INF250

6 Morphology

Morphology

- Morphological operations are used to modify the shape of an object, using local filter operations
- It can be used to remove unwanted effects in segmentation post-processing
  - Remove small objects (noise)
  - Smooth edges of larger objects
  - Fill in holes in objects
  - Link objects together
- It can be used as part of object description analysis
  - Locate boundary of objects
  - Thin objects
  - Locating objects within a certain structure
  - Locating patterns in an image
    - The operations are small and often very fast

Morphology and image types

- Morphology is used on binary images
- Filters can be used on gray level images
- Binary image
  - 1: Pixel in foreground (objects)
  - 0: Background
- Binary images are made through thresholding
- Morphology is about how to shrink or grow pixels of an object based on the location of the
- In this process a structure element is used as a filter

Median filter used as structuring element

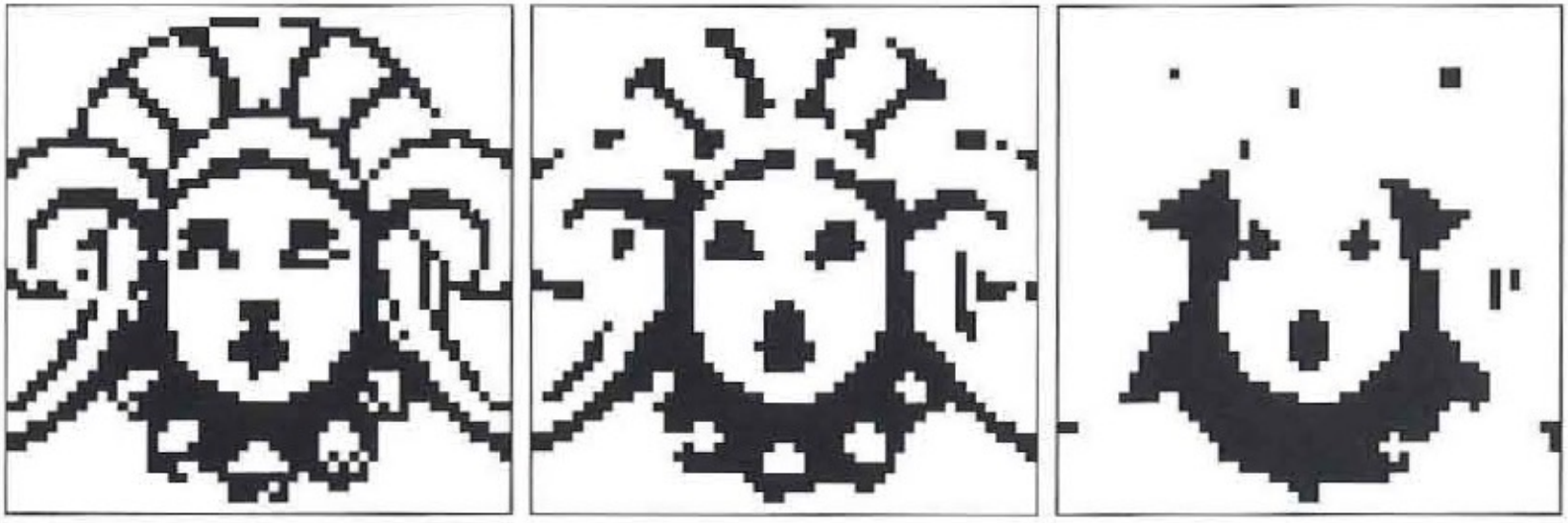
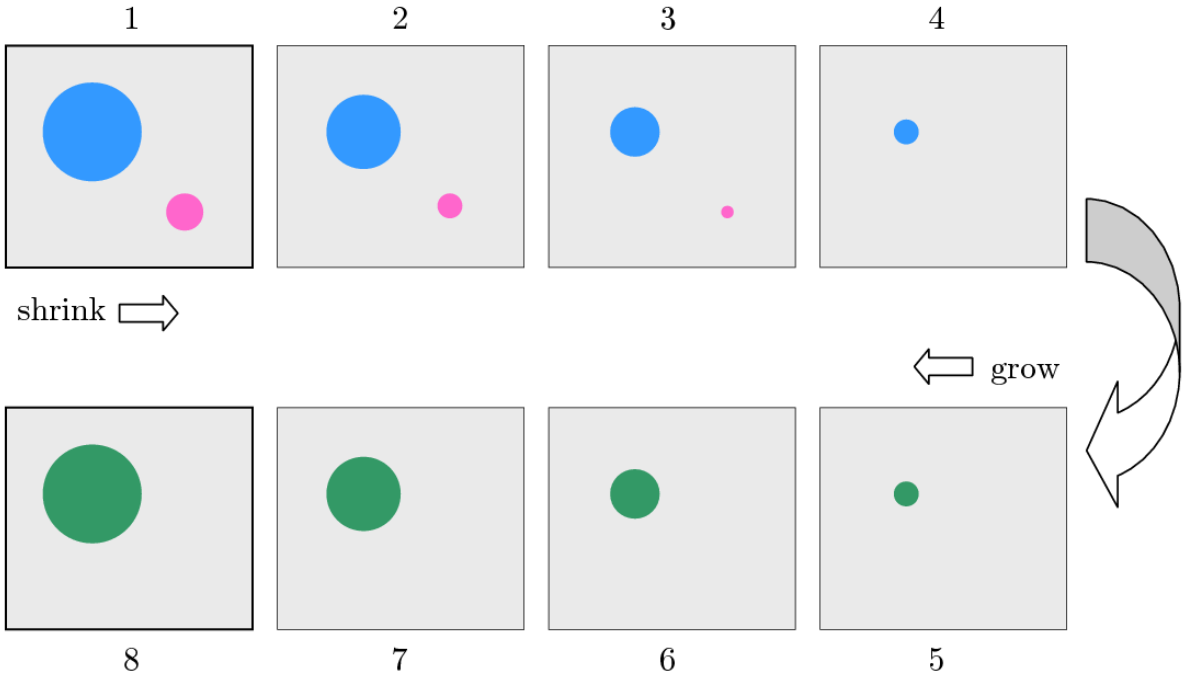


Figure 7.1 Median filter applied to a binary image: original image (a) and results from a 3×3 pixel median filter (b) and a 5×5 pixel median filter (c).

