计算机辅助手术讲座（7）
Image Guided Surgery (7)

二值形态学

Binary morphology

顾 力栩 (*Lixu Gu*)
上海交通大学 Med-X研究院

2009.12

形态学的二值操作

Source T

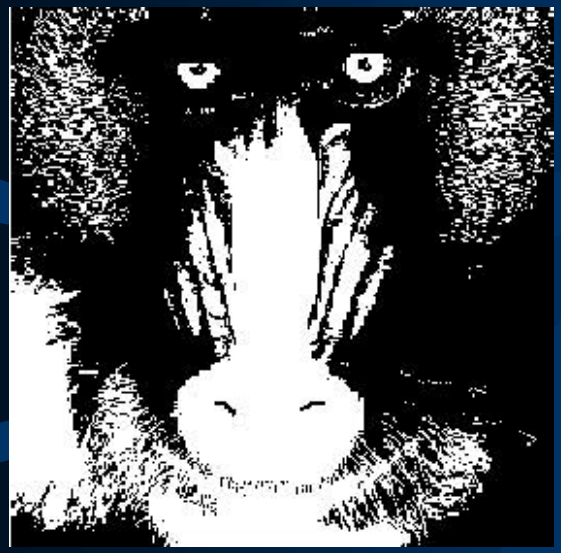
$T \oplus 2B$

$T \$ 2B$

$T \circ 2B$

$T \bullet 2B$

Binary Morphological Operations

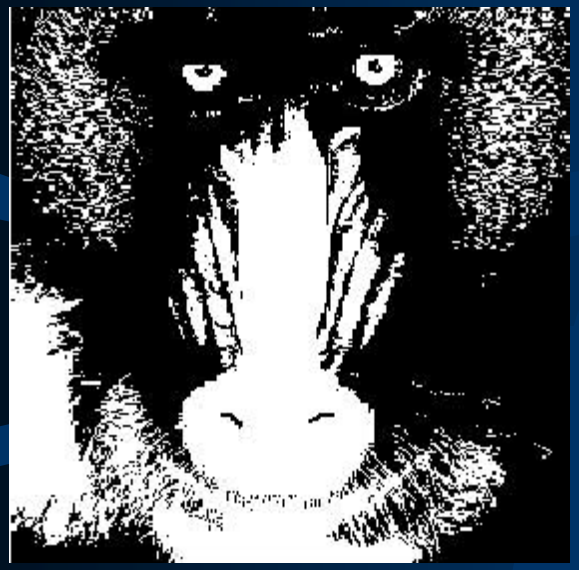


Source T

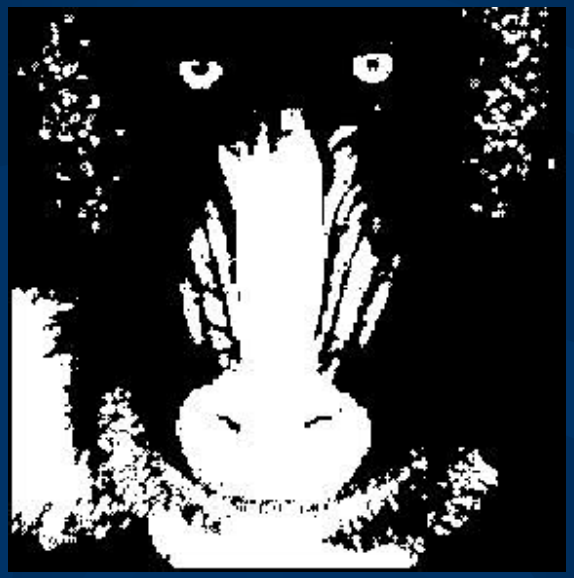
$T \oplus 2B$

$T \ominus 2B$

Binary Morphological Operations



Source T



$T \circ 2B$

$T \bullet 2B$

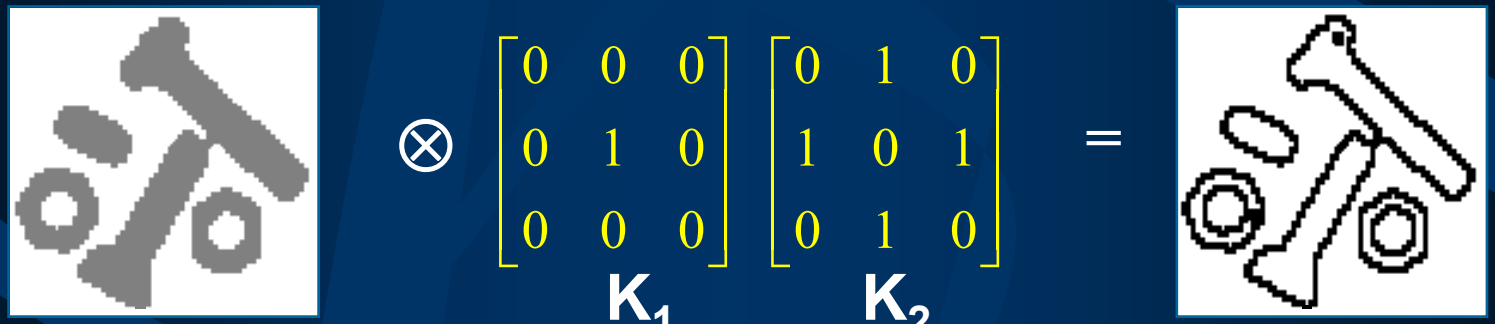
Hit-and-Miss

- Hit-and-miss: A morphological shape detector.

$$F \otimes K = (F \ominus K_1) \cap (F^c \ominus K_2)^c,$$

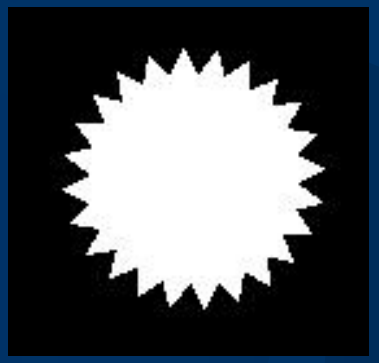
$$K_1 \cap K_2 = \varnothing,\ K_1 \in K,\ K_2 \in K$$

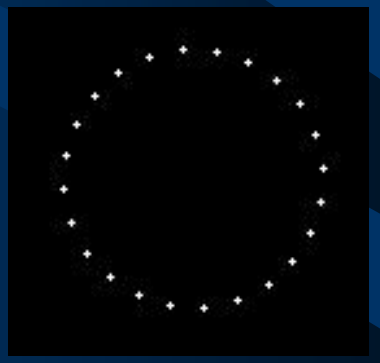
can be used to look for particular patterns of foreground and background pixels in an image



Hit-and-Miss

- “Does the first set fit the foreground while, simultaneously, the second set misses it (i.e., fits the background)?”
- Particular patterns:
  - Isolated pixels
  - End points
  - Contour points



