Fondamenti di Elaborazione di Immagini

Morfologia Matematica

Raffaele Cappelli

raffaele.cappelli@unibo.it

Contenuti

- Introduzione alla morfologia matematica
  - Notazione e concetti di base
- Gli operatori di base
  - Dilatazione, erosione
- Altre operazioni
  - Apertura, chiusura
  - Hit-or-Miss transform
  - Estrazione del bordo
- Morfologia in scala di grigio

Morfologia matematica

- Branca della matematica che si rivolge all’elaborazione delle immagini
  - Derivata dalla teoria degli insiemi
  - Fornisce strumenti utili per
    - estrarre informazioni utili a rappresentare e descrivere la forma (contorno, scheletro, ...)
    - rimuovere particolari irrilevanti mantenendo le informazioni importanti sulla forma degli oggetti
  - Lavora su immagini binarie (appartiene alla più generale disciplina della topologia digitale), ma esistono estensioni per immagini grayscale
- Elemento strutturante
  - Piccola immagine binaria (es. 3x3 o 5x5) che viene utilizzata come parametro nelle operazioni morfologiche (anch’essa considerata un insieme di pixel di foreground)
    - Tipicamente quadrata (lato dispari) e centrata rispetto all’origine

Notazione di base

- Considera immagini digitali binarie (2 soli livelli di grigio)
  - Foreground (in genere 255 o 0, nel seguito indicato con *foreground*)
  - Background (in genere 0 o 255, nel seguito indicato come ≠*foreground*)

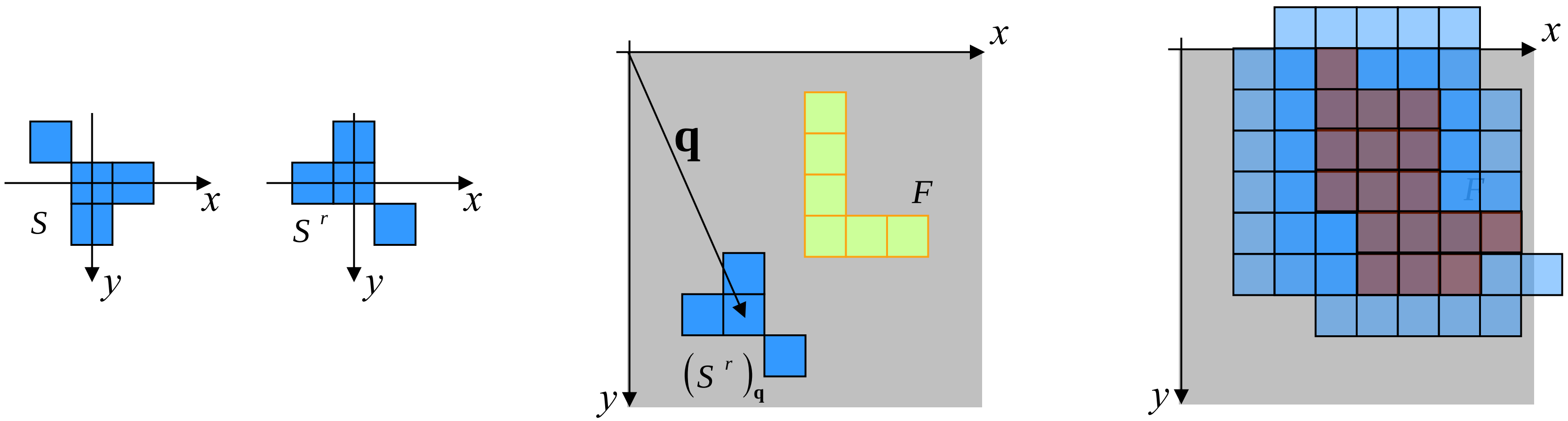
- Sia F l'insieme di tutti i pixel di foreground e F^* l'insieme di quelli di background di un'immagine Img:
  - $\mathrm{F} = \{\mathbf{p} \mid \mathbf{p} = [x, y]^{\mathrm{T}}, \mathrm{Img}[y, x] = \textit{foreground}\}$
  - $\mathrm{F}^* = \{\mathbf{p} \mid \mathbf{p} = [x, y]^{\mathrm{T}}, \mathrm{Img}[y, x] \neq \textit{foreground}\}$

- Operazioni di base
  - Intersezione e unione: $A \cap B = \{\mathbf{p} \mid \mathbf{p} \in A \wedge \mathbf{p} \in B\}$ $A \cup B = \{\mathbf{p} \mid \mathbf{p} \in A \vee \mathbf{p} \in B\}$
  - Complemento: $A^c = \{\mathbf{p} \mid \mathbf{p} \notin A\}$
  - Differenza: $A - B = \{\mathbf{p} \mid \mathbf{p} \in A \wedge \mathbf{p} \notin B\} = A \cap B^c$
  - Traslazione rispetto a un punto $\mathbf{q}$: $(A)_{\mathbf{q}} = \{\mathbf{p} \mid \mathbf{p} = \mathbf{a} + \mathbf{q}, \mathbf{a} \in A\}$
  - Riflessione rispetto all'origine: $A^r = \{\mathbf{p} \mid \mathbf{p} = -\mathbf{q}, \mathbf{q} \in A\}$