Shrink & Grow



- Can round off large structures and remove small structures
- Useful for preprocessing before further analysis

Neighborhood between pixles

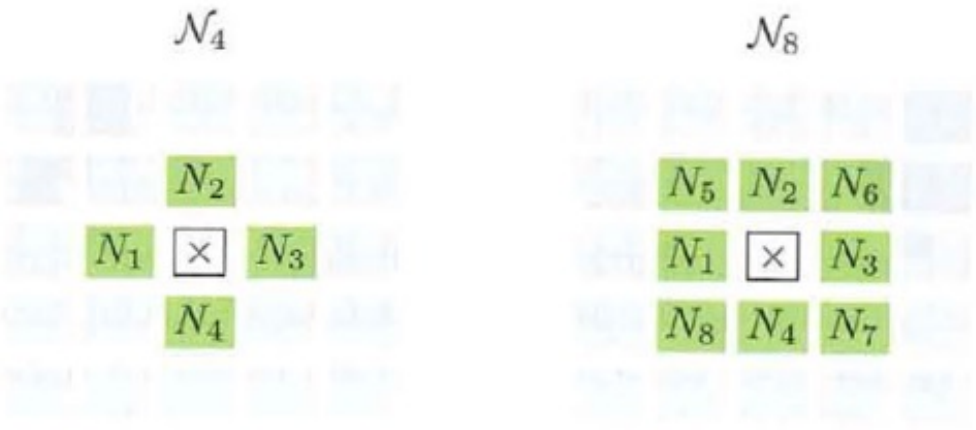


Figure 7.5 Definitions of "neighborhood" on a rectangular pixel grid: *4-neighborhood* $\mathcal{N}_4 = \{N_1, \ldots N_4\}$ (left) and *8-neighborhood* $\mathcal{N}_8 = \mathcal{N}_4 \cup \{N_5, \ldots N_8\}$ (right).

4 - neighborhood

8 - neighborhood

Morphological Operations

- In short: A set of operations that process images based on shapes. Morphological operations apply a *structuring element* to an input image and generate an output image.
- The most basic morphological operations are two: Erosion and Dilation.
  - Removing noise
  - Isolation of individual elements and joining disparate elements in an image.
  - Finding of intensity bumps or holes in an image

Dilation

- This operations consists of convoluting an image with some kernel (), which can have any shape or size, usually a square or circle.
- The kernel has a defined *anchor point*, usually being the center of the kernel.
- As the kernel is scanned over the image, we compute the maximal pixel value overlapped by and replace the image pixel in the anchor point position with that maximal value.
- As you can deduce, this maximizing operation causes bright regions within an image to “grow” (therefore the name *dilation*).

Erosion

- This operation is the sister of dilation. What this does is to compute a local minimum over the area of the kernel.
- As the kernel is scanned over the image, we compute the minimal pixel value overlapped by and replace the image pixel under the anchor point with that minimal value.

Shrink and grow

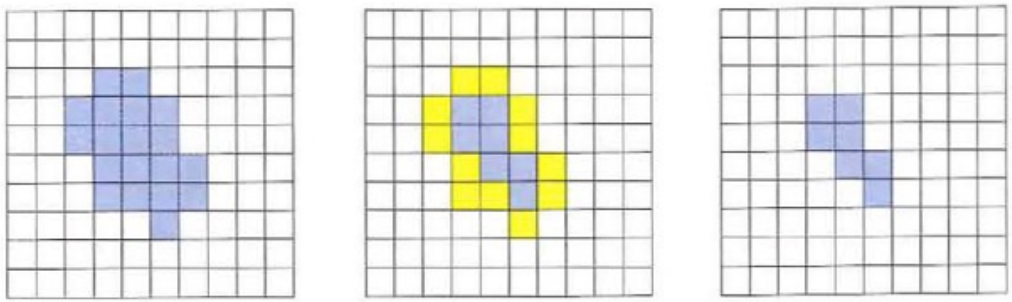


Figure 7.3 "Shrinking" a foreground region by removing a layer of border pixels: original image (a), identified foreground pixels that are in direct contact with the background (b), and result after shrinking (c).

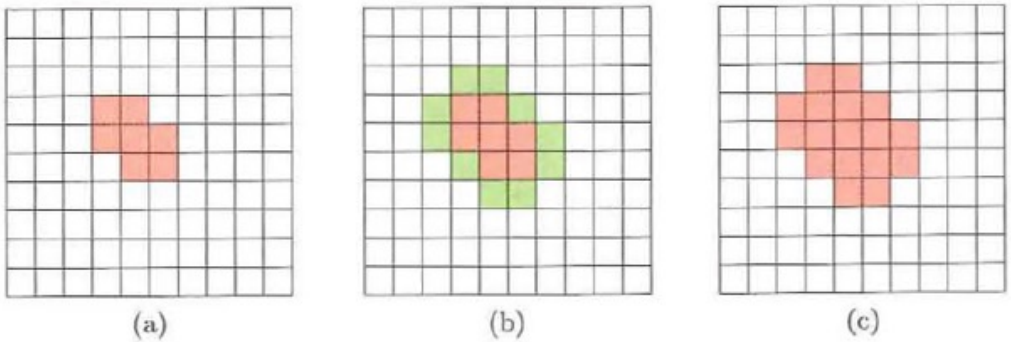
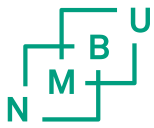
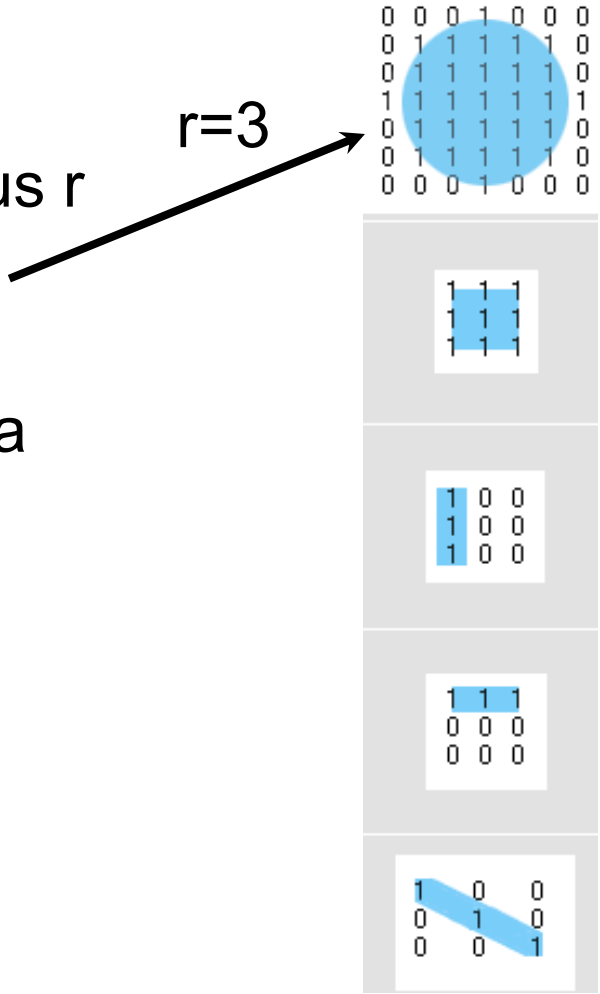


