alcune interruzioni

Matlab e Octave offrono alcune funzioni che possono determinare con buona approssimazione la regione interna completando il perimetro

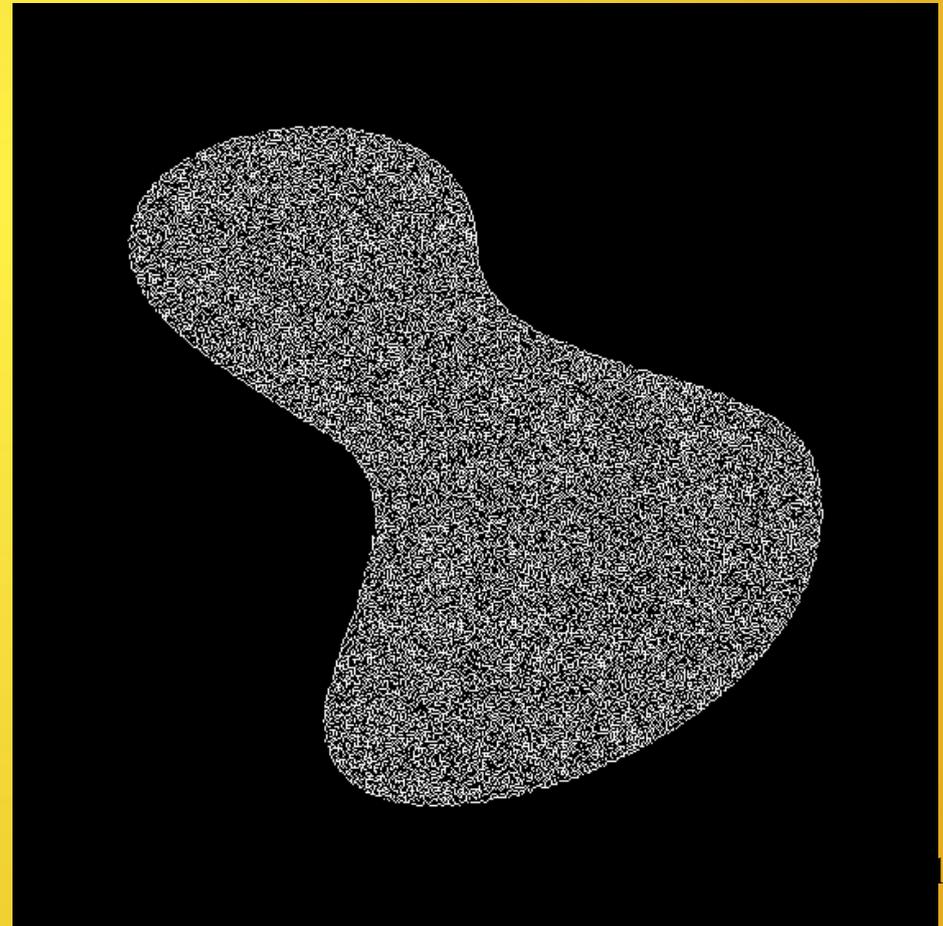
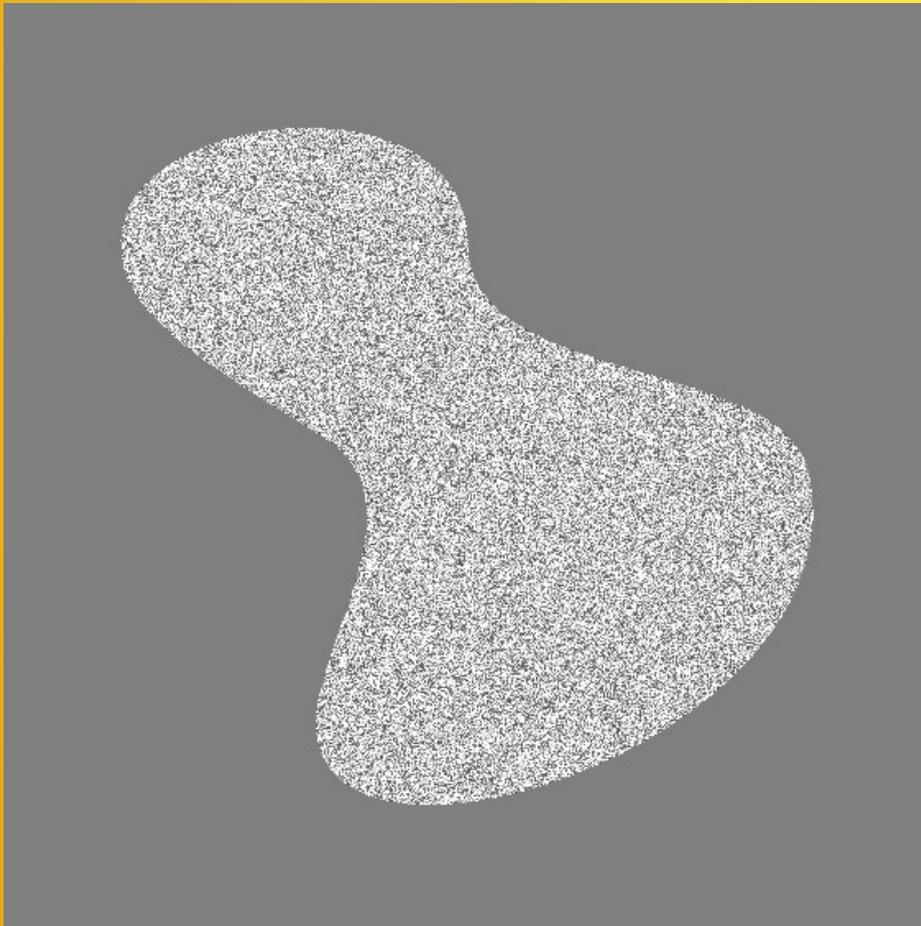
Nella slide successiva l'immagine di prova viene confrontata con la mappa binaria che identifica l'interno

