Nonlinear filtering of images

Paweł Forczmański
Chair of Multimedia Systems, Faculty of Computer Science and Information Technology

Agenda

1. nonlinear filtering: Median, maximal, minmal
2. morphological operation
3. thinning
4. Hough transform

DIP scheme

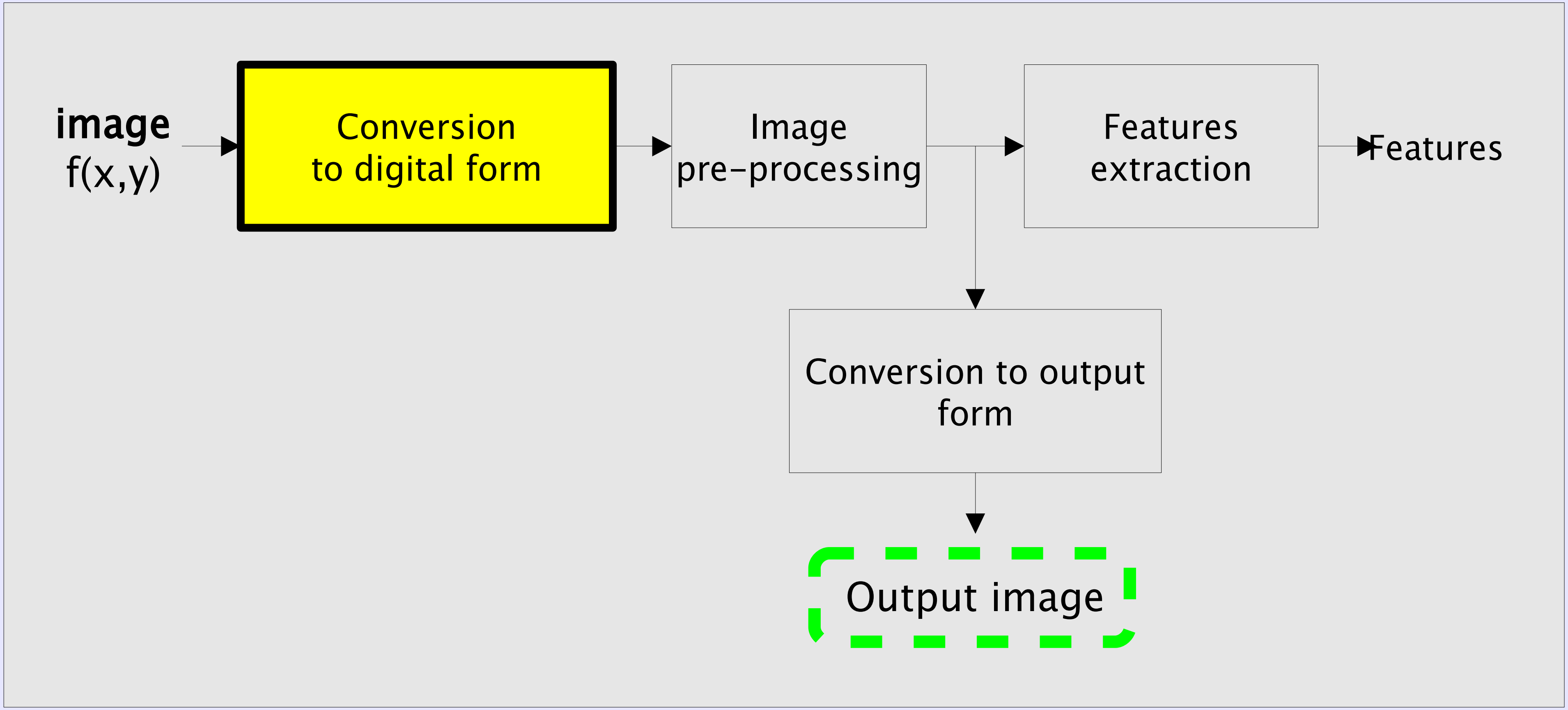


image
f(x,y)
Conversion
to digital form
Image
pre–processing
Features
extraction
Features
Conversion to output
form
Output image

Nonlinear filtering

Output image consists of a **limited number** of pixels from the input image

Output image's pixels result from a nonlinear transform of input image's pixels and a filter mask

Median filtering

In statistics and probability theory, the **median** is the numerical value separating the higher half of a data sample, a population, or a probability distribution, from the lower half.

The **median of a finite list of numbers** can be found by arranging all the observations from lowest value to highest value and picking the **middle one**

The median is of central importance in robust statistics, as it is the **most resistant statistic**, having a breakdown point of 50%: so long as **no more than half** the data is contaminated, the median will not give an arbitrarily large result.

A median is only defined on **ordered one-dimensional data**, and is independent of any distance metric.

