# Overview

**Last lecture**
- Practical Foundation of Digital Image Processing III
- Derivative Filtering
- Image Sharpening in Spatial Domain
- Edge Detection in Spatial Domain
- Filter Combination
- Multiple-Image Operation
- Frequency Domain Filtering

**Today’s lecture**
- Advanced Image Processing
- Edge-Preserving Smoothing
- Morphological Operation
- Connected Component Analysis

---

## Advanced Image Processing:

**Edge-Preserving Smoothing**

**Morphological Operations**

**Connected-Components Analysis**

CSC621-821
Biomedical Imaging and Analysis
Dr. Kazunori Okada
Review: Smoothing vs Sharpening

• We learned two concepts: **smoothing** for noise reduction and **sharpening** for edge detection

• We learned how to do them by both spatial and frequency domain image filtering
  – Smoothing **removes fine details** by blurring
  – Sharpening **highlights fine details** by differentiating

• Notice that they trade-off to each other…

• But can we do them together???

Gaussian Smoothing

• Reduce noise by locally averaging intensity values or by blurring an image

  ![Before](image1.png) ![After](image2.png)

• PROBLEM:
  details also gets lost, Edges are gone!!!
**Edge-Preserving Smoothing**

Can we smooth images **without** blurring edges? Can we combine sharpening & smoothing?

- Yes: Edge-Preserving Smoothing
  - Anisotropic Diffusion
  - Median Filtering
  - Rank-Order Filtering
  - Bilateral Filtering
  - Mean Shift Filtering
  - etc

**Diffusion and Gaussian Smoothing**

- Gaussian smoothing is equivalent to subject an image to the **heat diffusion**
  - More time of diffusion, more blurred image
  - Elapsed time $t = \text{Gaussian width } h$
  - Solution of the following **partial differential equation** (heat equation) is a **Gaussian image convolution**!

$$\text{div } I = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial y} + \frac{\partial I}{\partial z} \frac{\partial I(x, y, z)}{\partial t} = \text{div}[c \cdot \nabla I(x, y, z)]$$
**Anisotropic Diffusion**

- Make the diffusion coefficient $c$ as a function of the image gradient
- Set the coefficient to suppress blurring at high gradient points (edge!!!)

\[
\frac{\partial I(x, y, z)}{\partial t} = \text{div}\left[ g\left(\|\nabla I(x, y, z)\|\right) \cdot \nabla I(x, y, z) \right]
\]

**Example**

- Noise is removed AND edge is preserved!
Perona & Malik Anisotropic Diffusion

- A well-known non-linear diffusion
  - Gaussian of gradient for edge-stopping function
  - Less smoothing across high gradient
  - Contrast parameter $k$

$$\frac{\partial I}{\partial t} = \nabla \cdot \left( g(|\nabla I|) \nabla I \right)$$

$$g(x) = e^{-\frac{x^2}{k^2}}$$

Numerical Solution

- You need to solve the nonlinear PDE!
- Use **Euler method** to transform the problem to an **iterative** algorithm with **difference equation**
- ITK: itk::GradientAnisotropicDiffusionImageFilter

$$\frac{\partial I(x,y,z)}{\partial t} = \text{div}[g(|\nabla I(x,y,z)|) \nabla I(x,y,z)]$$
Image Segmentation: Overview

- Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image.
- Typically the first step in any automated image analysis and computer vision application.

Next Two Lectures!!!
Binary Segmentation

- Binary segmentation
  - **Foreground** is assigned by 1s / White
  - **Background** is assigned by 0s / Black
  - 0 and 1 are interchangeable

What is Morphological Operations?

- **Once binary segmentation is complete**, *morphological operations* can be used to remove imperfections in the segmented image and provide information on the form and structure of the image
  
  Cleaning the segmentation results!!!

- *Morphological image processing* (or *mathematical morphology*) describes a range of image processing techniques that deal with the shape of features in an image
Morphological Operations: Overview

- Input: binary segmentation result: 0s for background pixels and 1s for object pixels
- Similar to Spatial Filtering!

*Image after segmentation*  
*Image after segmentation and morphological processing*
Fundamental Operations

- Fundamentally morphological image processing is like spatial filtering
- The **structuring element** (filter) is moved across every pixel in the original image to give a pixel value in a new filtered image
- The value of this new pixel depends on the operation performed

Structuring Elements (SEs)

- SEs is the same as spatial filters with only 0 & 1!
- SEs can be any size and make any shape
- For simplicity we will use rectangular SE with their origin at the middle pixel
- **On-pixel** is those with value 1
**Fit & Hit & Miss**