# Edge Based Segmentation



# Edge Based Segmentation

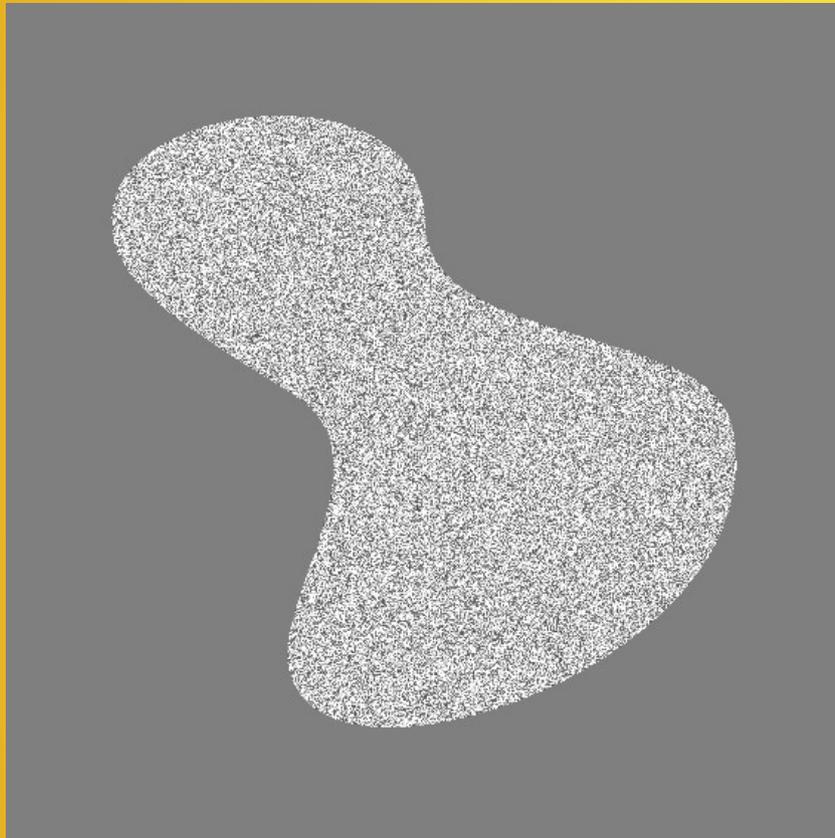
Questa tecnica può essere molto sensibile a immagini con bassa omogeneità e rapide variazioni che potrebbero confondere la determinazione del comportamento regolare del bordo della regione



# Image Segmentation

- Region based segmentation
  - Le regioni sono identificate per criteri di *uniformità* all'interno di esse
  - Possibili criteri possono essere basati su
    - Intervalli di intensità luminosa
    - Colore: regioni dello spazio RGB
    - Pattern: analisi specializzate su determinati pattern.