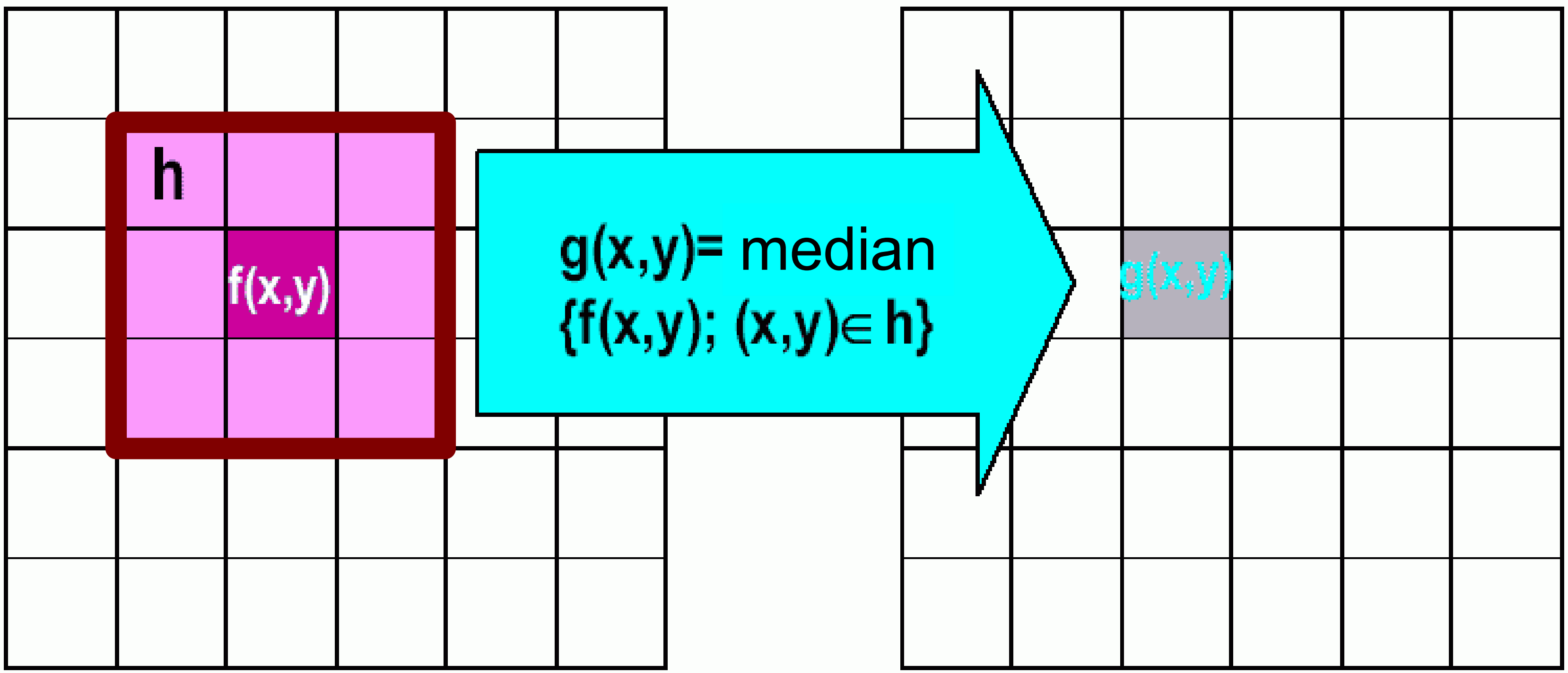
Median: example

Input set:

A={9, 88, 1, 15, 43, 100, 2, 34,102}

- Sort elements in A (increasing order): B=sort(A) → B={1, 2, 9, 15, 34, 43, 88, 100, 102}
- Select median of B (middle element): mediana(B)=34