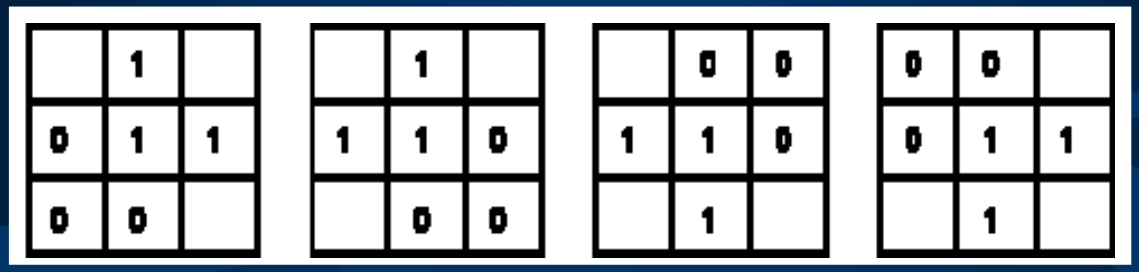


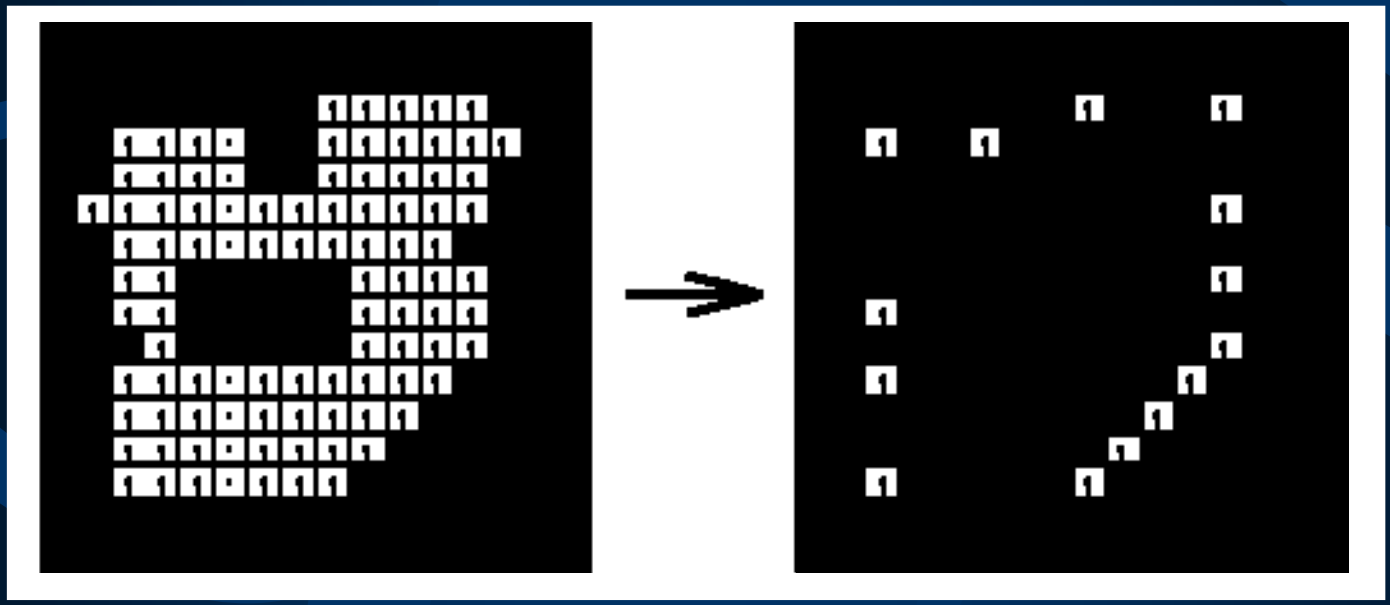
Hit-and-Miss

- Translating the origin of the structuring element to all points in the image
- Comparing the structuring element with the underlying image pixels.
  - If the foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image, then the pixel underneath the origin of the structuring element is set to the foreground color.
  - If it doesn't match, then that pixel is set to the background color.

Hit-and-Miss

- An example: corner detector





Pattern Spectrum

- Pattern Spectrum is known as granulometric size density. It is employed to measure the size distribution of an object.
- Pattern spectrum $PS_{rik}(F)$ of a set F in terms of SE r_ik is defined as:

$$PS_{r_iK}(F) = \begin{cases} Card((F \circ r_iK) - (F \circ r_{i+1}K)) & if\ i \geq 0 \\ Card((F \bullet r_{-i}K) - (F \bullet r_{-i-1}K)) & if\ i < 0 \end{cases}$$

Where, $Card(F)$ denotes the cardinality of set F

Pattern Spectrum

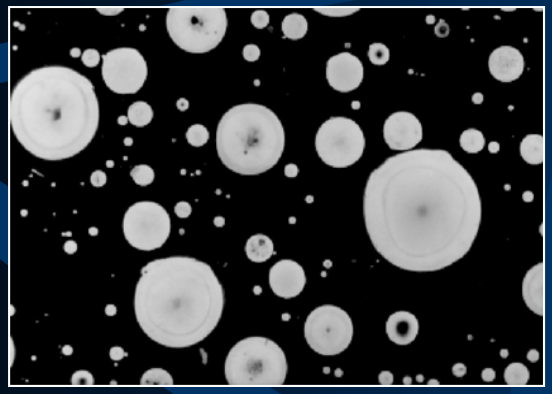
1. Take an opening of the image with an appropriate structuring element;
2. Distract this opening from the original image. This difference image is the *n*-th element of the so called *Distance Size Transformation* (DST);
3. In a binary image: count the number of object pixels; in a grayscale image: sum all the gray values;
4. Put this number in the *n*-th bin of the pattern spectrum (start with $n = 0$);

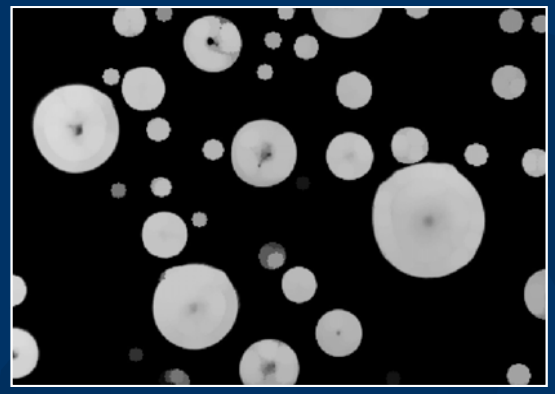
Pattern Spectrum

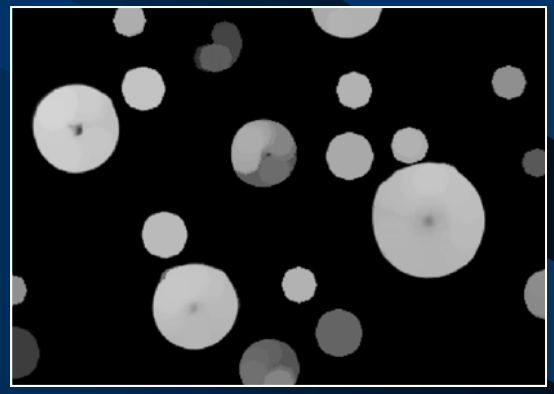
5. Repeat steps 1 through 4 with the opened image as the new original (*n* increases with one);
   Important: In the next cycle we need to take another structuring element. Take *nB*, this means that the past structuring element dilated with the original one is the new structuring element. This way the small objects will be filtered out first, and with increasing *n* the bigger objects will disappear.
6. In a binary image: stop when the image is empty;
   in a grayscale image: stop when the image is flat.

Pattern Spectrum

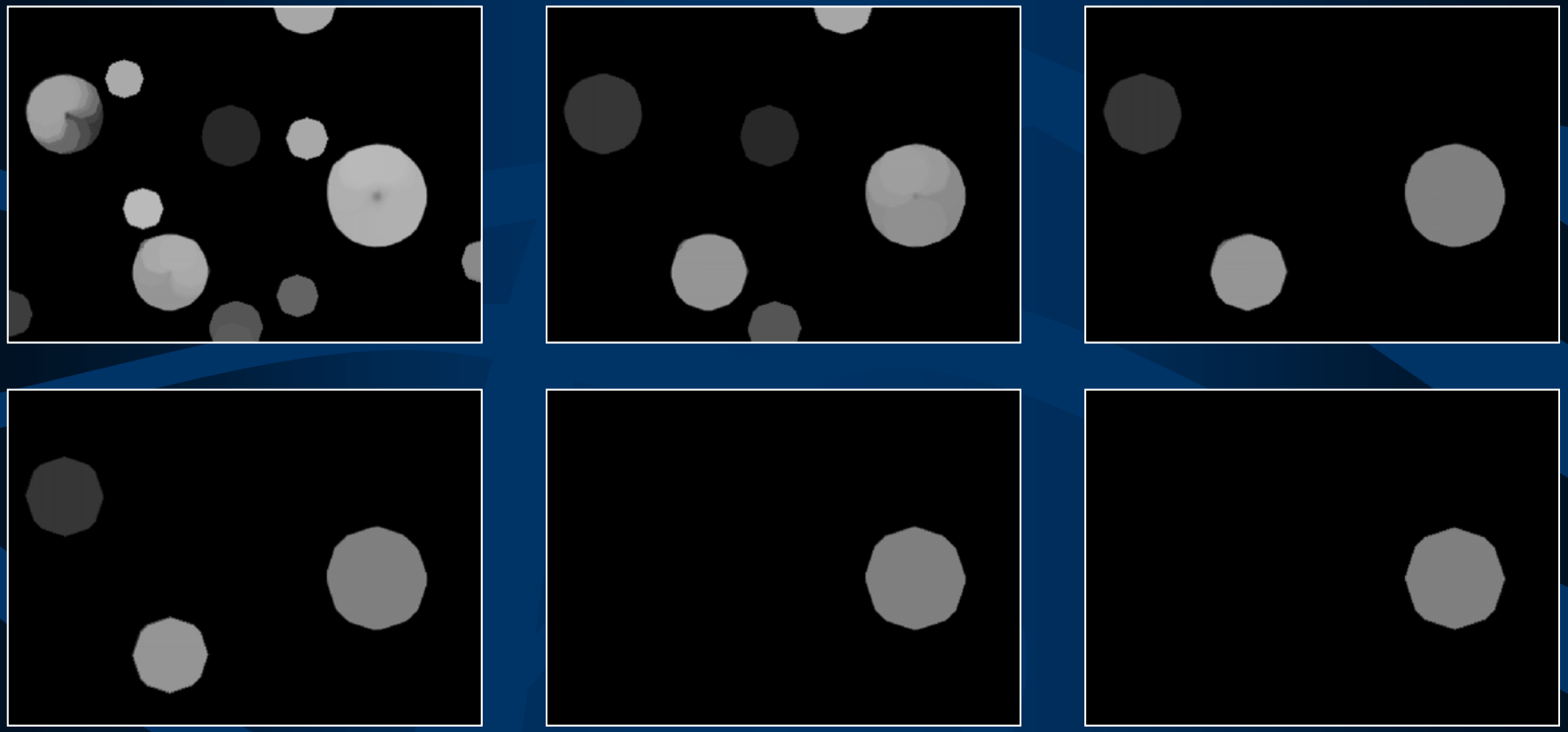
Example: The first figure is our original image. Because the present objects have a circular shape, we choose a disk shaped structuring element. We start with a disk with radius five, but the radius will increase with increasing *n*-value *nB*



