Topic 4 – Image Segmentation

What is Segmentation? Why?

- Segmentation → important contributing factor to the success of an automated image analysis process
- What is Image Analysis: Processing images to derive information about various structures within the image. (area of the object, perimeter, No. of objects): Object Recognition
- First Step in analysis - Segmentation
• Function of Segmentation:
  subdivides an image into its constituent regions or objects.
  → Level of division depends on the problem (Known and unknown image environment)
  → Subdivision should stop when the objects of interest have been isolated eg. skin region

• Image segmentation for monochrome image → based on 2 basic properties of intensity values:
  
  (I) **Discontinuity** → partition an image based on abrupt changes in intensity.
  
  Eg. Point, Line, Edge Detection in an image

  (II) **Similarity** → partition an image based on similarity to a set of predefined criteria
  
  Eg. Thresholding, region growing, split and merge.
**Detection of Discontinuities**

- 3 basic types of Discontinuities in Digital Image:
  - (i) Point
  - (ii) Line
  - (iii) Edge

- Most common method to detect discontinuities:
  - Run mask over the image (same as spatial filtering)

\[
R = w_1z_1 + w_2z_2 + w_3z_3 + \ldots + w_9z_9
\]

<table>
<thead>
<tr>
<th>(w_1)</th>
<th>(w_2)</th>
<th>(w_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(w_4)</td>
<td>(w_5)</td>
<td>(w_6)</td>
</tr>
<tr>
<td>(w_7)</td>
<td>(w_8)</td>
<td>(w_9)</td>
</tr>
</tbody>
</table>

**Point Detection**

- Isolated Point will be different from its surroundings and can be detected using this mask.
- Mask is same as one of the masks shown in 2D Laplacian but the difference:

\[
|R| \geq T
\]

; \(T = \) Threshold

**Detected Point**
Example of Point Detection

(a) X-ray Image of a Turbine Blade with a porosity in the upper right quadrant of the image

(b) Result of Point Detection

(c) Result of Using \( |R| \geq T \)

\( T = 90\% \) of the highest absolute pixel value of the image

Line Detection

<table>
<thead>
<tr>
<th>Horizontal</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 -1 -1</td>
<td>-1 2 -1</td>
</tr>
<tr>
<td>2 2 2</td>
<td>-1 2 -1</td>
</tr>
<tr>
<td>-1 -1 -1</td>
<td>-1 2 -1</td>
</tr>
</tbody>
</table>

- Mask can detect 1 pixel thick lines only
- If all the 4 masks are run individually through an image:
  \( |R_i| > |R_j|; j \neq i \)
  then that point is likely to be associated with the direction in mask \( i \)

Eg. \( R_i \) = Vertical Line Mask, \( R_j \) = The 3 other masks, so, the point will be likely to be associated to a Vertical Line
Example of Line Detection

- If we need to detect a line in a specific direction:
  (i) Run the mask for the line
  (ii) Threshold the absolute value of the result

**Detect -45° lines**

![Detect -45° lines](image)

- Absolute value of result after processing with -45° line detector
- Result of Thresholding the Absolute Value (Threshold = maximum value in the image)

Edge Detection

- Most common method to detect discontinuities
- Used to derive structural, shape information
- Basic approach – to compute local derivative → using 1\textsuperscript{st} and 2\textsuperscript{nd} order derivative
- Several method to detect edges - Problem
What is an Edge?

- Edge can be defined as the change in intensity profile

Roof Type
- Discontinuity in the gradient profile
- One peak in the 2nd derivative

Step Type
- Discontinuity in the intensity profile
- One peak in the gradient
- 2 peaks of the opposite signs in the 2nd derivative
  - i.e. a zero-crossing

Characterising an edge

- **Roof** Type
  - Discontinuity in the gradient profile
  - One peak in the 2nd derivative

- **Step** Type
  - Discontinuity in the intensity profile
  - One peak in the gradient
  - 2 peaks of the opposite signs in the 2nd derivative
    - i.e. a zero-crossing
• Thickness of the edge: determined by the length of the ramp
• Length of ramp: determined by the slope of the ramp (determined by degree of blurring)