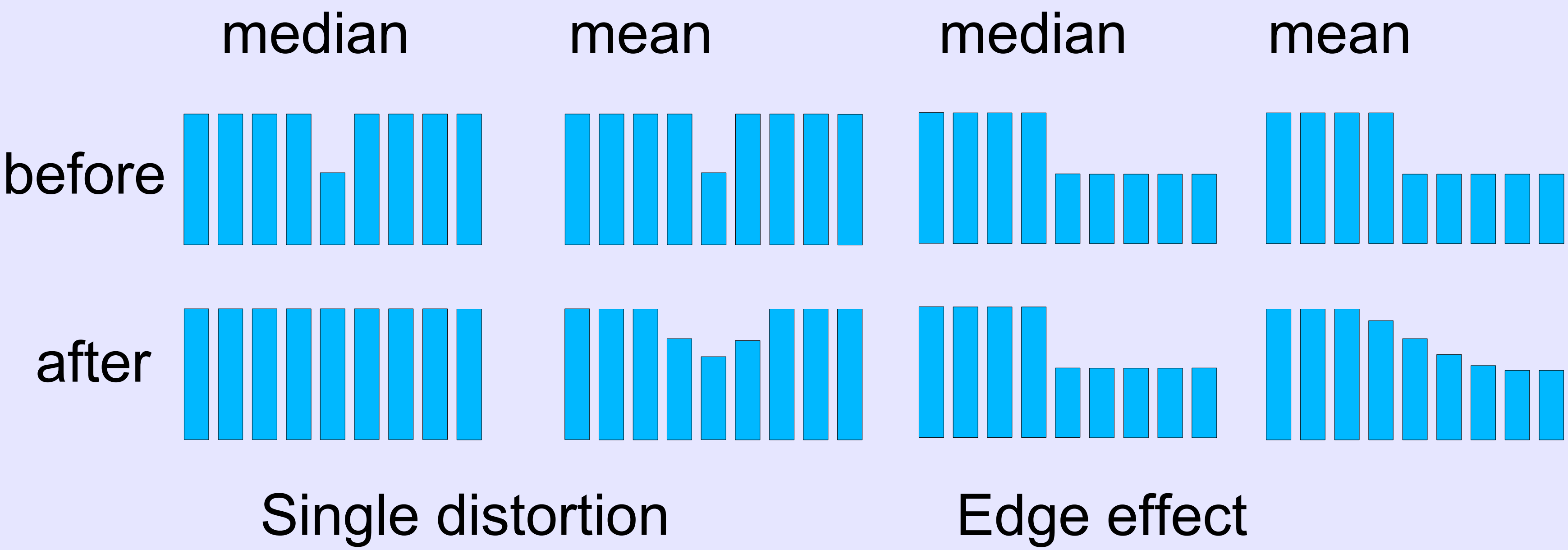
Implementation of an image



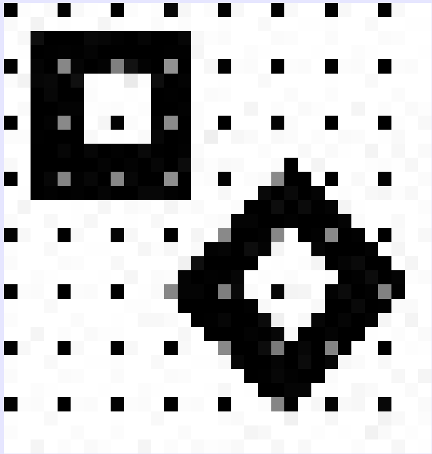
Input image f

output image g

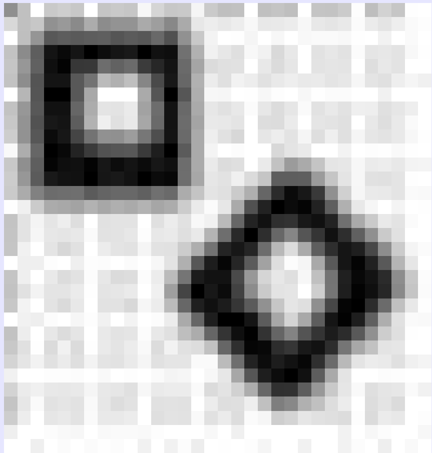
Interesting properties



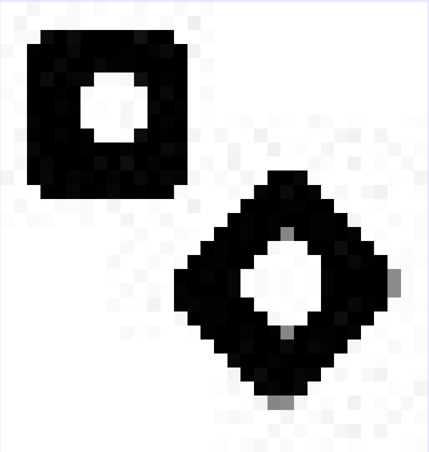
properties



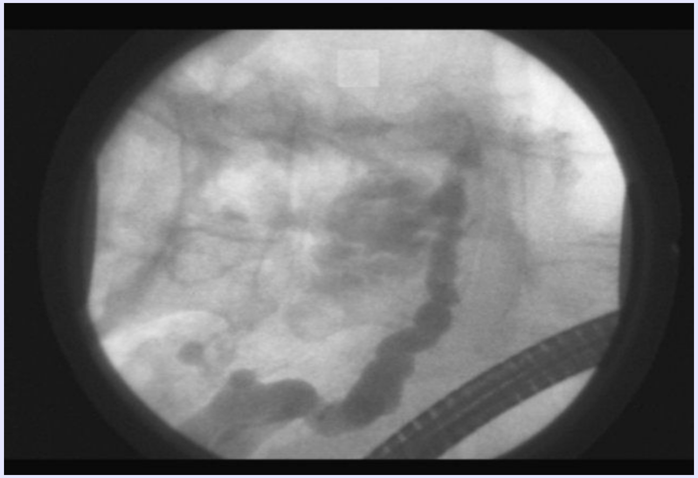
Input image



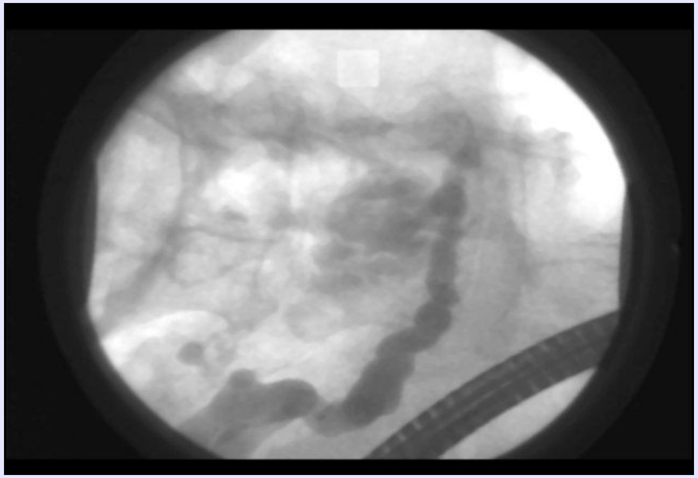
mean



median

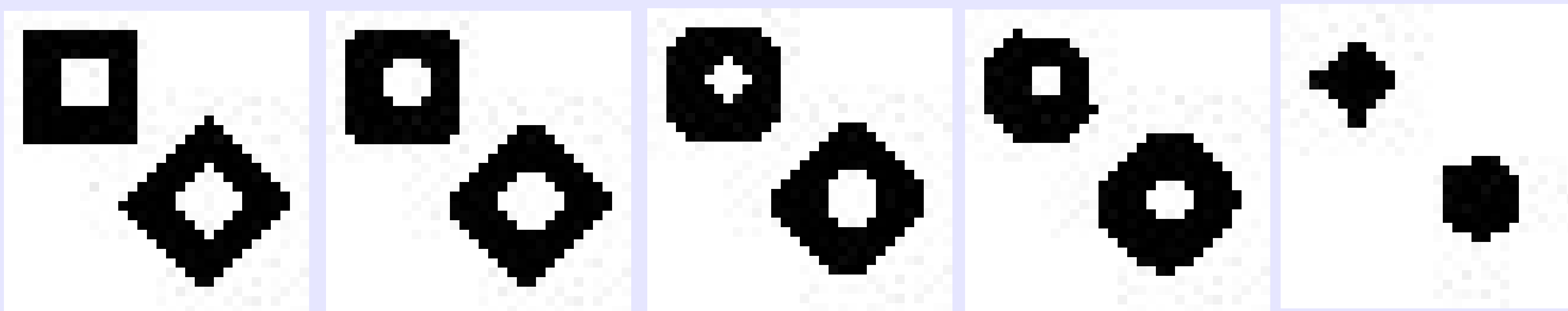


X-Ray image



After median filtering

properties



input 3x3 5x5 7x7 9x9

Other nonlinear filters

- maximal filtering

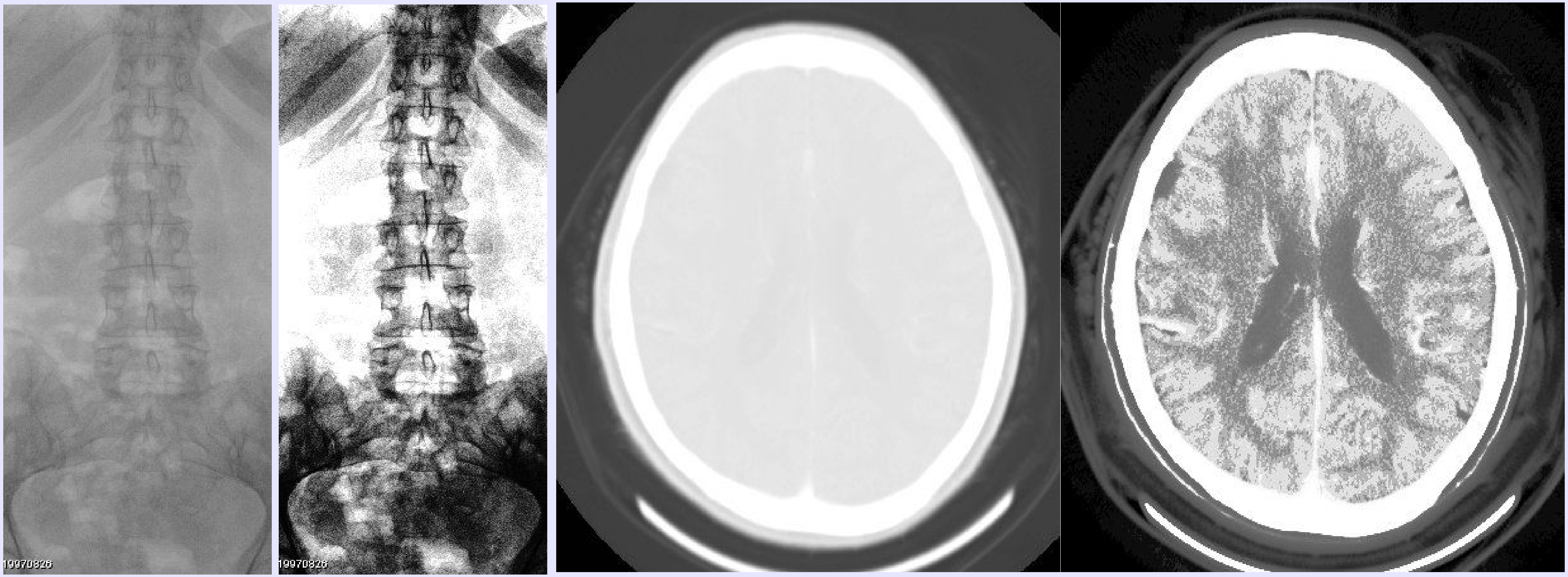
B=sort(A) → B={ 1, 2, 9, 15, 34, 43, 88, 100, **102** }

- minimal filtering

B=sort(A) → B={ **1**, 2, 9, 15, 34, 43, 88, 100, 102 }

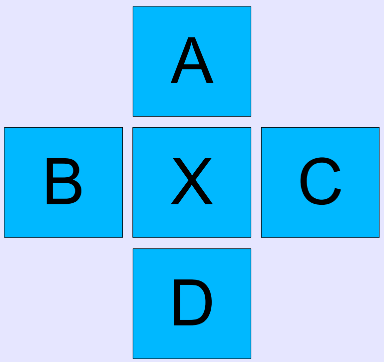
Context transformation

Change their effects depending on the image characteristics : **adaptive filtering**



Context transforms

Output pixel's value depend on the input image properties:



$$X' = \begin{cases} A & \text{for } A = D \\ X & \text{In other cases} \end{cases}$$

$$X' = \begin{cases} B & \text{for } B = C \\ X & \text{In other cases} \end{cases}$$

$$X' = \begin{cases} A & \text{for } A = B = C = D \\ X & \text{In other cases} \end{cases}$$

For gray-scale images:

$$X' = \begin{cases} A & \text{for } |A - D| < \varepsilon \\ X & \text{In other cases} \end{cases}$$

Morphology

Mathematical morphology (MM) is a theory and technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions.