**Fit:** All on-pixels in the structuring element cover on-pixels in the image

**Hit:** At least one on-pixel in the structuring element covers an on-pixel in the image

All morphological processing operations are based on these simple ideas

---

**Fitting & Hitting Example**

<table>
<thead>
<tr>
<th>0 0 0 0 0 0 0 0 0 0 0 0</th>
<th>1 1 1</th>
<th>Structuring Element 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1 1 0 0 0 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 B 1 1 1 1 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 1 1 1 1 1 1 1 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 1 1 1 1 1 1 1 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1 1 1 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1 1 1 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1 1 1 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1 1 1 0 0 0 0</td>
<td>1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 1 1 1 1 1 0 0</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td>Structuring Element 2</td>
</tr>
</tbody>
</table>

---
Basic Morphological Operations: Break

- Basic Operations
  - Erosion
  - Dilation

- Compound Operations (combinations of the basic operations)
  - Opening
  - Closing

Erosion

- Erosion of image $f$ by a SE $s$ is denoted by $f \ominus s$
- The SE $s$ is positioned with its origin at $(x, y)$ and the new pixel value $g(x, y)$ at the position is determined using the rule:

$$g(x, y) = \begin{cases} 
1 & \text{if } s \text{ fits } f \ominus s(x, y) \\
0 & \text{otherwise}
\end{cases}$$
Erosion Process

**Original Image**

**Processed Image With Eroded Pixels**

**Structuring Element**
Erosion Example 1

Original image

Erosion by 3*3 square structuring element

Erosion by 5*5 square structuring element

Watch out: In these examples a 1 refers to a black pixel!

Erosion Example 2

Original image

After erosion with a disc of radius 10

After erosion with a disc of radius 5

After erosion with a disc of radius 20
What is Erosion For?

- Split apart joined objects
- Strip away extrusions

Watch out: Erosion shrinks objects

Dilation

- Dilation of image $f$ by SE $s$ is denoted by $f \oplus s$
- The SE $s$ is positioned with its origin at $(x, y)$ and the new pixel value $g(x, y)$ at the position is determined using the rule:

$$g(x, y) = \begin{cases} 
1 \text{ if } s \text{ hits } f \text{ at } (x, y) \\
0 \text{ otherwise } \end{cases}$$
Dilation Process

Original Image

Processed Image

Structuring Element

Dilation Process

Original Image

Processed Image With Dilated Pixels

Structuring Element
Dilation Example 1

Original image  Dilation by 3x3 square structuring element  Dilation by 5x5 square structuring element

Watch out: In these examples a 1 refers to a black pixel!

Dilation Example 2

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Structuring element
What is Dilation For?

• Repair breaks

• Repair intrusions

Watch out: Dilation enlarges objects

Opening

• The opening of image \( f \) by SE \( s \), denoted \( f \circ s \), is defined as an erosion followed by a dilation

\[
f \circ s = (f \ominus s) \oplus s
\]

Original shape \( A \) After erosion \( A \ominus B \) After dilation (opening) \( A \cdot B = (A \ominus B) \oplus B \)

Note a disc shaped structuring element is used
Opening Example

Original Image

Image After Opening

Opening Process

Original Image

Processed Image

Structuring Element
Opening Process

Closing

- The closing of image $f$ by SE $s$, denoted $f \circ s$, is defined as a dilation followed by an erosion:

$$f \circ s = (f \oplus s) \ominus s$$

Note a disc shaped structuring element is used.
Closing Example

Original Image

Image After Closing

Closing Process

Original Image  Processed Image

Structuring Element
Closing Process

Original Image

Processed Image

Structuring Element
Compound Opening and Closing: Break

Morphological Algorithms

- Using the techniques we have looked at so far we can begin to consider some more interesting morphological algorithms
  - Boundary extraction
  - Region filling
  - Thinning/thickening
  - Skeletonisation
  - etc
### Review: Neighborhood and Connectivity

- **Neighbors**: a set of pixels in a rectangle region around a center pixel
- **Connectedness**: which pixels/voxels are neighbors?
- **Connectivity**: two pixels $p$ and $q$ are connected *iff* there is a path from $p$ to $q$, on which each pixel is connected to the next.

#### 4-connected 8-connected

```
  0 1 0 1
  1 0 1 0
  0 1 0 1
  0 0 0 0
```

### Region and Boundary

- **Region**: aka, Connected-Component
  - A maximal set of pixels which are all connected to each other

```
  o 1 o 1
  1 0 1 1
  o 0 1 0
  o 0 0 0
```

- **Boundary**
  - A subset of a connected-component which is connected to the background
  - A closed path under certain connectedness
Boundary Extraction

- Extract a boundary of binary segmented object
- This example with a square 3x3 SE

\[ \beta(A) = A - (A \ominus B) \]

Boundary Extraction Process

- The boundary (outline) of an object can be extracted by subtracting erosion result from the original image

\[ \beta(A) = A - (A \ominus B) \]
Region Filling

- You can fill a region with a hole inside a boundary with object pixels (1s) in semi-automatic fashion.

Given a point inside here, can we fill the whole circle?