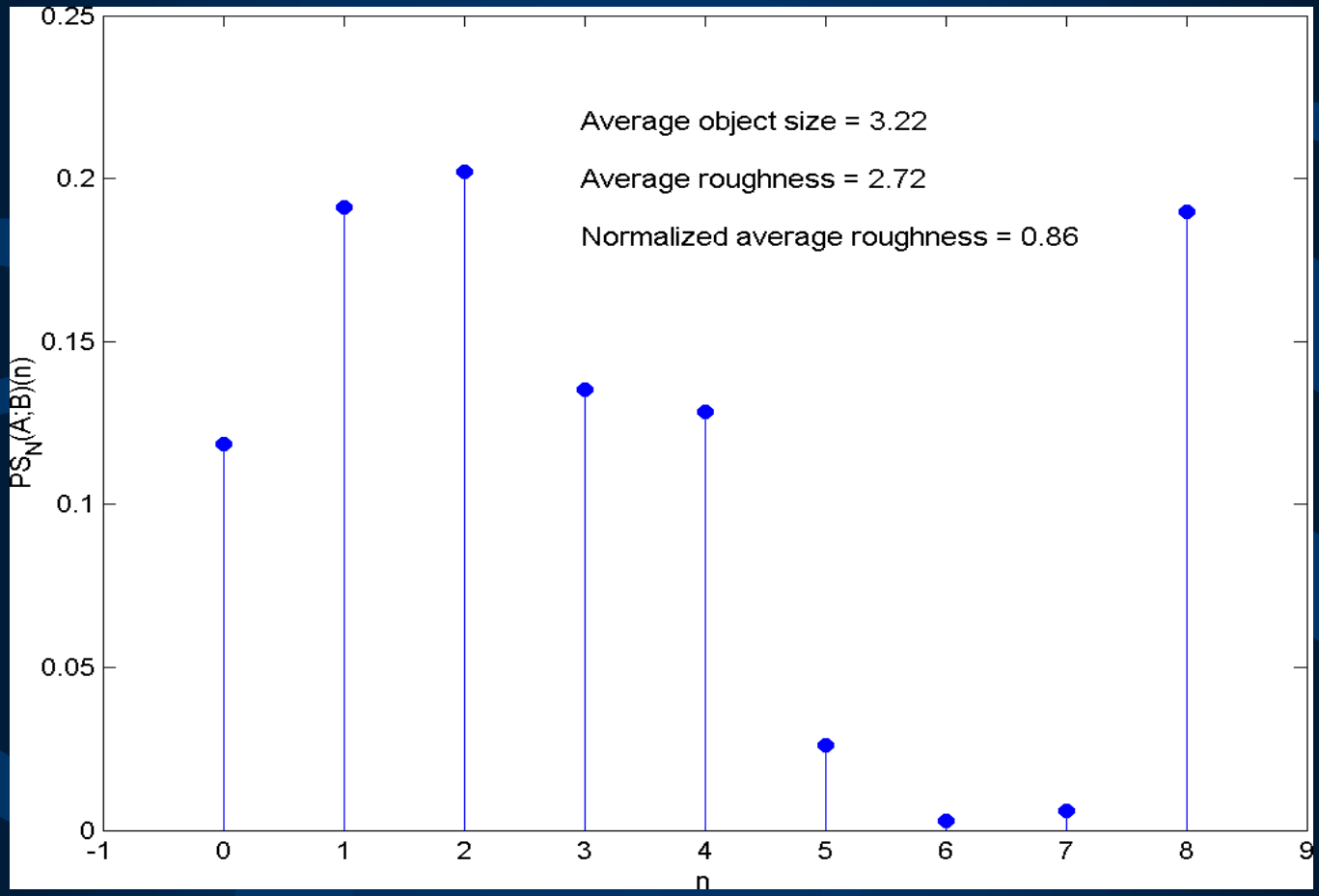




Pattern Spectrum

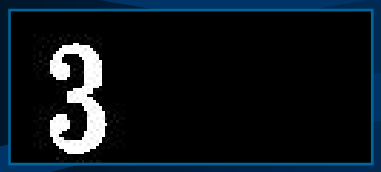


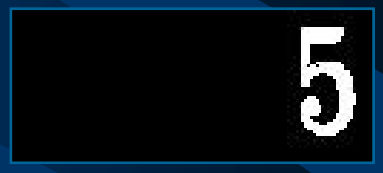
Pattern Spectrum

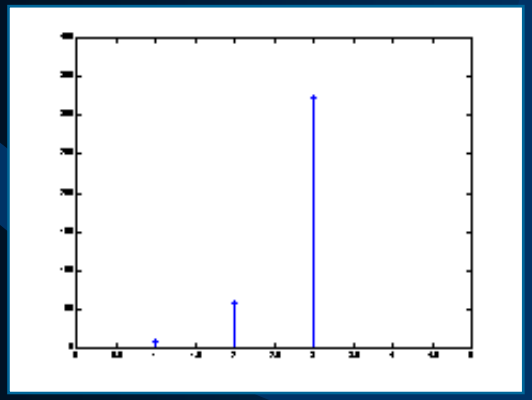


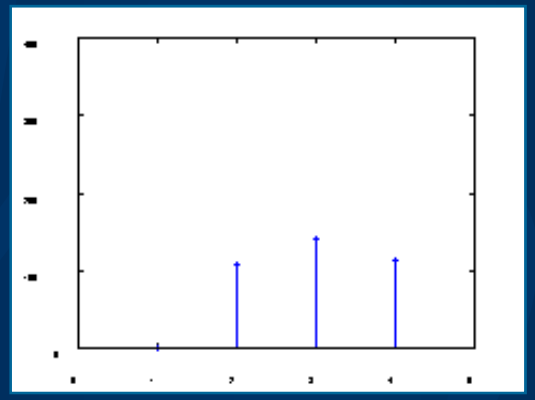
Pattern Spectrum

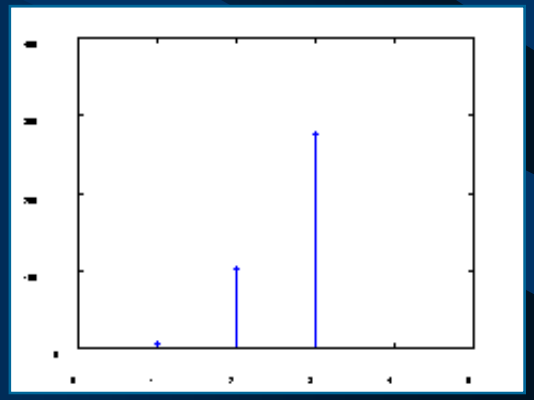
- Pattern spectrum not only can detect the size of parts in an image, but also can analyze the shapes of them.