MM is most commonly applied to digital images, but it can be employed as well on graphs, surface meshes, solids, and many other spatial structures.

MM is the foundation of morphological image processing, which consists of a set of operators that transform images.

MM was originally developed for binary images, and was later extended to grayscale functions and images.

Morphological operators

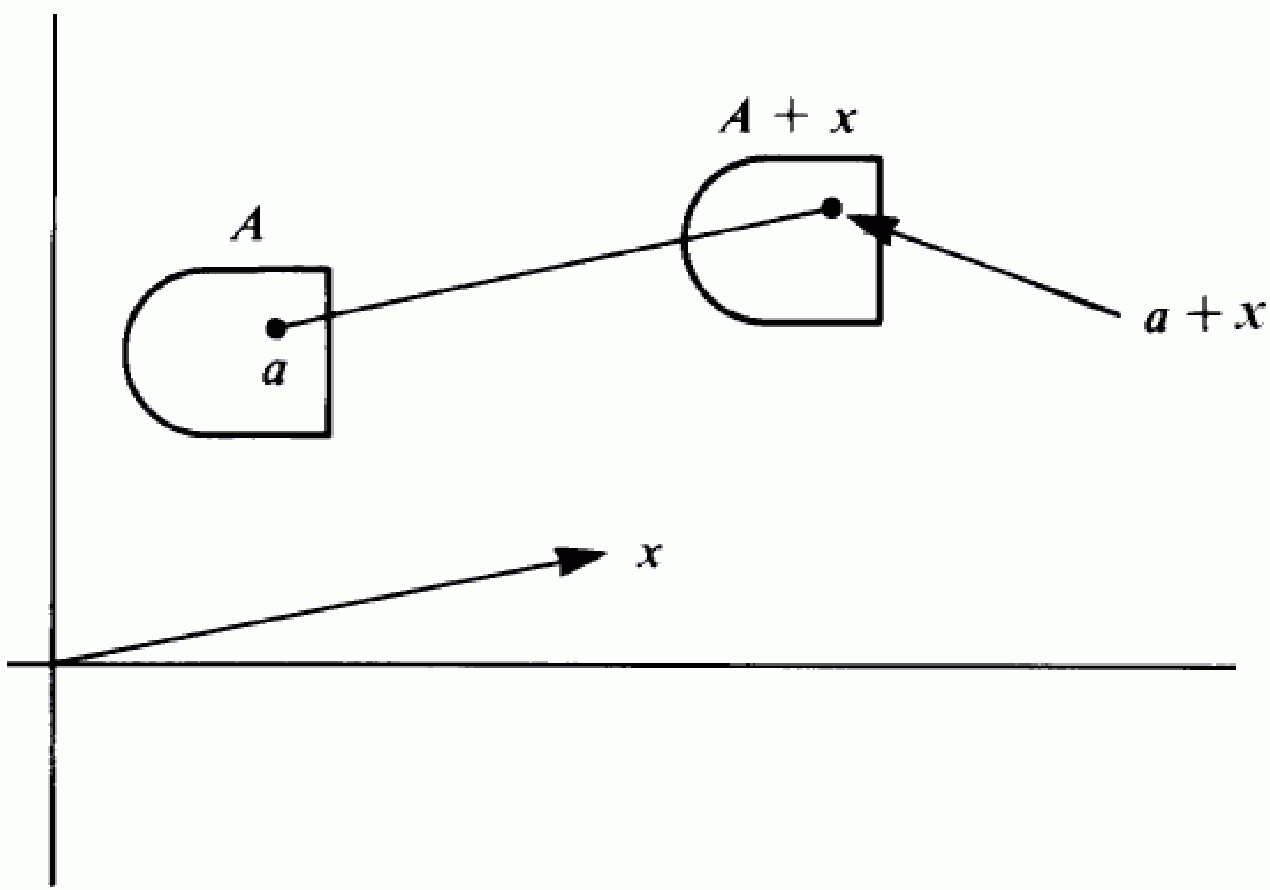
translation,

diltation,

erosion

Morphological operators

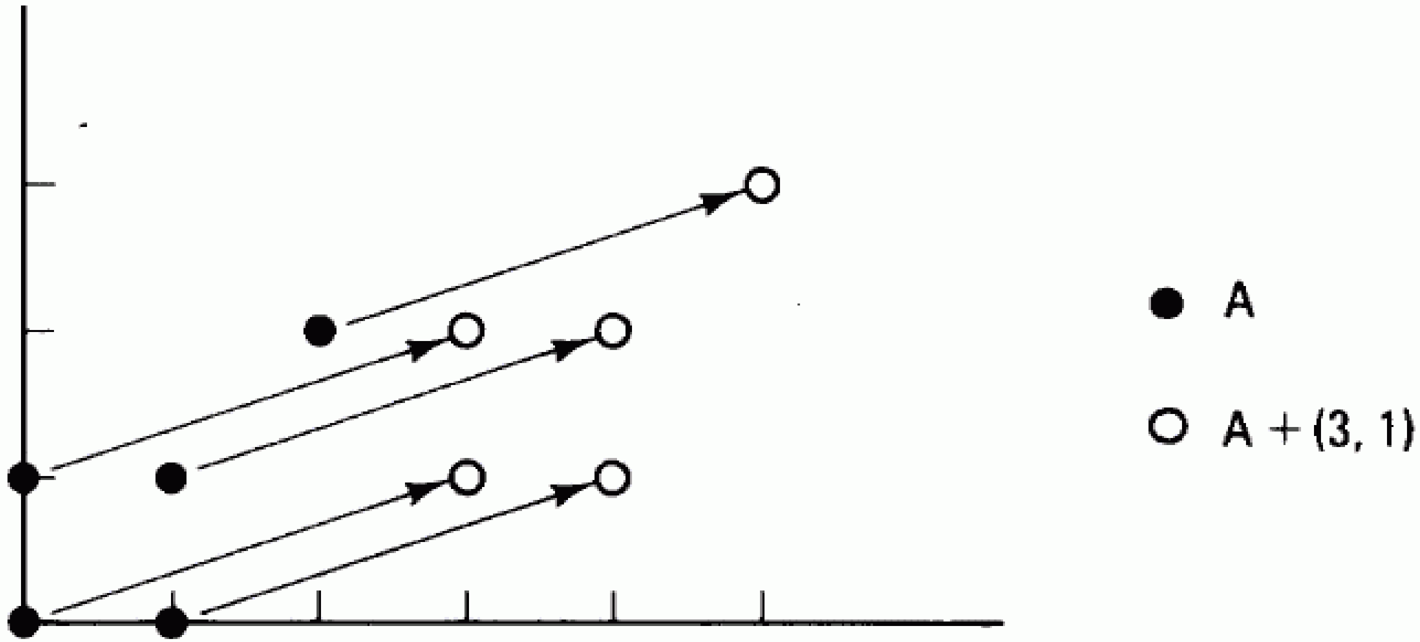
Translation of a set $A \subset R^2$ of vector $x \subset R^2$ is defined as: $A + x = \{a + x : a \in A\}$



Morphological operators

For a discrete set:

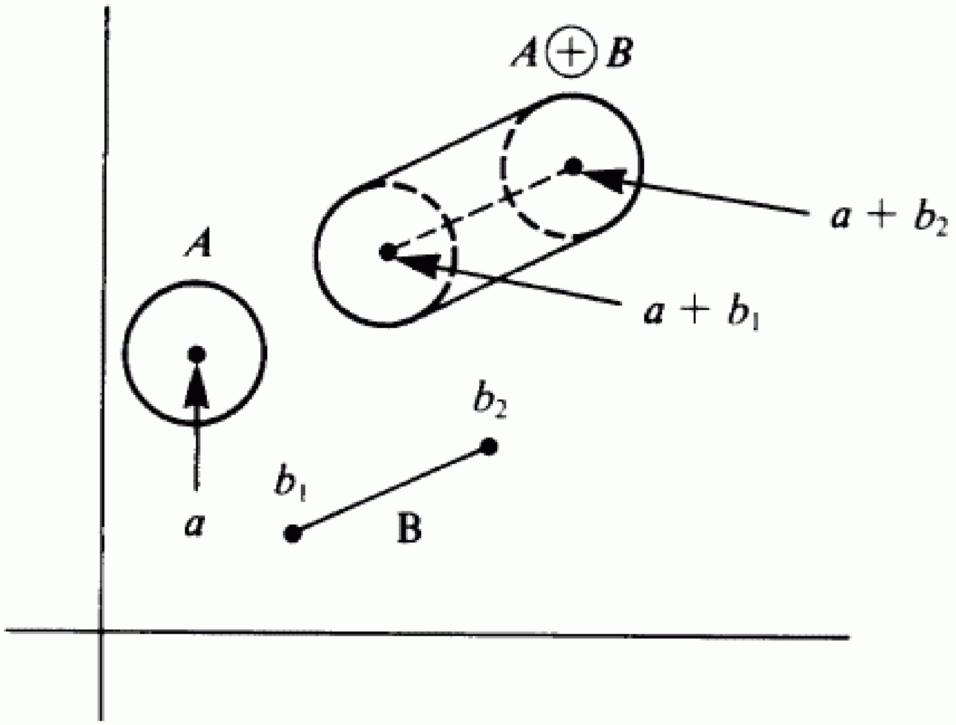
x=[3,1]