Figure 7.4 "Growing" a foreground region by attaching a layer of pixels: original image (a), identified background pixels that are in direct contact with the region (b), and result after growing (c).

Structuring element, SE (Morphological filter)



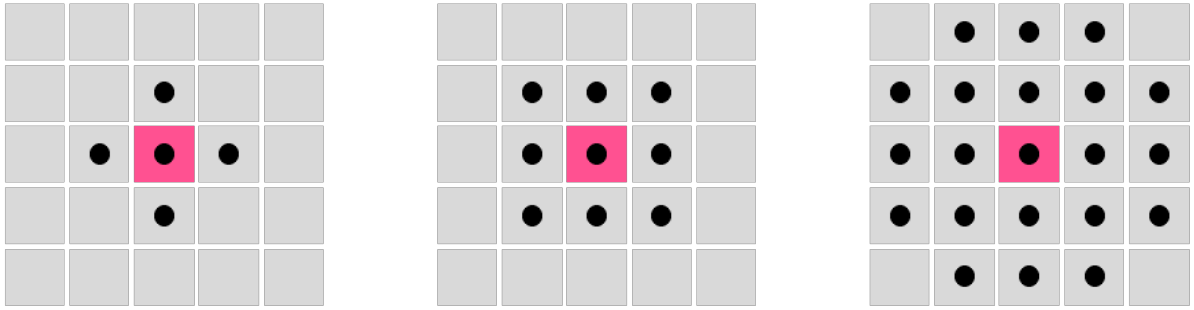
- A structuring element is applied
- Dilation with a circular SE with radius r adds a layer with thickness r to the foreground elements in the image
- Erosion with the same SE removes a layer of the same thickness



Structuring element (Morphology filter)

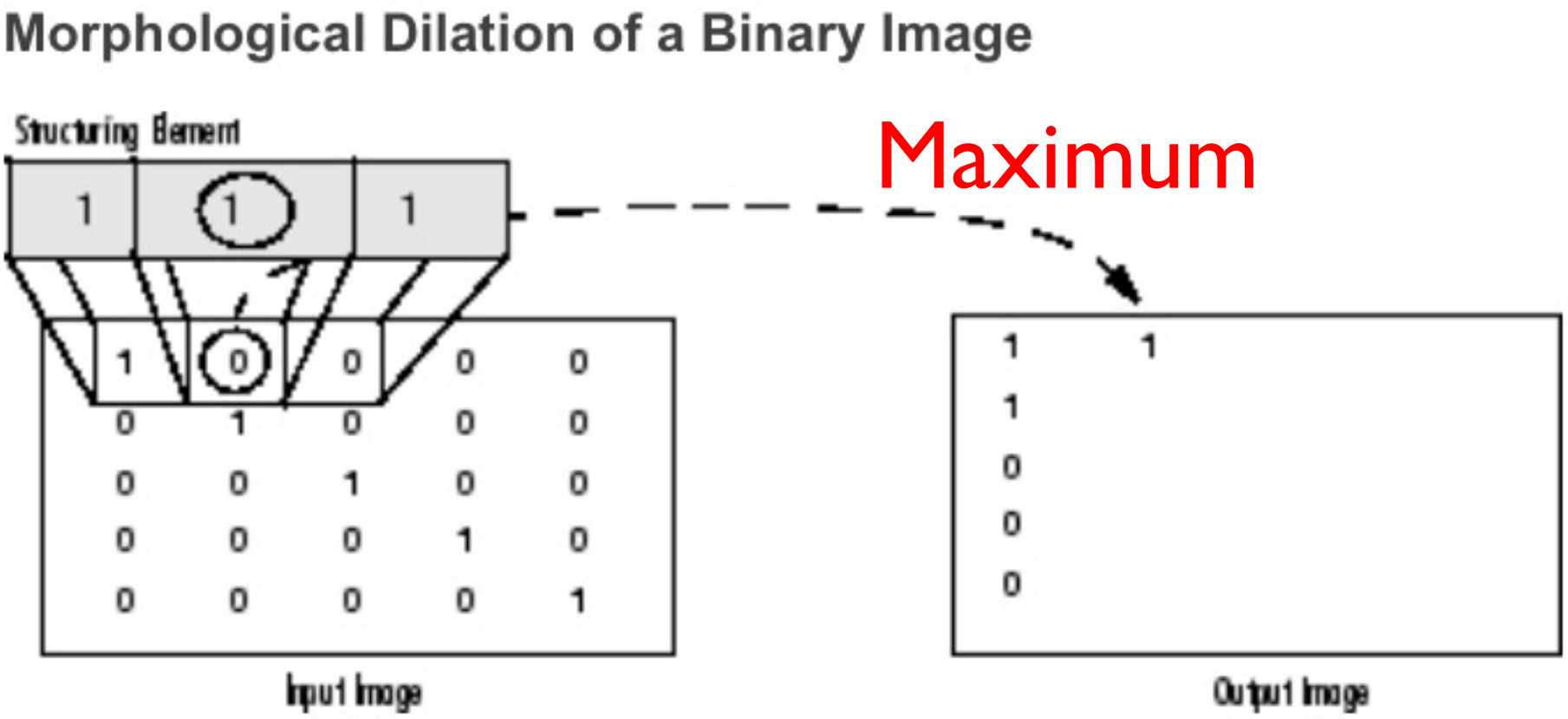
- A morphology filter is specified by
  - type of operation
  - Content of structuring element (SE)
- Size and form of SE is decided by
  - application
  - Form of object
  - structure in image, texture