# Region Based Segmentation



- la zona centrale è contraddistinta da variabilità
- Si calcola la media degli scarti quadratici (varianza) su regioni quadrate 9x9
- La zona centrale è determinata con una soglia sul valore della varianza

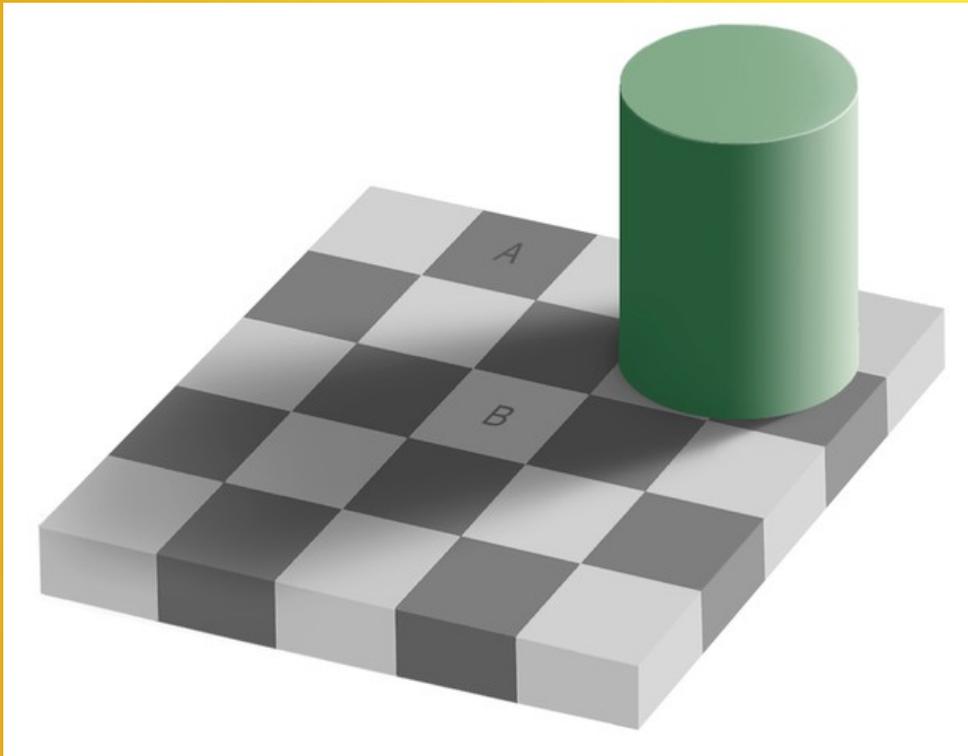
# Image Segmentation

- Thresholding
  - Selezione di regioni di un immagine in base ad un valore di soglia di luminosità (*threshold*)
    - Si controlla quindi che il valore di un pixel sia o meno maggiore di un valore stabilito per assegnarlo ad una delle regioni di segmentazione
    - In termini formali la segmentazione viene fatta con un criterio di confronto  $Q(R_i): p_{uv} > s$ , dove 's' è un valore soglia e u,v sono coordinate dei pixel della regione  $R_i$
  - Composizione di una immagine binaria: il confronto con la soglia ha come risultato un numero booleano, quindi vero o falso. Sulla base di questo un pixel viene catalogato all'interno di una regione
  - Gli esempi di thresholding che vedremo assegnano i pixel a 2 regioni: lo *sfondo* (background) oppure ad uno o più *oggetti* nel campo di osservazione (foreground)

# Problemi nella determinazione della Soglia

- Thresholding
  - Somiglianza tra zone appartenenti alle 2 regioni di classificazione
  - Problemi:
    - Variabilità di illuminazione delle varie parti del campo di osservazione
    - Rumore luminoso di vario tipo
    - Queste alterazioni rendono in generale difficile la determinazione di una soglia globale
    - Il problema della determinazione di una soglia globale talvolta non è evidente.