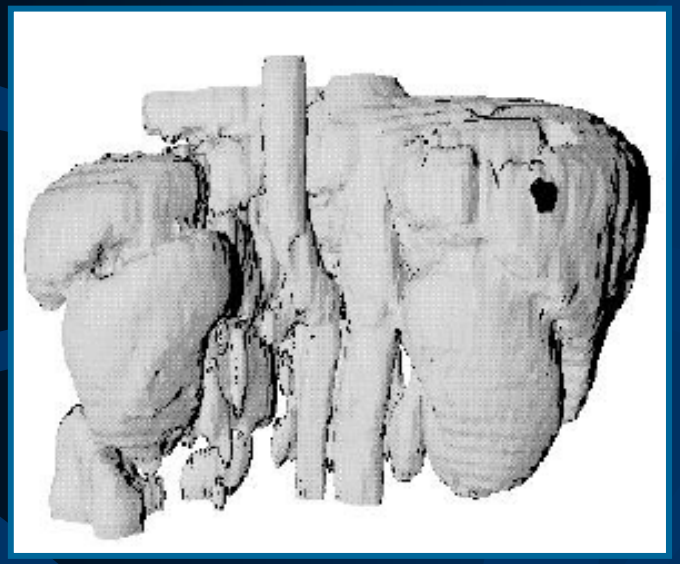


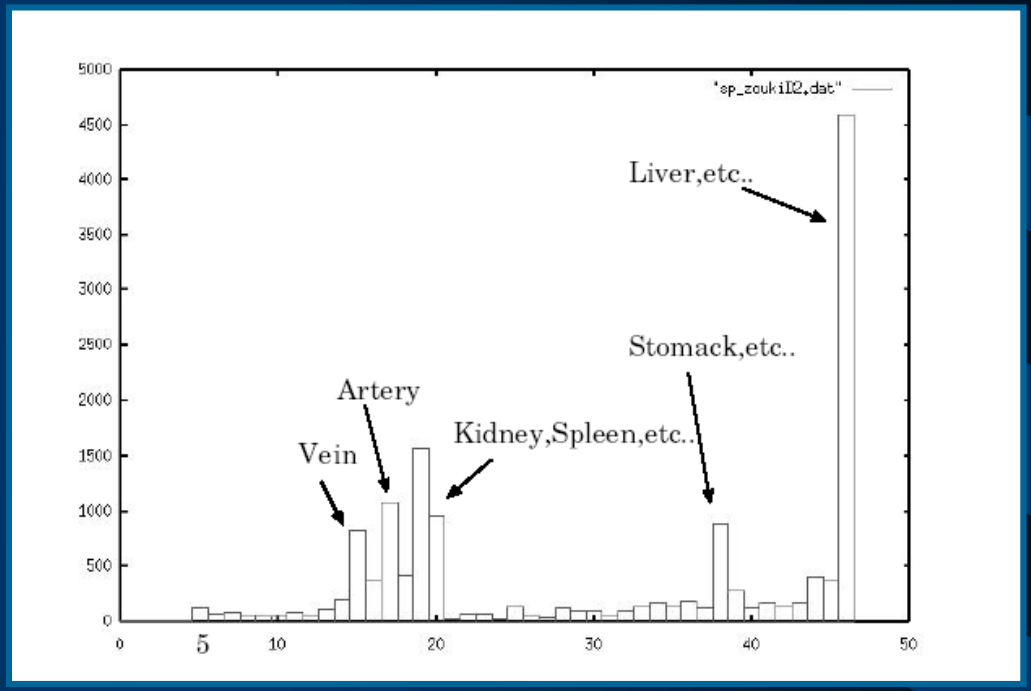




Pattern Spectrum

- Another example: size analysis





Organ mass

Pattern Spectrum analysis

Recursive Dilation

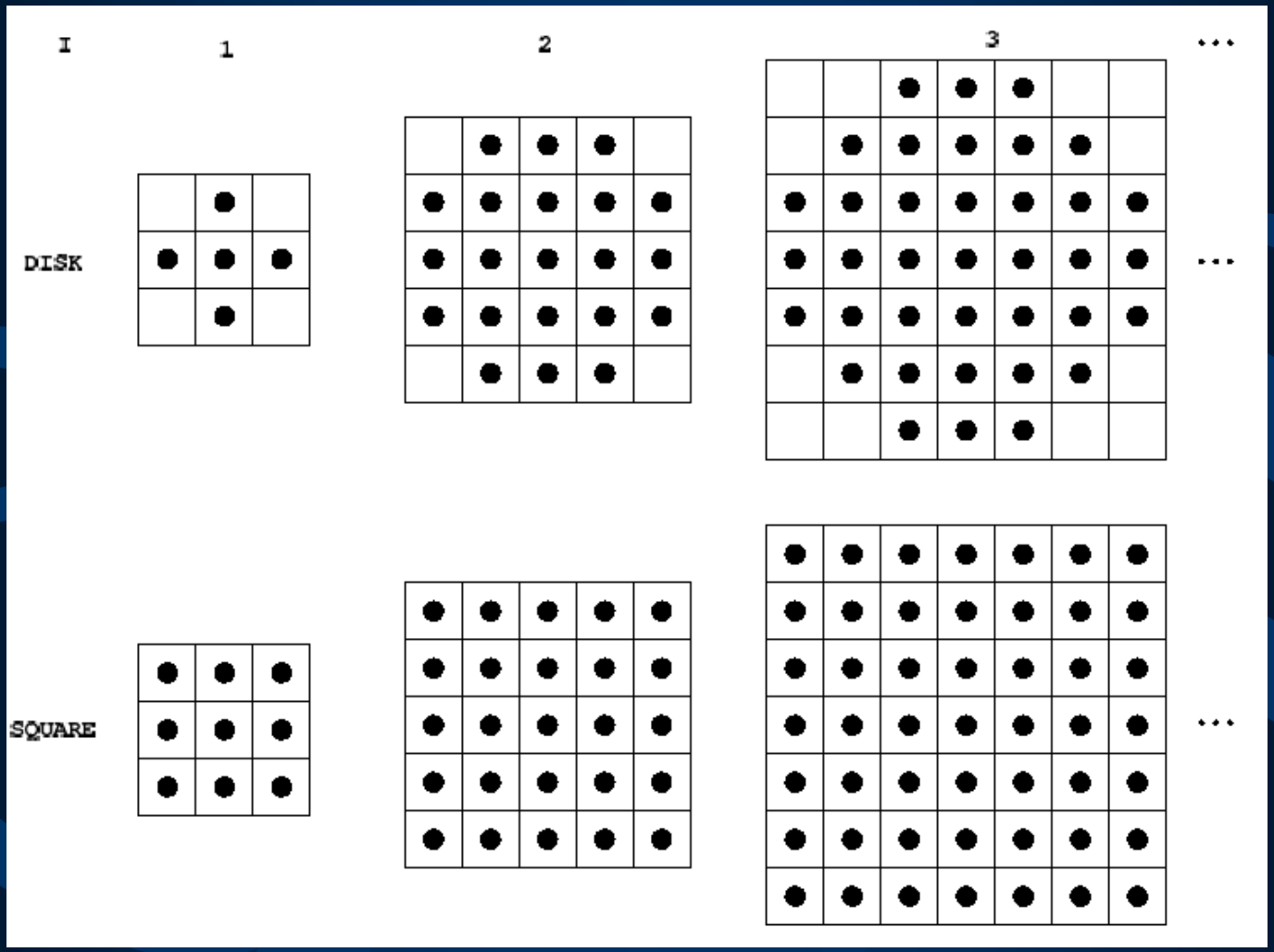
- Recursive Dilation is defined as:

$$F \overset{i}{\oplus} K = \begin{cases} F & if \;\; i = 0 \\ (F \overset{i-1}{\oplus} K) \oplus K & if \;\; i \geq 1 \end{cases}$$

where, i is defined as scalar factor and K as its base.

- Recursive Dilation is employed to compose SE series in the same shape but different sizes.

Recursive Dilation



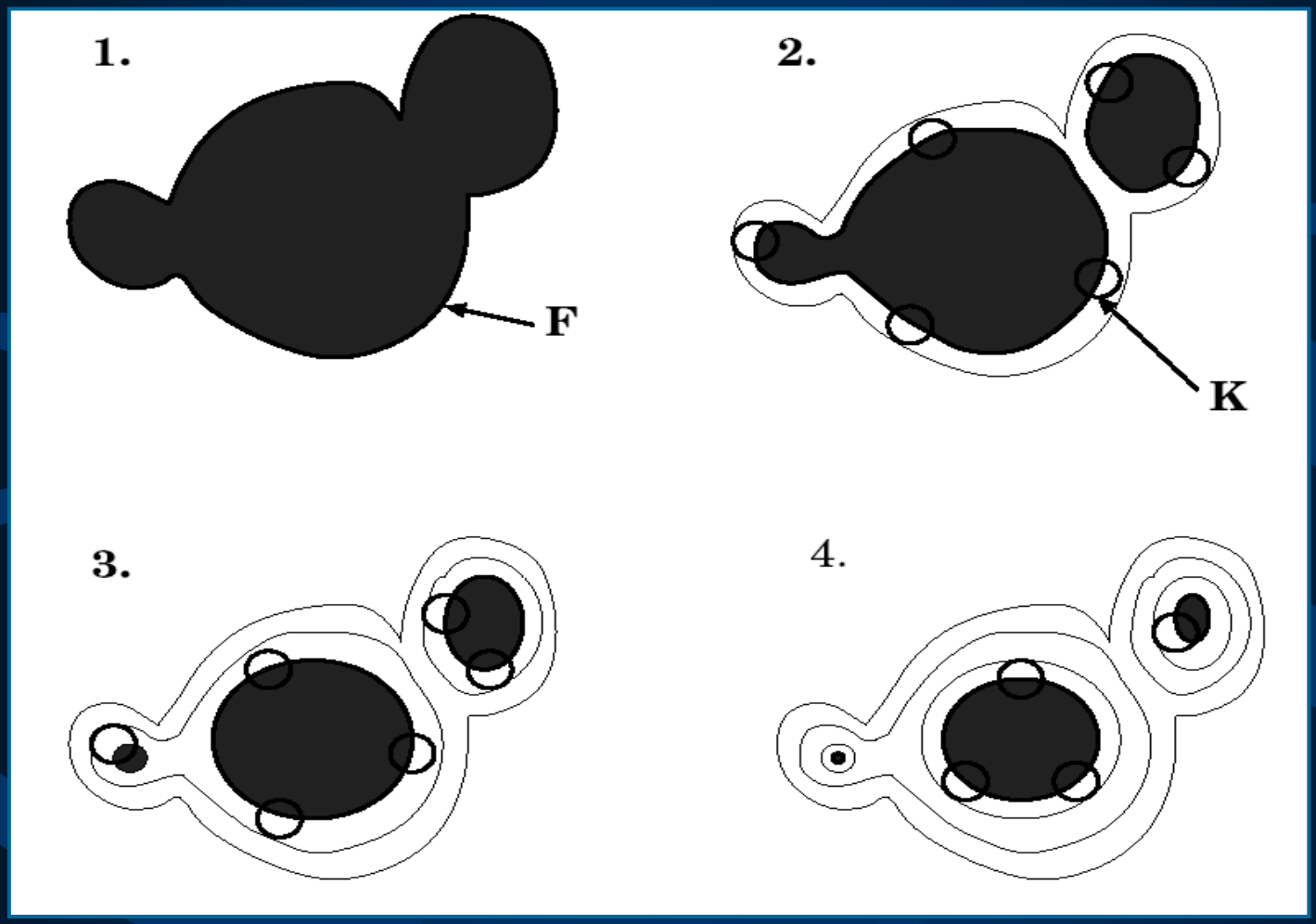
Recursive Erosion

- Recursive Erosion is also called *successive erosion which is defined as:*

$$F \overset{i}{\$} K = \begin{cases} F & if \ \ i = 0 \\ (F \overset{i-1}{\$} K) \$ K & if \ \ i \geq 1 \end{cases}$$