**Blurred Edges: THICK and Sharp Edges: THIN**

**Observation:**
1. Magnitude of the 1st Derivative can be used to detect if a point is on an edge
2. Sign of the 2nd Derivative can be used to find whether the edge pixel is on the dark or light side of an edge
3. 2nd Derivative produces 2 values for every edge and an imaginary straight line will cross zero near the midpoint of the edge (called the zero-crossing property of 2nd derivative useful for locating the centers of thick edges)
Edge Detection

• **Gradient** based
  - Compute first derivative (gradient) and threshold this image

• **Zero-crossing** based
  - Compute second derivative, locate zero crossings and threshold this image

---

Gradient based edge detection

• **Roberts**
  - 2x2 masks
  - Difference operation in 2 diagonal directions

• **Sobel**
  - gradient operator + smoothing

• **Prewitt**
  - 3x3
  - gradient operator + smoothing
Gradient operators

Roberts

\[
\begin{bmatrix}
0 & -1 \\
1 & 0
\end{bmatrix}
\]

Prewitt

\[
\begin{bmatrix}
-1 & 0 & 1 \\
-1 & 0 & 1 \\
-1 & 0 & 1
\end{bmatrix}
\]

Sobel

\[
\begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1
\end{bmatrix}
\]

-1 0 1
0 0 0
1 1 1

X-Direction

Y-Direction

X & Y-Direction

With Smoothing using Averaging Filter
Zero-crossing based edge detection

- **Laplacian**
  - 2nd derivative operator
  - Can only locate edges, can’t give edge direction
  - Very sensitive to noise thus, will produce undesirable double edges

- **Smoothed Laplacian**
  - Noise reduction through smoothing by using Gaussian function
  - Known as Laplacian of Gaussian (LoG)
  - Approximated as difference of Gaussians (DoG)
    - used in HVS (simple cells)

Second derivative operators

<table>
<thead>
<tr>
<th>Masks</th>
<th>Mexican Hat</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 -1 0</td>
<td>1 -2 1</td>
</tr>
<tr>
<td>-1 4 -1</td>
<td>-2 4 -2</td>
</tr>
<tr>
<td>0 -1 0</td>
<td>1 -2 1</td>
</tr>
</tbody>
</table>

\[
\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]

Laplacian of a Gaussian (LoG)

\[
(\nabla^2 G) * f(x, y)
\]
Canny Edge Detector

- An optimal edge detector
  - optimal in detection, localization and giving single response
- **Main proc. stages**
  - Gaussian smoothing – removes noise
  - Edge direction estimation – from gradient image
  - Non-maxima suppression – by following edge direction to get single response to edges
  - Thresholding (hysteresis) – dual threshold to obtain clean contours

Canny edge detector

- **Performance**
  - Very good results
  - Computationally very expensive
Edge based segmentation

Compute gradient and threshold the gradient image

Problems in edge based segmentation

Sensitivity to noise
Comparisons of edge detection methods

- Coin Image
- Prewitt
- LoG
- Canny

Thresholding

- Original Image and Histogram
- After Thresholding
Thresholding

- Division of image into 2 uniform regions
  - Object vs background
- Types of thresholding
  - Interactive (w. mouse input)
  - Automatic
    - Fixed
    - Adaptive
      - global
      - local
      - dynamic

Fixed thresholding

- **When to use**: grey values of object vs background are known
  
  \[ x[m,n] = 1 \text{ for } g_1 < x < g_2 \]
  
  \[ = 0 \text{ otherwise} \]

- **Variation**: fixed grey value AND some other criterion
  
  - Ex. Proximity of the candidate to an object pixel
Adaptive Thresholding

- **Threshold**, \( t = f\{[m,n], p_l, x[m,n]\} \)
  - \( x \) is the given image
  - \( x[m,n] = \) gray-level values
  - \( p_l \) is the local property eg. of local property \( \bar{\text{average gray value of a neighborhood centered on (m,n)}} \)

  (i) **Global**: \( t = f\{x[m,n]\} \)

  (ii) **Local**: \( t = f\{p_l, x[m,n]\} \)