Examples of structuring elements - Neighborhood

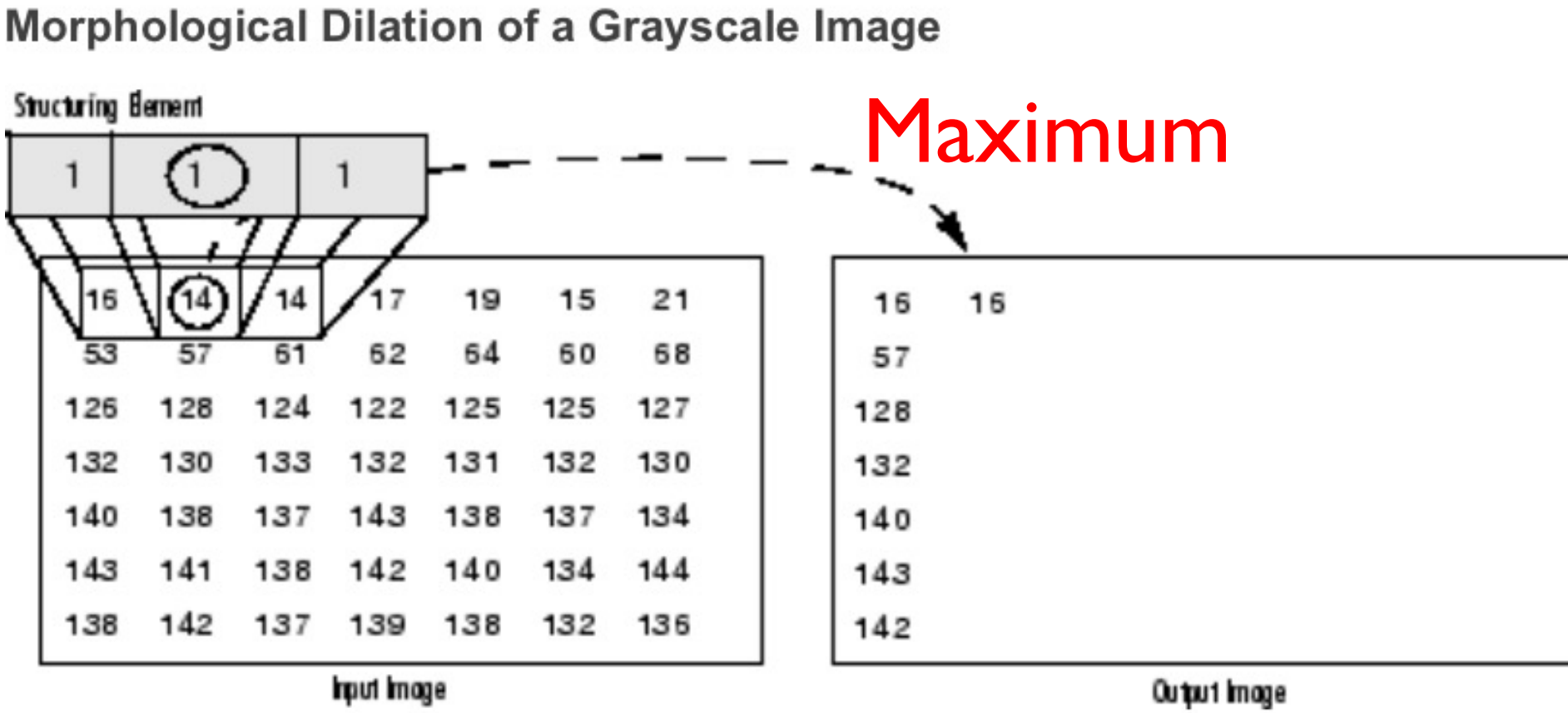


- (a) 4-neighborhood
- (b) 8-neighboorhood
- (c) small circle with radius 2

Graphical examples of dilation (maximum) (erosion is minimum)



Graphical example of dilation (maximum) (erosion is minimum)

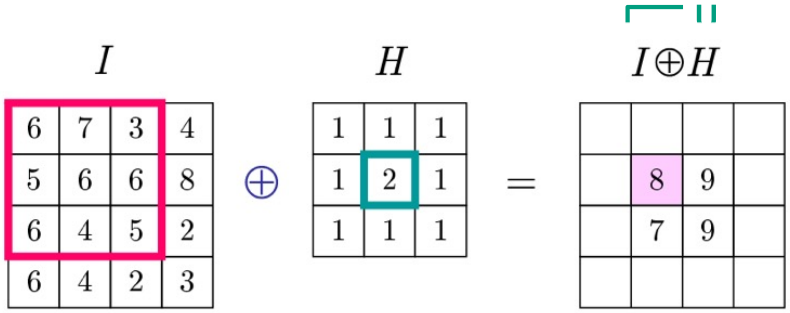


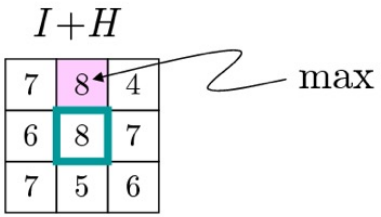
Gray level morphology

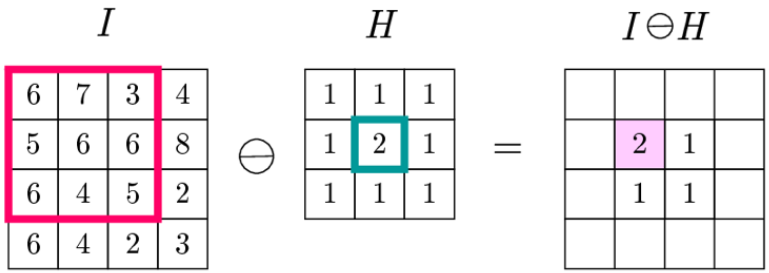
- The structuring element is not only binary
- values $H(i,j) \in \mathbb{R}\,, \qquad \text{for} \;\; (i,j) \in \mathbb{Z}^2$
- Dilation $(I \oplus H)(u,v) = \max\limits_{(i,j)\in H} \{I(u+i, v+j) + H(i,j)\}$
- Erosion $(I \ominus H)(u,v) = \min\limits_{(i,j)\in H} \{I(u+i, v+j) - H(i,j)\}$