Morfologia matematica – Operatori di base

- Due operatori di base: Dilatazione ed Erosione
  - I due operatori su cui si basano la maggior parte delle operazioni morfologiche più complesse
- Dilatazione (Dilation)
  - La nuova immagine è l'insieme dei pixel tali che, traslando in essi S^r, almeno uno dei suoi elementi è sovrapposto a F

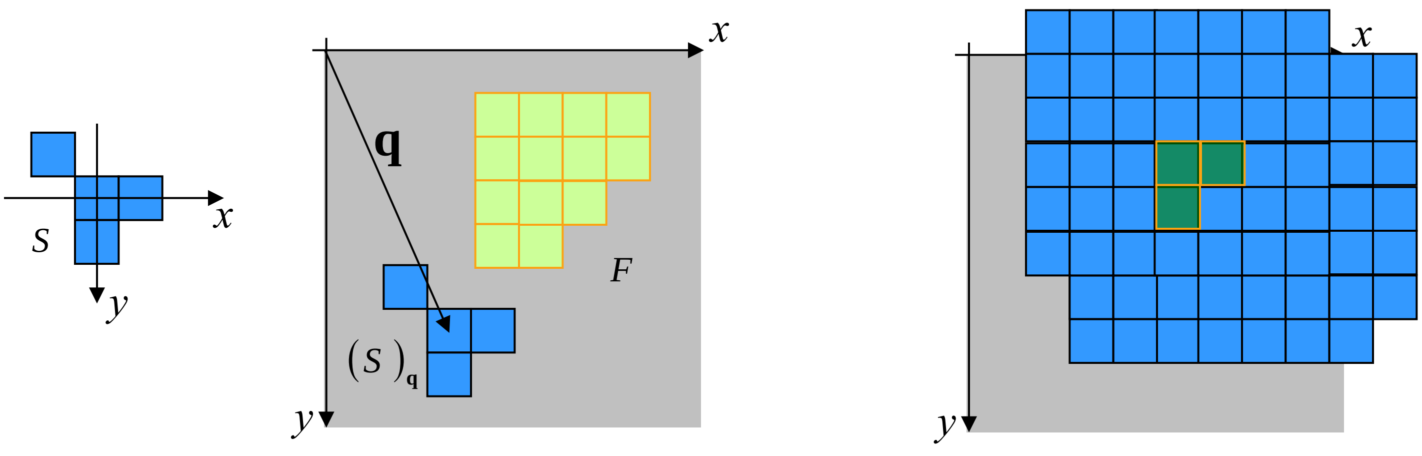
$$F \oplus S = \left\{ \mathbf{q} \mid \left(S^r \right)_{\mathbf{q}} \cap F \neq \varnothing \right\}$$



