Morphological Image Processing and Image Segmentation UNIT-IV

P. Balamurugan, Assistant Professor
PG & Research Department of Computer Science
Government Arts College, Coimbatore -641018
Email: spbalamurugan@rediffmail.com

Morphological Operations

- Basic Morphological Operations
    1. Dilation
    2. Erosion
    3. Opening
    4. Closing
    5. Hit-or-Miss Transformation
- **Dilation** adds pixels to the boundaries of objects in an **image**, while erosion removes pixels on object boundaries.
- Dilation of A by B is $A \oplus B$.
- **Dilation** (usually represented by $\oplus$) is one of the basic operations in mathematical morphology. Originally developed for binary images, it has been expanded first to grayscale images, and then to complete lattices.

Morphological Operations

- Erosion (usually represented by $\ominus$) is one of two fundamental operations (the other being dilation) in morphological image processing from which all other morphological operations are based.
- Erosion of A by B is $A \ominus \mathbf{B}$
- The number of pixels added or removed from the objects in an **image** depends on the size and shape of the structuring element used to **process** the **image**.
- **Opening** is the dilation of the erosion of a set A by a structuring element B: $A \circ B = (A \ominus B) \oplus B$

Morphological Operations

- In **image processing**, **closing** is, together with opening.
- Opening removes small objects, while **closing** removes small holes.

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

- Hit-or-Miss Transform (HMT) - To detect an object in an image:
- Basic Idea: Use the object as Structural Element (se) for erosion of A and detect possible fits.
- Use the neighborhood of the object as Structural Element (se) for erosion of A^c and find over fits.
- Combine the two to detect exact fits.

Morphological Algorithms

- Boundary Extraction

  Boundary of A is computed as: $\beta(A) = A - (A \; B)$

- Region Filling

  Fills a regions, whose boundary is given as 8-connected neighbours:

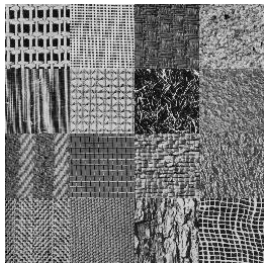
  $X_k = (X_{k-1} \oplus B) \cap A^c, \; k = 1, 2, 3$

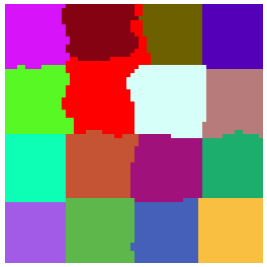
- Connected Components
- Convex Hull
- Thinning
- Thickening
- Skeletonization
- Pruning

Image Segmentation

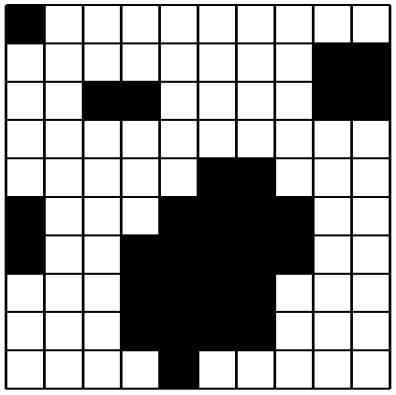
What is Image Segmentation?

- subdivide an image into its component regions or objects.





- assigns a label to every pixel in an image such that pixels with the same label share certain visual characteristics.



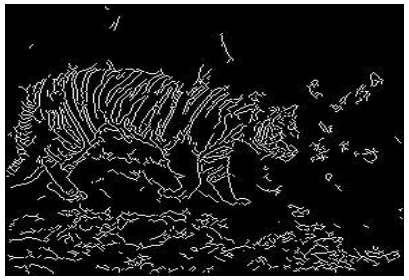
1									
								5	5
		3	3					5	5
					4	4			
2				4	4	4	4		
2			4	4	4	4	4		
			4	4	4	4			
			4	4	4	4			
				4					