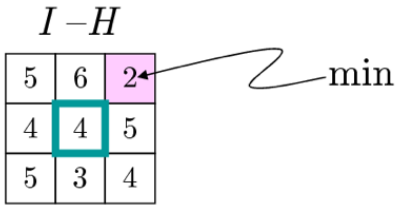
Gray level morphology

- Burger and Burge describes gray level dilation (erosion) by summing (subtracting) the structuring element with corresponding pixles and replace the center pixel with max (min) value



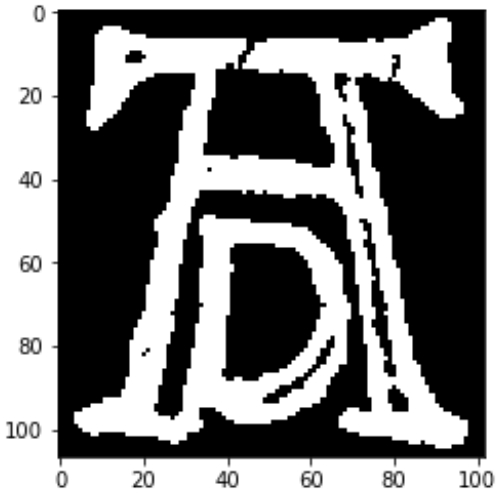


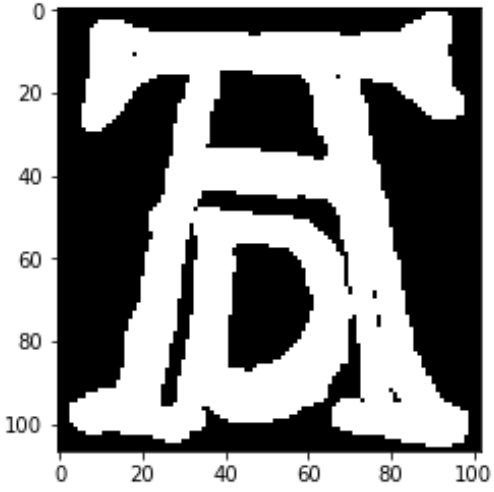


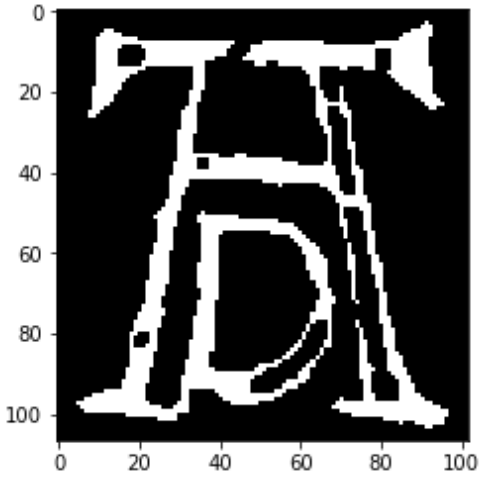


Dilation and erosion

```
from skimage.morphology import skeletonize, dilation, opening, square
from skimage.morphology import erosion, closing
from skimage import io

filename = 'rhino_detail.tif'

rhino = io.imread(filename)
plt.imshow(rhino,'gray')

rhinodil = dilation(rhino,square(3))
plt.imshow(rhinodil,'gray')

rhinoerode = erosion(rhino,square(3))
plt.imshow(rhinoerode,'gray')
```