# Checker Shadow Illusion



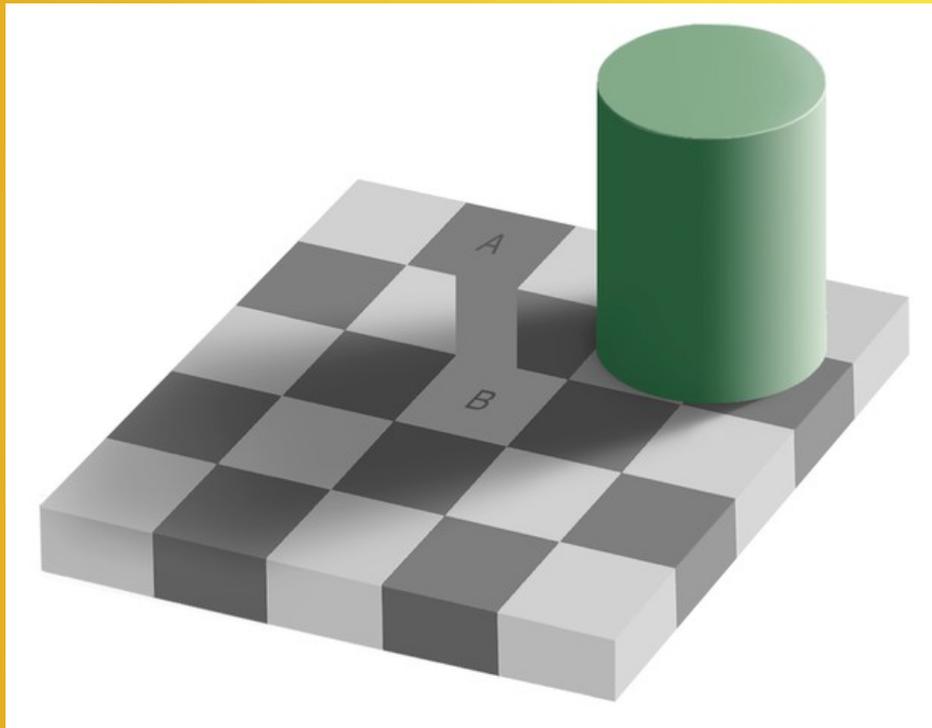
L'immagine a fianco è una rappresentazione realistica di un piano a scacchi chiari e scuri alternati. Ci sono caselle della scacchiera che hanno una riflettività intrinseca più alta (quindi appaiono in generale più chiare) alternate a caselle con riflettività più bassa

Il cilindro rende più scure tutte le caselle che stanno all'interno della sua ombra. Questo crea un'ambiguità alla nostra percezione

Possiamo concordare che la casella B sia intrinsecamente più chiara della casella A, ma essendo nell'ombra del cilindro continua ad apparire più chiara? Potendo esprimere la luminosità apparente con i valori assegnati ai pixel, quanto appare più chiara?

La risposta nella slide successiva

# Checker Shadow Illusion



La casa B e la casa A hanno una luminosità identica. Tuttavia continuiamo a vedere A più scura di B

La segmentazione di un'immagine simile, avente un'illuminazione variabile, fallirebbe con una singola soglia globale.

[https://en.wikipedia.org/wiki/Checker\\_shadow\\_illusion](https://en.wikipedia.org/wiki/Checker_shadow_illusion)

# Determinazione della Soglia

La distribuzione dei valori di intensità (istogramma) è uno strumento di analisi che ci aiuta a capire il problema della determinazione della soglia.