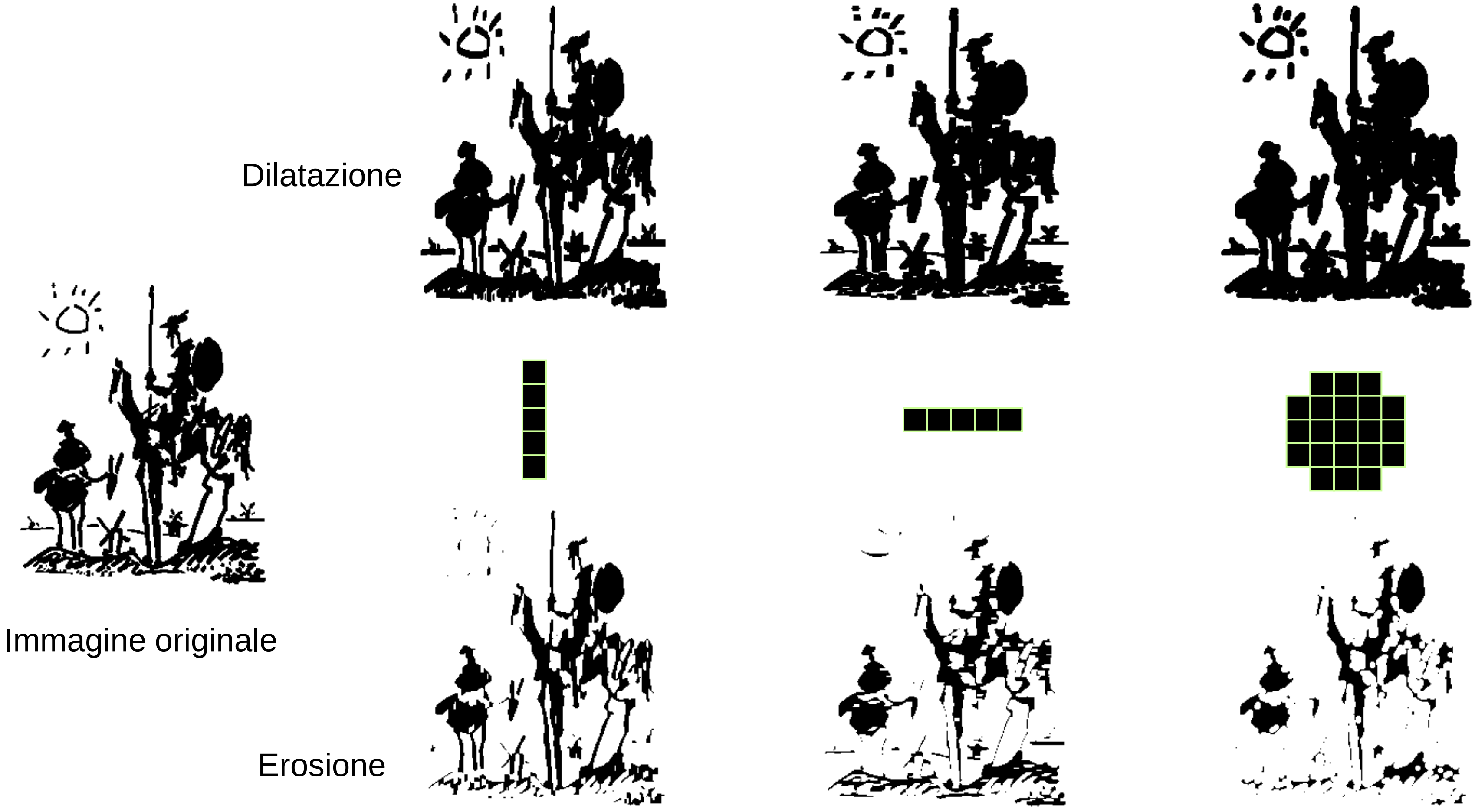
Morfologia matematica – Operatori di base (2)

- Erosione (Erosion)
  - La nuova immagine è l'insieme dei pixel tali che, traslando in essi S, l'intero elemento strutturante è contenuto in F

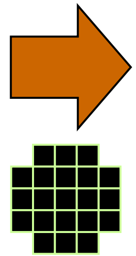
$$F \ominus S = \{\mathbf{q} \mid (S)_{\mathbf{q}} \subseteq F\}$$



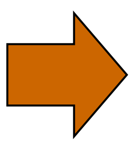
Dilatazione ed Erosione – Esempi



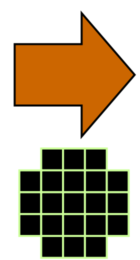
Dilatazione ed Erosione – Esempi (2)