Region Filling Process

- Iterative algorithm given $X_0$

\[ X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \ldots. \]

where $X_0$ is a binary image with only a single point inside boundary is on, B is a simple structuring element and $A^c$ is the complement of input A

- This equation is applied repeatedly until $X_k$ is equal to $X_{k-1}$

- Finally the result is union-ed with the original input image of boundary

\[ (X^c \cup A) \]
Region Filling Process Cond.

\[ X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \ldots \]

Counting the Number of Regions

- After binary segmentation & morphological operations to clean the seg. results,
- You have distinct regions corresponding to many objects in your image
  - Car, Person, Tree, etc
  - Liver, Kidney, Bones, Stomach, etc
- **How many objects are there?**
- **Where is each object?**
Connected-Components Analysis: Overview

- AKA: Connected-Components Labeling
- Goal: identify distinct regions

![Binary image](image1.png) ![Connected components labeling](image2.png)

Sequential Algorithm

Note: We want to label A. B, C, D are already labeled.

4-connected: consider B & C (prior and superior)
8-connected: consider B & C & D
Sequential Algorithm Cond.

- Process the image from left to right, top to bottom:
  1.) If the next pixel to process is 1
     i.) If only one of its neighbors (top or left) is 1, copy its label.
     ii.) If both are 1 and have the same label, copy it.
  iii.) If they have different labels
       - Copy the label from the left.
       - Update the equivalence table.
     iv.) Otherwise, assign a new label.

- Re-label with the smallest of equivalent labels

Example

- Cellular Image, How many nucleus?

<table>
<thead>
<tr>
<th>Original Image</th>
<th>Binary Segmentation Result</th>
<th>Connected-Component Labeling Result</th>
<th>Color Coding</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Original Image" /></td>
<td><img src="image2.png" alt="Binary Segmentation Result" /></td>
<td><img src="image3.png" alt="Connected-Component Labeling Result" /></td>
<td><img src="image4.png" alt="Color Coding" /></td>
</tr>
</tbody>
</table>

163
What is Segmentation?

- Segmentation attempts to **partition** the pixels of an image into groups that strongly correlate with the objects in an image.

![Segmentation example](image)

What for?

- **Typical scenario**
  - Separate an object from a distinct background

![Segmentation example](image)

- To obtain compact object representation
- **Applications:** Finding tumors, veins, etc. in medical images, finding targets in satellite/aerial images, finding people in surveillance images, summarizing video, etc.
**Why Partitioning is Difficult?**

- Simple strategy: try every possible partitions and pick the best one you like! Why not?
- How many possible partitions given an image?
- A partition of a set with $n$ elements ($n$ pixels/voxels)
- Bell number: $B(n) =$ # of all possible partitions
  - $B(1) = 1$
  - $B(3) = 5$
  - $B(5) = 52$
  - $B(10) = 115975$ (if it takes one second per case, 1.3 days)
  - $B(20) = 5.17 \times 10^{13}$ (well, $6 \times 10^8$ days = 1.6 million years)
  - $B(100) = 4.8 \times 10^{115}$ (this is just a 10 by 10 pixel image...)
- **Too many possibilities!!! We must find a smart way to find a good partition!**

---

**Dimensionality**

- 2D algorithms presuppose 2D images as input
- 3D algorithms presuppose 3D images as input
- Time & Space Complexity
  - 3D algorithms have significantly higher complexities both in time and space (*any double-loop becomes triple-loop*)
- Algorithms that work for both 2D & 3D
  - Methods that rely solely on image intensities are often independent of the image domain dimensionality (e.g., approach based on intensity histograms).
- Compatibility
  - 2D methods can be applied to 3D image slice by slice
  - But this is to be avoided since this approach ignores continuity along z-axis
Partial Volume Effect

- Partial-volume effects are artifacts that occur where multiple tissues types contribute to a single discrete pixel, resulting in blurring of intensities.

![Illustration of partial-volume effect. (a) Ideal image. (b) Acquired image.](image)

Manual Interaction

- The type of interaction can vary from completely manual delineation of an anatomical structure to the selection of a seed point for region growing.
  - No interaction: **Automatic Segmentation**
    - E.g., Thresholding with automatic threshold estimation
  - Minimal interaction: **Semi-Automatic Segmentation**
    - E.g., Region growing with initial seeds
  - Fully manual: Manual Segmentation

- Even automated segmentation requires specification of some initial parameters
In Medical Image Segmentation?

• Manual interaction is an important consideration in medical image segmentation
  – Less interaction is preferred toward reducing labor
  – More interaction can improve accuracy
  – Manual delineation is hard in 3D images
  – Manual delineation by MDs is considered to be most accurate! (ground-truth/gold standard)
  – Building a semi-automatic system in which expert’s manual interaction is a part of system to improve overall accuracy?

Summary

• Advanced Image Processing
  – Edge-Preserving Smoothing
  – Morphological Operation
  – Connected Component Analysis
  – Segmentation: Overview

• Next Lecture
  – Medical Image Segmentation I
  – Thresholding
  – Region growing
  – Watersheds