Morphological operators

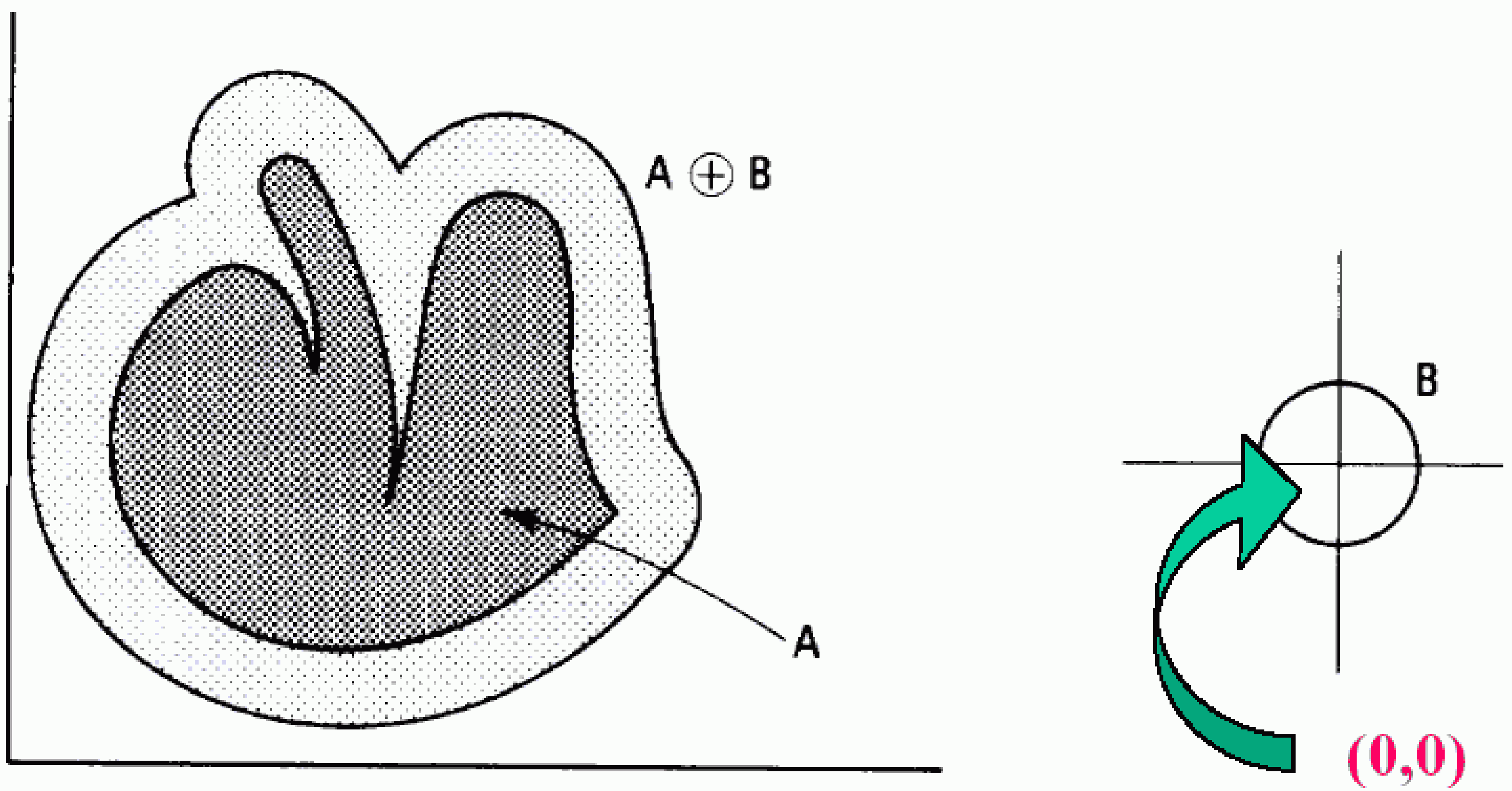
Dilation of a set A by B $(A,B \subset R^2)$**, where:**
B is called a structuring element is defined as:

$$A \oplus B = \bigcup_{b \in B} (A + b)$$



Morphological operators

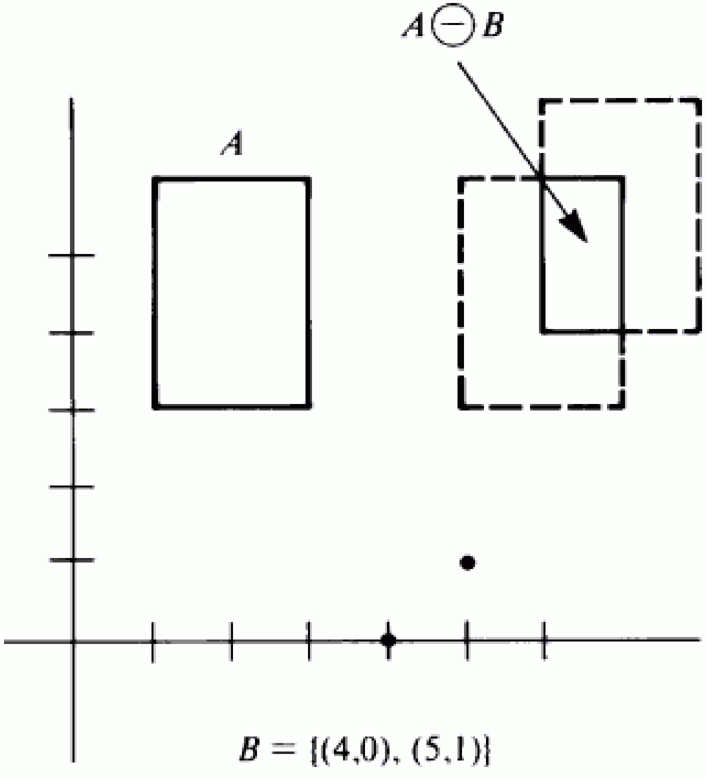
For a continious set:



