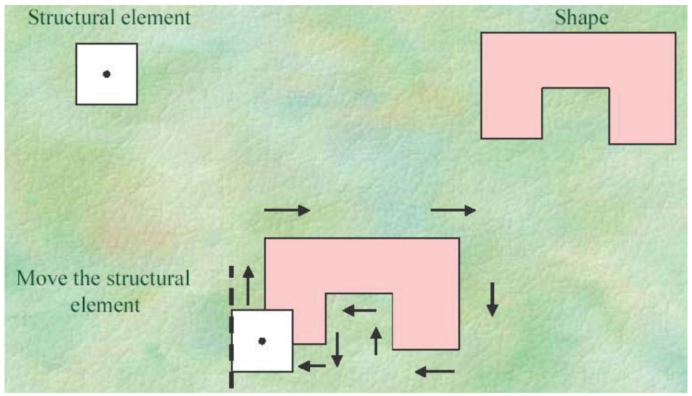
Morphology : *Britannica*

- Morphology (in biology): the study of the size, shape, and structure of animals, plants, and microorganisms and the relationships of their internal parts.
- Morphology (in linguistics): study of the internal construction of words.
- Image processing: The basis of mathematical morphology is the description of image regions as sets.

A	the (usually binary) image
A^C	the complement of the image (inverse)
$A \cup B$	the union of images A and B
$A \cap B$	the intersection of images A and B
$A - B = A \cap B^C$	the difference between A and B (pixels in A not in B)
$\#A$	the cardinality of A (area of the object)

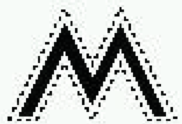
Mathematical morphology

- A morphological operator is defined by a structuring element (or kernel) of size $n \times n$ and a set operator.
- Kernel is shifted over the image and its elements are compared with the underlying pixels.
- If the two sets of elements match the condition defined by the set operator, the pixel underneath the center of the structuring element is set to a pre-defined value (0 or 1 for binary images).

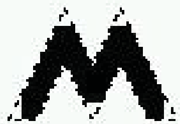


Morphological operators

Dilation - grow image regions

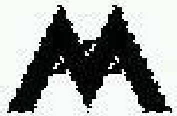


Erosion - shrink image regions

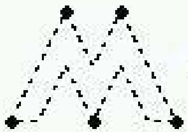


Opening - structured removal of image region boundary pixels

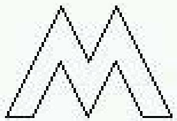
Morphological operators



Closing - structured filling in of image region boundary pixels

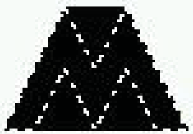


Hit and Miss Transform - image pattern matching and marking

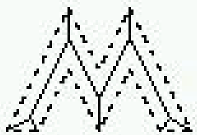


Thinning - structured erosion using image pattern matching

Morphological operators



Thickening - structured dilation using image pattern matching



Skeletonization/Medial Axis Transform - finding skeletons of binary regions

Dilation (Minkowski addition)

- The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically).
- Areas of foreground pixels grow in size, while holes in regions become smaller.

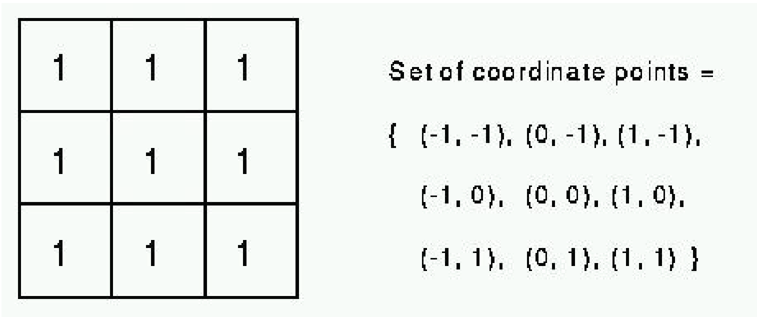
  Let A and B denote sets in $\mathbb{R}^n$ or $\mathbb{Z}^n$ with elements a and b.