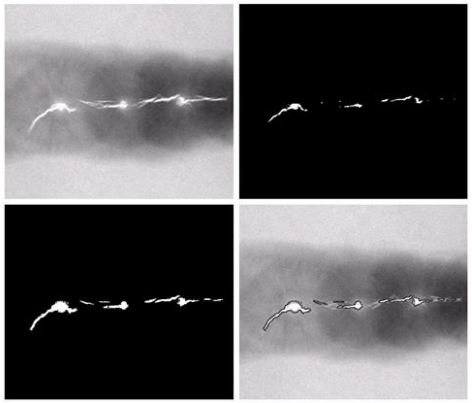
Segmentation - Applications

- Medical imaging
  - Locate tumors and other pathologies
  - Measure tissue volumes
  - Computer-guided surgery
  - Diagnosis
  - Treatment planning
  - Study of anatomical structure
- Locate objects in satellite images (roads, forests, etc.)
- Face recognition
- Iris recognition
- Fingerprint recognition
- Traffic control systems
- Agricultural imaging – crop disease detection

Segmentation - Types

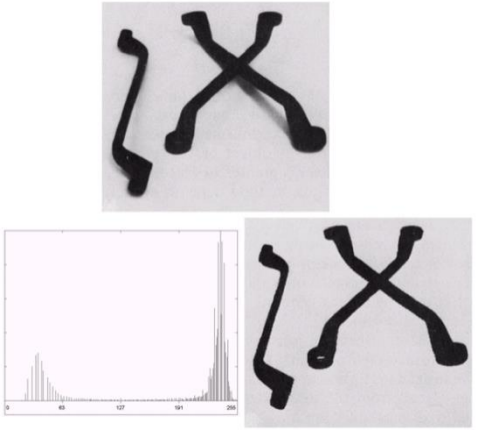
Edge-based - Search for discontinuities , and try to connect objects or borders

<<  



>> *Region-based – Group similar pixels:*
*region gr*owing, merge & split

Thresholding - Based on pixel intensities, often using the shape of the histogram



Edge function

>>[g, t]=edge(f, ‘method’, parameters)

f is the input image, method is one of the approaches listed in table

Edge Detector	syntax
sobel	[g,t]=edge(f,’sobel’,T,dir), T is the specified threshold and dir specifies the preferred direction of the edges which can be ‘horizontal’, ‘vertical’ or ‘both’ (default)
prewitt	[g,t]=edge(f,’prewitt’,T,dir)
roberts	[g,t]=edge(f,’roberts’,T,dir)
Laplacian of gaussian(LoG)	[g,t]=edge(f,’log’,T,sigma), sigma is the standard deviation whose default value is2
canny	[g,t]=edge(f,’canny’,T,sigma)