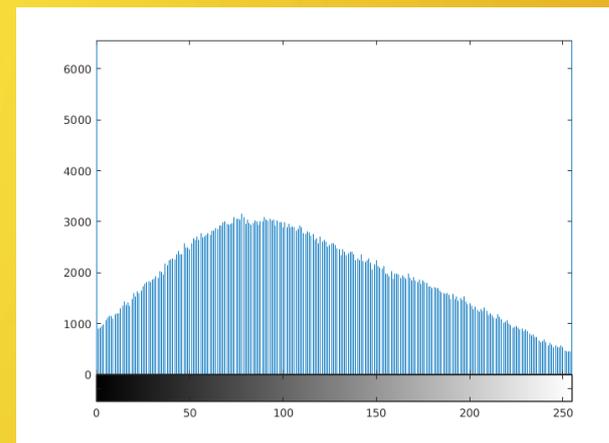
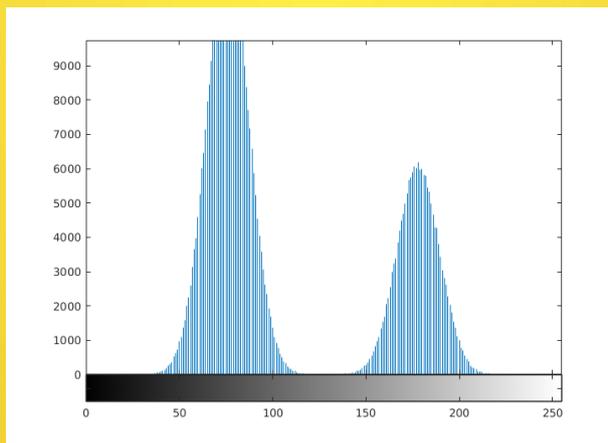
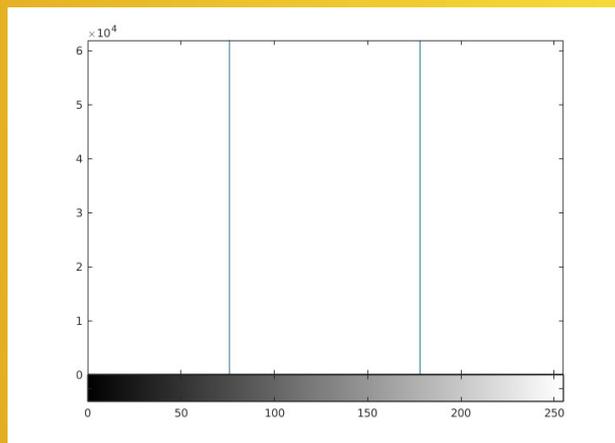
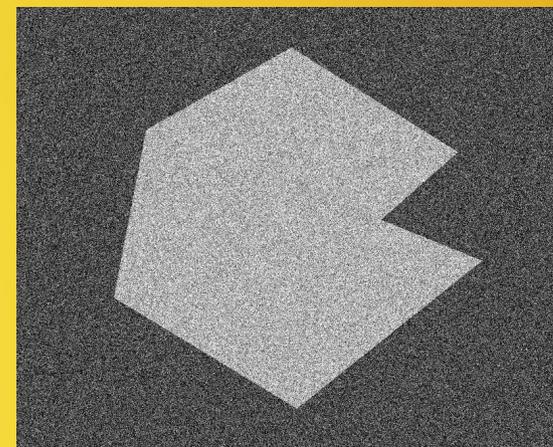
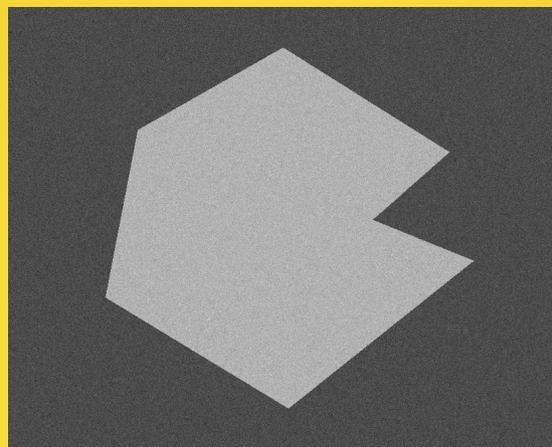
Se la distribuzione dei valori dello sfondo e degli oggetti dell'immagine fossero sufficientemente distinte l'istogramma potrebbe dare indicazione diretta della soglia

Più frequentemente le due popolazioni hanno distribuzione di luminosità sovrapposta

La slide successiva mostra un'immagine di prova (sinistra in alto) e due immagini derivate ottenute aggiungendo livelli di rumore crescenti. In basso gli istogrammi di queste 3 immagini

- La distribuzione del rumore aggiunto ha una distribuzione gaussiana (Matlab/Octave hanno funzione in grado di generare sequenze di numeri casuali con una data distribuzione di probabilità).

# Immagini con Rumore Gaussiano



$$\tilde{x}(u, v) = x(u, v) + \eta(u, v)$$

# Immagini con Rumore Gaussiano

- Il primo istogramma a sinistra ha solo 2 valori di intensità (circa 75 il valore inferiore e 180 quello superiore). Sarebbe naturale assegnare come soglia qualsiasi valore di intensità intermedio tra questi
- Il secondo istogramma mostra due distribuzioni centrate sui valori precedenti. Le due distribuzioni riflettono la variabilità indotta dal rumore gaussiano
  - Un valore di soglia continua ad essere identificabile nell'intervallo compreso tra le due distribuzioni gaussiane
- L'istogramma della terza immagine non permette più l'identificazione non ambigua di un valore soglia essendo il livello di rumore abbastanza alto da confondere le due distribuzioni gaussiane. In altre parole: il rumore ha un livello sufficientemente alto da assegnare a molti pixel del background valori di intensità che sono tipici del foreground e viceversa
- Si noti che indipendentemente dal livello di rumore la nostra percezione continua a vedere immagini apparentemente semplici da suddividere in regioni
- La determinazione della soglia non è, in molti casi pratici, possibile se non attraverso metodi analitici per cercare di minimizzare il numero di false attribuzioni alle regioni.