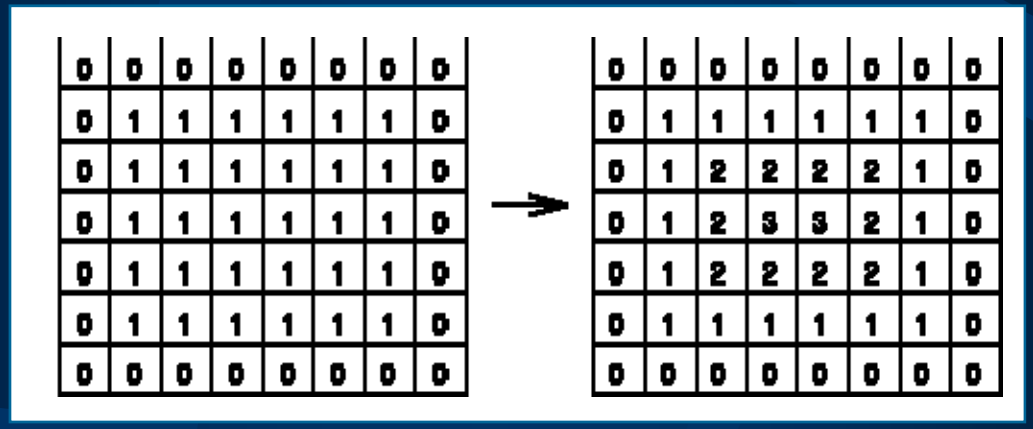
- When performing recursive erosions of an object, its components are progressively shrunk until completely disappeared.
- Useful for distance transform and segmentation

Recursive Erosion

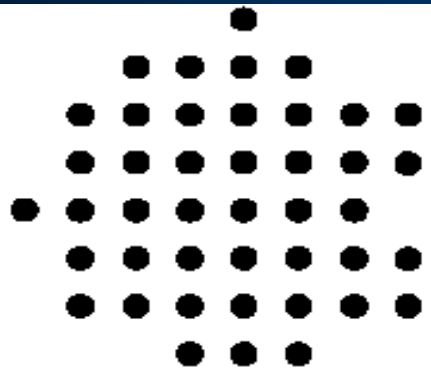


Distance Transform

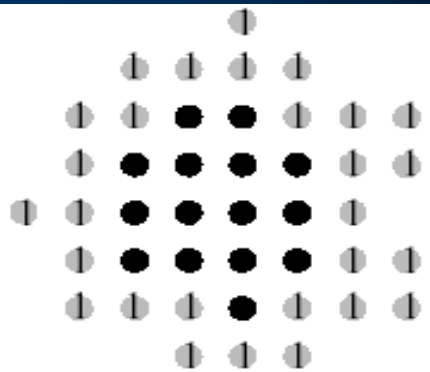
- The distance transform is an operator normally only applied to binary images.
- The result of the transform is a greylevel image showing the distance to the closest boundary from each point.



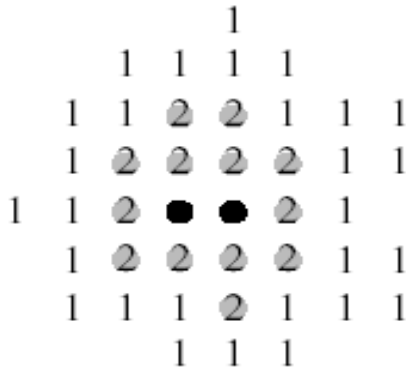
Distance Transform



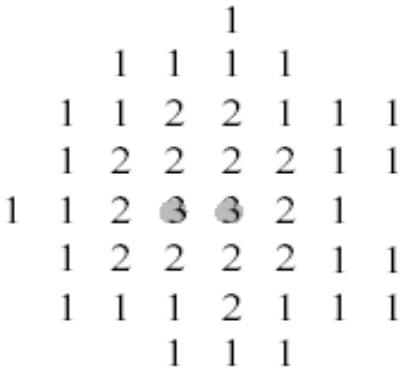
1. Original image



2. "1's" are assigned to pixels which are on the 8-neighborhood boundary.



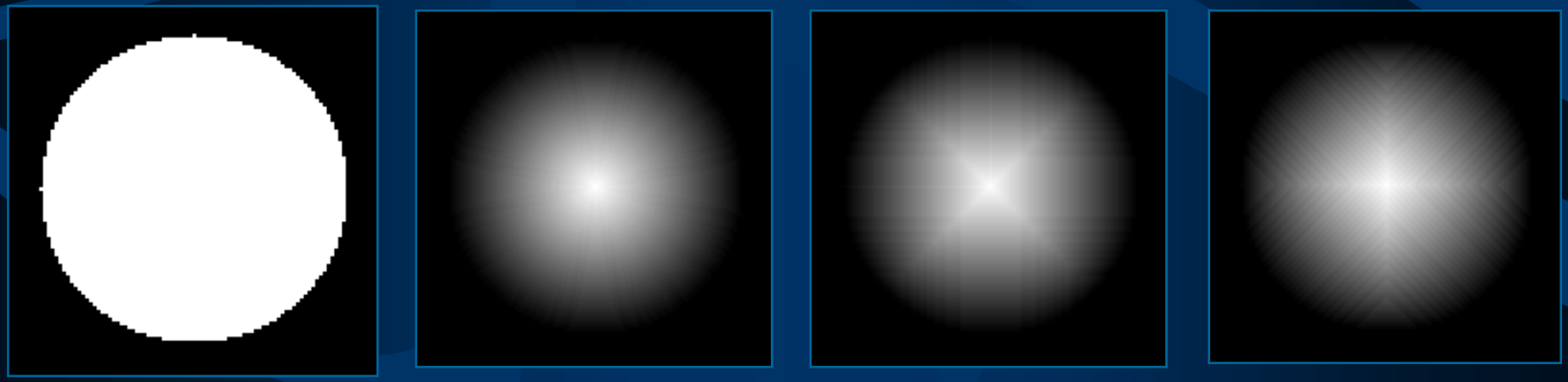
3. "2's" are assigned to pixels which are connected to the pixels with "1."



4. "*n*'s" are assigned to pixels which are connected to the pixels with "*n*–1."

Distance Metrics

- **Euclidean Distance** $D_{Euclid} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- **City Block Distance (N_8)** $D_{City} = |x_2 - x_1| + |y_2 - y_1|$
- **Chessboard Distance (N_4)** $D_{Chess} = \max(|x_2 - x_1|, |y_2 - y_1|)$

