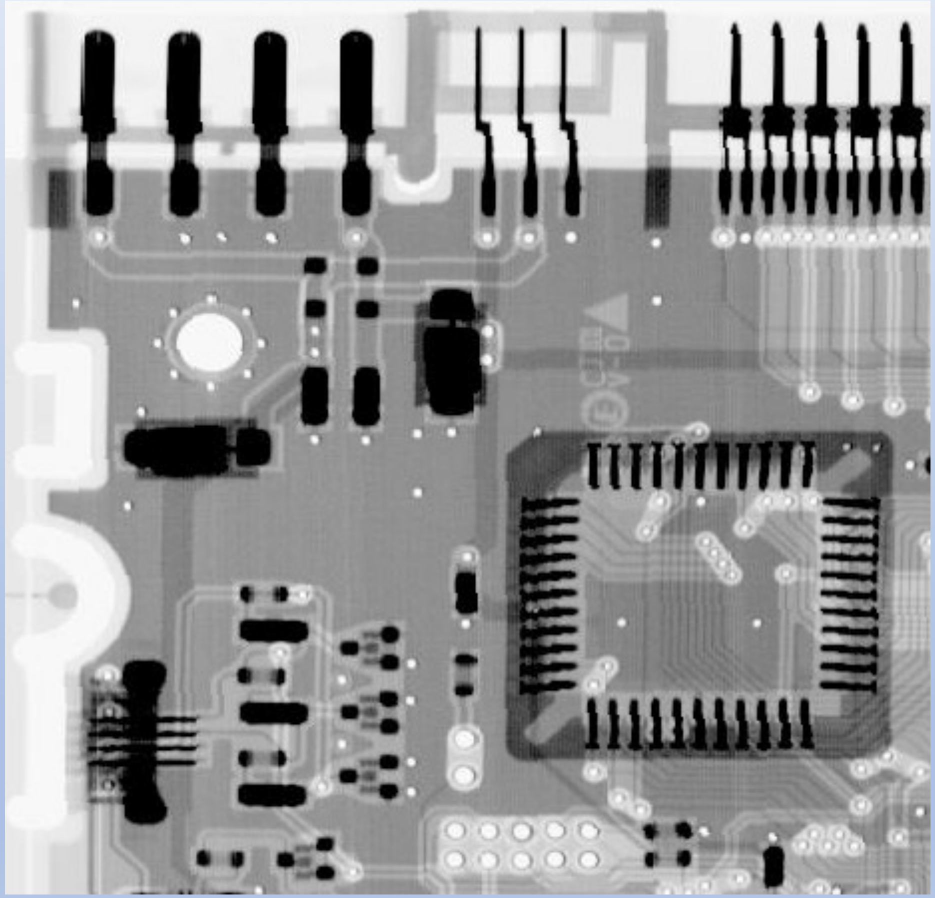
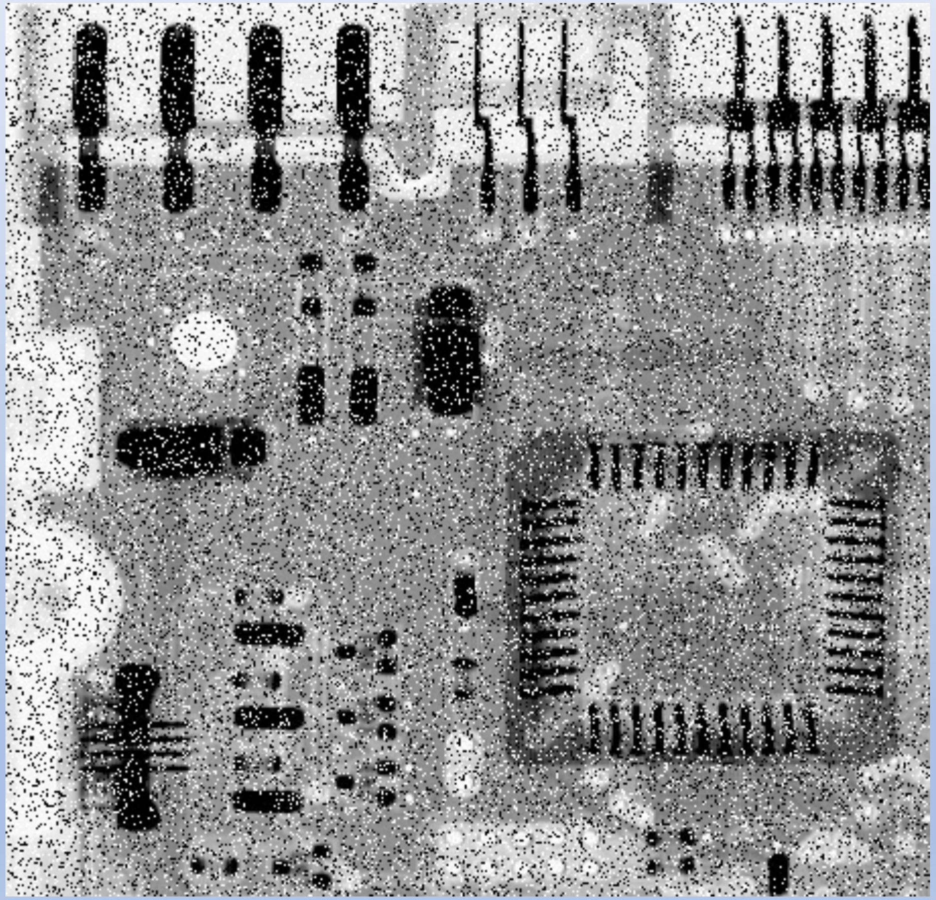
3. Image Enhancement—nonlinear processing

Observe an image and its noise contaminated version

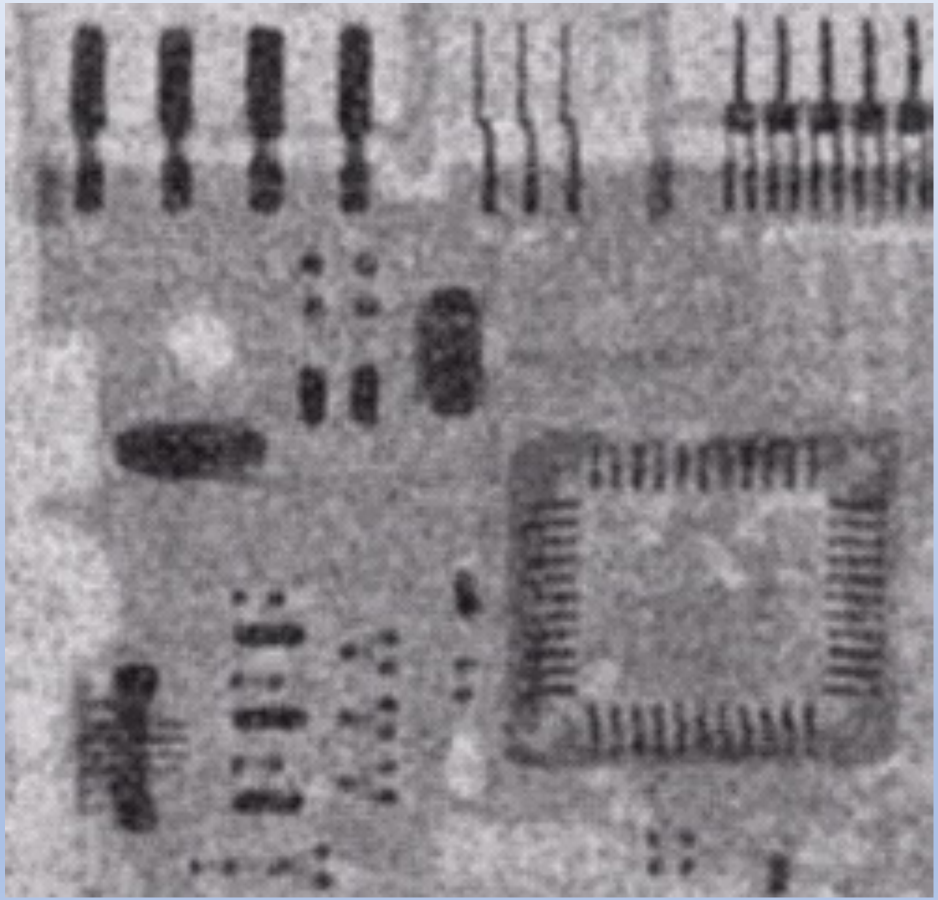


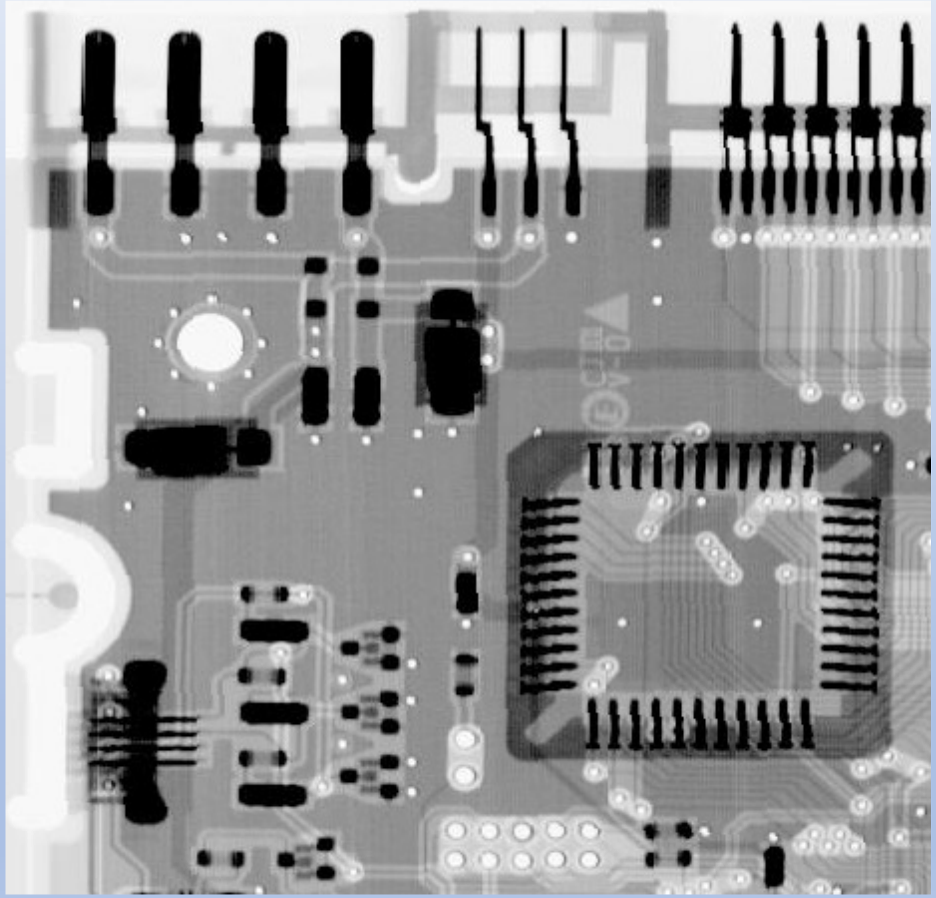


What are the noise characteristics? How to remove such noise?

