Binary image processing
Issues
Issues
Issues

- How to demarcate multiple regions of interest?
  - Count objects
Issues

• How to demarcate multiple regions of interest?
  – Count objects
  – Compute further features per object

• What to do with “noisy” binary outputs?
Issues

• How to demarcate multiple regions of interest?
  – Count objects
  – Compute further features per object

• What to do with “noisy” binary outputs?
  – Holes
Issues

• How to demarcate multiple regions of interest?
  – Count objects
  – Compute further features per object

• What to do with “noisy” binary outputs?
  – Holes
  – Extra small fragments
Connectedness

When are two pixels connected?

4-connected neighbors of *

8-connected neighbors of *
**Connected Component:** Set of pixels for which there exists a connected path between all pairs

We need to define 8 or 4-connectedness
Two-pass algorithm for Connected Components

1. Scan the image from left to right, top to bottom; if the pixel is 1 then
   a) if only one of the upper or left pixels has a label, copy this label to current pixel
   b) if both have the same label, copy this label
   c) if they have different labels, copy one label and mark these two labels as equivalent
   d) if there are no labeled neighbors, assign it a new label

2. Scan the labeled image and replace all equivalent labels with a common label
Two-pass algorithm for Connected Components

How many unique labels are assigned in first pass?

How to implement 2-pass algorithm for 8-connectedness?
Dilation

Dilation: “I should turn on if any of my neighbors are turned on”

Neighborhood is defined by a binary mask or “structuring element” (SE)
Example: finding borders of objects

Image
Dilated with 3x3 mask
Dilated - Image
Grayscale dilation

\[ F(x, y) = \max_{(u,v) \in SE} I(x + u, y + v) \]
Erosion

“I should turn off if any of my neighbors are off”

\[ F(x, y) = \min_{(u,v) \in SE} I(x + u, y + v) \]
Opening

- Erode, then dilate
- Remove small objects, keep original shape

Before opening

After opening
Closing

- Dilate, then erode
- Fill holes, but keep original shape
Application: segmentation of a liver

Application by Jie Zhu, Cornell University
Matlab functions

• `bwconnectedcomponent`
  - (compute connected components)

• `bwmorph`
  - (apply erosion, dilation, closing, opening, etc.)

• `regionprops`
  - (compute properties of each con. comp, such as area, centroids, etc.)
Distance Transform Formula

- Set of points, $P$, and measure of distance
  \[ DT(P)[x] = \min_{y \in P} \text{dist}(x,y) \]
- For each location $x$ distance to nearest point $y$ in $P$
Different Distance Measures

- Euclidean distance ($L_2$ norm)
  \[
  \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots}
  \]

- City block distance ($L_1$ norm)
  \[
  |x_1 - y_1| + |x_2 - y_2| + \ldots
  \]

- Chessboard distance ($L_\infty$ norm)
  \[
  \max(|x_1 - y_1|, |x_2 - y_2|, \ldots)
  \]
L₁ Distance Transform Algorithm

- Two pass O(n) algorithm for 1D L₁ norm
  (just distance and not source point)

1. **Initialize**: For all j
   \[ D[j] \leftarrow 1_{p}[j] \]

2. **Forward**: For j from 1 up to n-1
   \[ D[j] \leftarrow \min(D[j], D[j-1]+1) \]

3. **Backward**: For j from n-2 down to 0
   \[ D[j] \leftarrow \min(D[j], D[j+1]+1) \]
**L₁ Distance Transform**

- 2D case analogous to 1D
  - Initialization
  - Forward and backward pass
    - Forward pass adds one to closest above and to left, takes min with self
    - Backward pass analogous below and to right
Chamfer distance

- Average distance to nearest feature