Image Processing I
Exercise Class

WS 2017/2018

Noha Sarhan

Cognitive Vision Group, Department of Informatics
University of Hamburg, Germany

Talk on Monday 20.11.

- "Analyzing Human Behavior in Video Sequences" by Prof. Dr. **Jürgen Gall**, University of Bonn
- On **Monday 20. Nov at 17:15**
- Room **B-201**
- Talk will be in English

strel class

- To create a morphological *structuring element (SE)*
- `SE = strel('disk', R, N)`
  - Creates a disk-shaped SE. **R** specifies the radius. **N** is the number of SEs used to approximate the disk shape

de.mathworks.com

strel class

- To create a morphological *structuring element (SE)*
- `SE = strel('disk', R, N)`
  - Creates a disk-shaped SE. **`R`** specifies the radius. **`N`** is the number of SEs used to approximate the disk shape
- `SE = strel('line', len, deg)`
  - Creates a linear SE. **`deg`** specifies the angle in degrees counterclockwise from the x-axis. **`len`** is the length of the line

de.mathworks.com

strel class

- To create a morphological *structuring element (SE)*
- `SE = strel('disk', R, N)`
  - Creates a disk-shaped SE. **`R`** specifies the radius. **`N`** is the number of SEs used to approximate the disk shape
- `SE = strel('line', len, deg)`
  - Creates a linear SE. **`deg`** specifies the angle in degrees counterclockwise from the x-axis. **`len`** is the length of the line
- `SE = strel('rectangle', MN)`
  - Creates a rectangular structuring element, where **`MN`** specifies the size. **`MN`** is a two-element vector of nonnegative integers

de.mathworks.com

Erosion

- The value of the output value is the *minimum* value of all the pixels in the input pixel's neighborhood.
  - Pixels beyond image border are assigned the *maximum* value of the data type

Erosion

- The value of the output value is the *minimum* value of all the pixels in the input pixel's neighborhood.
  - Pixels beyond image border are assigned the *maximum* value of the data type
- `img2 = imerode (img, SE)`

Erosion

- The value of the output value is the *minimum* value of all the pixels in the input pixel's neighborhood.
  - Pixels beyond image border are assigned the *maximum* value of the data type
- `img2 = imerode (img, SE)`
- Load the image 'text.png', erode it with a vertical line SE of length 11.

Erosion

- The value of the output value is the *minimum* value of all the pixels in the input pixel's neighborhood.
  - Pixels beyond image border are assigned the *maximum* value of the data type
- `img2 = imerode (img, SE)`
- Load the image 'text.png', erode it with a vertical line SE of length 11.
  - ```
    img = imread('text.png')
    se = strel('line', 11, 90);
    imgErode = imerode(img, se);
    ```
    ```

Dilation

- The value of the output value is the *maximum* value of all the pixels in the input pixel's neighborhood.
  - Pixels beyond image border are assigned the *minimum* value of the data type
- `img2 = imdilate (img, SE)`

Dilation

- The value of the output value is the *maximum* value of all the pixels in the input pixel's neighborhood.
  - Pixels beyond image border are assigned the *minimum* value of the data type
- `img2 = imdilate (img, SE)`
- Load the image 'text.png', dilate it with a vertical line SE of length 11.

Dilation

- The value of the output value is the *maximum* value of all the pixels in the input pixel's neighborhood.
  - Pixels beyond image border are assigned the *minimum* value of the data type
- `img2 = imdilate (img, SE)`
- Load the image 'text.png', dilate it with a vertical line SE of length 11.
  - ```
    img = imread('text.png')
    se = strel('line', 11, 90);
    imgDilate= imdilate(img, se);
    ```
    ```

Morphological opening

- Removes small objects from the image

Morphological opening

- Removes small objects from the image
- Preserves shape and size of larger objects

Morphological opening

- Removes small objects from the image
- Preserves shape and size of larger objects
- Is an erosion followed by a dilation, using the same SE for both operations

Morphological opening

- Removes small objects from the image
- Preserves shape and size of larger objects
- Is an erosion followed by a dilation, using the same SE for both operations
- `img2 = imopen(img1, SE)`

Morphological opening

- Removes small objects from the image
- Preserves shape and size of larger objects
- Is an erosion followed by a dilation, using the same SE for both operations
- `img2 = imopen(img1, SE)`
- Load the image 'snowflakes.png', remove the snowflakes that have a radius less than 5 pixels by opening it with a disk-shaped SE

Morphological opening

- Removes small objects from the image
- Preserves shape and size of larger objects
- Is an erosion followed by a dilation, using the same SE for both operations
- `img2 = imopen(img1, SE)`
- Load the image 'snowflakes.png', remove the snowflakes that have a radius less than 5 pixels by opening it with a disk-shaped SE
  - ```
    img = imread('snowflakes.png')
    se = strel('disk', 4);
    imgOpen= imopen (img, se);
    ```
    ```

Morphological closing

- Fills up (removes) small holes

Morphological closing

- Fills up (removes) small holes
- Is dilation followed by erosion, using the same SE for both operations

Morphological closing

- Fills up (removes) small holes
- Is dilation followed by erosion, using the same SE for both operations
- `img2 = imclose(img1, SE)`

Morphological closing

- Fills up (removes) small holes
- Is dilation followed by erosion, using the same SE for both operations
- `img2 = imclose(img1, SE)`
- Load the image ‘circles.png’, fill the gaps in the image. Use a disk-shaped SE to preserve the circular nature of the object

Morphological closing

- Fills up (removes) small holes
- Is an dilation followed by a erosion, using the same SE for both operations
- `img2 = imclose(img1, SE)`
- Load the image 'circles.png', fill the gaps in the image. Use a disk-shaped SE to preserve the circular nature of the object
  - ```
    img = imread('circles.png')
    se = strel('disk', 10);
    imgClosed= imclose (img, se);
    ```
    ```