3. Image Enhancement—Problems of Linear Filter

Observe the Image smoothed by a linear low pass filter

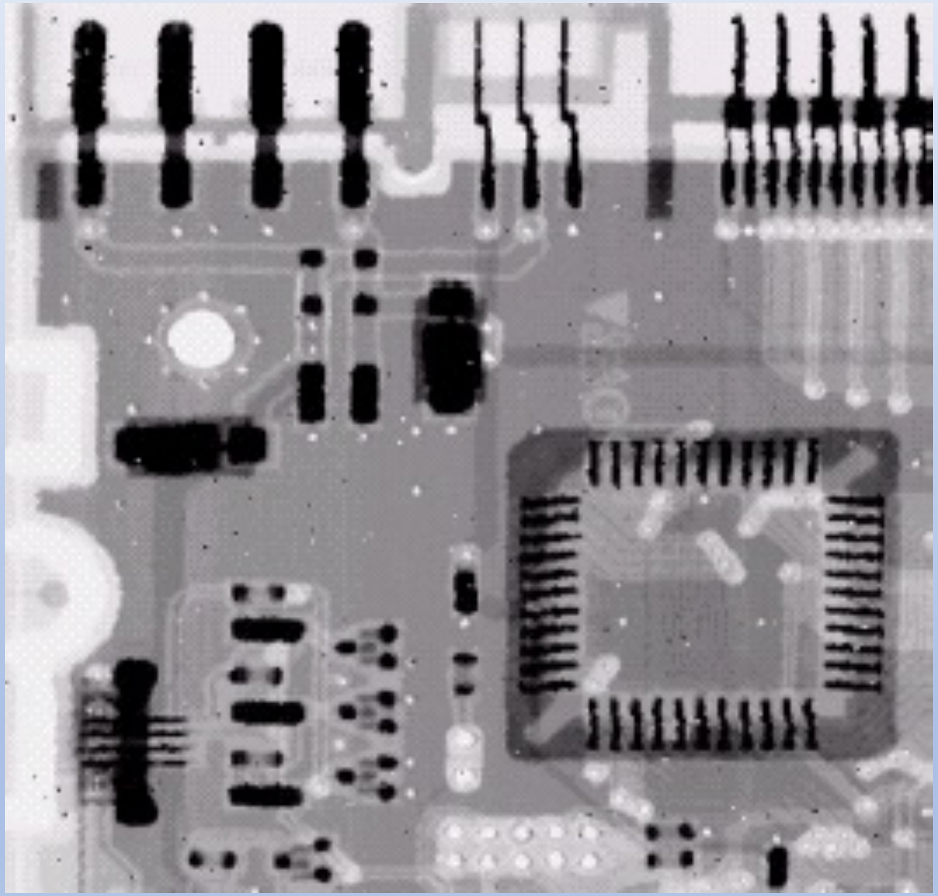


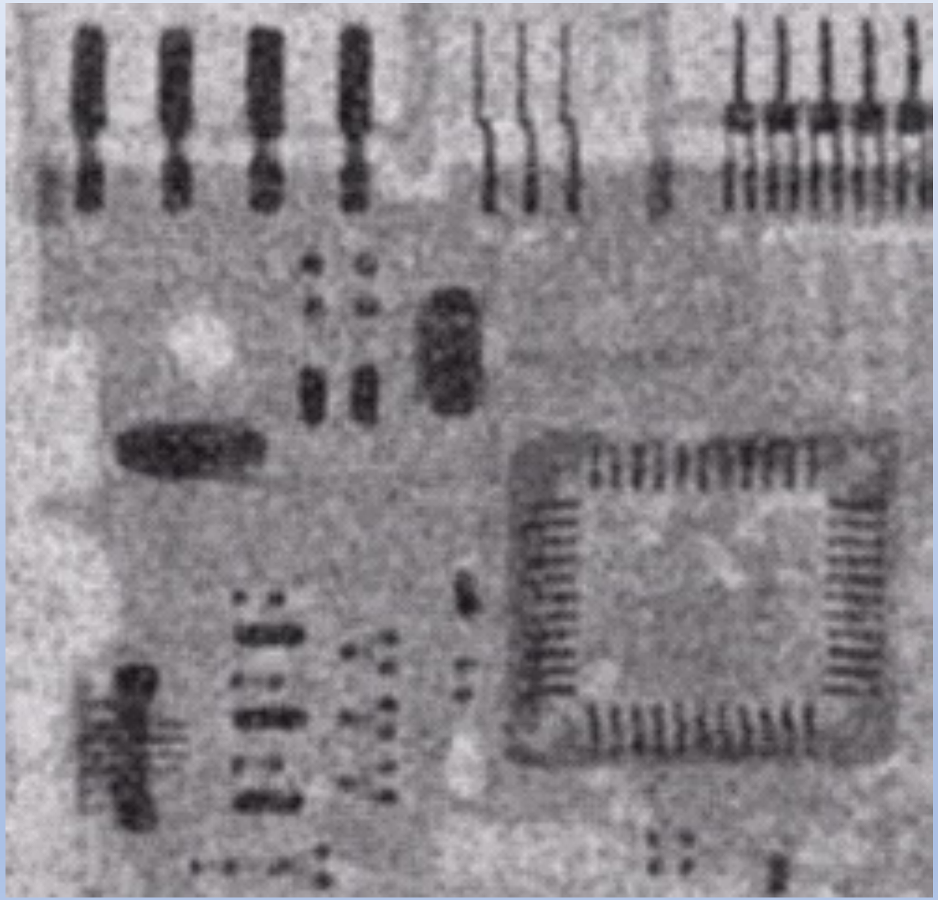


What are its problems comparing to the original image? Why?

3. Image Enhancement—Problems of Linear Filter

See another smoothed image comparing to the previous one





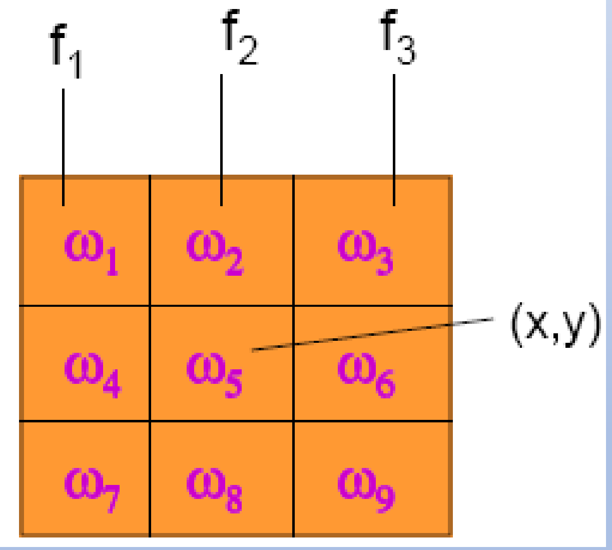
How is this smoothed image much better than the previous one?

3. Image Enhancement—Problems of Linear Filter

- Any linear filter output is a weighted average of the input pixels

$$\hat{f}(x,y) = h(x,y) * f(x,y) = \sum_{i=-a}^{a}\sum_{j=-b}^{b} h(i,j) f(x-i, y-j)$$

$$= \sum_{(s,t)\in S_{xy}} \omega(s,t) f(s,t)$$



➢ What are problems of the average of pixel grey values?

image blurring, sharpness details are lost, difficult to smooth strong noise

3. Image Enhancement—Order-Statistic Filters

- The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter.

- The best-known example is median filter, which replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel.

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{f(s,t)\}$$

10	20	20
20	15	20
25	20	100

$\longrightarrow$

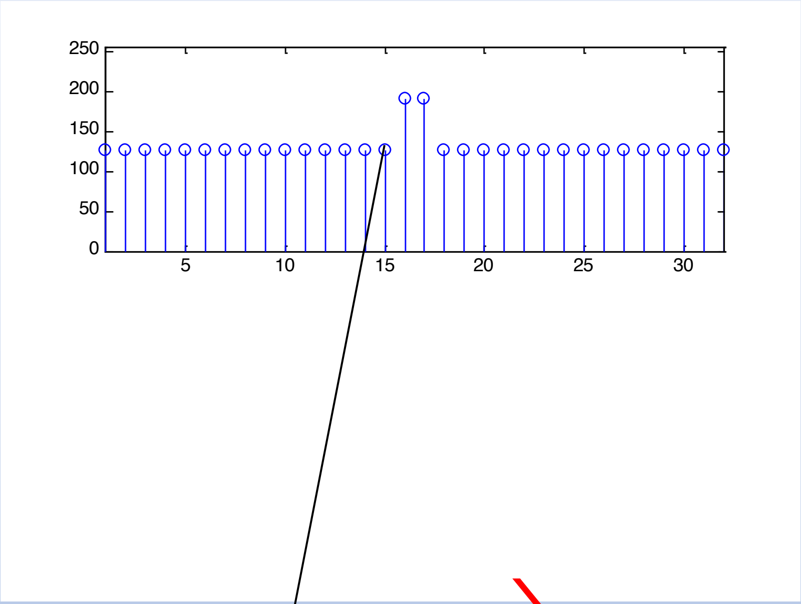
(10,15,20,20,20,20,20,25,100)

Median=20

So replace (15) with (20)

3. Image Enhancement—Mean vs. Median Filter

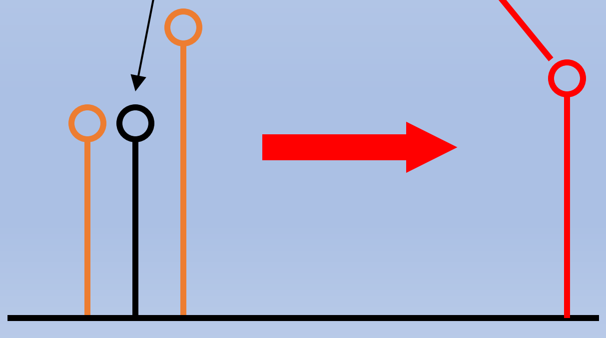
- Consider a uniform 1-D image with a pulse function.



- Pulse function corresponds to fine image detail such as lines and curves.

➢ Mean filter ‘blurs’ the image details.

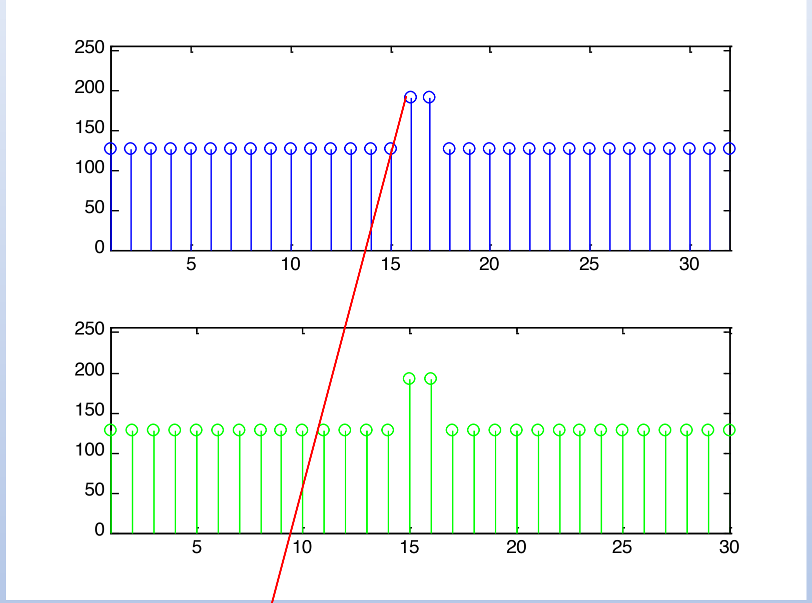
➢ If the pulse is noise, mean filter suppress it only for some extent but spread the noise.