Region-Based Approach

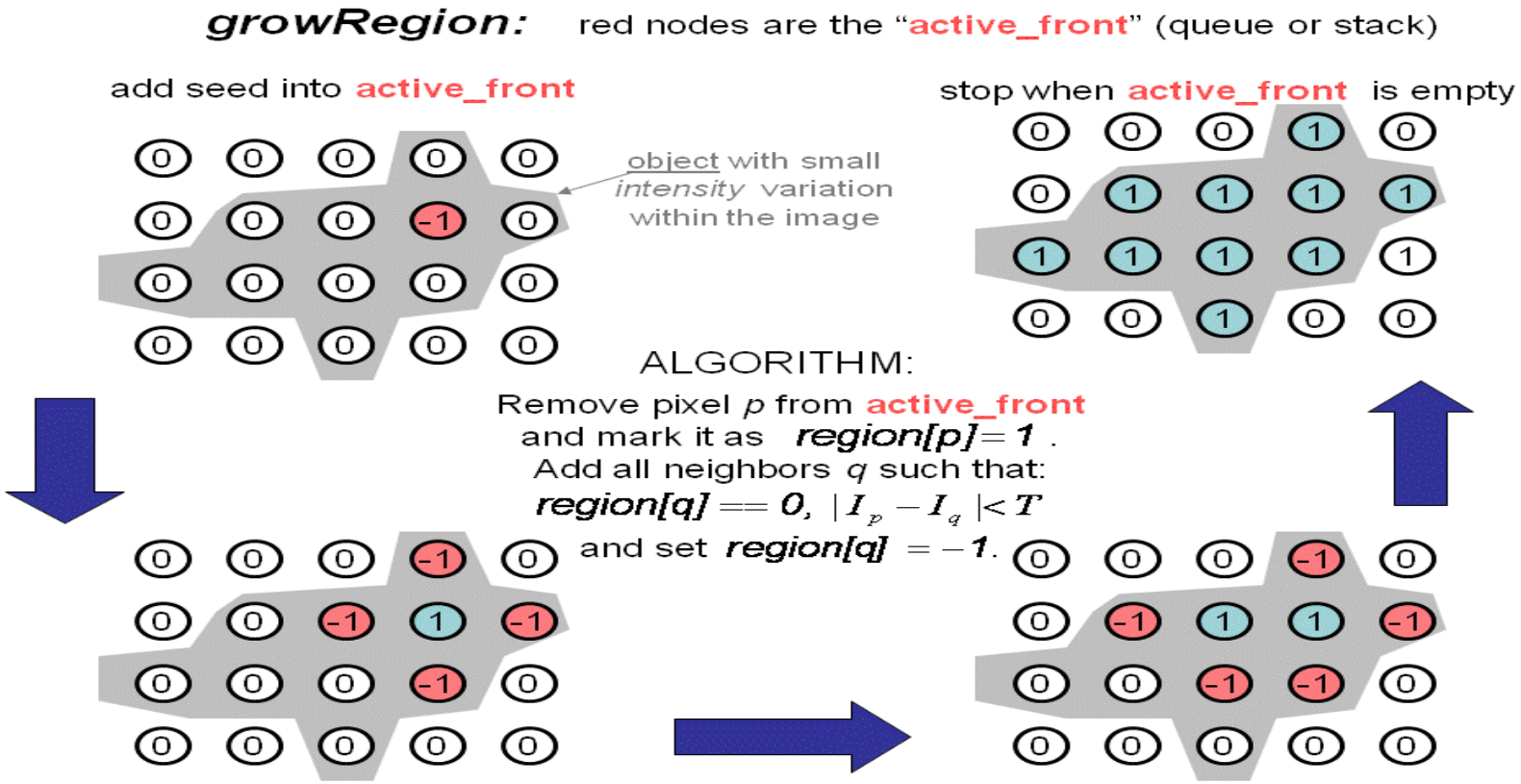
- Pixels corresponding to an object grouped together and marked.
- Region - A group of connected pixels with similar properties.

Region Growing – Step 1

- start with a set of "**seed**" points (starting pixels)
  - Predefined seeds
  - All pixels as seeds
  - Randomly chosen seeds

Region Growing – Step 2

- Growing by appending to each seed those neighbors that have similar properties (bottom-up method)
- Include neighboring pixels with similar features (gray level, texture, color) A similarity measure must be selected



Thresholding - Foundation

- Dark objects on the light background
- One obvious way to extract the objects from the background is to select a threshold 'T' that separates these modes
- Then any point (x,y) for which f(x,y) < T is called an object point, otherwise, the point is called a background point

A few unanswered questions to address

- How do we define similarity measure *S?*
- What threshold *T do we use? Does it change or stay constant?*