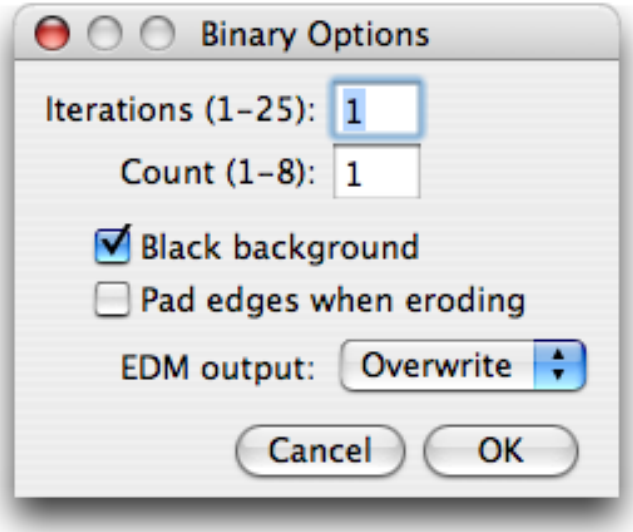


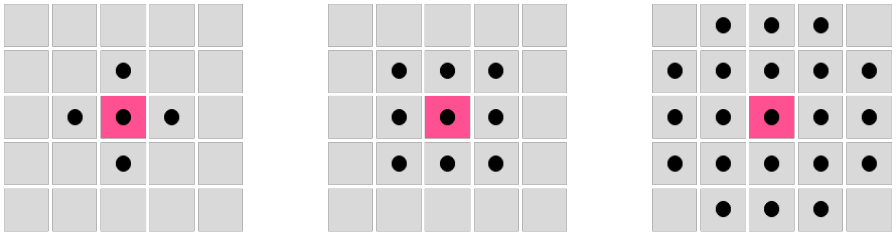


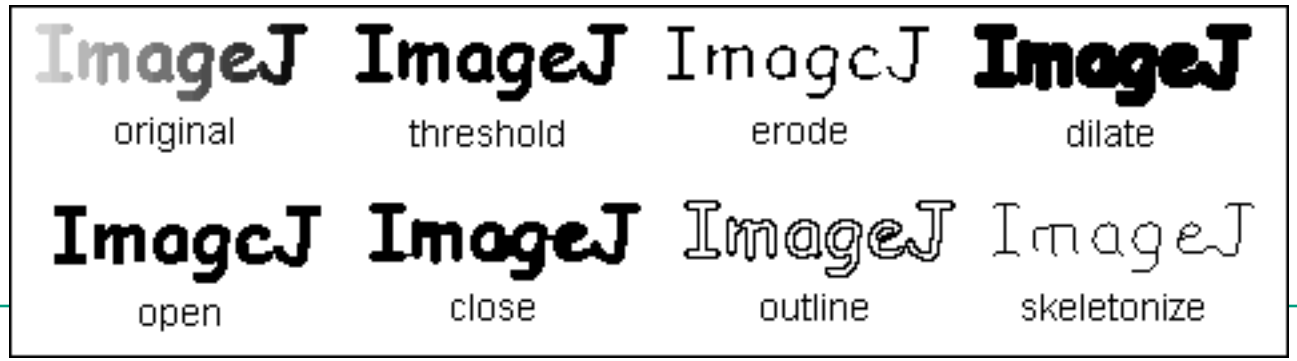


ImageJ

- ImageJ has implemented a 8-neighborhood structuring element (b).



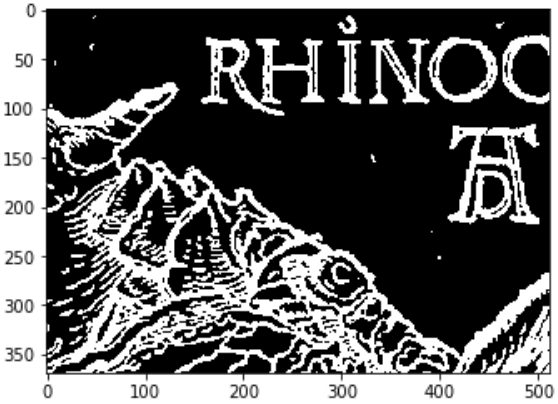




ImageJ binary menu

- **Convert to Mask**
- Converts the image to black and white based on the current threshold settings (if set) or on a threshold calculated by analyzing the histogram. The mask will have an inverting LUT (white is 0 and black is 255) unless "Black Background" is checked in the *Process>Binary>Options* dialog box.
- **Erode**
- Removes pixels from the edges of black objects. Use *Process>Filters>Minimum* to do grayscale erosion.
- **Dilate**
- Adds pixels to the edges of black objects. Use *Process>Filters>Maximum* to do grayscale dilation.
- **Open**
- Performs an erosion operation, followed by dilation. This smoothes objects and removes isolated pixels.
- **Close**
- Performs a dilation operation, followed by erosion. This smoothes objects and fills in small holes.