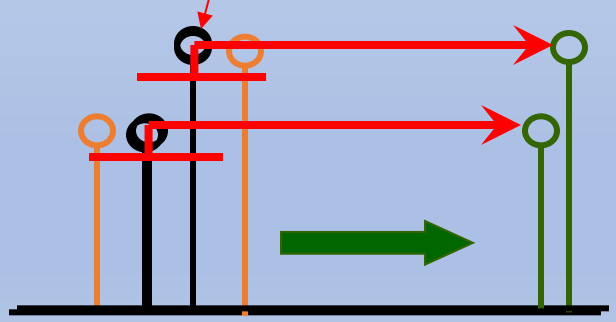


3. Image Enhancement—Mean vs. Median Filter

- Consider a uniform 1-D image with a pulse function.
- Pulse function corresponds to fine image detail such as lines and curves.
- Median filter does not 'blur' the edge.

➢ If the pulse is noise, 5X5 median filter totally remove such noise.





3. Image Enhancement—Median Filter

- Edge is a basic and significant structure of an image.

What is the outputs of a mean filter?

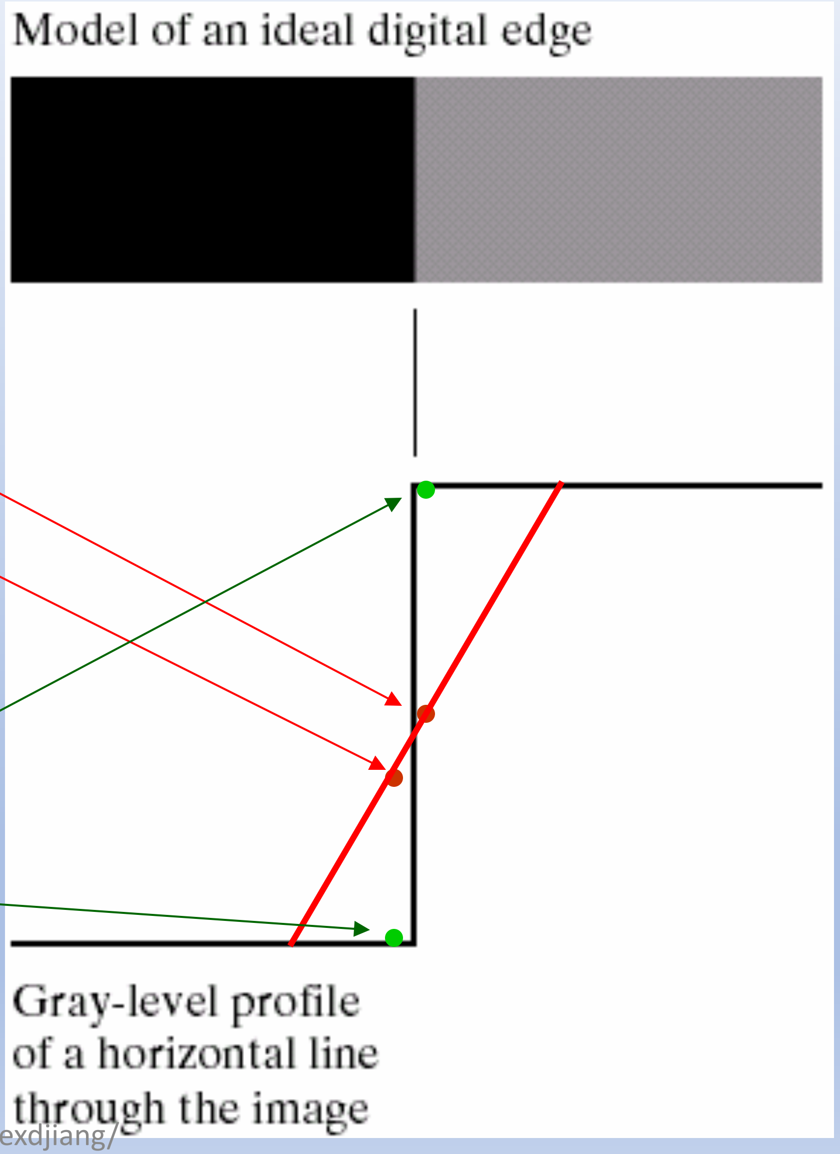
$$mean\{0,0,0,\underset{\leftrightarrow}{1},1,1,1\} = 0.57$$

$$mean\{0,0,0,\underset{\leftrightarrow}{0},1,1,1,\} = 0.43$$

What is the outputs of a median filter?

$$median\{0,0,0,\underset{\leftrightarrow}{1},1,1,1\} = 1$$

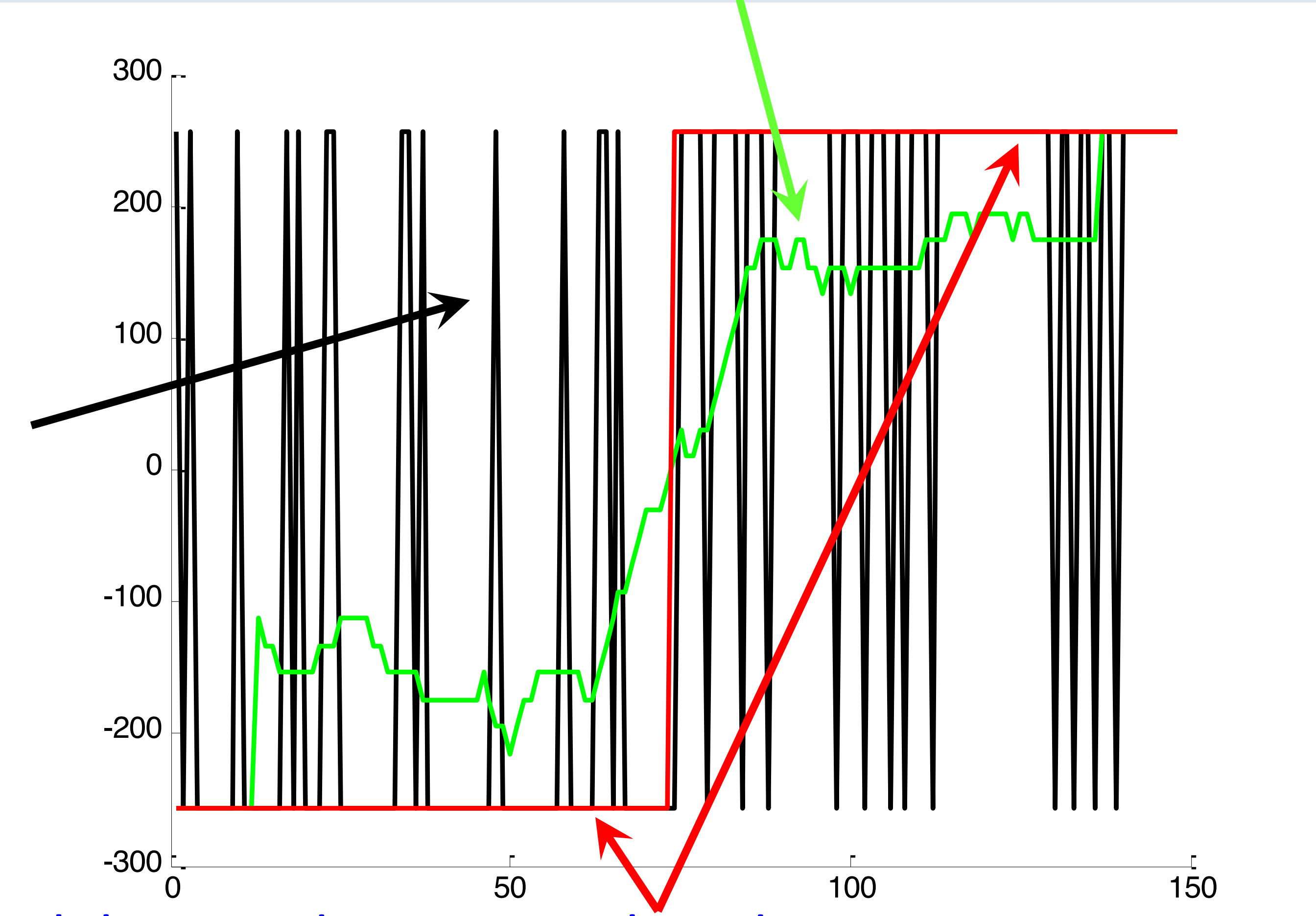
$$median\{0,0,0,\underset{\leftrightarrow}{0},1,1,1,\} = 0$$



3. Image Enhancement—Mean vs. Median Filter

A simple MATLAB program can show: Mean filter is ineffective to attenuate impulsive noise and blurs the edge.

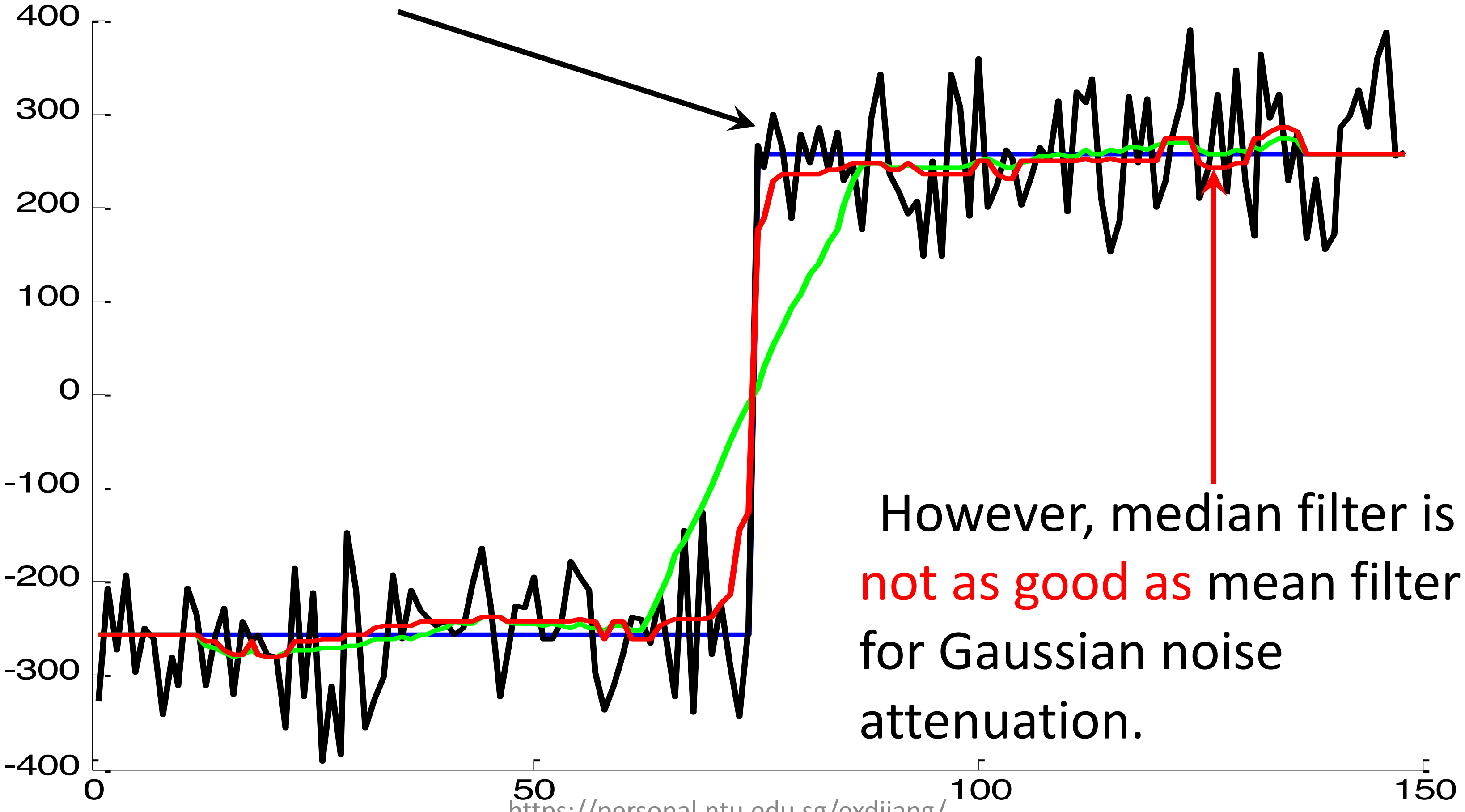
Impulsive noise is strong in amplitude and spatial sparse



Median filter provides excellent noise-reduction capabilities and preserves the edge.