# Determinazione Iterativa di una Soglia

Esempio di algoritmo iterativo per la determinazione della soglia per il quale è disponibile un'implementazione in Matlab e Octave alla pagina <http://imaging.biol.unipr.it/static/calcolosoglia.html>

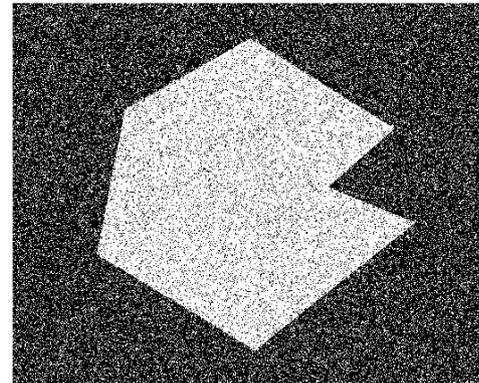
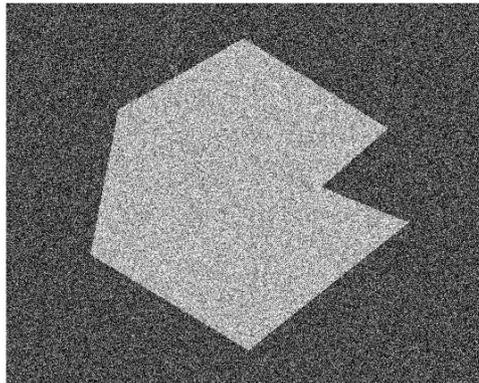
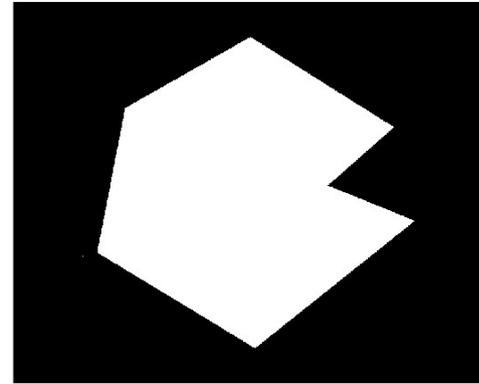
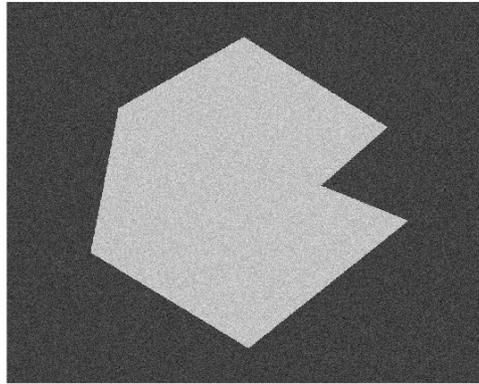
L'algoritmo iterativo è una successione passi computazionali che viene eseguita ripetutamente e si arresta quando una determinata condizione si verifica.

# Determinazione Iterativa di una Soglia

L'algoritmo iterativo può essere suddiviso in 6 punti

- 1) Viene stabilita una soglia iniziale  $T$ . Questo valore potrebbe essere la media, la mediana o anche un valore intermedio determinato osservando l'istogramma.
- 2) I valori dei pixel dell'immagine  $\{p_{ij}\}$  vengono confrontati con  $T$ , cioè si esegue il confronto  $\{p_{ij}\} > T$ . Il risultato è un'immagine binaria dove i pixel 1 sono quelli per cui la condizione è vera, 0 gli altri. Quindi avremo una nuova immagine  $\{b_{ij}\}$  della stessa dimensione di  $\{p_{ij}\}$  costituita da valori booleani (1: vero, 0: falso)
- 3) Con questa immagine binaria, usata come 'mappa' logica possiamo calcolare il valore di intensità medio  $m_1$  dei pixel  $\{p_{ij}\}$  al foreground (quelli per cui il valore corrispondente  $\{b_{ij}\} = 1$ ) e il valore  $m_2$  dei rimanenti pixel (quelli per cui il valore corrispondente  $\{b_{ij}\} = 0$ ).
- 4) Salviamo il valore appena usato della soglia (che chiamiamo  $T_{prev}$ ) e calcoliamo un nuovo valore di  $T$  come media di  $m_1$  e  $m_2$
- 5) Se la differenza tra  $T$  e  $T_{prev}$  in valore assoluto è superiore di un valore dato allora torniamo al punto 2
- 6) Se il valore assoluto della differenza tra  $T$  e  $T_{prev}$  è inferiore allora l'algoritmo si arresta e l'ultimo valore di  $T$  è il risultato