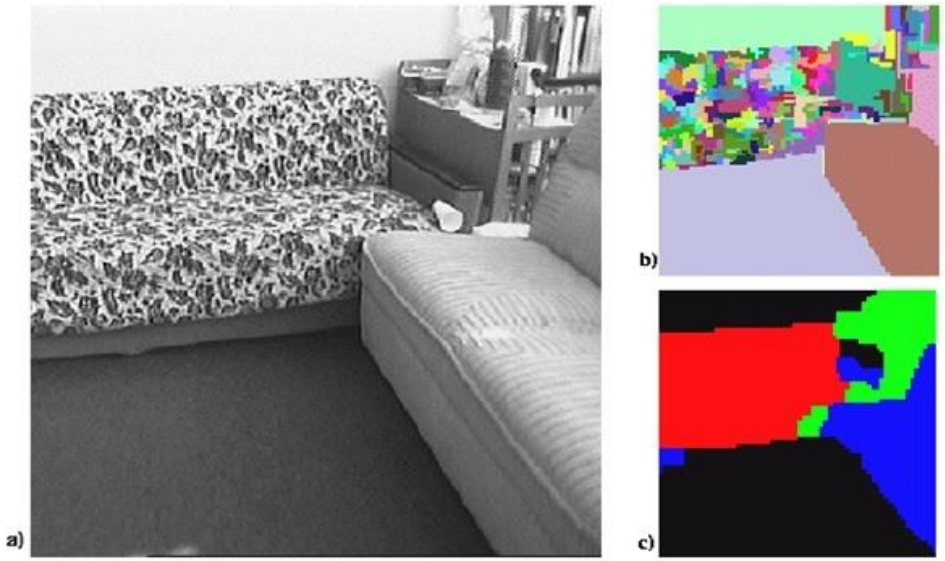
Similarity Measures

Gray Value

- Gray value differences
- Gray value variance

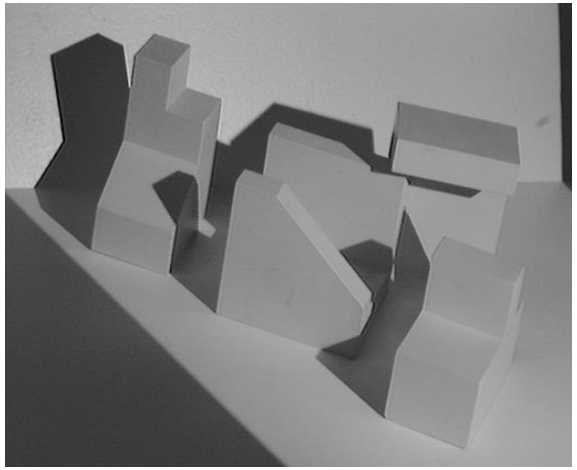
Texture

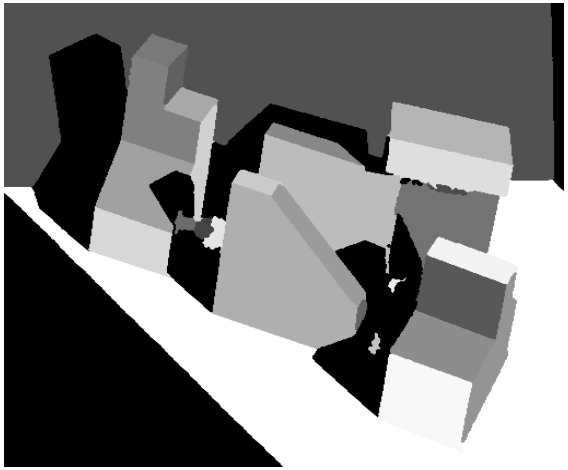
- Enables object surfaces with varying patterns of grey to be segmented



Similarity measure – Depth

- This example shows a range image, obtained with a laser range finder
- A segmentation based on the range (the object distance from the sensor)





Watershed Segmentation

- The **watershed** is a classical **algorithm** used for **segmentation**, that is, for separating different objects in an **image**. Starting from user-defined markers, the **watershed algorithm** treats pixels values as a local topography (elevation).

Step- 1: Read in the Color Image and Convert it to Grayscale. ...

Step- 2: Use the Gradient Magnitude as the **Segmentation** Function.

Step- 3: Mark the Foreground Objects.

Step- 4: Compute Background Markers.

Step- 5: Compute the Watershed Transform of the **Segmentation** Function.

Step 6: Visualize the Result.

References

- https://www.slideshare.net/CristinaPrezBenito/simultaneous-smoothing-and-sharpening-of-color-images.
- https://www.javatpoint.com/digital-image-processing-tutorial.
- https://www.tutorialspoint.com/dip/index.html.

TEXT BOOKS

1) Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Second Edition, PHI/Pearson Education.
2) Alexander M., Abid K., "OpenCV-Python Tutorials", 2017.

REFERENCE BOOKS

1) B. Chanda, D. Dutta Majumder, "Digital Image Processing and Analysis", PHI, 2003.
2) Nick Efford, "Digital Image Processing a practical introducing using Java", Pearson Education, 2004.