3. Image Enhancement—Mean vs. Median Filter

A simple MATLAB program can show: Mean filter attenuates additive Gaussian noise but blurs the edge. Median filter attenuates Gaussian noise and preserves the edge.



However, median filter is not as good as mean filter for Gaussian noise attenuation.

3. Image Enhancement—Median Filter

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{median}\{f(s,t)\}$$

- Median filter forces the points with distinct gray levels to be more like their neighbors.
- Isolated clusters of pixels that are lighter or darker with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n x n median filter.
- eliminated = forced to have the value equal the median intensity of the neighbors.
- Larger clusters are affected considerably less.

3. Image Enhancement—Mean vs. Median Filter

Original and noise corrupted images

impulse noise ⇨ salt and pepper noise.

3. Image Enhancement—Mean vs. Median Filter

Example outputs of

mean filter and median filter.

3 Nonlinear Image Smoothing–Med. Filter Properties

- Linear filter has established theory to analyze its properties, especially in the frequency domain.
- However, It is difficult to analyze Median filter and other order-statistic filters due to their nonlinearity.

➢ Repeated applications of median filter to a signal result in an invariant signal called the "root signal". A root signal is invariant to further application of the median filter.

➢ Example: 1-D signal: Median filter length = 3

0 0 0 1 2 1 2 1 2 1 0 0 0

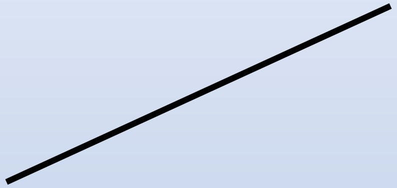
0 0 0 1 1 2 1 2 1 1 0 0 0

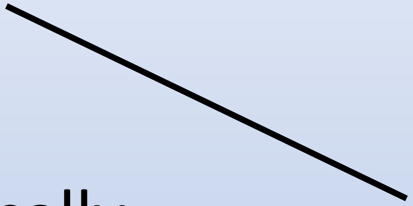
0 0 0 1 1 1 2 1 1 1 0 0 0

0 0 0 1 1 1 1 1 1 1 0 0 0 ~ root signal

3. Image Enhancement—Med. Filter Properties

- Invariant signals to a median filter:





Constant

Monotonically

increasing decreasing

length?

3. Image Enhancement—Other Order-stati. Filter

- Simple extension of the median filter

  - Max filter
$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{f(s,t)\}$$

  - Min filter
$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{f(s,t)\}$$

  - Midpoint filter
$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{f(s,t)\} + \min_{(s,t)\in S_{xy}} \{f(s,t)\}\right]$$

3. Image Enhancement—Limitation and Solution

- Although Median filter preserves image edges, it removes image details such as corner, thin lines / curves and other fine details.

- How to design a rank order filter that can effectively removes impulsive noise and preserves these image details at the same time?

- The research work on this topic can be found in the research publication:

  X.D. Jiang, "Image Detail-Preserving Filter for Impulsive Noise Attenuation," *IEE Proceedings: Vision, Image and Signal Processing*, Vol. 150, No. 3, pp. 179-185, June 2003.

3. Image Enhancement—Other Order-stati. Filter

- As median filter underperforms mean filter in attenuating short-tailed noise, e.g. Gaussian noise, filters that own merits of the both mean and median filters have been developed:
- Alpha-trimmed mean filter

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} f_r(s,t)$$

where $f_r(s,t)$ are the remaining $mn-d$ pixels around median.

- Iterative Truncated Arithmetic Mean Filter

X.D. Jiang, "Iterative Truncated Arithmetic Mean Filter And Its Properties," ***IEEE Transactions on Image Processing***, vol. 21, no. 4, PP. 1537-1547, April 2012.

Z. Miao and X.D. Jiang, "Further Properties and a Fast Realization of the Iterative Truncated Arithmetic Mean Filter" ***IEEE Transactions on Circuits and Systems-II***, vol. 59, no. 11, pp. 810-814, Nov 2012.

Z. Miao and X.D. Jiang, "Weighted Iterative Truncated Mean Filter," ***IEEE Transactions on Signal Processing***, Vol. 61, no. 16, pp. 4149-4160, Aug 2013.

Z. Miao and X.D. Jiang, "Additive and exclusive noise suppression by iterative trimmed and truncated mean algorithm," ***Signal Processing***, vol. 99, pp. 147-158, June 2014.

Iterative Truncated Arithmetic Mean Filter

$$\mu = \arg\min_{\varphi} \sum_{i=1}^{n} (x_i - \varphi)^2 \qquad \phi = \arg\min_{\varphi} \sum_{i=1}^{n} | x_i - \varphi |$$

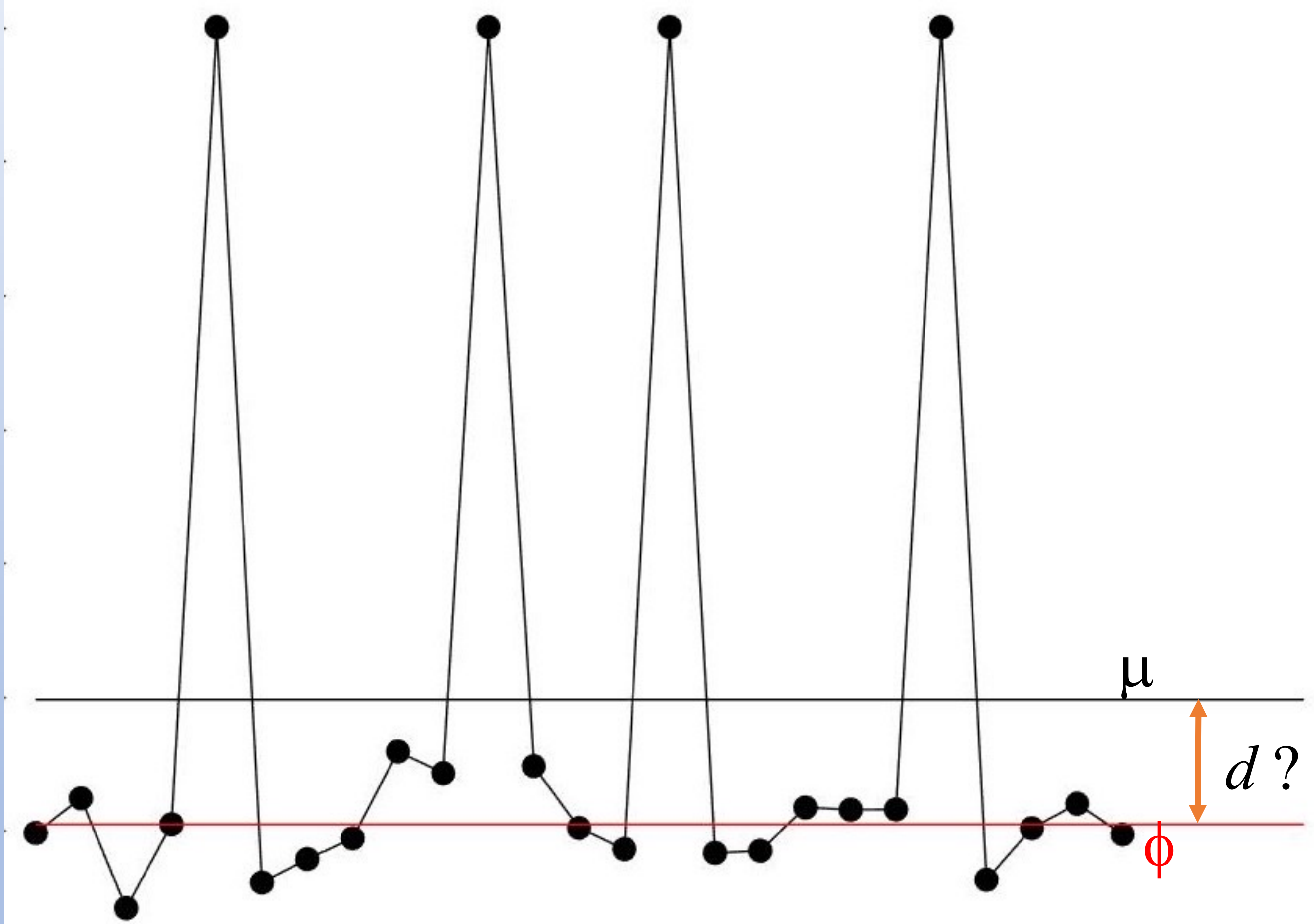
- As both mean and median have their own merits and limitations, how to find a solution between them that inherits the merits of the both operations?

- AS the computation of median is inefficient, how to use the simple arithmetic mean to approach the order statistic median?

- To achieve these, we need first explore the relation between arithmetic mean and order statistic median

Iterative Truncated Arithmetic Mean Filter

- Relation between arithmetic mean and order statistic median



$$\tau_1 = \tfrac{1}{2}(\mu_h - \mu_l)$$

$$\tau_2 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

$$\tau_3 = \frac{1}{n}\sum_{i=1}^{n}|x_i - \mu|$$

$$d = |\phi - \mu| ?$$

- For some data distribution, mean and median are close to each other while for some other data distribution, they are apart very far away.

Iterative Truncated Arithmetic Mean Filter

- **Theorem 1:** The distance between the median and the mean of any finite data set is never greater than one sample standard deviation, τ_2, never greater than the mean absolute deviation of the data from the mean, τ_3, and never greater than the half distance between the upper mean and lower mean τ_1.

$$|\phi - \mu| \leq \tau_1, \quad |\phi - \mu| \leq \tau_2, \quad |\phi - \mu| \leq \tau_3$$

- **Theorem 2:** The mean absolute deviation of the data from the mean is the tightest bound of the distance between the median and the mean of any finite data set.