Morphological operators

Erosion of a set A by B $(A,B \subset R^2)$, **Where:**

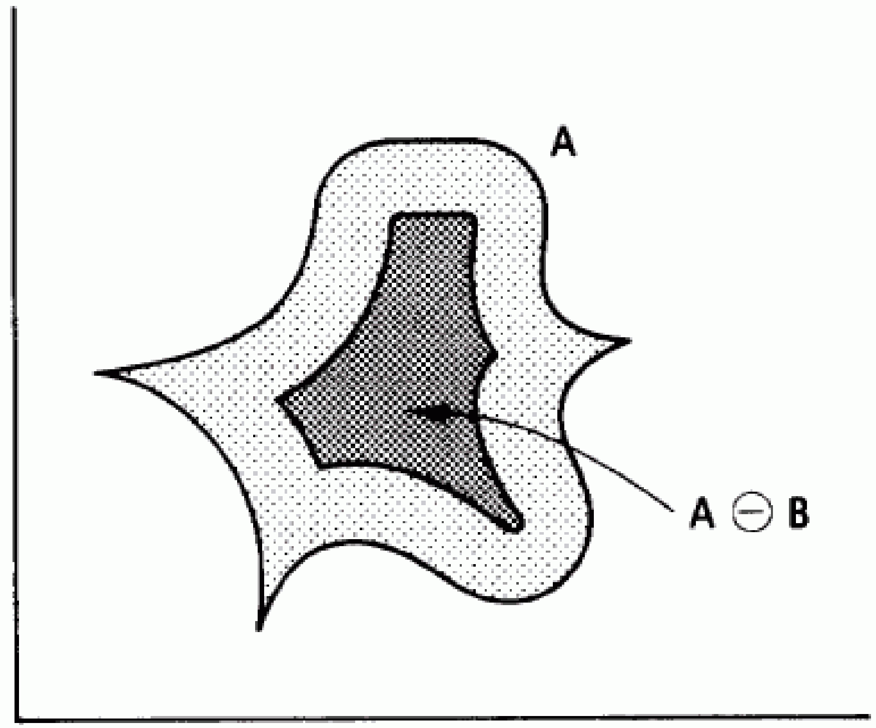
B is a structuring element is defined as:

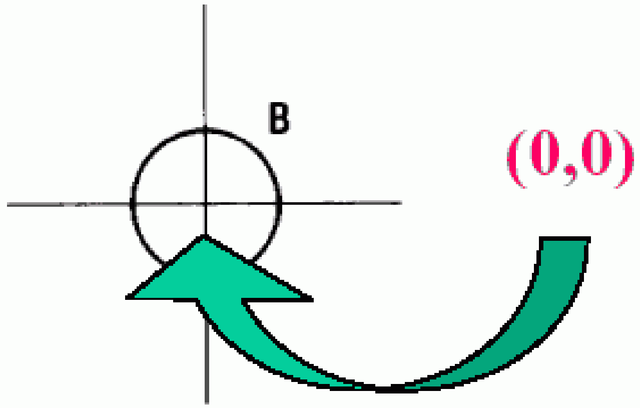


$$A \otimes B = \bigcap_{b \in B} (A + b)$$

Morphological operators

For a continious set:

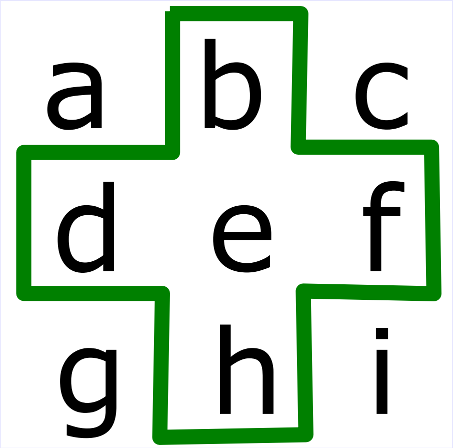




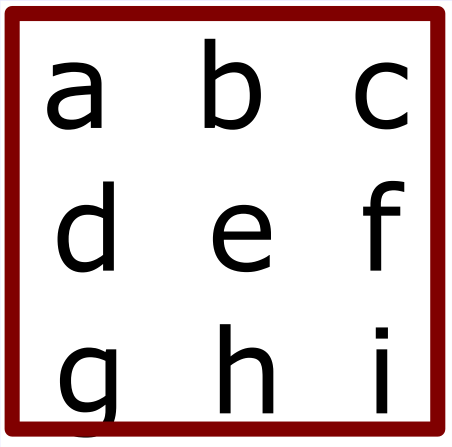
For binary images

Structuring element B is moved to the subsequent pixles. In each location it is used to perform a specific operation:

4 neighbors



8 neighbors



Erosion

$$e' = b \wedge d \wedge e \wedge f \wedge h$$

$$e' = a \wedge b \wedge c \wedge d \wedge e \wedge f \wedge g \wedge h \wedge i$$

Dilation

$$e' = b \vee d \vee e \vee f \vee h$$

$$e' = a \vee b \vee c \vee d \vee e \vee f \vee g \vee h \vee i$$

Denoising

$$e' = ((b \wedge d \wedge f \wedge h) \vee e) \wedge (b \vee d \vee f \vee h)$$

$$e' = ((a \wedge b \wedge c \wedge d \wedge f \wedge g \wedge h \wedge i) \vee e) \wedge (a \vee b \vee c \vee d \vee f \vee g \vee h \vee i)$$