Image Segmentation

• An important step in image analysis is to segment the image.
• Segmentation: subdivides the image into its constituent parts or objects.
• Autonomous segmentation is one of the most difficult tasks in image processing.
• Segmentation algorithms for monochrome images generally are based on two basic properties of gray-level values:
   1. Discontinuity
   2. Similarity
Image Segmentation

• Discontinuity: the image is partitioned based on abrupt changes in gray level.
• Similarity: partition an image into regions that are similar
  – Main approaches are thresholding, region growing and region splitting and merging.
Detection of Discontinuities

- **Line detection**: If each mask is moved around an image it would respond more strongly to lines in the mentioned direction.

![Figure 10.3](image)

**FIGURE 10.3** Line masks.
Detection of Discontinuities

- There are 3 basic types of discontinuities in digital images:
  1. Point
  2. Line
  3. Edges.

Mask: \[ R = \sum_{i=1}^{9} w_i z_i \]
Detection of Discontinuities

- Point detection: detect an isolated point

\[ |R| \geq T \]

<table>
<thead>
<tr>
<th>-1</th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>8</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

**FIGURE 10.2**
(a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)
FIGURE 10.4
Illustration of line detection.
(a) Binary wire-bond mask.
(b) Absolute value of result after processing with −45° line detector.
(c) Result of thresholding image (b).
Edge detection

- **Edge**: boundary between two regions with relatively distinct gray levels.

---

**FIGURE 10.5**
(a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.
Edge detection

- Basic idea: computation of a local derivative operator.

**FIGURE 10.6**
(a) Two regions separated by a vertical edge.
(b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.
Edge detection

- The magnitude of the first derivative can be used to detect an edge.
- The sign (zero crossing) of the second derivative can be used to detect an edge.
- The same idea can be extended into 2-D. 2-D derivatives should be used.
- The magnitude of the gradient and sign of the Laplacian are used.
Gradient Operator

Gradient
\[ \nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \]

Gradient magnitude
\[ \nabla f = \text{mag}(\nabla f) = \sqrt{\left(G_x^2 + G_y^2\right)} \quad \nabla f \approx |G_x| + |G_y| \]

Gradient direction
\[ \alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \]
FIGURE 10.8
A $3 \times 3$ region of an image (the $z$’s are gray-level values) and various masks used to compute the gradient at point labeled $z_5$. 
**FIGURE 10.9** Prewitt and Sobel masks for detecting diagonal edges.
FIGURE 10.10
(a) Original image. (b) $|G_x|$, component of the gradient in the x-direction.
(c) $|G_y|$, component in the y-direction.
(d) Gradient image, $|G_x| + |G_y|$. 
FIGURE 10.11
Same sequence as in Fig. 10.10, but with the original image smoothed with a $5 \times 5$ averaging filter.
FIGURE 10.12
Diagonal edge detection.
(a) Result of using the mask in Fig. 10.9(c).
(b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).
Laplacian

• Laplacian in its original form is not used for edge detection because:
  1. It is very sensitive to noise
  2. It’s magnitude produces double edges
  3. Unable to detect the direction of an edge.
• To solve the first problem, the image is low-pass filtered before using the Laplacian operator.

\[
f(x,y) \xrightarrow{h(x,y)} \nabla^2 \xrightarrow{} g(x,y)
\]
Laplacian

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

\[ \nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \]

\[ \nabla^2 f = 8z_5 - \sum_{i=1}^{8} z_i \]
Laplacian

Convolution: \( g(x, y) = f(x, y) \ast \nabla^2 h(x, y) \)

Gaussian function: \( h(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \)

\( r^2 = x^2 + y^2 \)

Second derivative: \( \nabla^2 h(r) = \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \)

- Cross section of \( \nabla^2 h \) has a Mexican hat shape.
- The average \( \nabla^2 h \) is zero. The average of image convolved with \( \nabla^2 h \) is also zero.
- We will have negative pixel values in the result.
Laplacian