  Then

$$A \oplus B = \{c \in \mathbb{R}^n : c = a + b\}$$

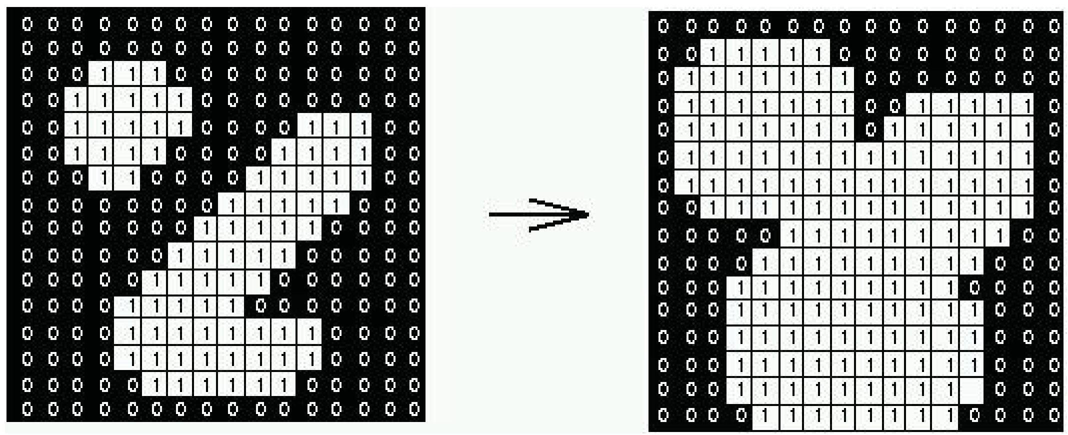
- Typically: A = binary image, B = mask (structuring element)

  Example: 3×3 structuring element



Dilation (Minkowski addition)

- If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value.
- If all the corresponding pixels in the image are background, however, the input pixel is left at the background value.

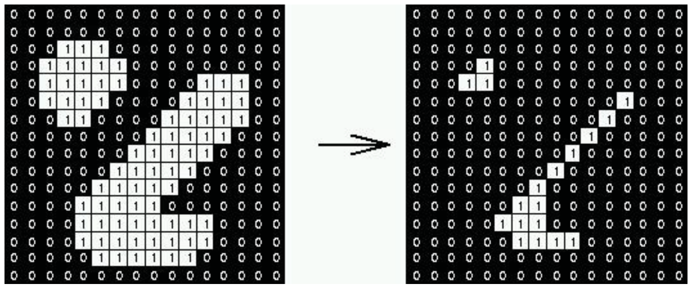


Erosion

- The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels.
- Thus areas of foreground pixels shrink in size, and holes within those areas become larger.

$$A \ominus B = \{c \in \mathbb{R}^n : c + b \in A, \forall b \in B\}$$

- If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is.
- If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value.



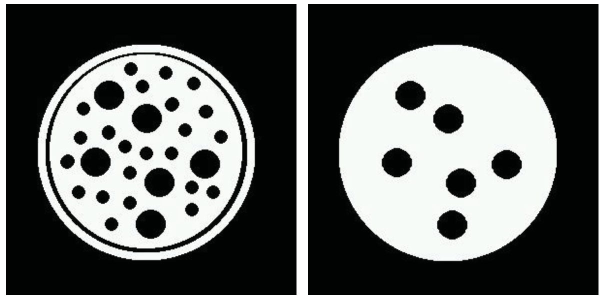
Operators

- Opening: an opening is defined as an erosion followed by a dilation using the same structuring element for both operations

$$A \circ B = (A \underbrace{\ominus}_{erosion} B) \underbrace{\oplus}_{dilation} B$$

- Closing:

$$A \bullet B = (A \oplus B) \ominus B$$



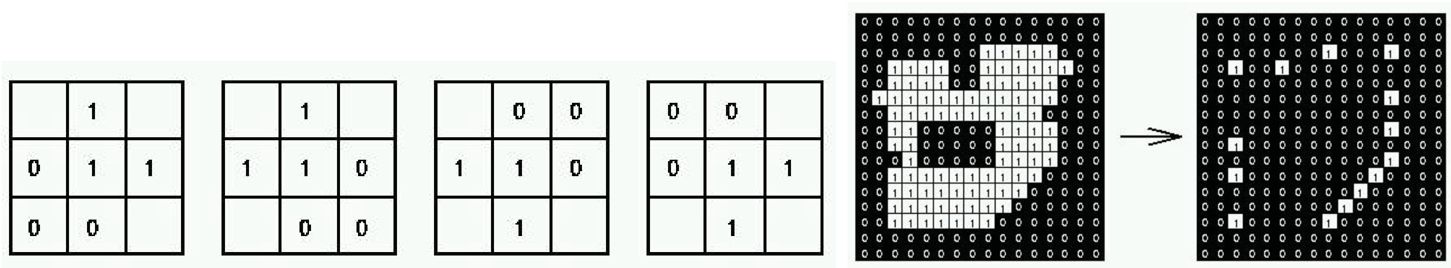
- Opening is the dual of closing: opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element.

Hit and miss transform

- Purpose: Extract objects of certain shape.
- Let J and K be structuring elements with $J \cap K = \emptyset$
- Hit-and-miss transform can then be defined as

$$A \otimes (J, K) = (A \ominus J) \cap (A^C \ominus K)$$

- Contains all points where
  - J matches the object
  - K matches background
- Example: detect corners in a binary image.



Thinning

- The hit-and-miss transform has many applications in more complex morphological operations.
- It is used to construct the thinning and thickening operators.
- The thinning operation can be written in terms of a hit-and-miss transform

  thin(A,B) = A - hit-and-miss(A,B)