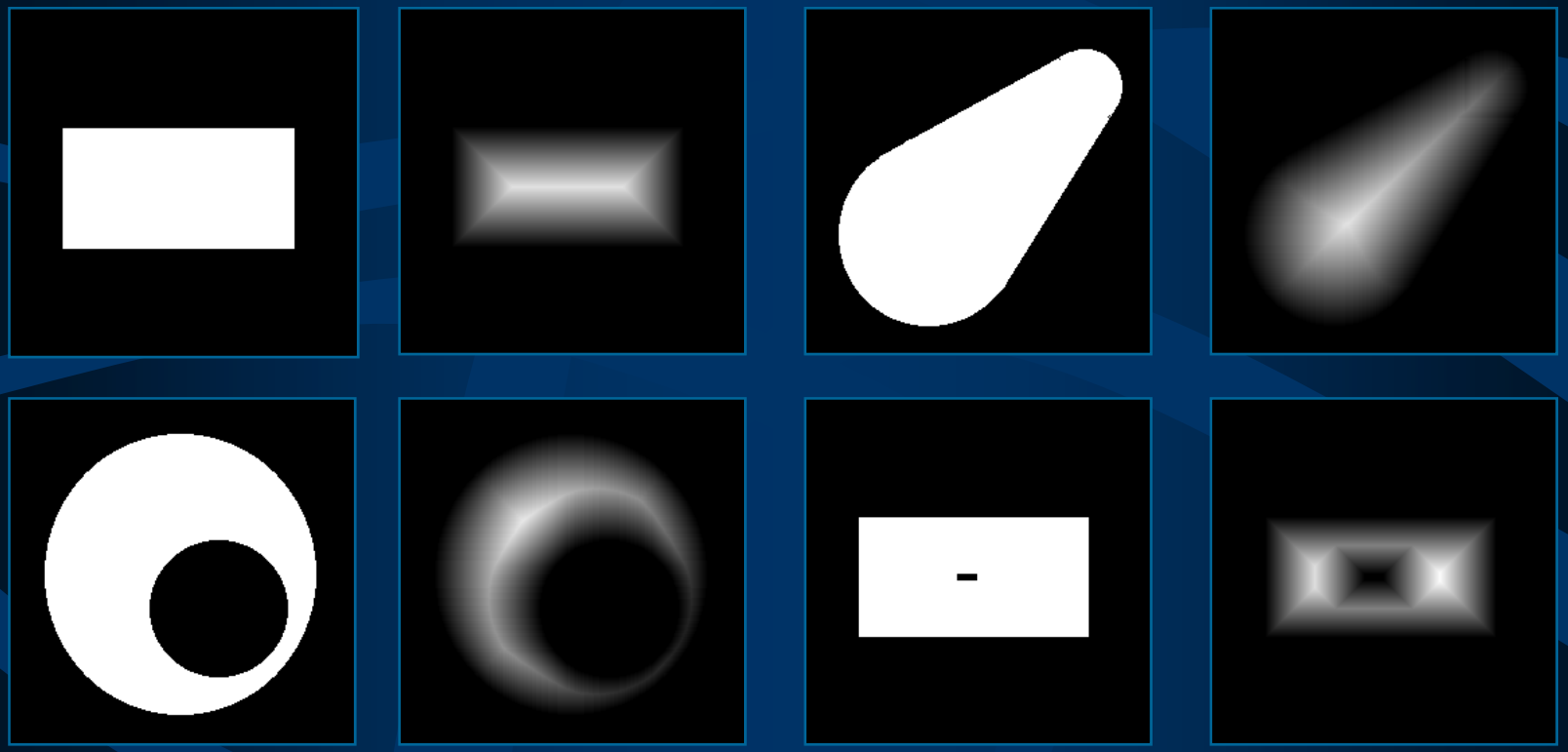


Original | Euclidean metric | City block metric | Chessboard metric

Distance Transform

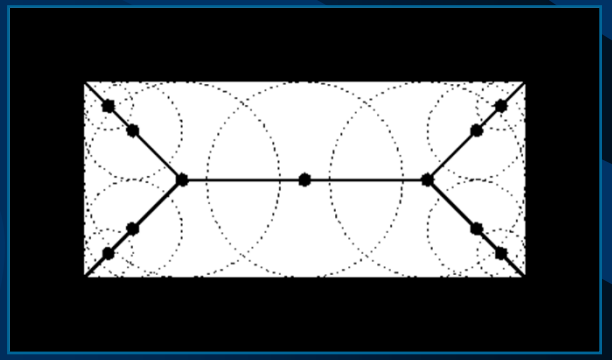
- Perform multiple recursive erosions with a suitable SE until all foreground regions of the image have been eroded away.
- Label each pixel with the number of erosions that had been performed before it disappeared, then get the distance transform result.
- Suitable SE for different distance metrics:
  - A square SE gives the chessboard distance transform
  - A cross shaped SE gives the city block distance transform
  - A disc shaped SE gives the Euclidean distance transform.

Distance Transform



Skeleton

- Skeleton is a process for reducing foreground regions in a binary image
- preserves the topology (extent and connectivity) of the original region while throwing away most of the original foreground pixels
- *locus of centers* of bi-tangent circles that fit entirely within the foreground region



Skeleton

- **Grass fire:** Imagine that the foreground regions in the input binary image are made of some uniform slow-burning material. Light fires simultaneously at all points along the boundary of this region and watch the fire move into the interior. At points where the fire traveling from two different boundaries meets itself, the fire will extinguish itself and the points at which this happens form the so called `quench line'. This line is the skeleton.

Skeleton

- *Skeleton subset* $S_i(F)$ is defined as:

$$S_i(F) = (F \ominus r_i K) - [(F \ominus r_i K) \circ K] \quad i = 0, 1, 2, \ldots, n$$

where n *is the largest value of* i *before the set* $S_i(F)$ *becomes empty. SE* K *is chosen to approximate a disc*

- *Skeleton* is then the union of the skeleton subsets:

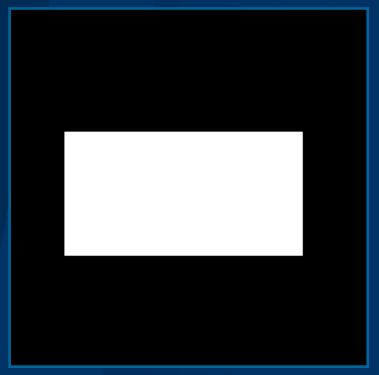
$$S(F) = \bigcup_{i=0}^{n} S_i(F)$$

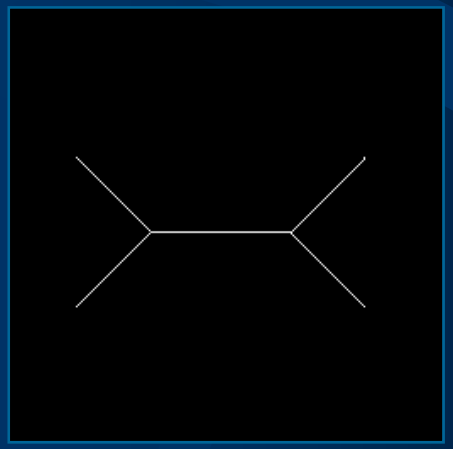
Skeleton Reconstruction

- *Reconstruction*: the original object can be reconstructed by given knowledge of the skeleton subsets $S_i(F)$, the SE K, and i:

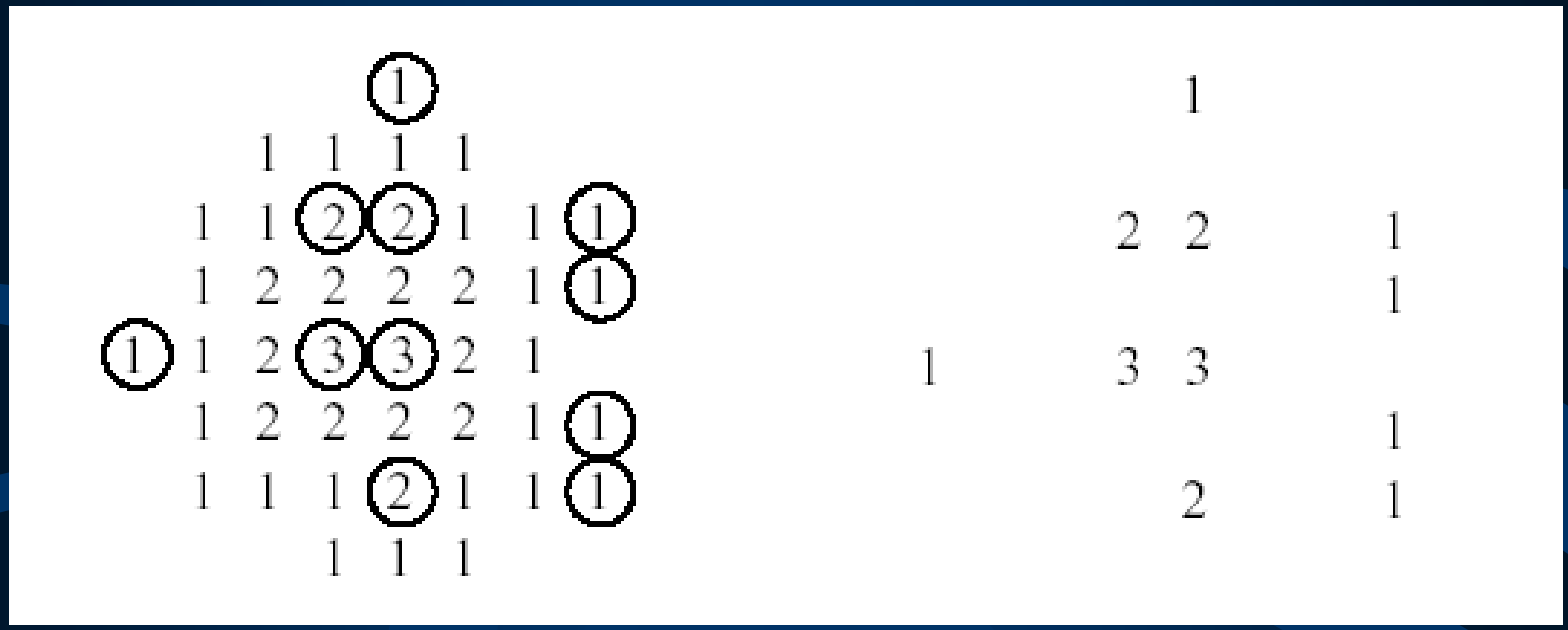
$$F = \bigcup_{i=0}^{n} (S_i(F) \oplus r_i K)$$

- Examples of skeleton:





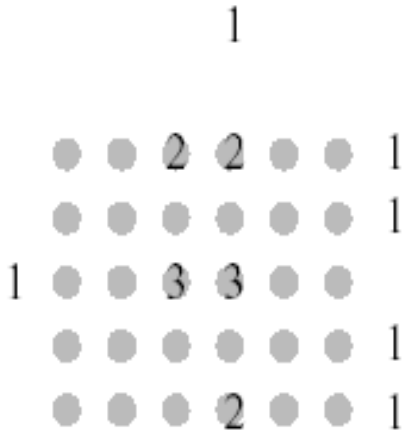
Skeleton *vs.* DT



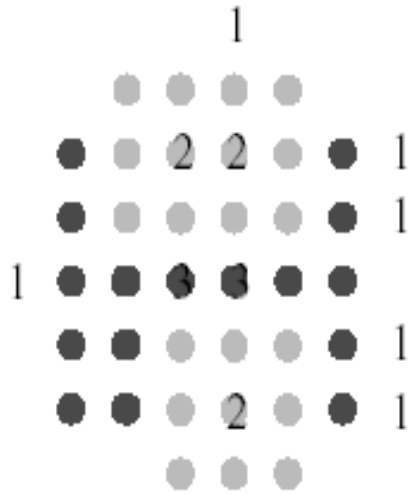
pixels whose values are greater than or equal to every pixel in their 8-neighborhood.(local maxima of pixel values)

skeleton = set of maxima.

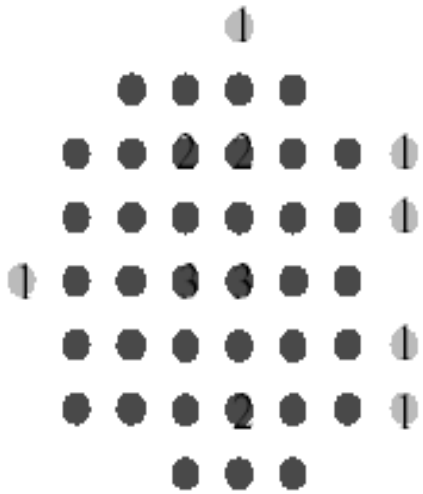
Skeleton Reconstruction



1. squares of "size 2" are centered on the pixels with "3."