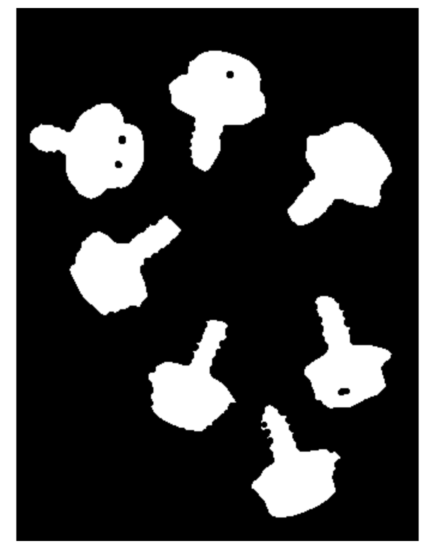


La dilatazione può aiutare a riempire "buchi" e altre simili imperfezioni



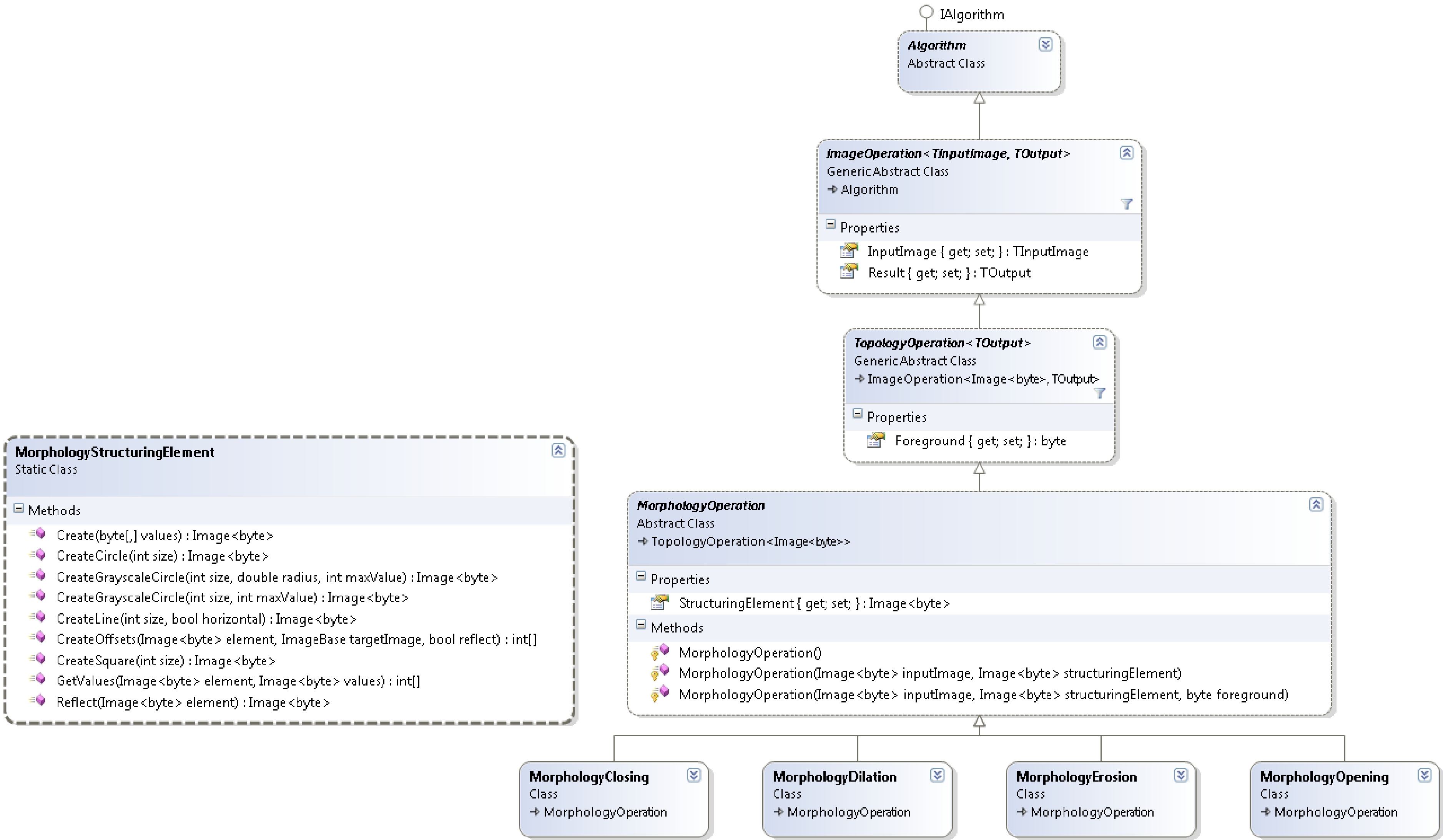
Binarizzazione





L'erosione può aiutare ad eliminare il rumore

Morfologia matematica: classi nella libreria



Dilatazione – Implementazione di base

```
// Costruisce l'array degli offset dell'elemento strutturante riflesso
int[] elementOffsets = MorphologyStructuringElement.CreateOffsets(
                                         StructuringElement, InputImage, true);

// Crea un cursore per scorrere l'immagine escludendo i pixel di bordo
var pixelCursor = new ImageCursor(
                 StructuringElement.Width / 2,
                 StructuringElement.Height / 2,
                 InputImage.Width - 1 - StructuringElement.Width / 2,
                 InputImage.Height - 1 - StructuringElement.Height / 2,
                 InputImage);
do
{
  foreach (int offset in elementOffsets)
  {
    if (InputImage[pixelCursor + offset] == Foreground)
    {
      Result[pixelCursor] = Foreground;
      break; // esce dal foreach
    }
  }
} while (pixelCursor.MoveNext());
```

Morfologia matematica – Altre operazioni

- Apertura (Opening)
  - Erosione seguita da dilatazione
  - Separa oggetti debolmente connessi e rimuove regioni piccole

$$F \circ S = (F \ominus S) \oplus S$$

- Chiusura (Closing)
  - Dilatazione seguita da erosione
  - Riempie buchi e piccole concavità e rafforza la connessione di regioni unite debolmente

$$F \bullet S = (F \oplus S) \ominus S$$

- Hit-or-Miss Transform
  - Localizza i punti in cui S_1 è contenuto nel foreground e S_2 nel background
  - È un'operazione di pattern matching