Morphological opening

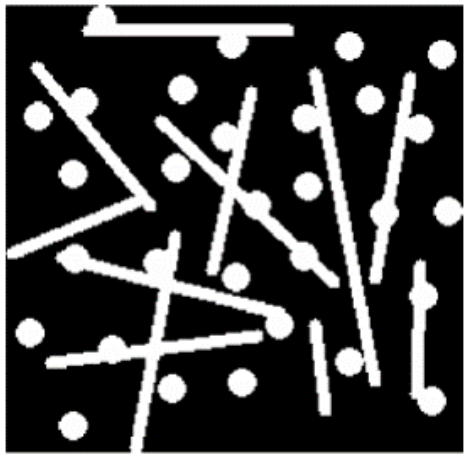
- Remove all the circuit lines from the original circuit image 'circbw.tif' using erosion and dilation operations, creating an output image that contains only the rectangular shapes of the microchips
  - Note: the SE should be large enough to remove the lines during erosion, yet not remove the rectangles

Opening = erosion followed by dilation

de.mathworks.com

Sticks and dots

- Get this image from CommSy
- Target: suppress all dots so the resulting image would only contain all the lines.
- Play around with morphological operations and see what you get!



Assignment #4

1. Binarize the image on the right using Otsu's method. Find the threshold **T** that minimizes the within-class variance of the two classes separated by **T**. Work only on the two thresholds T ≤ 2 and T ≤ 6.
   Also sketch the final binary image, clearly marking pixels values (0 and 1).

4	4	3	3
5	5	5	5
7	5	5	5
2	2	1	7
2	2	9	7

$$\sigma^2_{global} = \sigma^2_{within} + \sigma^2_{between}$$

$$\sigma^2_{within} = P_1\sigma^2_1 + P_2\sigma^2_2$$

$$\begin{aligned}\sigma^2_{between} &= \sigma^2_{global} - \sigma^2_{within} \\ &= P_1(\mu_1 - \mu_{global})^2 + P_2(\mu_2 - \mu_{global})^2 \\ &= P_1P_2(\mu_1 - \mu_2)^2\end{aligned}$$

Assignment #4

2. Enhance the image on the right using morphological operations (erosion, dilation, opening, and closing).
   Make sure the resulting image does not contain any noise nor any holes/gaps, without losing any important parts of the image.

Assignment #4

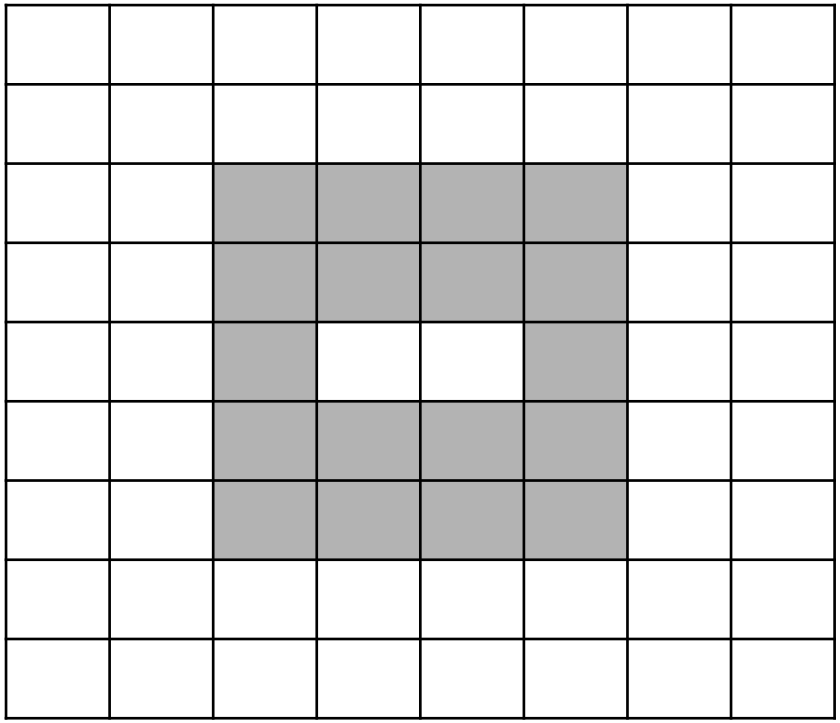
3.

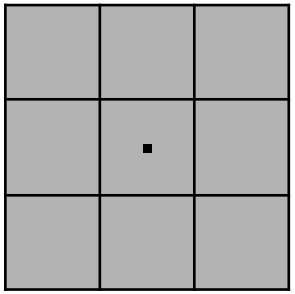
a) Given a structuring element described as SE = {(-1,0), (0,-1), (0,0), (0,1), (1,0)}, with the origin at (0,0). **Sketch** the structuring element SE.

b) Using the below 3x3 square structuring element, perform morphological **dilation** to the below binary image. Each shaded square is a member of the set. Sketch the output binary in a grid similar to that of the image.

Assignment #4

3.





Structuring Element