$$\tau_3 \leq \tau_1, \quad \tau_3 \leq \tau_2$$

Iterative Truncated Arithmetic Mean Filter

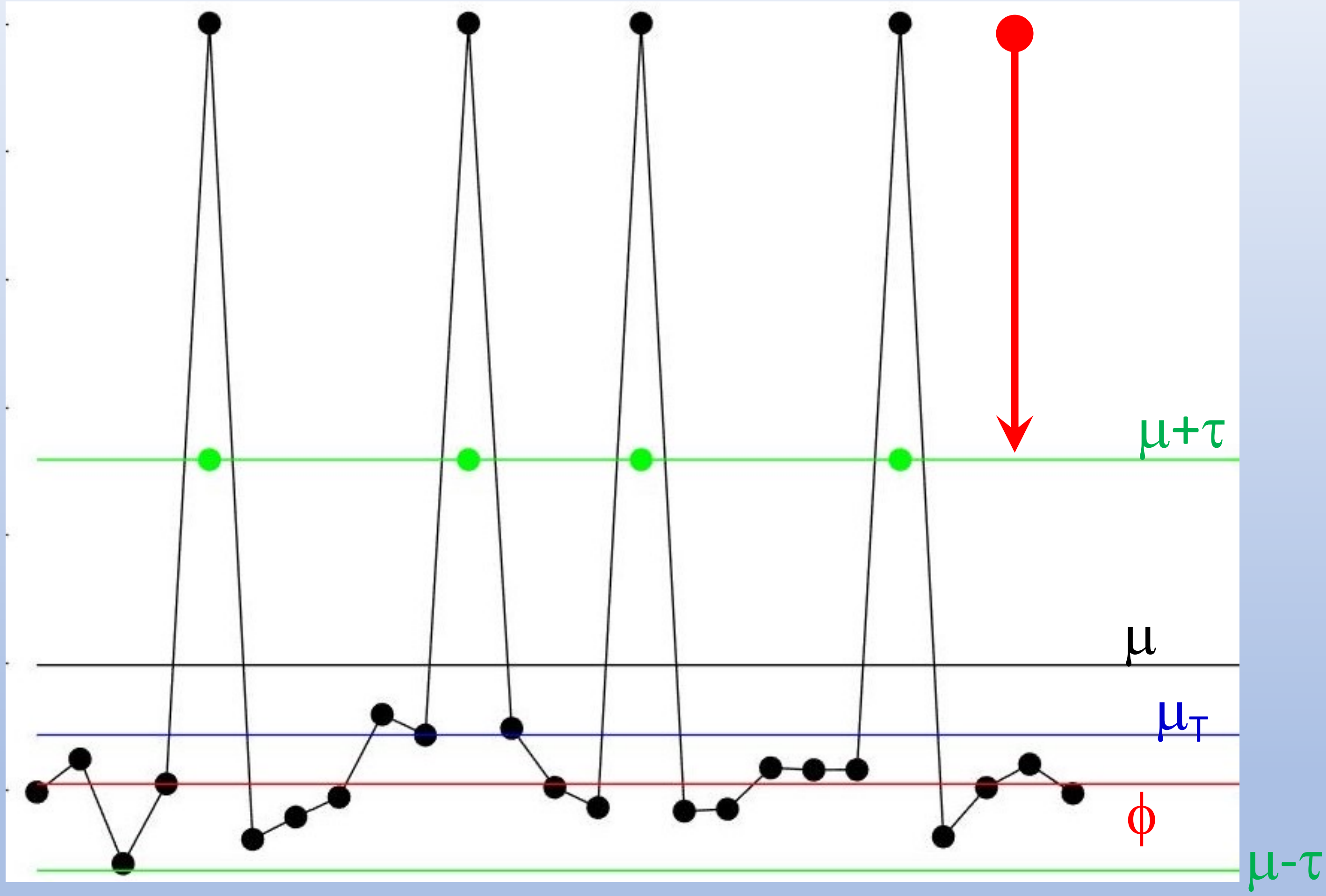
Outline of the ITM algorithm:

1) Compute the mean

2) Compute threshold and truncate input data

$$x_i = \begin{cases} \mu + \tau, & \text{if } x_i > \mu + \tau \\ \mu - \tau, & \text{if } x_i < \mu - \tau \end{cases}$$

3) Return to step 1) if stopping criterion is violated. Otherwise, terminate iteration.



Iterative Truncated Arithmetic Mean Filter

- **Theorem 3:** For any finite data set, there exists at least one sample whose distance from the sample mean is greater than the mean absolute deviation of the samples from the mean if the sample median deviates from the sample mean, i.e., letting

$$\exists x_i, x_i \in \mathbf{x}, \quad \text{that } |x_i - \mu| > \tau_2, \ \text{ if } \ \phi \neq \mu$$

- **Theorem 4:** The ITM algorithm decreases truncation threshold monotonically to zero if the mean deviates from the median.

$$\tau_2(k) < \tau_2(k-1), \qquad \lim_{k \to \infty} \tau_2(k) = 0, \ \text{ if } \ \phi \neq \mu$$

- **Theorem 5:** The truncated mean of the ITM algorithm approaches to median arbitrarily close.

Iterative Truncated Arithmetic Mean Filter

- Mean and median are two fundamental data operations that have different characteristics. It’s desirable to have merits of the both.
- Comparing with the arithmetic operation, data sorting required by computing median is a complex process and is intractable.
- This work discovers the relation between the two fundamental statistics, the arithmetic mean and the order statistical median.
- Based on this discovery, ITM filter is developed that circumvents the data-sorting process to approach the median.
- Proper termination of the proposed ITM algorithm enables the filters to own merits of the both mean and median and, hence, outperform both the filters in many image denoising applications.
- Although it is an iterative algorithm, only few iterations are required to achieve good results, It is faster than the median computation.

4 Morphological Image Processing –Outline

- Introduction
- Set Theory and Logic Operation
- Dilation and Erosion
- Opening and Closing
- Morphological Algorithm and applications

4 Morphological Image Processing –Introduction

Looking at these images……

What is interesting, important or useful information we care about?

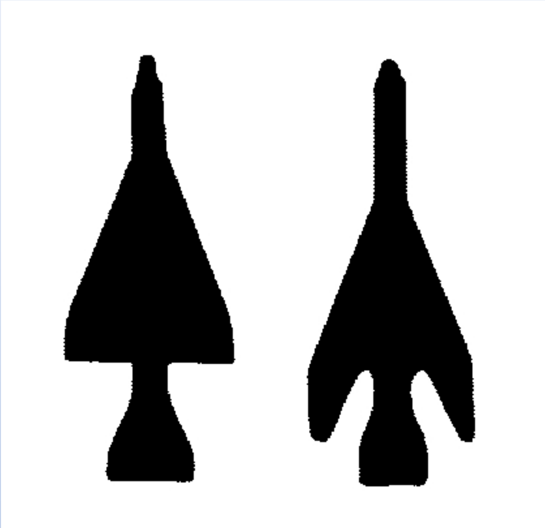
The grey value of the image is not important as there are only two different grey values.

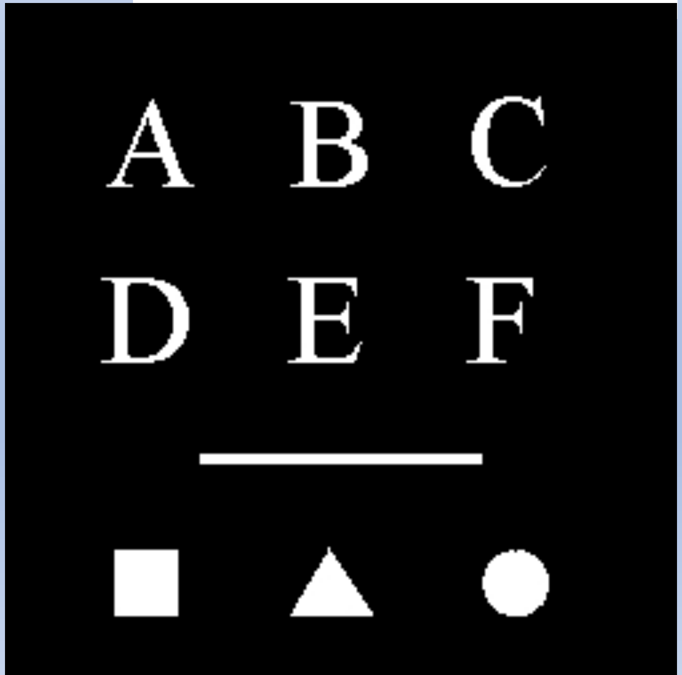
- Region shape and boundaries of object are important.

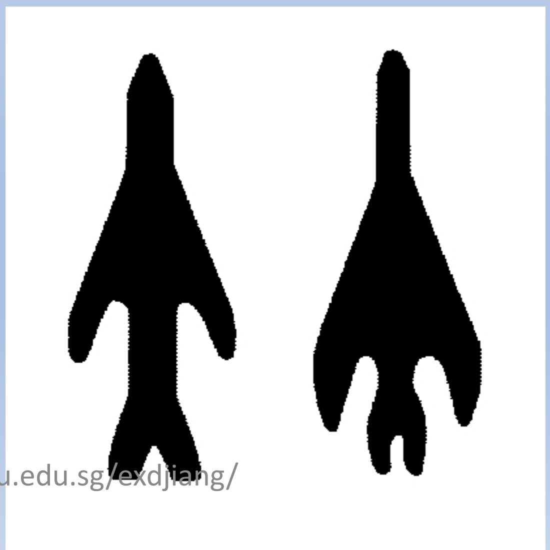
- A binary image can be represented by object pixel set.

$$A = \{a \mid a=(x,y), f(x,y)=1\}$$

$f(x,y)$







4 Morphological Image Processing –Introduction

- Morphology deals with form and structure
- Mathematical morphology is a tool for extracting image components useful in:
  - representation and description of region shape (e.g. boundaries)
  - pre- or post-processing (filtering, thinning, etc.)
- Morphological operations are powerful tools in image analysis. They usually operate on binary images and thus often follow a segmentation task or an edge detection task.
- Based on set theory and logic operations

4 Morphology –Dilation

- Dilation of A by B, denoted by $A \oplus B$, is defined as:

$$A \oplus B = \left\{ z \mid \left[\left(\hat{B} \right)_z \cap A \right] \neq \varnothing \right\}$$

- Interpretation:
Obtaining the reflection of B about its origin and then shifting this reflection by z. Dilation of A by B then is the set of all z displacements such that the shifted $\hat{B}$ and A overlap by at least one nonzero element.

- B is called the structuring element in Dilation.

4 Morphology –Dilation

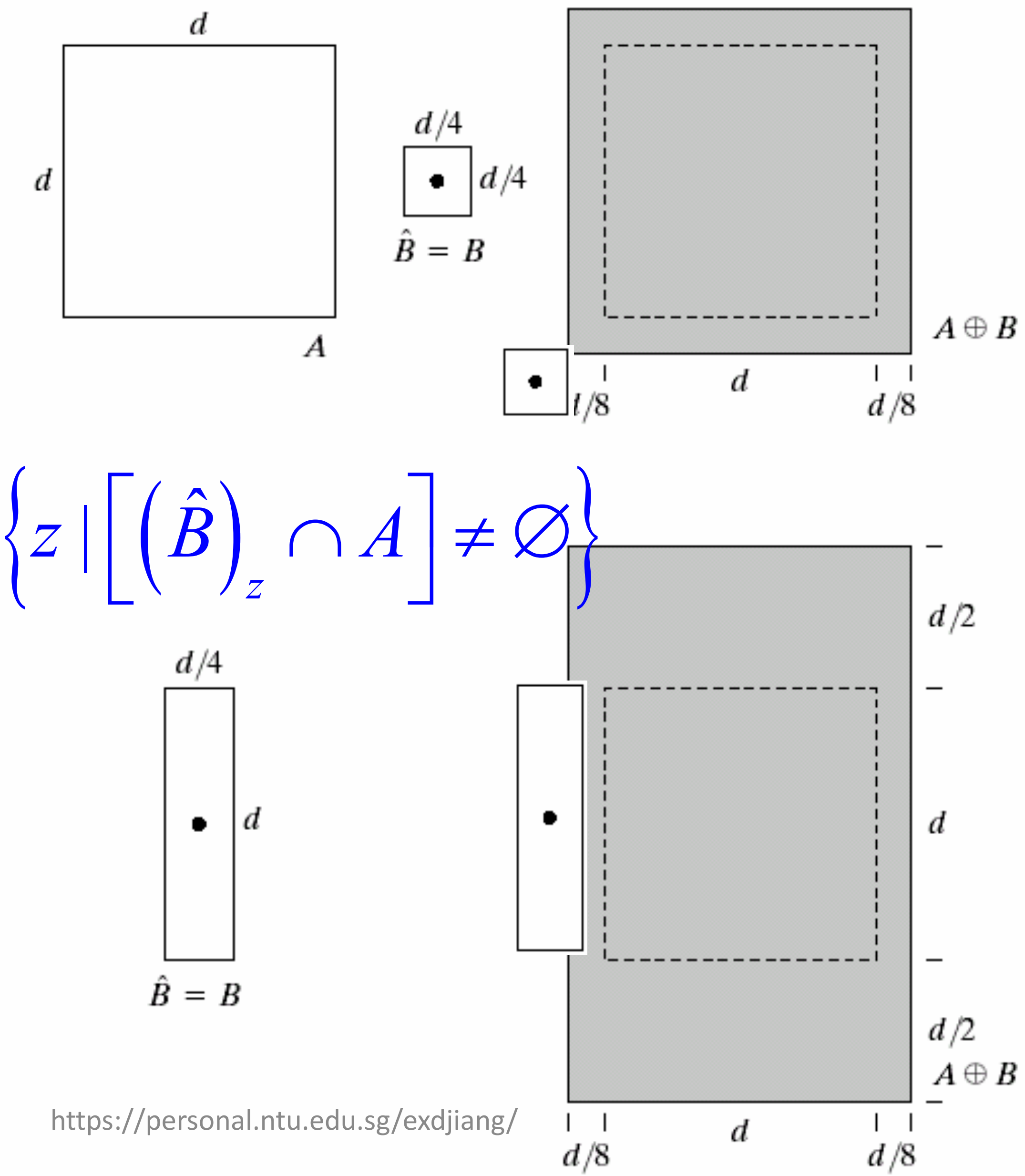
- Dilation of A by B can also be expressed as:

$$A \oplus B = \left\{ z \mid \left[\left(\hat{B} \right)_z \cap A \right] \subseteq A \right\}$$

- Further Interpretation:
  Set B can be viewed as a convolution mask. The basic process of "flipping" B and then successively displace it so that it slides over set (image) A is analogous to the convolution.

4 Morphology –Dilation

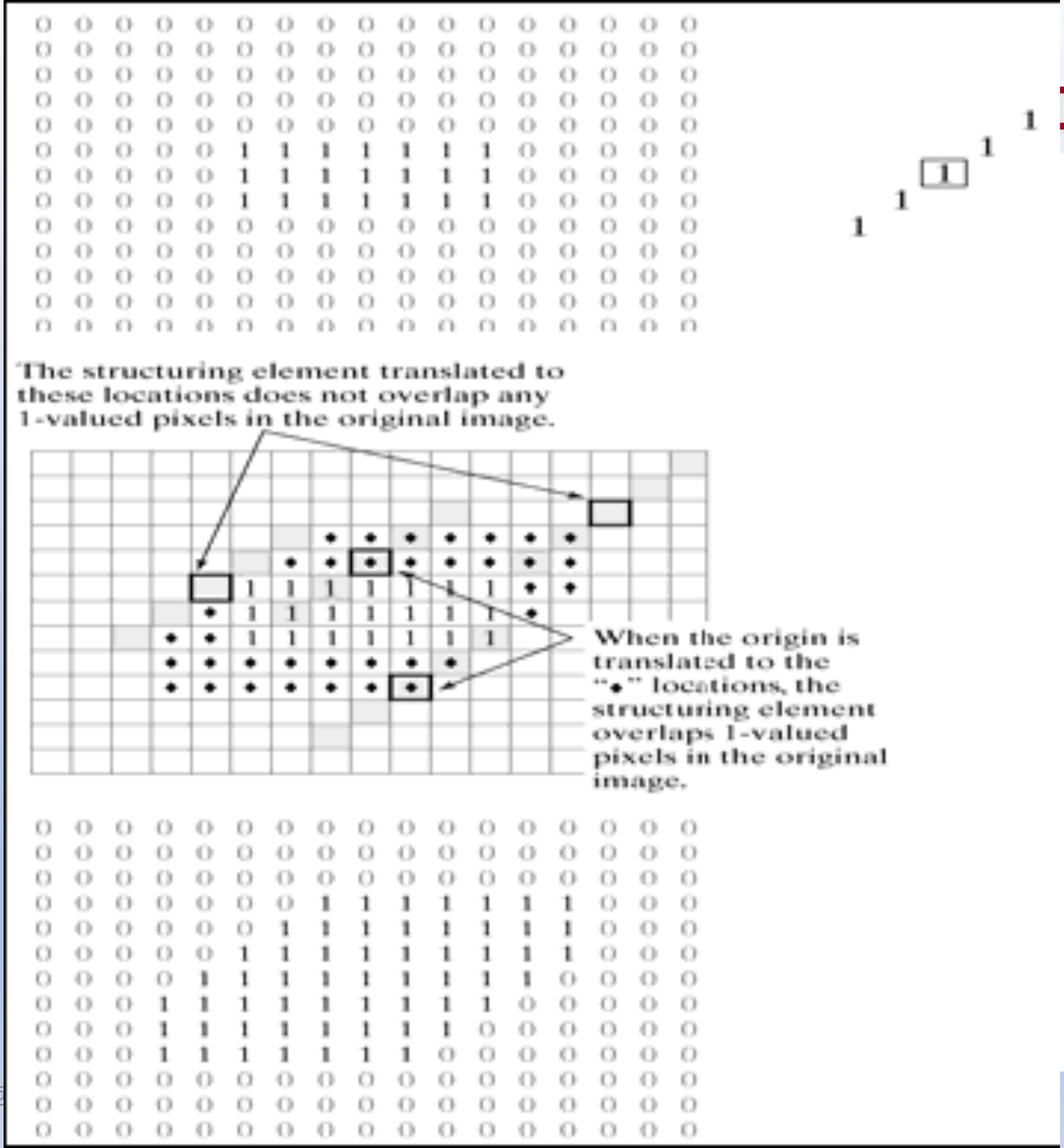
$$A \oplus B = \left\{ z \mid \left[\left(\hat{B} \right)_z \cap A \right] \neq \varnothing \right\}$$



a b c
d e

(a) Set A. (b) Square structuring element (dot is the center). (c) Dilation of A by B, shown shaded. (d) Elongated structuring element. (e) Dilation of A using this element.

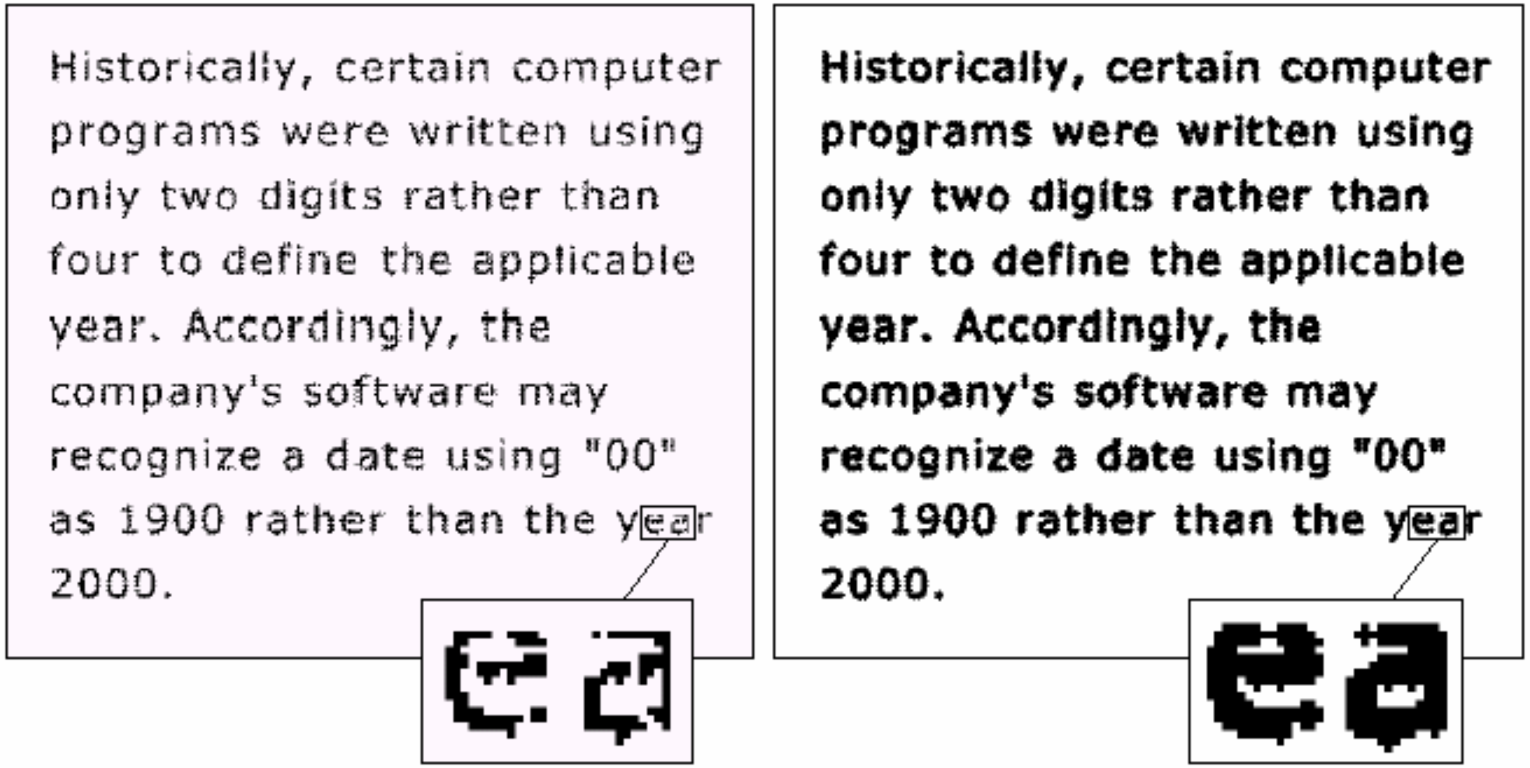
Dilation



a b
c
d

Illustration of dilation. (a) Original image with rectangular object. (b) Structuring element with five pixels arranged in a diagonal line. The origin of the structuring element is shown with a dark border. (c) Structuring element translated to several locations on the image. (d) Output image.