# Applicazione all'immagine di prova



# Determinazione Analitica della Soglia

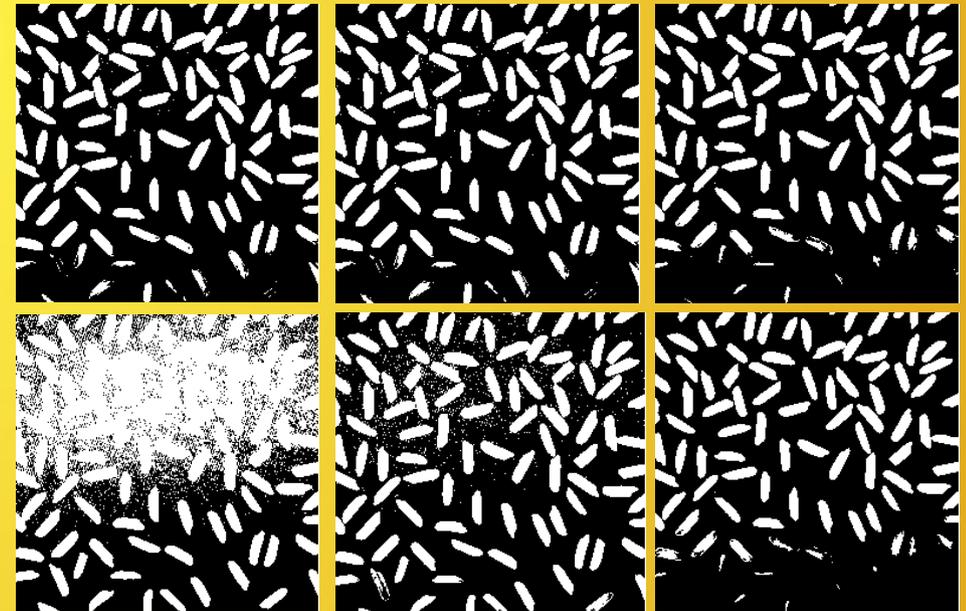
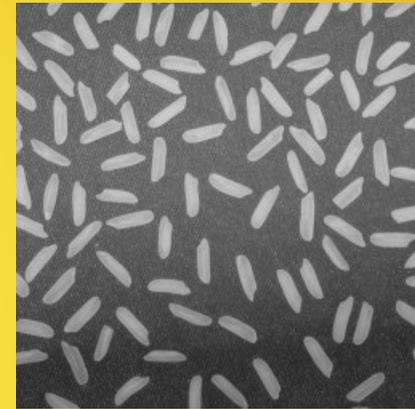
- Un importante metodo analitico di determinazione della soglia è il metodo di Otsu
- Si assume l'istogramma essere la sovrapposizione di 2 contributi generati da
  - pixel di background
  - pixel di foreground
- Ciascun dei due gruppi ha una propria distribuzione di luminosità
- La soglia calcolata con il metodo di Otsu è il valore di intensità che massimizza la varianza interclasse  $\sigma_B$  (e minimizza la varianza intraclasse)

$$\sigma_B = P_1 P_2 (m_1 - m_2)^2$$

$P_1$  e  $P_2$  sono le probabilità che un pixel di intensità  $p_{ij}$  venga assegnato ad una delle due classi. Necessariamente  $P_2 = 1 - P_1$ .  $P_1$  e  $P_2$  vengono stimate dall'istogramma