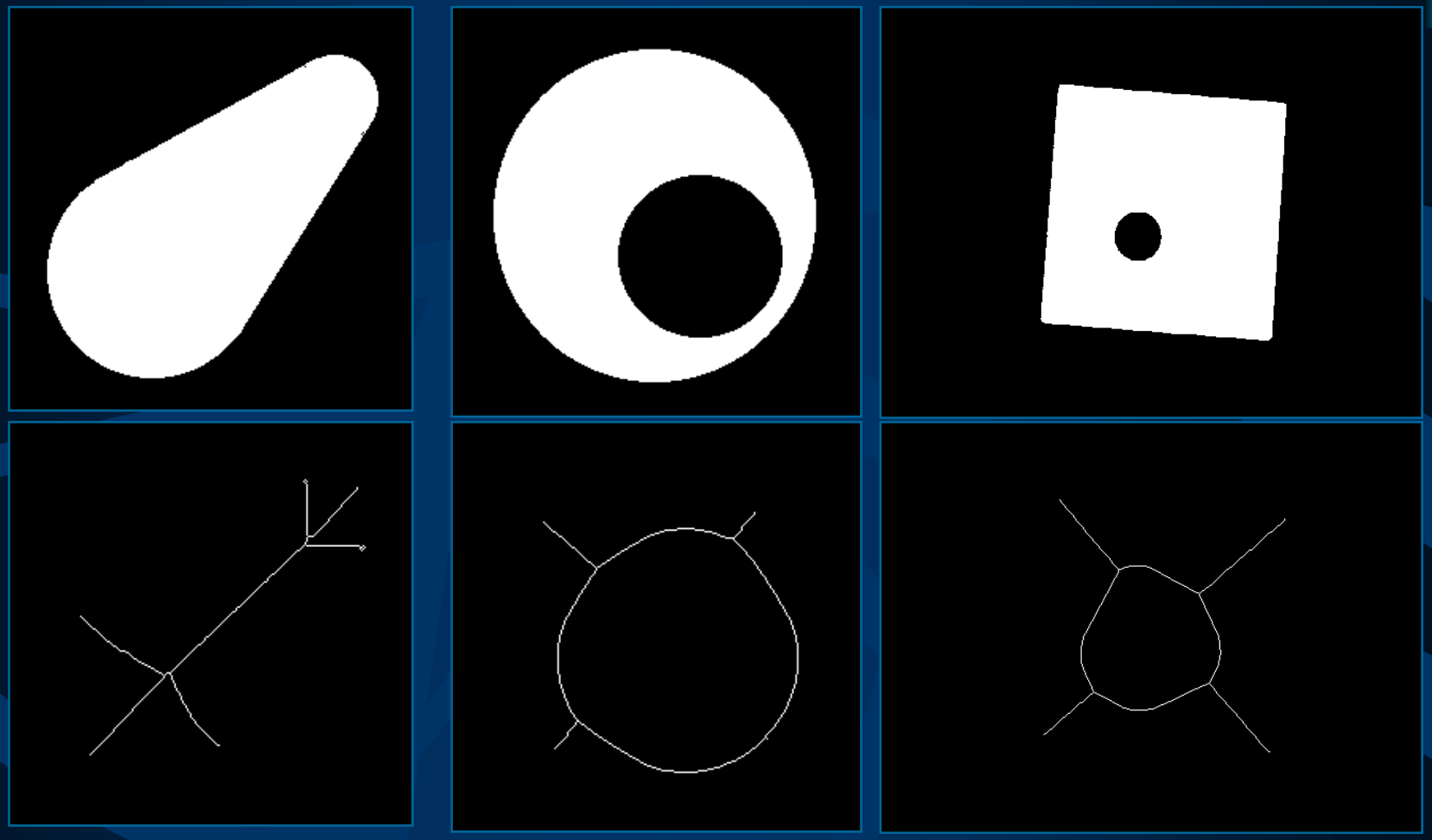


2. squares of "size 1" are centered on the pixels with "2."

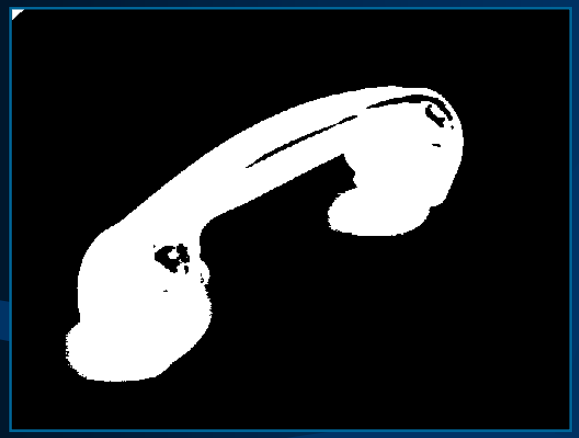


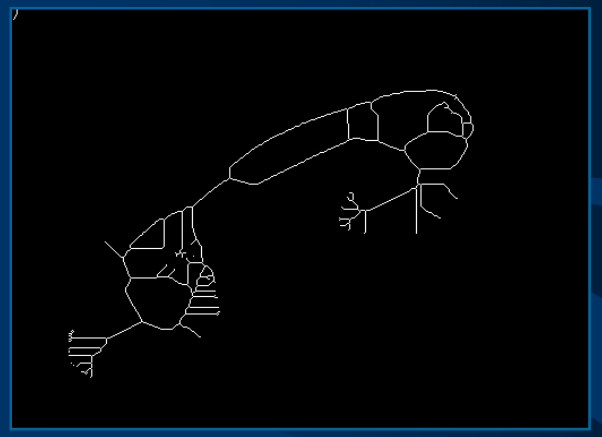
3. one-pixel points are centered on the pixels with "1."

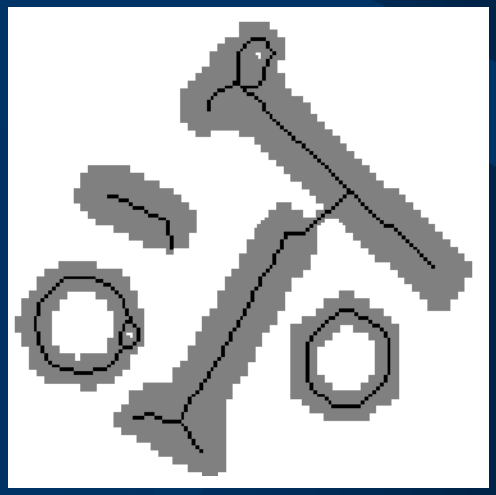
Skeleton



Skeleton







VTK 的相关类

- VTK:
  - ✓ vtkImageDilateErode3D()
  - ✓ vtkImageOpenClose3D()
  - ✓ vtkImageEuclideanDistance ()
  - ✓ vtkImageCityBlockDistance ()
  - ✓ vtkImageSkeleton2D ()

Projects

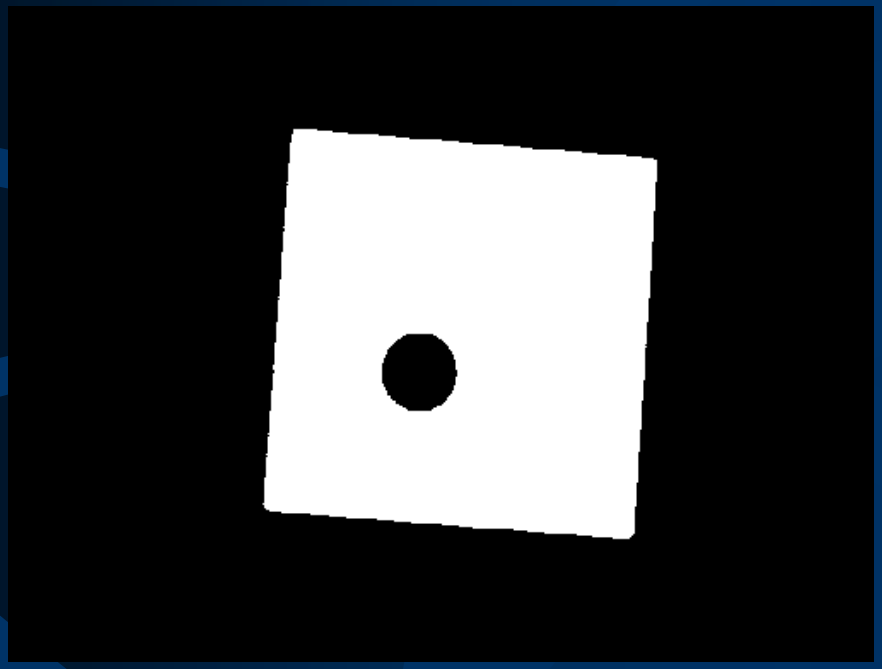
Project-3

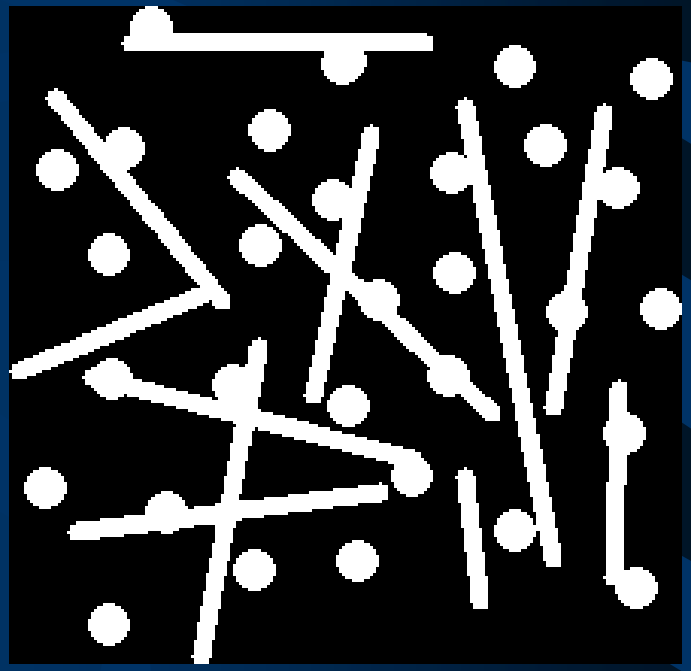
- Write your own code to realize binary dilation, erosion,opening and closing operations.
- Requirement:
  - Design your own UI and display I/O images
  - Try to apply fast operations in case

