UNIT–4 : Image Segmentation (PYQ Solution)

AKTU · Image Processing (BCS057) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. Define erosion and dilation. (2023, 2019)

Dilation of A by structuring element B, $A \oplus B = \{z \mid (\hat{B})z \cap A \neq \varnothing\}$, grows or thickens objects and fills small holes and gaps. Erosion, $A \ominus B = \{z \mid (B)z \subseteq A\}$, shrinks objects, removes small protrusions and noise and widens holes. They are dual operations; dilation after erosion gives closing and opening in the reverse order.

Q2. Explain the concept of region growing in image segmentation. (2019)

Region growing is a region-based segmentation method that starts from seed pixels and grows regions by appending neighbouring pixels that satisfy a similarity criterion (e.g. intensity difference below a threshold T). Growth continues until no more pixels qualify. Choice of seeds, the criterion and stopping rule decide the result; it works well on connected, homogeneous areas.

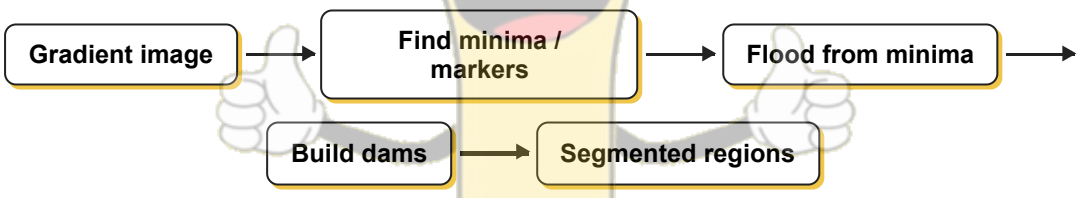
SECTION B Answers (≈250 Words Each)

Q1. Explain the watershed segmentation algorithm in detail. (2023)

Watershed segmentation is a region-based method that views a gray-scale image as a **topographic surface**: pixel intensity is the height. **Catchment basins** are regions around local minima, and **watershed lines** (ridges) separate them. Imagine water rising from the minima; where waters from two basins would merge, a **dam** is built, and the dams are the segmentation boundaries.

Algorithm (flooding):

1. Compute the gradient magnitude of the image, so object edges form ridges and homogeneous regions form basins.
2. Find **regional minima** (or select markers) and label each as a separate basin.
3. Flood the surface gradually by raising the water level n from min to max.
4. At each level, pixels at that height are assigned to the basin of adjacent labelled pixels.
5. If a pixel touches two different basins, mark it as a **dam (watershed line)**.
6. Continue until all pixels are labelled; the dams form closed, connected boundaries.



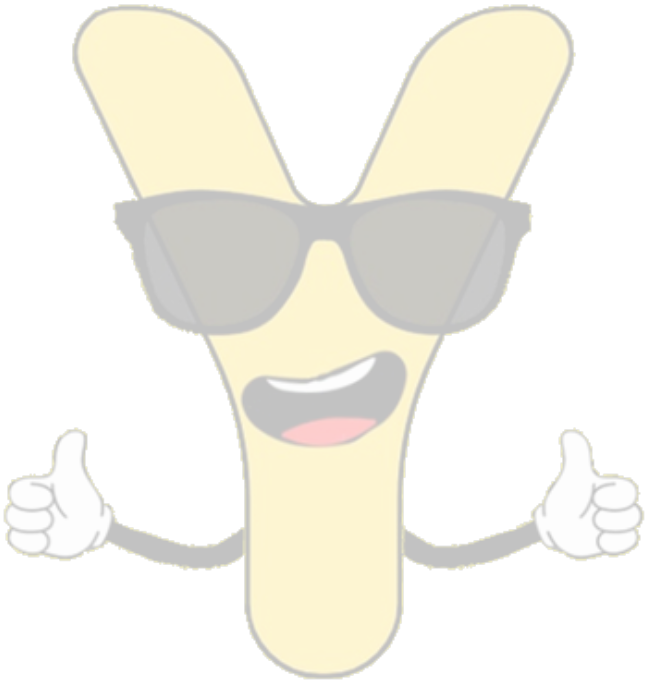
Watershed algorithm

Marker-controlled watershed: the plain algorithm over-segments because of noise and many spurious minima. Internal markers (inside objects) and external markers (background) are specified, and the gradient is modified so only these are minima. This gives meaningful segments.

Advantages: produces closed, one-pixel-wide boundaries; separates touching objects; combines edge and region ideas. **Disadvantages:** over-segmentation from noise and sensitivity to gradient quality.

Applications: medical imaging (cells), counting touching coins or grains, and separating overlapping objects.

Conclusion: watershed segments by flooding catchment basins, and markers make it practical.



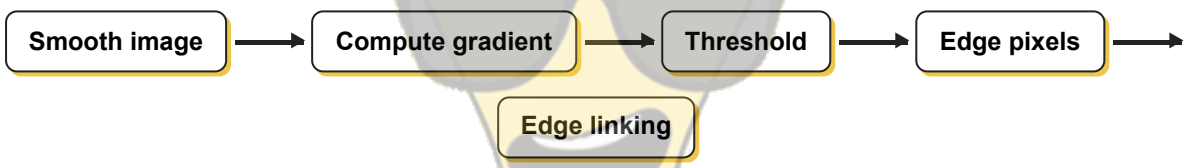
SECTION C Answers (≈500 Words Each)

Q1. Explain edge detection and edge linking. Differentiate between them. (2023)

Edge detection finds pixels where intensity changes abruptly, marking object boundaries. Edges are detected using **derivatives**: the first derivative (gradient) is large at an edge, and the second derivative (Laplacian) has a **zero crossing** at the edge.