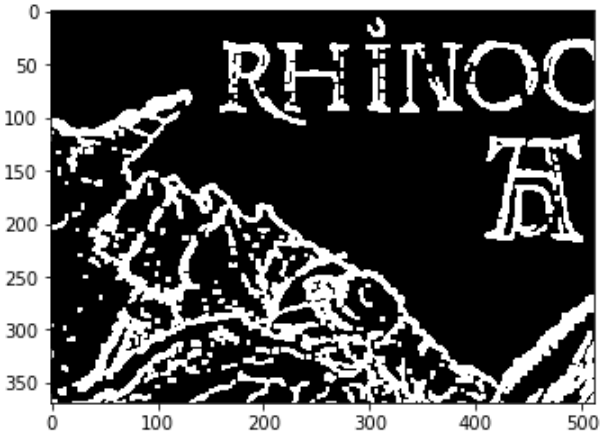
Example

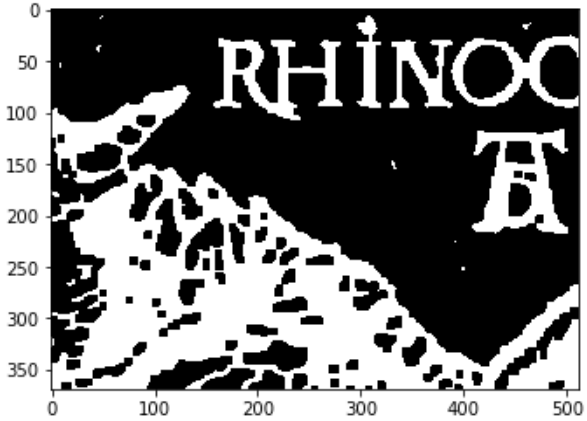


dilation

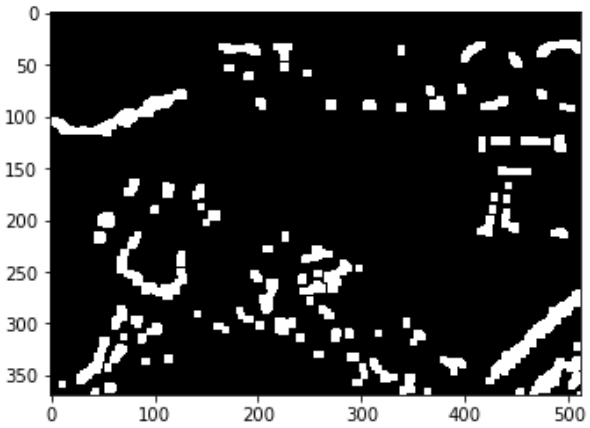
Square 4

erosion





Square 6



Example

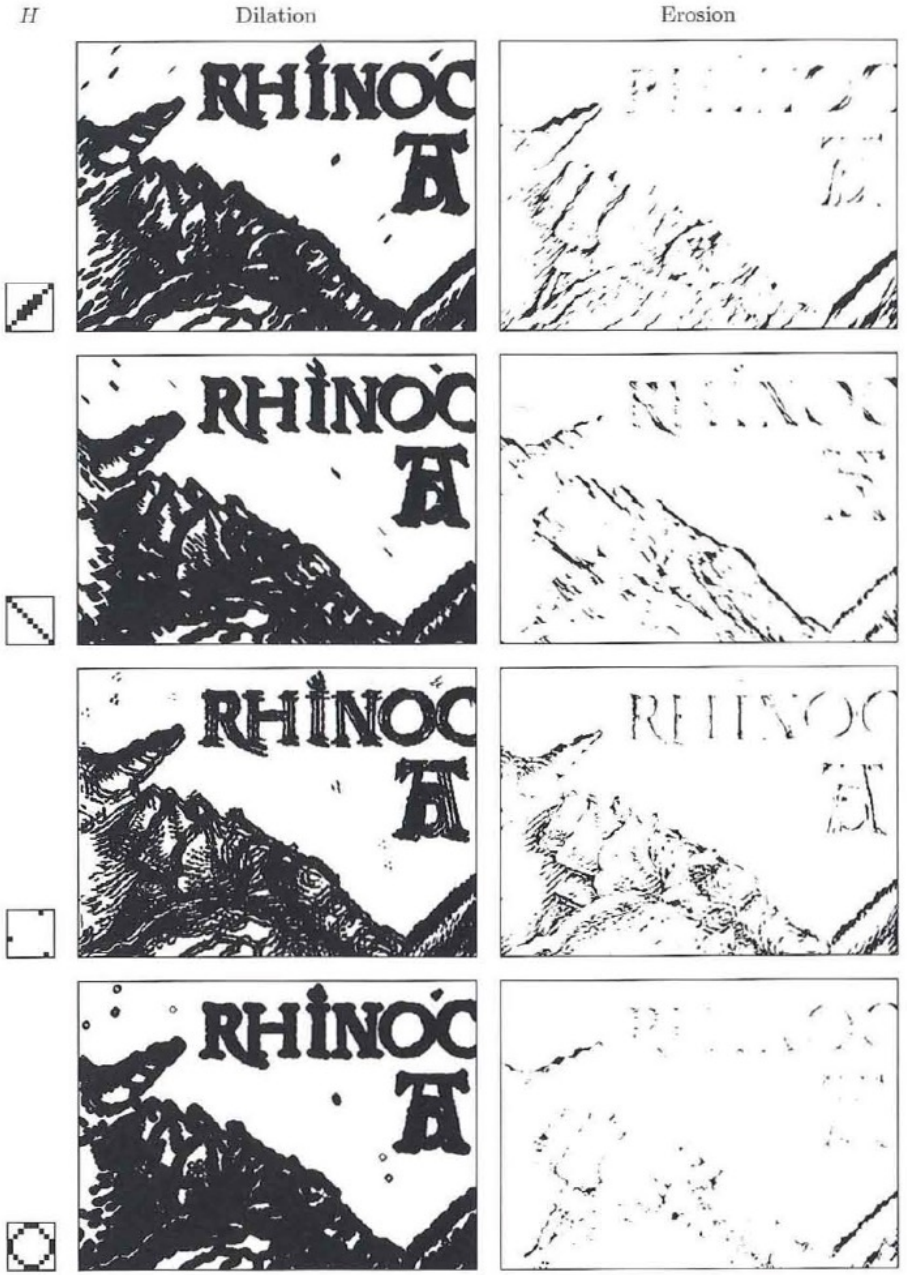


Figure 7.14 Examples of binary dilation and erosion with various free-form structuring elements. The structuring elements H are shown in the left column (enlarged). Notice that the dilation expands every isolated foreground point to the shape of the structuring element,

Example

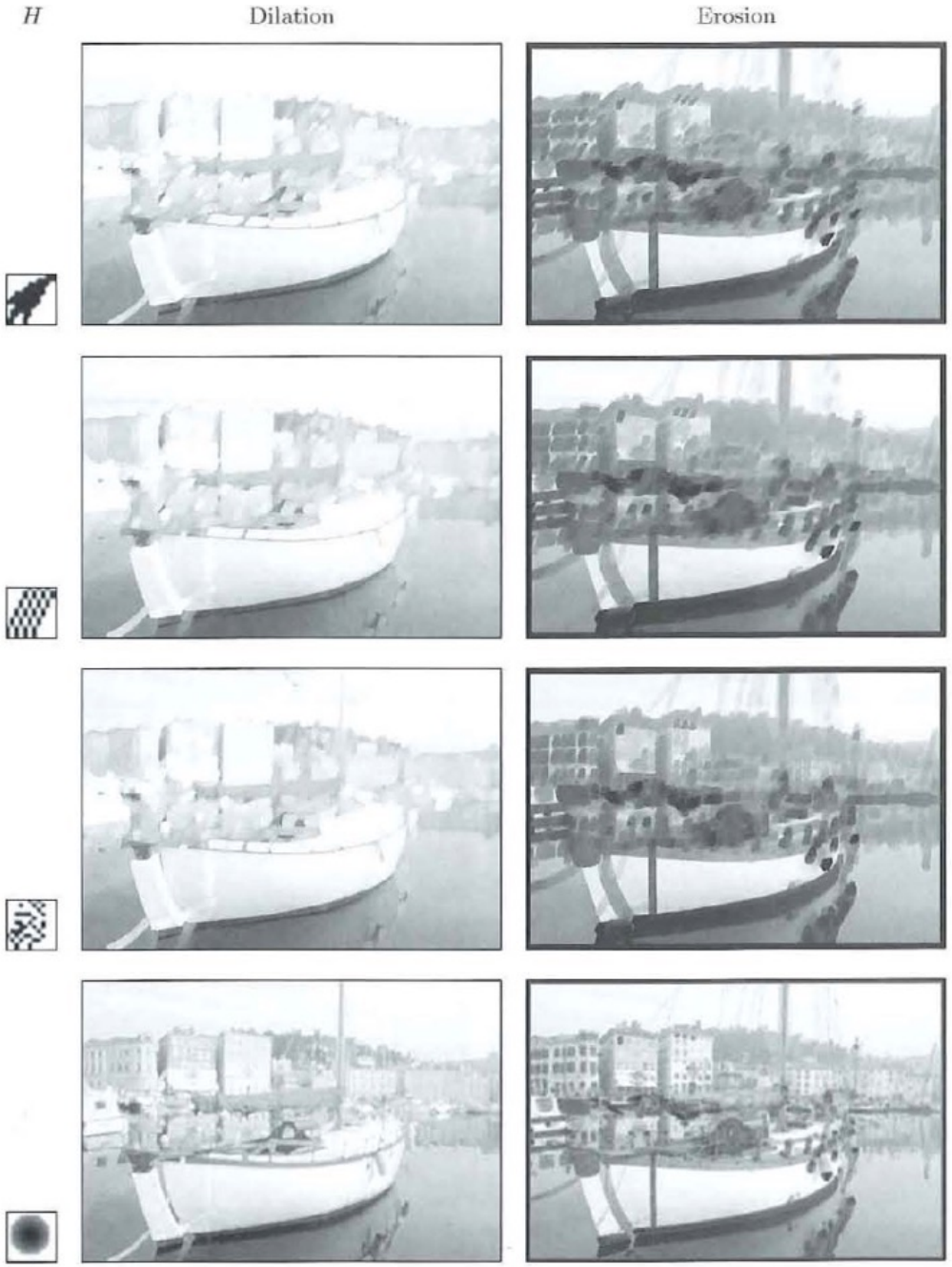
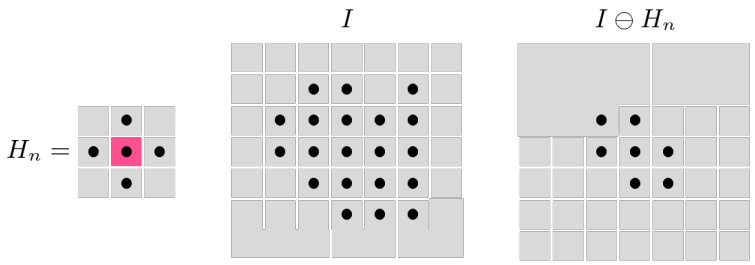