- To solve the problem of double edges, zero crossing of the output of Laplacian operator is used.
FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)
FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)
Edge linking and Boundary Detection

• The output of discontinuity detection stage seldom characterizes a boundary completely.
• This is because of noise, breaks in the boundary and other effects.
• Edge detection algorithms typically are followed by linking and boundary detection procedures.
• These procedures assemble edge pixels into meaningful boundaries.
Edge linking and Boundary Detection

- All the points that are similar are linked forming a boundary of pixels.
- Two principle properties used for establishing similarity of edge pixels in this kind of analysis are:
  1. The strength of the response of the gradient operator used to produce the edge pixels
  2. The direction of the gradient

\[
\left| \nabla f(x, y) - \nabla f(x_0 + y_0) \right| \leq E
\]
\[
|\alpha(x, y) - \alpha(x_0, y_0)| < A
\]
FIGURE 10.16
(a) Input image.
(b) $G_y$ component of the gradient.
(c) $G_x$ component of the gradient.
(d) Result of edge linking. (Courtesy of Perceptrics Corporation.)
Hough Transform

- \((x_i, y_i)\): all the lines passing this point \(y_i = ax_i + b\)
- \(b = y_i - ax_i\): point \((x_i, y_i)\) maps to a single line in \(ab\) plane.
- Another point \((x_j, y_j)\) also has a single line in \(ab\) plane
  \[ b = y_j - ax_j \]
- \(a'\) and \(b'\) are the slope and intercept of the line containing both \((x_i, y_i)\) and \((x_j, y_j)\).
Hough transform

- \((a_{min}, a_{max})\): expected range of slope
- \((b_{min}, b_{max})\): expected range of intercepts
- \(A(i,j)\): the number of points in the cell at coordinates \((i,j)\) in the \(ab\) plane.

For every point in the image plane, we let the value of \(a\) equal each of the allowed subdivisions and find the corresponding \(b\) from \(b = y_i - ax_i\). If for \(a_p\) we get \(b_q\), the \(A(p,q)\) is incremented.
Hough transform

- The cell with the largest value shows the parameters of the line that contains the maximum number of points.
- Problem with this method: $\alpha$ approaches infinity as the line gets perpendicular to the $x$ axis.
- Solution: use the representation of the line as:
  \[ x \cos \theta + y \sin \theta = \rho \]
Hough transform

- A point in $xy$ plane is mapped into a sinusoidal curve in $\rho \theta$ plane.

*FIGURE 10.19*
(a) Normal representation of a line.
(b) Subdivision of the $\rho \theta$-plane into cells.
FIGURE 10.20
Illustration of the Hough transform.
(Courtesy of Mr. D. R. Cafe, Texas Instruments, Inc.)
Hough transform

- Hough transform is applicable to any function of the form $g(v,c) = 0$, where $v$ is the vector of coordinates and $c$ is the vector of coefficients.

- Exp: $(x - c_1)^2 + (y - c_2)^2 = c_3^2$

- 3 parameters $(c_1, c_2, c_3)$, 3-D parameter space, cube-like cells, accumulators of the form $A(i,j,k)$.

- Procedure:
  1. Increment $c_1$ and $c_2$
  2. Solve for $c_3$
  3. Update the accumulator associated with $(c_1, c_2, c_3)$
Thresholding

• Thresholding is one of the most important approaches to image segmentation.
• Gray level histogram of an image $f(x,y)$ composed of a light object on a dark background.
• To extract the object: select a threshold $T$ that separates the gray levels of the background and the object.
Thresholding

- Single threshold: points with \( f(x,y) > T \) belong to object; other points belong to background.
- Multiple thresholds: points with \( f(x,y) > T_2 \) belong to object; points with \( f(x,y) < T_1 \) belong to background.