Gradient operators: $\nabla f = [G_x, G_y]$, magnitude $M = \sqrt{(G_x^2 + G_y^2)} \approx |G_x| + |G_y|$, direction $\alpha = \tan^{-1}(G_y/G_x)$.

- **Roberts** (2 × 2), **Prewitt** and **Sobel** (3 × 3: Gx = [−1 0 1; −2 0 2; −1 0 1]).
- **Laplacian:** $\nabla^2 f$; it is noise sensitive, so the **Laplacian of Gaussian (LoG)** smooths first.
- **Canny:** smoothing, gradient, non-maximum suppression, hysteresis thresholding.



Edge detection to linking

Edge linking joins the detected edge pixels into meaningful, continuous boundaries, since noise and uneven illumination leave gaps.

- **Local processing:** a pixel (x′, y′) in the neighbourhood of (x, y) is linked if $|M(x, y) - M(x', y')| \le E$ (similar magnitude) and $|\alpha(x, y) - \alpha(x', y')| \le A$ (similar direction).
- **Global processing:** the **Hough transform** links points lying on a line or curve.
- Edge thinning and graph-theoretic methods are other options.

Basis	Edge detection	Edge linking
Purpose	Locate points of sharp intensity change	Connect edge points into boundaries
Input	Gray image	Edge image
Method	Gradient, Laplacian	Local or global (Hough)
Output	Isolated, possibly broken edges	Continuous contours
Order	First	After detection

Conclusion: detection finds candidate edge pixels, and linking turns them into complete boundaries.

Q2. Explain image segmentation and its fundamental conditions. (2023)

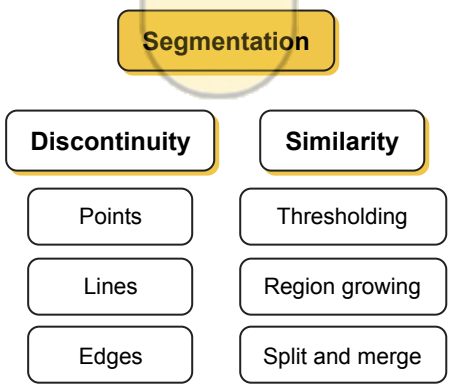
Image segmentation divides an image R into n subregions R1, R2, …, Rn, so that each region corresponds to an object or part of interest. It is the first step of image analysis; segmentation of non-trivial images is among the hardest tasks.

Fundamental conditions:

1. **Completeness:** ∪ Ri = R; every pixel belongs to some region.
2. **Connectedness:** each Ri is a connected set.
3. **Disjointness:** Ri ∩ Rj = ∅ for i ≠ j; regions do not overlap.
4. **Homogeneity:** Q(Ri) = TRUE; all pixels in a region satisfy a property (e.g. same gray level).
5. **Distinctness:** Q(Ri ∪ Rj) = FALSE for adjacent regions Ri and Rj; adjacent regions differ.

Approaches:

- **Discontinuity-based:** detect points, lines and edges (gradient, Laplacian, Hough), then link.
- **Similarity-based: thresholding** (global T: g = 1 if f > T), **region growing**, **region splitting and merging**.
- **Clustering and watershed** methods.



Segmentation approaches

Applications: medical image analysis, object recognition, satellite imagery, character recognition.

Conclusion: segmentation partitions the image into disjoint, homogeneous, connected regions that together cover the image.

Q3. Explain the Hough transform for edge linking. How can it detect lines in an image? (2019)

The **Hough transform** is a global method that links edge points lying on a particular shape by mapping them to a parameter space and voting.

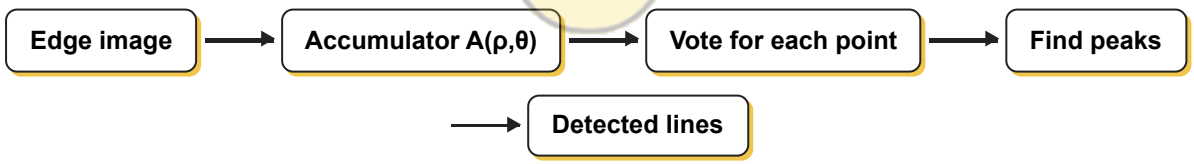
Line detection: a line $y = mx + c$ passes through (x_i, y_i) if $y_i = m\,x_i + c$. In **(m, c) space**, each image point becomes a line, and points collinear in the image give lines that intersect at one point (m, c). Because m is infinite for vertical lines, the **normal form** is used:

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

where ρ is the perpendicular distance of the line from the origin and θ its angle. Each image point corresponds to a sinusoidal curve in (ρ, θ) space; curves of collinear points meet at one (ρ, θ).

Algorithm:

1. Detect edge points (e.g. Canny).
2. Quantise the (ρ, θ) space into an **accumulator array** A(ρ, θ), initialised to zero.
3. For each edge point and each θ, compute $\rho = x \cos\theta + y \sin\theta$ and increment A(ρ, θ).
4. Find **peaks** in A above a threshold; each peak is a line.
5. Link edge points belonging to each line, filling gaps shorter than a limit.



Hough transform for lines

Example: three points (1, 1), (2, 2), (3, 3) with θ = 135° give $\rho = x\cos135° + y\sin135° = 0$ for all, so A(0, 135°) = 3, a peak, and the line $y = x$ is detected.

Advantages: robust to gaps and noise, finds multiple lines; can be extended to circles and ellipses. **Disadvantages:** high memory and computation cost.

Conclusion: the Hough transform detects and links collinear edge points through voting in parameter space.