Project-4

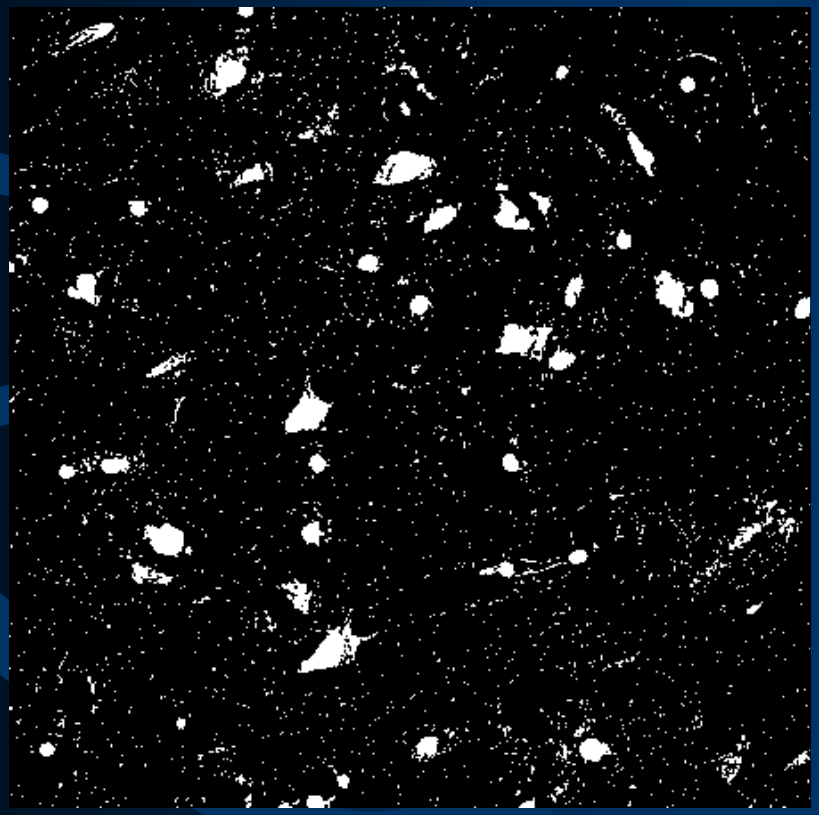
- Write code to realize the next functions:
  - Morphological distance transform
  - Morphological skeleton
  - Morphological skeleton restoration
- Requirement:
  - Design your own UI and display I/O images

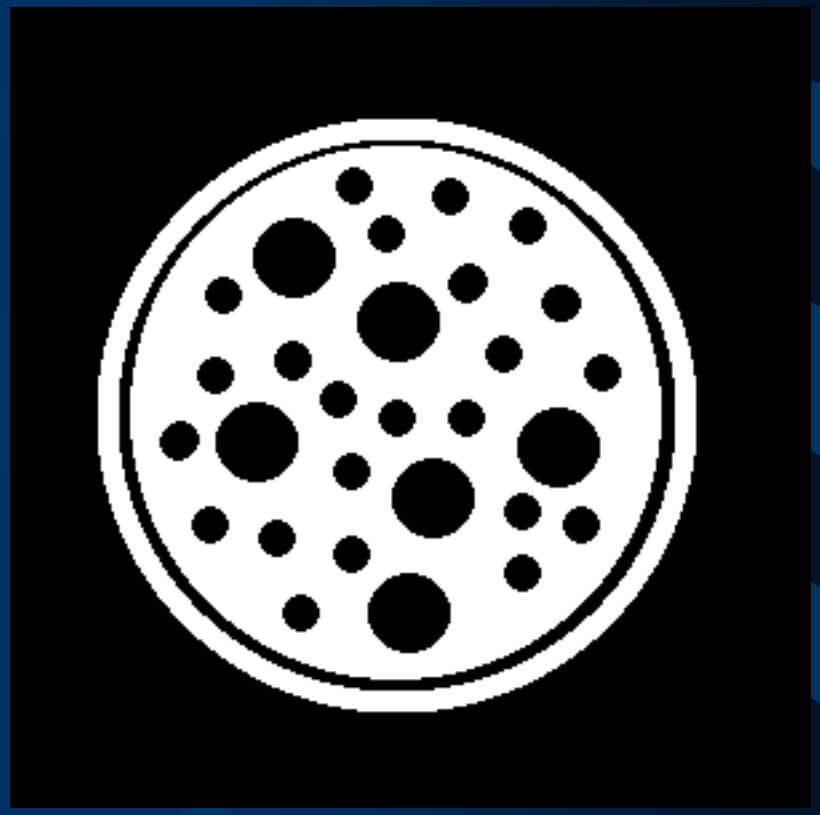
Test Images



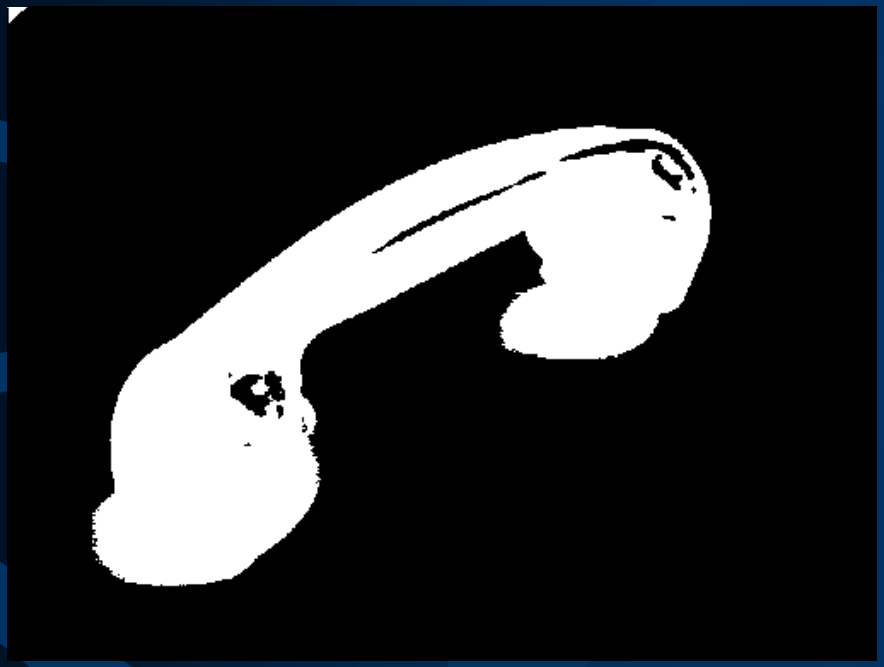


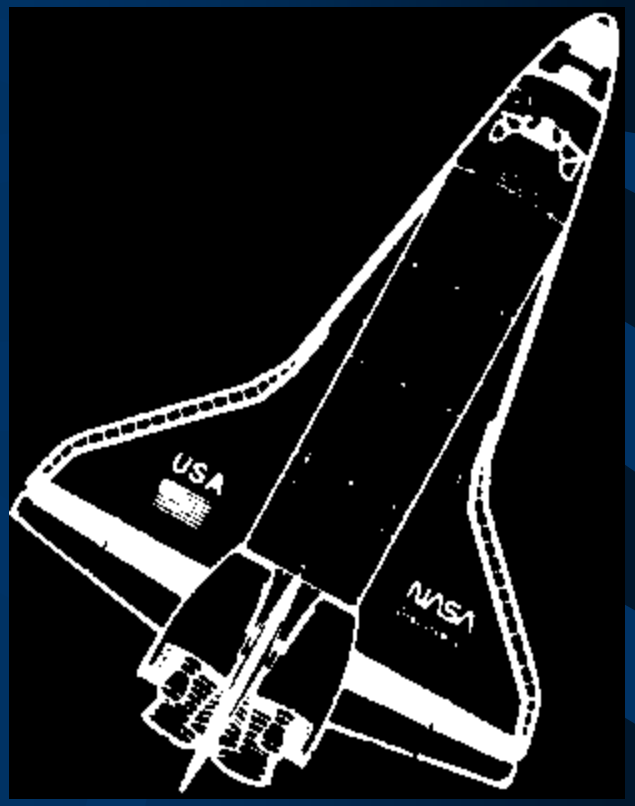
Test Images





Test Images





Discussion