**Figure 10.26** (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.
Thresholding

• Threshold in general can be calculated as:
  \[ T = T(x, y, p(x, y), f(x, y)) \]
  \( f(x, y) \): gray level at \((x, y)\)
  \( p(x, y) \): some local property of the point \((x, y)\) (e.g., the average gray level of a neighborhood centered on \((x, y)\)).

  \( T \) depends only on \( f(x, y) \): global threshold
  \( T \) depends on \( f(x, y) \) and \( p(x, y) \): local threshold
  \( T \) depends on \( f(x, y) \) and \( p(x, y) \) and \( x, y \): dynamic threshold
Thresholding

FIGURE 10.28
(a) Original image, (b) Image histogram, (c) Result of global thresholding with $T$ midway between the maximum and minimum gray levels.
Thresholding

MR brain image (top left), its histogram (bottom) and the segmented image (top right) using a threshold $T=12$ at the first major valley point in the histogram.
Thresholding

Two segmented MR brain images using a gray value threshold $T=166$ (top right) and $T=225$ (bottom)
Basic Global Thresholding

1. Select an initial estimate for $T$
2. Segment the image using $T$. This will produce two group of pixels $G_1$ (with gray level greater than $T$) and $G_2$ (with gray level less than $T$)
3. Compute average gray level values $\mu_1$ and $\mu_2$ for pixels in regions $G_1$ and $G_2$
4. Compute a new threshold: $T = (\mu_1 + \mu_2)/2$
5. Repeat step 2 through 4 until the difference in $T$ in successive iterations is smaller than a predefined value
Basic Global Thresholding

FIGURE 10.29
(a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)
Basic Adaptive (Local) Thresholding

FIGURE 10.30
(a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.
Optimal Global Thresholding

\[ p(z) = P_1 \cdot p_1(z) + P_2 \cdot p_2(z) \]

\[ P_1 + P_2 = 1 \]

**Problem:** how to optimally determine \( T \) to minimize the segmentation error?
Optimal Global Thresholding

Error 1: \( E_1(T) = \int_{-\infty}^{T} p_2(z)dz \)

Error 2: \( E_2(T) = \int_{T}^{\infty} p_1(z)dz \)

Total error: \( E(T) = P_1 E_2(T) + P_2 E_1(T) \)

Goal: what’s the value of \( T \) to minimize \( E(T) \)?

Result: \( P_1 p_1(T) = P_2 p_2(T) \)
Optimal Global Thresholding: Gaussian PDFs

\[ p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(z - \mu_1)^2}{2\sigma_1^2}\right) + \frac{P_2}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(z - \mu_2)^2}{2\sigma_2^2}\right) \]

Solution: \[ AT^2 + BT + C = 0 \]

\[ A = \sigma_1^2 - \sigma_2^2 \]

\[ B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2) \]

\[ C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln(\sigma_2P_1 / \sigma_1P_2) \]
Region Based Segmentation

- Let $R$ represent the entire image. Segmentation is a process that partitions $R$ into $n$ subregions $R_1, R_2, \ldots, R_n$ such that:

  a) $\bigcup_{i=1}^{n} R_i = R$
  
  b) $R_i$ is a connected region.
  
  c) $R_i \cap R_j = \emptyset$
  
  d) $P(R_i) = True$
  
  e) $P(R_i \cup R_j) = False$
Region Growing

- Pixel aggregation: starts with a set of seed points and from these grows regions by appending to each seed point those neighboring pixels that have similar properties (e.g., gray-level, texture, color).
Region Growing

- Center Pixel
- Pixels satisfying the similarity criterion
- Pixels unsatisfying the criterion
- 3x3 neighborhood
- 5x5 neighborhood
- 7x7 neighborhood

Segmented region
Region Growing

A T-2 weighted MR brain image (left) and the segmented ventricles (right) using the region-growing method.
Region growing

• Problems:
  1. Selection of initial seeds that properly represent regions of interest
  2. Selection of suitable properties
  3. Formulation of a stopping rule.
Seeded Region Growing
Region Growing in a Diffusion Weighted Image
Region Splitting and Merging

- Sub-divide an image into a set of disjoint regions and then merge and/or split the regions in an attempt to satisfy the condition (P).