Figure 7.22 Grayscale dilation and erosion with various free-form structuring elements.

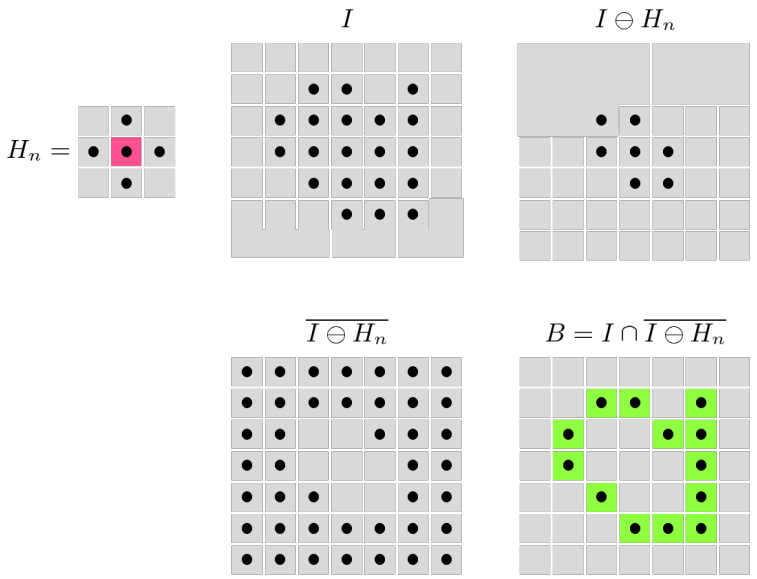
Outlining

- Dilation and erosion are combined with mathematical operations
- Example: contours, outlining
- Neighborhood kept (different from Sobel/Prewitt filter)



Outlining

- Dilation and erosion are combined with mathematical operations
- Example: contours, outlining
- Neighborhood kept (different from Sobel/Prewitt filter)



Opening

- Opening: $I \circ H = (I \ominus H) \oplus H$
- A binary opening is defined by erosion followed by a dilation
- Main effect: all foreground structures smaller than the structuring element will be eliminated in the erosion. The leftover structures will be smoothed in the following dilation
- In other words:
  - Shrinking followed by growth will eliminate the small structures

Closing

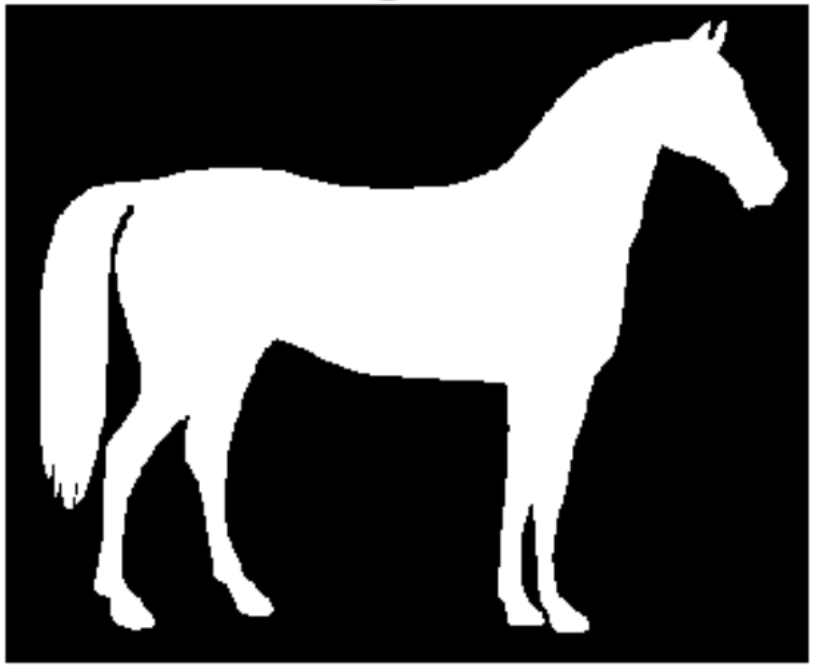
- Closing: $I \bullet H = (I \oplus H) \ominus H$
- A binary closing is defined by a dilation followed by an erosion
- Main effect: al holes and “impurities” smaller then the structuring element will be eliminated
- In other words:
  - Growth followed by shrinking will eliminate holes and small structures

python

http://scikit-image.org/docs/dev/api/skimage.morphology.html?highlight=dilation#skimage.morphology.dilation

Thinning (Skeletonize)

original



skeleton