$$F * S = (F \ominus S_1) \cap (F^c \ominus S_2)$$

$$S = (S_1, S_2),\ S_1 \cap S_2 = \varnothing$$

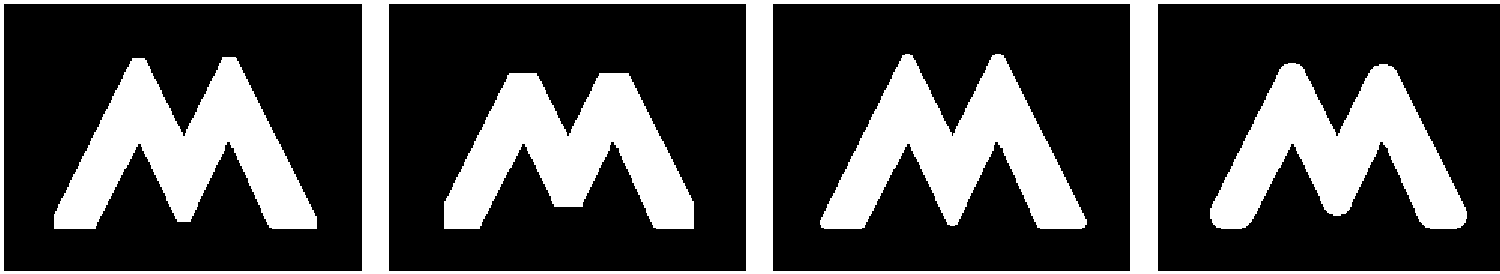
Apertura – Implementazione di base

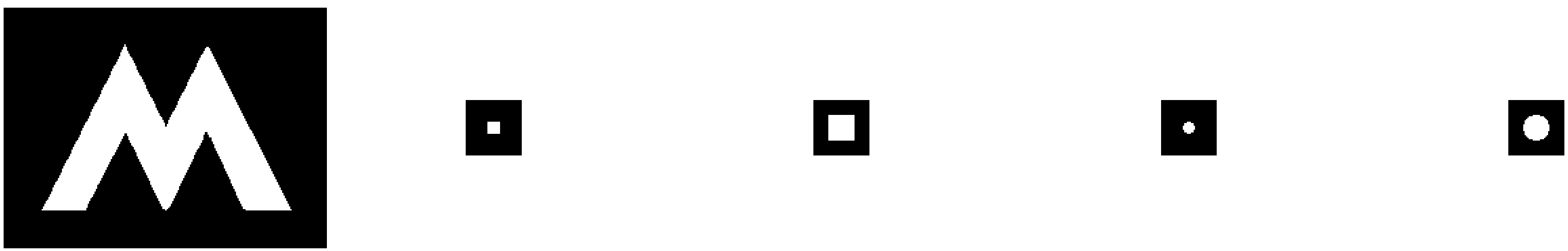
```
[AlgorithmInfo("Opening", Category = "Binary morphology")]
public class MorphologyOpening : MorphologyOperation
{
  public MorphologyOpening(Image<byte> inputImage, Image<byte>
                                      structuringElement, byte foreground)
    : base(inputImage, structuringElement, foreground)
  {
  }

  public override void Run()
  {
    var erosion = new MorphologyErosion(InputImage,
                                          StructuringElement, Foreground);
    var dilation = new MorphologyDilation(erosion.Execute(),
                                          StructuringElement, Foreground);
    Result = dilation.Execute();
  }
}
```

Apertura e Chiusura – Esempi

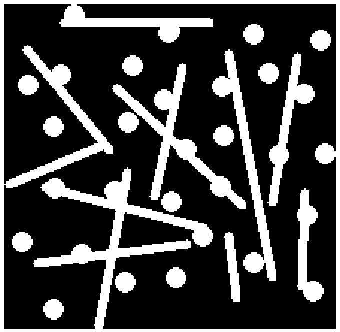
Apertura



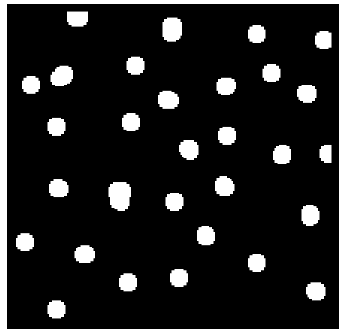


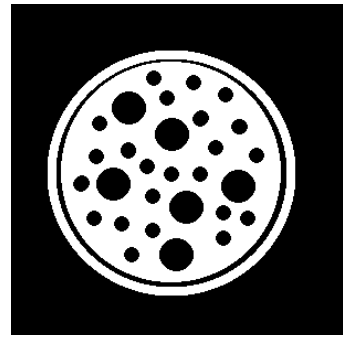
Chiusura

Apertura e Chiusura – Esempi (2)