  (iii) **Dynamic**: \( t = f\{[m,n], p_l, x[m,n]\} \)

Global Thresholding

- Applied to the whole image
- Uses image data to determine the threshold
  - Brightness histograms \( h[g] \) are popular
  - Ideally need well defined peaks and troughs
  - Single or dual thresholds
  - Computationally less intensive
Global Thresholding

- Applied after dividing x[m,n] into uniformly brightness patches
- Patch boundaries need to be post processed
- Combinations of local statistics and other derived information are used to find \( t \)
  - ex. Texture, entropy etc.
- Very effective for handling non-uniform illumination
- Computationally more intensive

Local Thresholding
Selecting the Threshold

Number of peaks and definition of troughs influence threshold selection

Difficult Case

Histogram Smoothing

- Histogram smoothing is used to select thresholds when histograms are ragged
- Threshold selection is made easier
- Care must be taken not to shift peaks
Thresholding - Example

Scanned doc

After global thresholding

After adaptive thresholding (local)

\[ t = \text{mean in 7x7 window} \]

\[ t = \text{local mean} - k \]
Thresholding Example

Threshold Selection on Boundary Characteristics

- To obtain a good segmentation, the selection of threshold is easy if the histogram peaks are tall and separated by deep valleys.

- **Solution**: to consider only the **pixels that lie on or near the edges between objects and background** (using pixels that satisfy some simple measures based on gradient and Laplacian operators).

\[
s(x, y) = \begin{cases} 
0 & \text{if } \nabla f < T \\
+ & \text{if } \nabla f \geq T \text{ and } \nabla^2 f \geq 0 \\
- & \text{if } \nabla f \geq T \text{ and } \nabla^2 f < 0 
\end{cases}
\]

- Pixels not on an edge
- Pixels on the dark side of an edge
- Pixels on the light side of an edge
Local Thresholding

**FIGURE 10.36**
Image of a handwritten stroke coded by using Eq. (10.3-16). (Courtesy of IBM Corporation.)

**FIGURE 10.37**
(a) Original image. (b) Image segmented by local thresholding. (Courtesy of IBM Corporation.)

**FIGURE 10.38**
Histogram of pitches with gradients greater than 5. (Courtesy of IBM Corporation.)
Region-based approaches

- Previously: Edge detection → segmentation by finding the boundaries between regions based on discontinuities in gray levels
- Thresholding → segmentation based on the distribution of pixel properties eg. gray-level values
- In this section: Region-based approach → segmentation based on finding the regions directly (based on similarity)
- Suitable for segmenting non-uniform regions that are perceived to be uniform
- Grouping by similarity and spatial proximity

Region Growing

- Region growing:
  a procedure that groups pixels or subregions into larger regions based on predefined criteria
- Basic Approach to implement Region Growing:
  - Start with ‘seed’ points
  - From the ‘seed’ points, grow the regions by adding to each seed those neighboring pixels that have similar properties to the seed (specific range of gray level)
- Stopping Rule for Region Growing: growing of a region should stop when no more pixels satisfy the criteria for inclusion in that region
- Additional criteria (besides gray-level) like size and shape can be included in region growing to obtain better segmentation
Region Growing Example

2 criteria for growing (including a pixel in the seed region):

a) Absolute Gray-level Difference between any pixel and seed must be < 65
b) Pixels had to be 8-connected to at least one pixel in that region

Region Splitting and Merge

- Region Split and Merge:
  a procedure that subdivide an image into a set of disjointed regions and later, merge and/or split the regions according to certain conditions

- Basic Approach to implement Region Split and Merge:
  - Split into 4 disjoint quadrants any region \( R_i \) for which \( P(R_i) = \text{FALSE} \)
  - Merge any adjacent regions, \( R_j \) and \( R_k \) for which \( P(R_j \cup R_k) = \text{TRUE} \)
  - Stop when no further merging or splitting possible
Region Splitting and Merge

In a block, Split the block if

Threshold, \( T = g_{\text{max}} - g_{\text{min}} > 0.27 \); \( g \) = gray-level

In MATLAB, function ‘qtdecomp’