**FIGURE 10.42**
(a) Partitioned image.
(b) Corresponding quadtree.

<table>
<thead>
<tr>
<th></th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_3$</th>
<th>$R_{41}$</th>
<th>$R_{42}$</th>
<th>$R_{43}$</th>
<th>$R_{44}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>$R_{41}$</td>
<td>$R_{42}$</td>
<td>$R_{43}$</td>
<td>$R_{44}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_2$</td>
<td>$R_{41}$</td>
<td>$R_{42}$</td>
<td>$R_{43}$</td>
<td>$R_{44}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_3$</td>
<td>$R_{41}$</td>
<td>$R_{42}$</td>
<td>$R_{43}$</td>
<td>$R_{44}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_4$</td>
<td>$R_{41}$</td>
<td>$R_{42}$</td>
<td>$R_{43}$</td>
<td>$R_{44}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

McMaster University
Region Splitting and Merging

- **Procedure:**
  1. Split into 4 disjoint quadrants any region $R_i$ where $P(R_i) = \text{False}$.
  2. Merge any adjacent regions $R_j$ and $R_k$ for which $P(R_j \cup R_k) = \text{True}$.
  3. Stop when no further merging or splitting is possible.
Region Splitting and Merging

FIGURE 10.43
(a) Original image. (b) Result of split and merge procedure. (c) Result of thresholding (a).
Mathematical Morphology

• Mathematical morphology involves a convolution-like process using various shaped kernels, called structuring elements.

• The structuring elements are mostly symmetric: squares, rectangles, and circles.

• Most common morphological operations are
  – Erosion
  – Dilation
  – Open
  – Close

• The operations can be applied iteratively in selected order to effect a powerful process.
Erosion Functions

• Erosion function is a reduction operator.
• It removes noise and other small objects, breaks thin connections between objects, removes an outside layer from larger objects, and increases the size of holes within an object
• For binary images, any pixel that is 1 and has a neighbor that is 0, is set to 0
• The minimum function is the equivalence of an erosion
• The neighbors considered are defined by the structuring element
Illustration of Erosion Function

Erosion with a 3x3 square structuring element
Example of Erode Function
Erosion Example

Input image

Eroded image
Dilation Function

• The dilation function is an enlargement operator, the reverse of erosion
• For a binary data set, any 0 pixel that has a 1 neighbor, where the neighborhood is defined by the structuring element, is set to 1
• For gray scale data, the dilation is a maximum function
• The dilation fills small holes and cracks and adds layers to objects in a binary image
Example of Dilation
Dilated Image
Erosion – Dilation Functions

• Erosion and dilation are essentially inverse operations, they are often applied successively to an image volume

• An erosion followed by a dilation is called an open

• A morphological open will delete small objects and break thin connections without loss of surface layers

• A dilation followed by an erosion is called close

• The close operation fills small holes and cracks in an object and tends to smooth the border of an object
Example of Open Operation
Example of Open Operation

[Images of an example of an open operation on an image]
Example of Close Operation
Example of Close Operation
An Automated Segmentation

(a) original image, (b) thresholding, (c) erosion, (d) dilation, (e) closing, (f) mask rendering, (g) volume rendering
Active Contours (Snakes)

• Segmenting an object in an image with active contours involves minimizing a cost function based on certain properties of the desired object boundary and contrast in the image.

• Smoothness of the boundary curve and local gradients in the image are usually considered.

• Snake algorithms search the region about the current point and iteratively adjust the points of the boundary until an optimal, low cost boundary is found.

• It may get caught in a local minimum (initial guess).
Example of A Snake Algorithm
Active Contour with Level-Set Method
End of Lecture