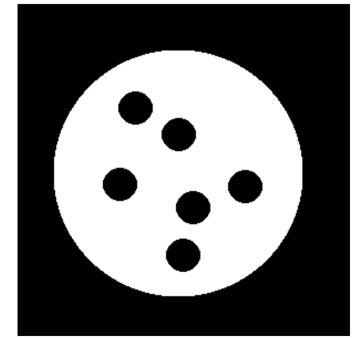


Apertura

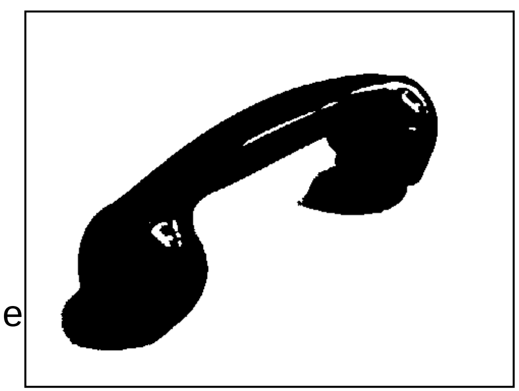




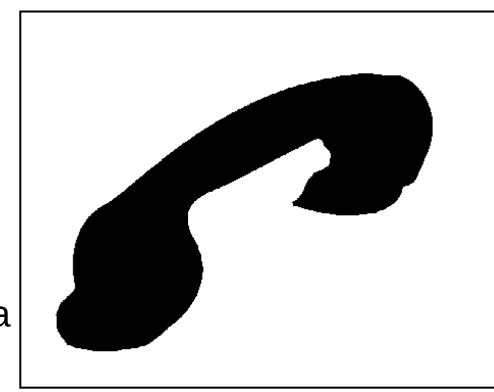
Chiusura



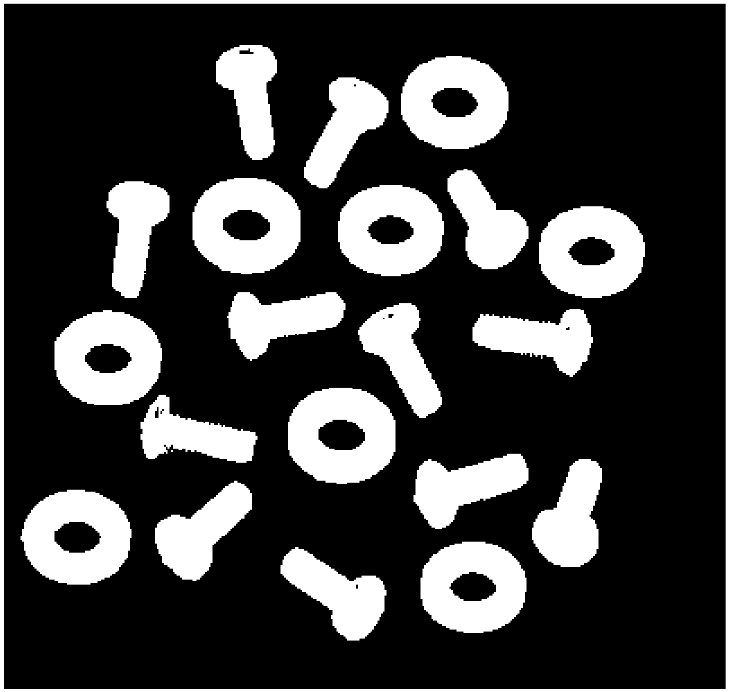
Binarizzazione

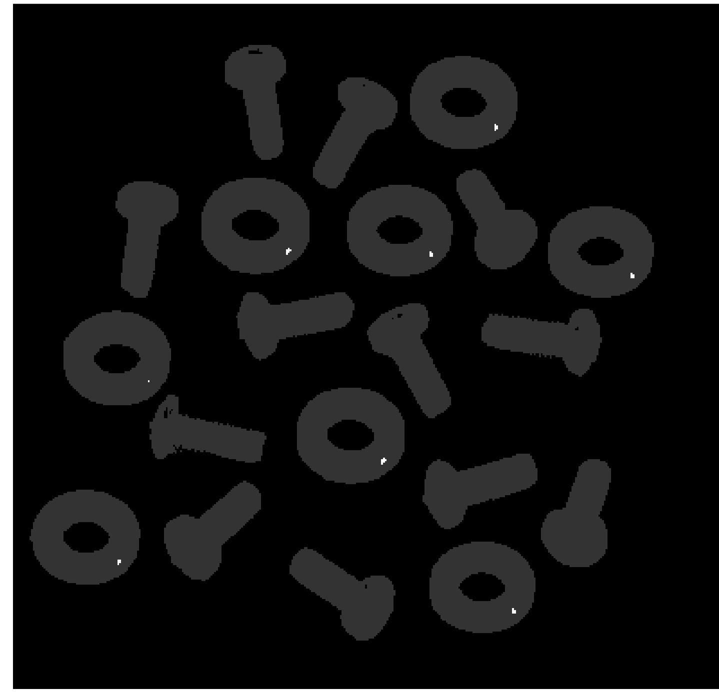


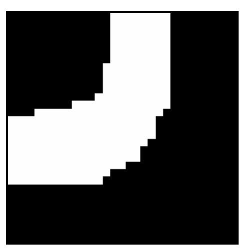
Chiusura



Hit-or-Miss Transform – Esempio







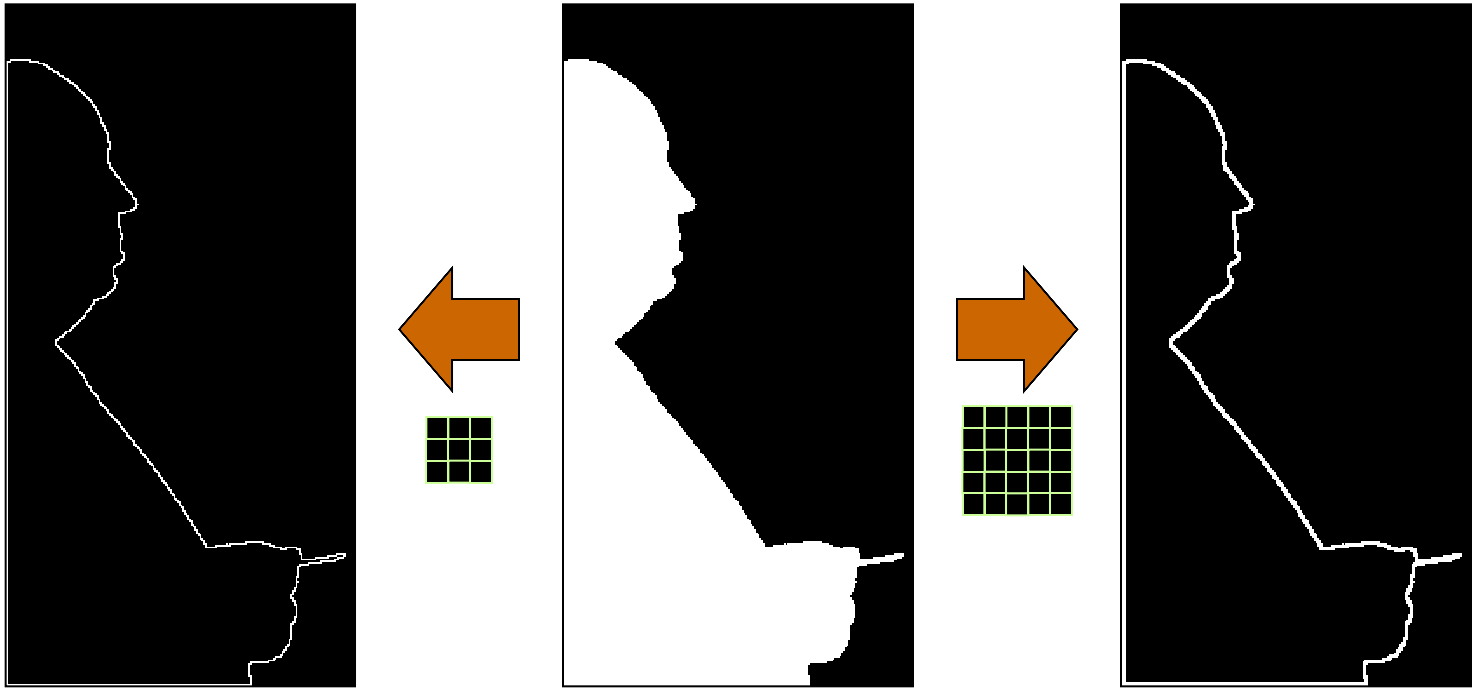
$$S = (S_1, S_2)$$

S_1 (*foreground*) S_2 (*background*)

Morfologia matematica: estrazione del contorno

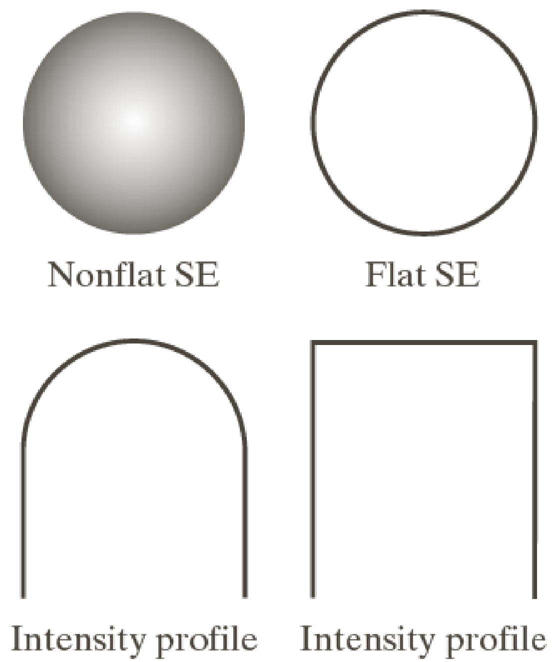
- È possibile ottenere un'immagine contenente solo i pixel appartenenti al contorno di F sottraendo il risultato dell'erosione all'immagine stessa
  - L'elemento strutturante determina lo spessore del contorno

$$\beta_S(F) = F - (F \ominus S)$$



Morfologia in scala di grigio

- È possibile estendere le operazioni di base alle immagini grayscale
- Definizioni:
  - $f(x,y)$: immagine in scala di grigio
  - $b(x,y)$: elemento strutturante (Flat o Non-flat) con origine posta nel centro