4 Morphology –Dilation



0	1	0
1	1	1
0	1	0

a c
b

(a) Sample text of poor resolution with broken characters (magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

4 Morphology –Erosion

- Erosion of A by B, denoted $A \ominus B$, is defined as:

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

- Erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

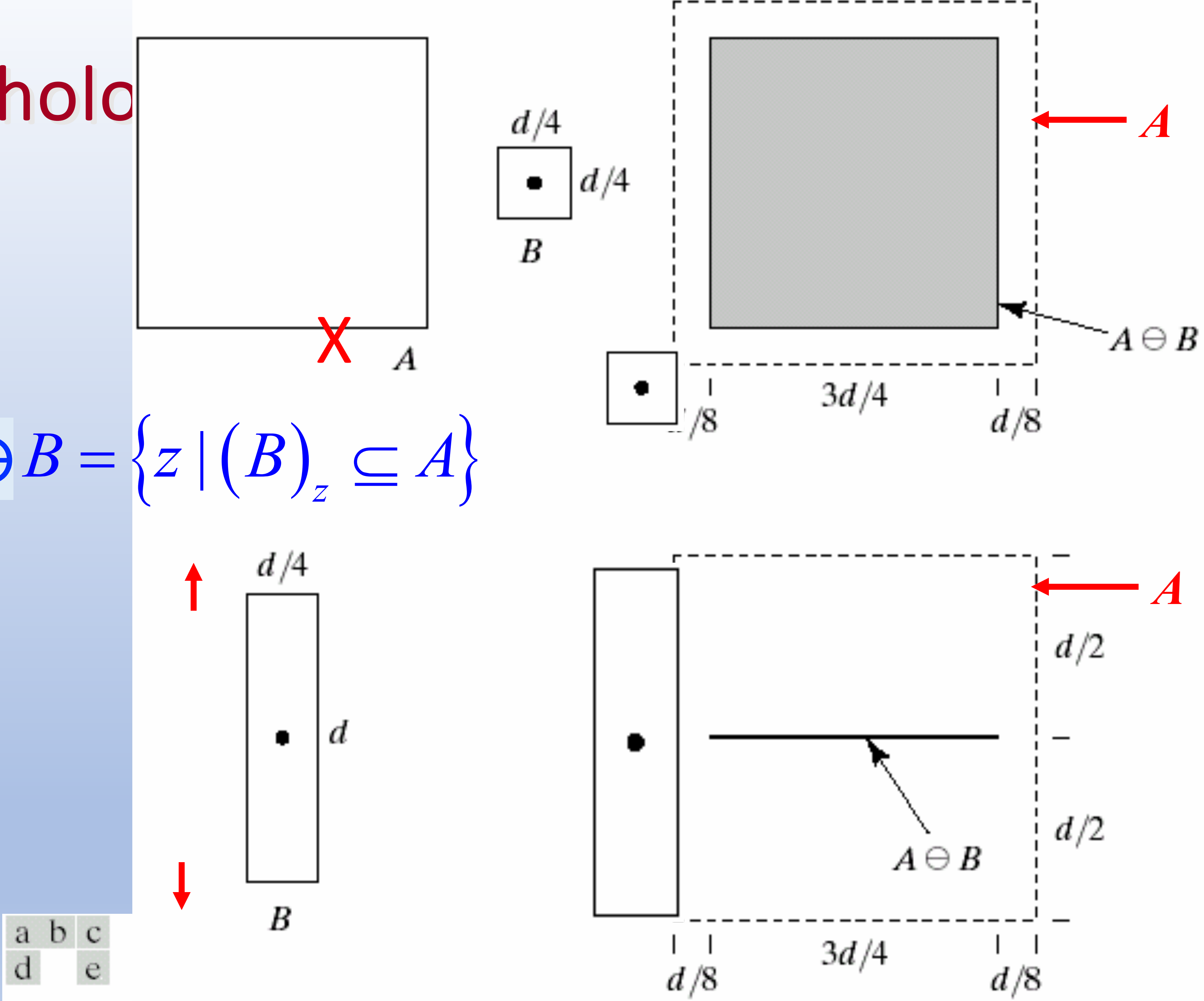
- Comparing with the Dilation:

$$A \oplus B = \left\{z \mid \left[\left(\hat{B}\right)_z \cap A\right] \subseteq A\right\}$$

- Dilation and erosion are duals of each other with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

4 Morpholo



$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

(a) Set A. (b) Square structuring element. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A using this element.

4 Morphology –Erosion

(a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

4 Morphology –Opening

- Compound operations – Opening

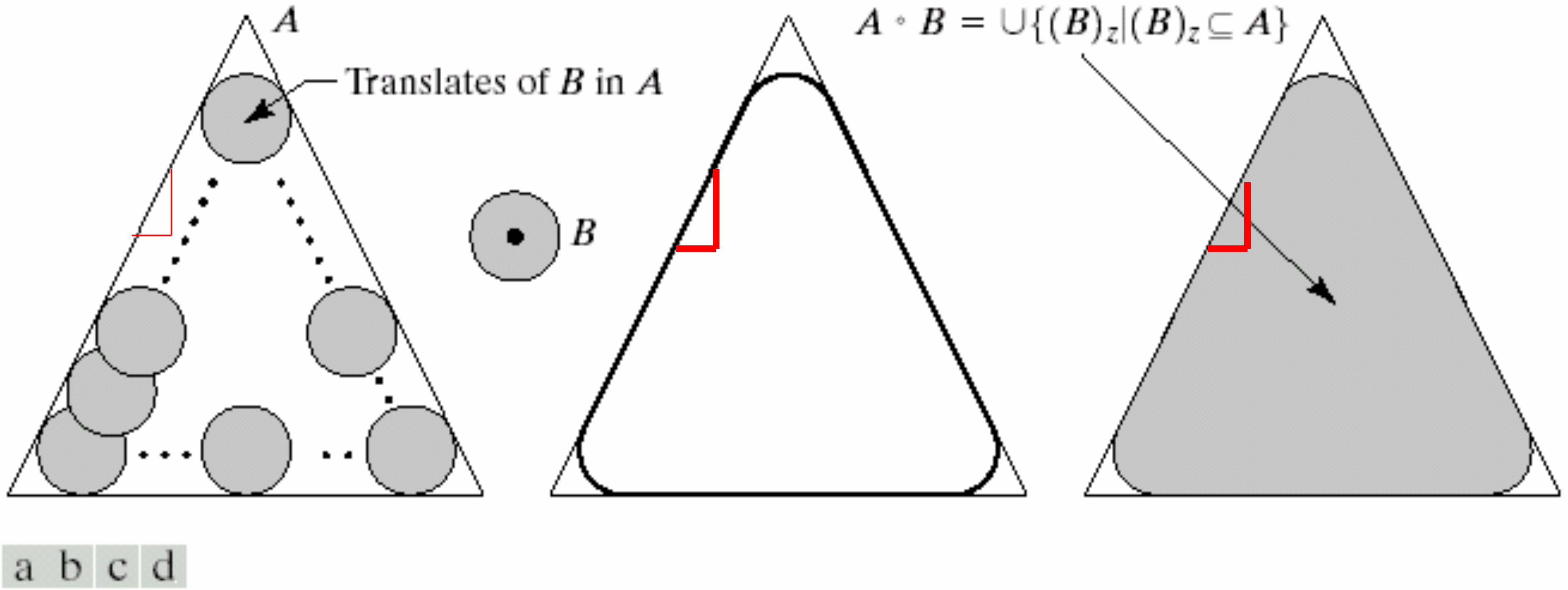
- A compound operation is when two or more morphological operations are performed in succession. A common example is opening which is an erosion followed by a dilation:

$$A \circ B = (A \ominus B) \oplus B$$

➢ The opening A by B is obtained by taking the union of all translates of B that fit into A. This can be expressed as a fitting processing such that:

$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\} \qquad A \ominus B = \{z \mid (B)_z \subseteq A\}$$

4 Morphology –Opening



(a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

- Note that the outward pointing corners are rounded, where the inward pointing corners remain unchanged.

4 Morphology –Opening

$$A \circ B = (A \ominus B) \oplus B \quad A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

- Opening is often performed to clear an image of noise whilst retaining the original object size. Care must be taken that the operation does not distort the shape size of the object if this is significant.
- The opening operation tends to flatten the sharp peninsular projections on the object.
- A useful way to see the effects of an opening operation is to look for differences between the original image and the image after opening by projecting these differences onto the original image.

4 Morphology –Closing

- Compound operations – Closing

- Closing is the complementary operation of opening, defined as dilation followed by erosion.

$$A \bullet B = \left(A \oplus B \right) \ominus B$$

> ➢ Opening and closing are duals of each other as:

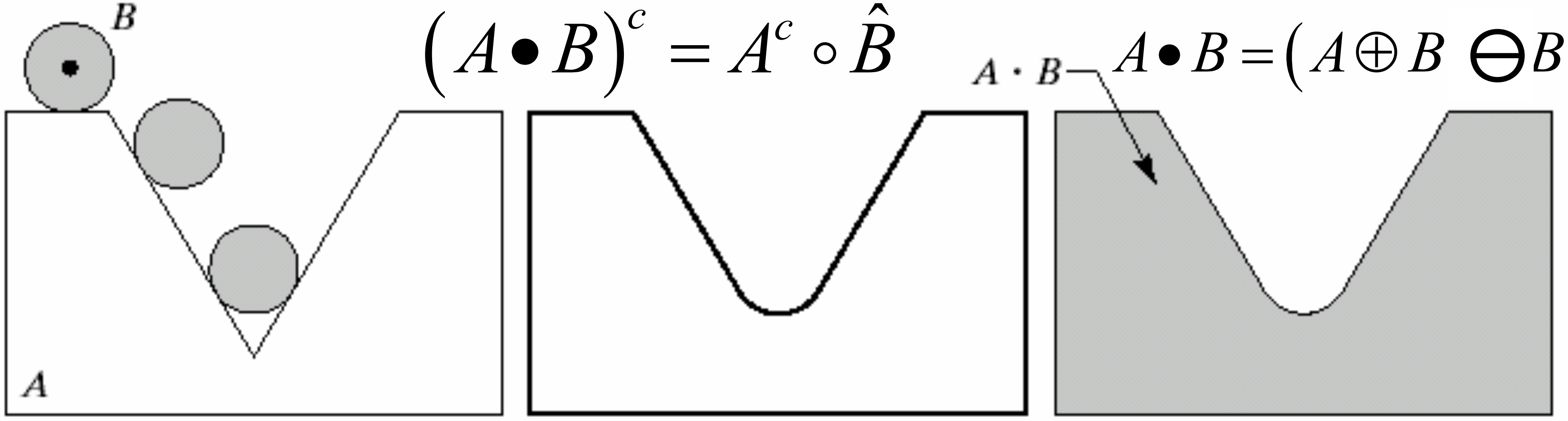
$$\left(A \bullet B \right)^c = A^c \circ \hat{B}$$

$$\text{Or:} \quad A \bullet B = (A^c \circ \hat{B})^c$$

4 Morphology –Closing

$$(A \bullet B)^c = A^c \circ \hat{B}$$

$$A \bullet B = (A \oplus B) \ominus B$$



a b c

(a) Structuring element B "rolling" on the outer boundary of set A. (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

- Note that the inward pointing corners are rounded, where the outward pointing corners remain unchanged.

