Module 5: Mathematical Morphology

Basics of mathematical morphology

- Mathematical Morphology is a technique used in image processing to analyze and manipulate the shape and structure of objects within images.
- It involves operations like erosion, dilation, opening, and closing, which modify the shape of objects based on their neighbourhood.
- The language of mathematical morphology is set theory, where images are represented as sets of coordinates or pixels.
- Basic concepts in set theory, such as subsets, union, intersection, complement, and difference, are fundamental to understanding mathematical morphology.
- Logic operations like AND, OR, and NOT are commonly used in image processing and are analogous to set operations like intersection, union, and complement.

Applications of mathematical morphology

- **Morphological Image Filtering**: Used for tasks like boundary extraction, skeletonization, convex hull computation, thinning, and pruning.
- **Boundary Extraction**: Identifying and extracting the outlines or boundaries of objects in an image.
- **Skeletons**: Extracting the central, often thin, structure of objects, useful for shape analysis.
- **Convex Hull**: Finding the smallest convex shape that encloses a given object.
- **Morphological Filtering**: Filtering or modifying the shape and structure of objects in an image.

- **Thinning**: Reducing the thickness of objects in an image while preserving their connectivity.
- **Pruning:** Removing small or insignificant structures from an image while retaining important features.

➢ Set theory basics

- Set theory provides the foundation for mathematical morphology, where images are treated as sets of pixels or coordinates.
- Basic set operations include subset, union, intersection, complement, and difference.

Subset

$A \subseteq B$

Union

$A \cup B$

Intersection

$A \cap B$

disjoint / mutually exclusive $A \cap B = \emptyset$

Complement $A^c \equiv \{ w \mid w \notin A \}$

Difference $A - B \equiv \{ w \mid w \in A, w \notin B \} = A \cap B^c$

Reflection $B \equiv \{ w \mid w = -b, \quad \forall b \in B \}$

Translation $(A)z \equiv \{ c \mid c = a + z, \quad \forall a \in A \}$

- In the context of image processing, logic operations like AND, OR, and NOT correspond to set operations like intersection, union, and complement.
- Set theory allows for the representation and manipulation of binary images (black and white) as well as grayscale images (with different intensity levels).

➢ Five Binary Morphological Operations:

- ⊖ • Erosion
- ⊕ • Dilation
- ○ • Opening
- ● • Closing
- ⊗ • Hit-or-Miss transform

➢ Erosion

Imp

- -Erosion is a morphological operation used to **shrink or thin** objects in binary images.
- It's achieved by applying a structuring element (B) to the input image (A), where the structuring element defines the shape of the erosion.
- The erosion of A by B, denoted as **A Θ B**, is the set of all points where B, when translated by z, is entirely contained within A.
- Mathematically, erosion is defined as $A \ominus B=\{z|(B)z \subseteq A\}$

➢ Dilation

Imp

- - Dilation is a morphological operation used **to expand or thicken objects** in binary images.
- It's achieved by applying a structuring element (B) to the input image (A), where the structuring element defines the shape of the dilation.
- The dilation of A by B, denoted as **A ⊕ B**, is the set of all displacements z such that B and A overlap by at least one element.
- Mathematically, dilation is defined a $\boldsymbol{A \oplus B=\{z|(B)z \cap A \neq \emptyset\}}$

Imp

➢ Erosion vs dilation

- Erosion **shrinks or thins** objects, while dilation **grows or thickens** them.
- Erosion removes image components by **eroding away the boundaries of foreground regions**, causing areas to shrink and holes to enlarge.
- Dilation fills in gaps and small holes within objects, **bridging small separations between foreground pixels.**
- Both operations use structuring elements to define the shape of the transformation, with erosion looking for containment and dilation looking for overlap.

Imp

➢ Opening and closing

1. Opening:

- Opening is a morphological operation performed by first applying erosion followed by dilation, **denoted as ∘**
- **It is used to eliminate protrusions and smoothning the contours of objects in binary images.**
- Mathematically, opening is defined as $A \circ B = (A \ominus B) \oplus B$, where $(A \ominus B)$ represents the erosion of A by B, and $\oplus$ represents dilation.

2. Closing:

- Closing is a morphological operation performed by first applying dilation followed by erosion, **denoted as •**
- It is used to **smooth contours, fuse narrow breaks, eliminate small holes, and fill gaps in the contours** of objects in binary images.
- Mathematically, closing is defined as $A \bullet B = (A \oplus B) \ominus B$, where $(A \oplus B)$ represents the dilation of A by B.

Hit or miss transform

- The hit-or-miss transformation is a morphological operation used for **pattern detection** in binary images.
- It involves matching a specific pattern, represented by a structuring element, with the input image.
- The **structuring element is composed of foreground (1's) and background (0's) pixels** arranged in a specific configuration.
- The purpose of using such structuring elements is based on the assumption that objects of interest are distinct only if they are disjoint sets.
- In some applications, the goal is to detect certain patterns or combinations of 1's and 0's rather than individual objects.
- This pattern detection scheme simplifies the operation to a form of erosion, where the goal is to find matches between the structuring element and the input image.

  Mathematically: $A \circledast B = (A \ominus B1) \cap (A^c \ominus B2)$

  A represents the input binary image.

  B is the structuring element composed of two parts, $B1$ and $B2$, representing foreground and background

  $A \ominus B1$ denotes erosion of the input image A by the foreground pattern $B1$.

  $A^c \ominus B2$ denotes erosion of the complement of A (background) by the background pattern $B2$.
- The hit-or-miss transformation returns the intersection of the eroded versions of A, representing the locations where both **foreground and background patterns match.**