Elementi strutturanti flat: operatori di base

- Dilatazione (Dilation)
  - Definita, per ogni posizione (x,y), come il valore massimo dell'immagine indicata da b quando l'origine di b si trova in (x,y)

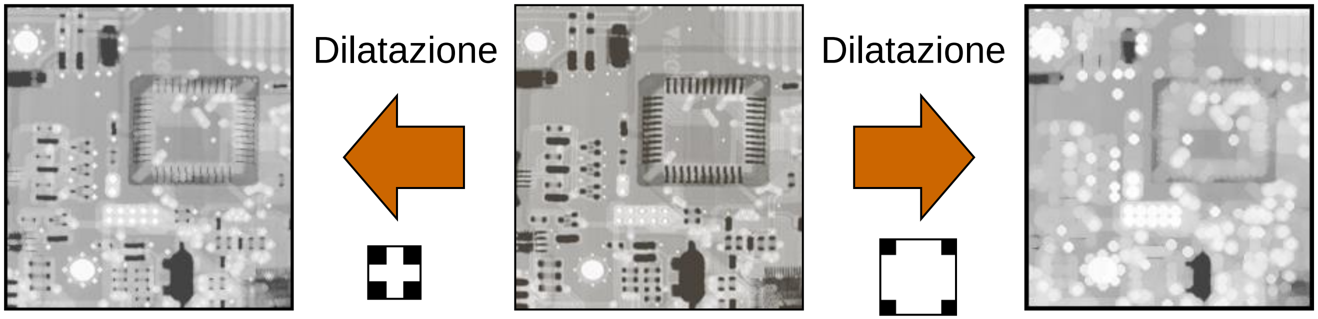
$$[f \oplus b](x,y) = \max_{(s,t)\in b} \{ f(x-s, y-t) \}$$

- Erosione (Erosion)
  - Definita, per ogni posizione (x,y), come il valore minimo dell'immagine indicata dalla riflessione di b quando l'origine di b si trova in (x,y)

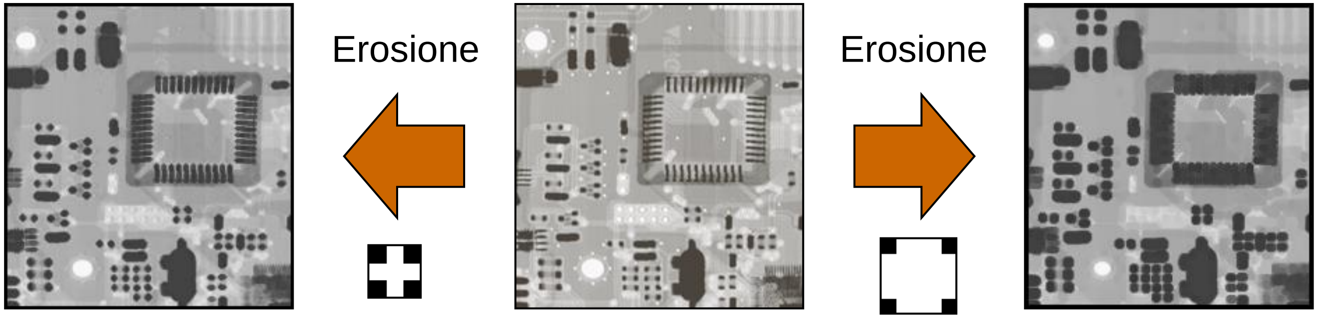
$$[f \ominus b](x,y) = \min_{(s,t)\in b} \{ f(x+s, y+t) \}$$

Elementi strutturanti flat: esempio

- Dilatazione
  - Le dimensioni delle componenti chiare vengono aumentate, mentre le dimensioni delle componenti scure vengono ridotte.



- Erosione
  - Risultato opposto a quello della dilatazione.



Elementi strutturanti non-flat: operatori di base

- Dilatazione (Dilation)

$$[f \oplus b_{NF}](x,y) = \max_{(s,t)\in b_{NF}} \{ f(x-s, y-t) + b_{NF}(s,t) \}$$

- Erosione (Erosion)

$$[f \ominus b_{NF}](x,y) = \min_{(s,t)\in b_{NF}} \{ f(x+s, y+t) - b_{NF}(s,t) \}$$

- Attenzione:
  - Al contrario degli elementi strutturanti flat, il risultato di questi operatori non è necessariamente limitato dai valori di f, cosa che può portare a problemi nell'interpretazione dei risultati

Morfologia binaria e in scala di grigio: classi