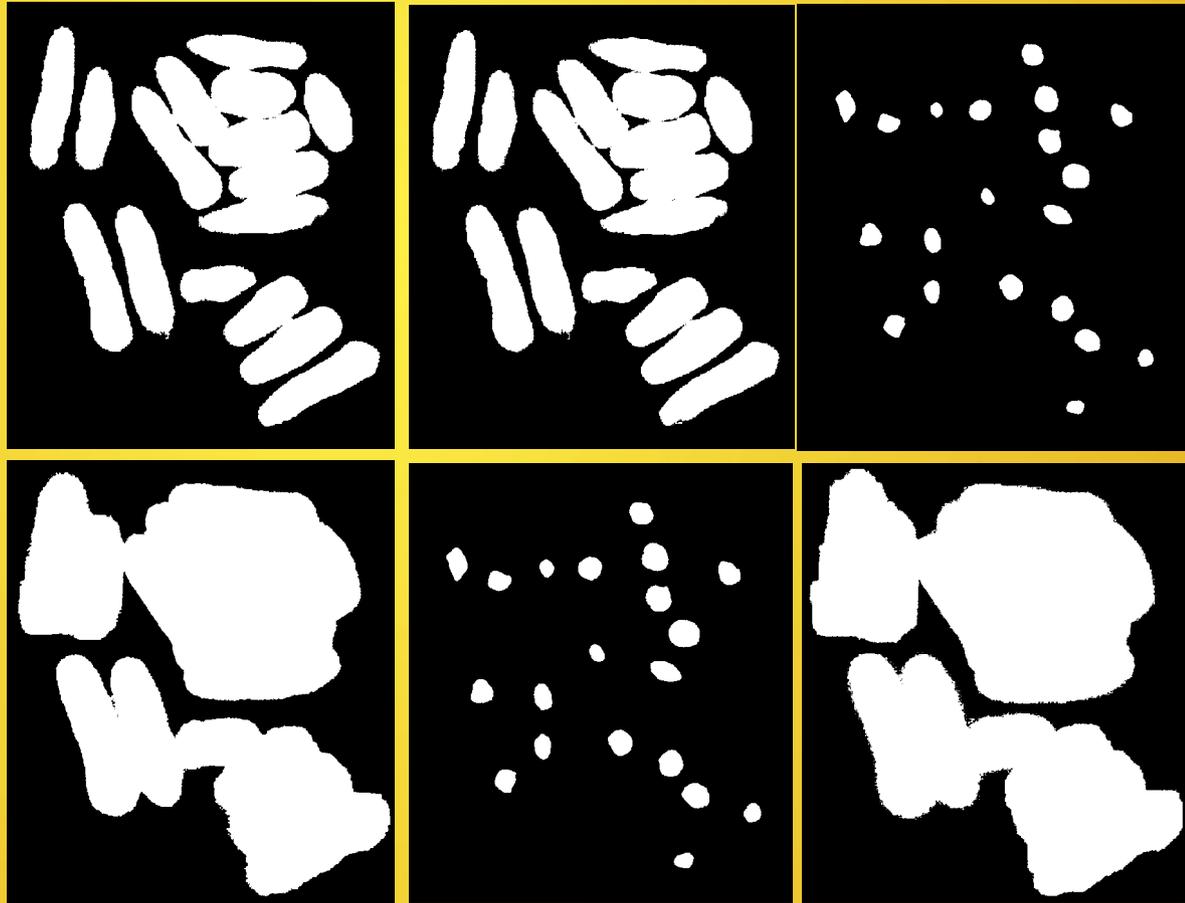
# Thresholding: confronto di vari metodi

- Confronto tra immagine originaria e thresholding con diversi metodi
- In alto un'immagine grayscale di grani di riso (immagine ricorrente usata come benchmark di questi metodi)
- In basso la segmentazione ottenuta con le soglie calcolate con vari metodi
  - Metodo Iterativo
  - Otsu
  - Intermodos
  - Concavity
  - MaxEntropy
  - Max Likelihood
- La documentazione di Octave ha riferimenti verso la letteratura dove si trova definizione e implementazione di ogni metodo



# Thresholding

Metodi diversi si adattano a risolvere strutture diverse di un'immagine. In questo caso un'immagine di lieviti al microscopio. La successione dei metodi è la stessa della slide precedente



# Thresholding

La segmentazione tramite soglia globale è semplice e computazionalmente veloce.

- Diversi metodi per il calcolo della soglia danno risultati differenti.
- Non necessariamente un metodo fornisce la migliore prestazione in tutti i casi.
- Metodi diversi possono rivelarsi adatti a costruire mappe di diverse strutture nella stessa immagine
- Anche i metodi che apparentemente hanno i risultati peggiori nelle 2 slide precedenti non vanno trascurati. Per esempio nella segmentazione dei lieviti il metodo della concavità fornisce una mappa accurata di agglomerati di lieviti

# Soglie Variabili

In immagini con illuminazione variabile può essere talvolta necessario usare soglie variabili per arrivare ad una segmentazione accettabile

Con soglie variabili un'immagine viene suddivisa in un numero di quadranti all'interno dei quali l'illuminazione è sufficientemente uniforme

Per ogni quadrante viene calcolata una soglia globale e il risultato è una mappa binaria specifica del quadrante

La mappa complessiva si costruisce ricomponendo le matrici delle mappe dei quadranti

Alla pagina <http://imaging.biol.unipr.it/static/varthresh.html> è disponibile un esercizio da fare con Matlab/Octave