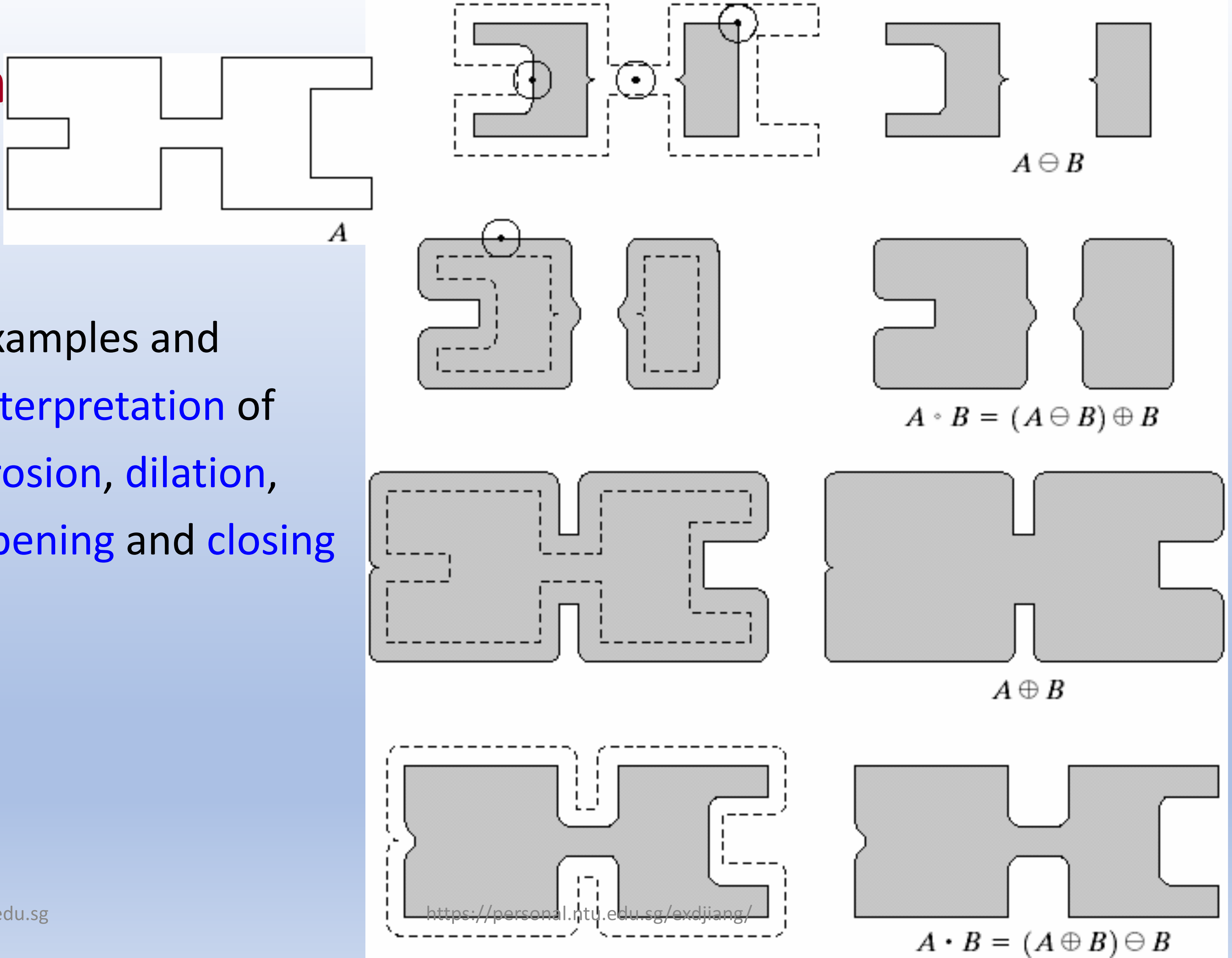
4 Morphology –Closing

- The classic application of closing is to fill holes in a region whilst retaining the original object size.

- Dilation fills the holes and erosion restores the original region size.

- In addition to filling holes the closing operation tends to fill the 'bays' on the edge of a region.

4 Morph



Examples and Interpretation of erosion, dilation, opening and closing

4 Morphology –Opening and Closing

- The opening operation satisfies the following properties:
  - $A \circ B$ is a subset (subimage) of A
  - If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$
  - $(A \circ B) \circ B = A \circ B$

➢ Similarly, the closing operation satisfies the following properties:

A is a subset (subimage) of $A \bullet B$

If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$

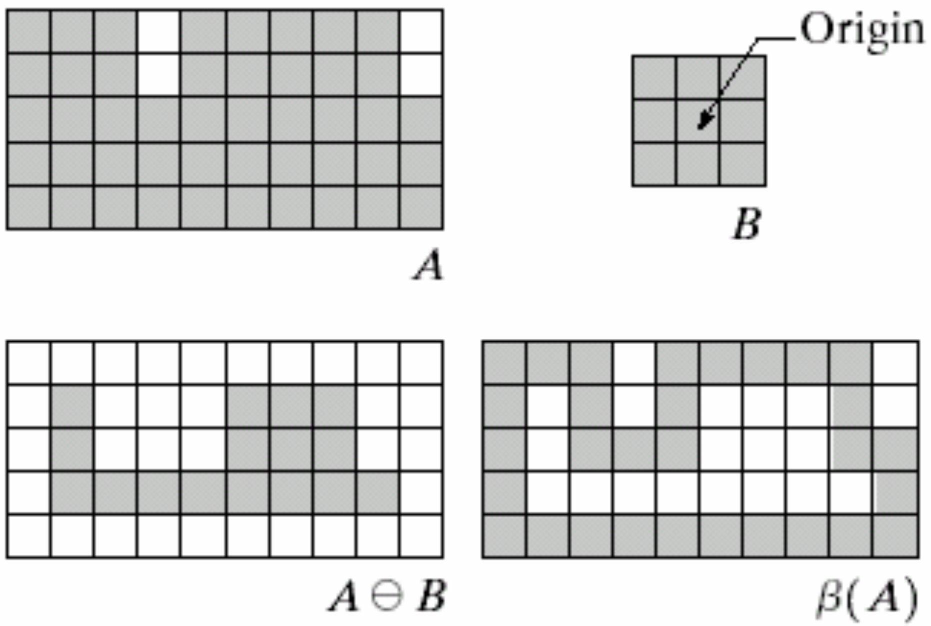
$$(A \bullet B) \bullet B = A \bullet B$$

4 Morphology –Algorithms and Applications

- **Boundary Extraction**:

  The boundary of a set A, denoted by $\beta(A)$, can be obtained by:

$$\beta(A) = A - (A \ominus B)$$



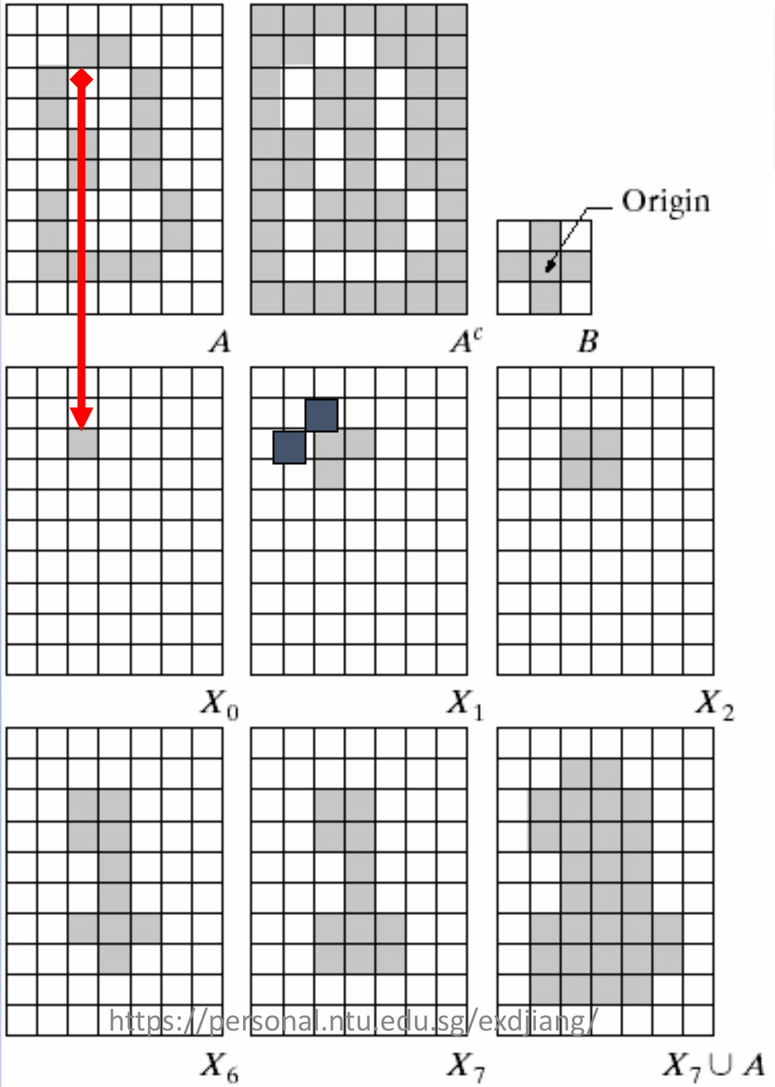
(a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.

4 Morphology –Algorithms and Applications

- Region Filling: $X_k = (X_{k-1} \oplus B) \cap A^c, \quad k = 1, 2, 3, \ldots\ldots$

$$A^F = X_k \cup A$$

Beginning with a point X_0 inside the boundary, the entire region inside the boundary is filled by the above procedure.

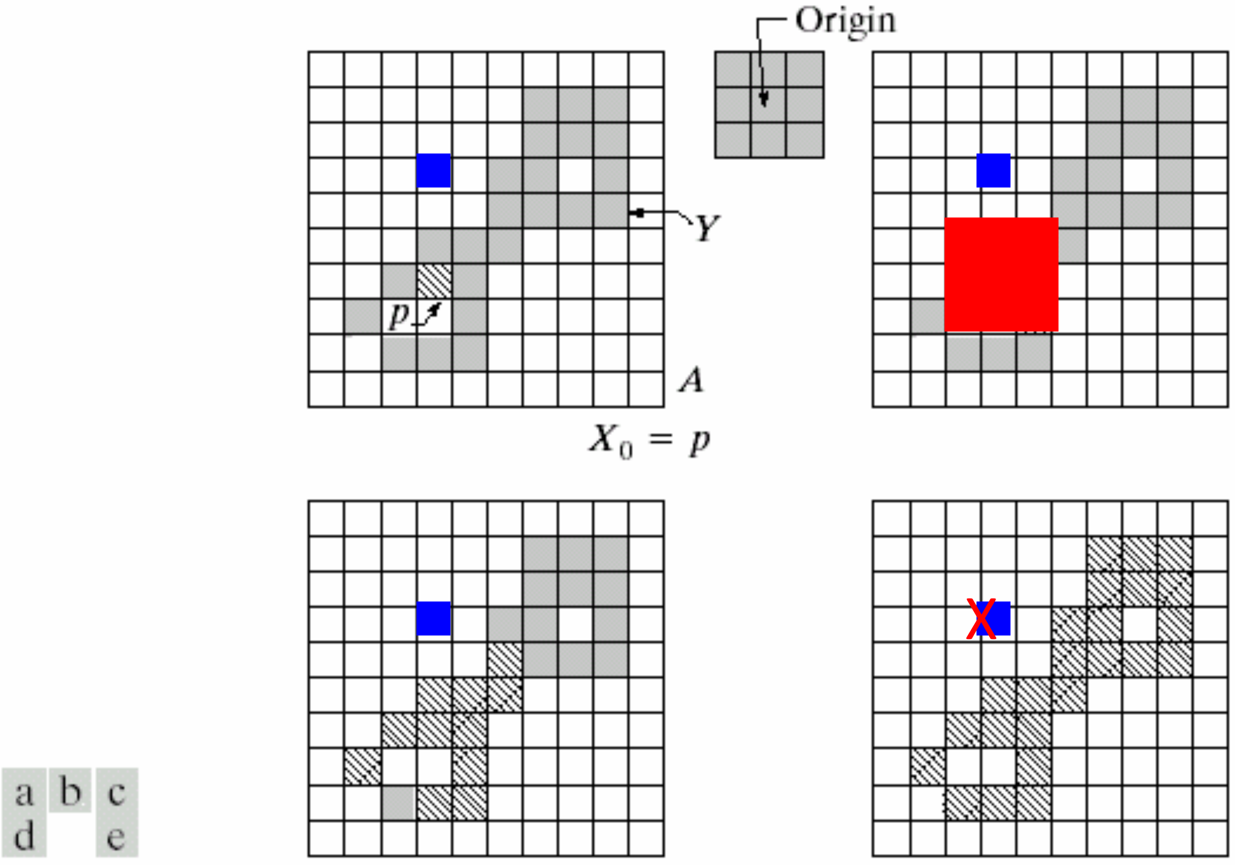


Region filling. (a) Set *A*. (b) Complement of *A*. (c) Structuring element *B*. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

4 Morphology –Algorithms and Applications

- Extract connected components:

$$X_k = \left(X_{k-1} \oplus B\right) \cap A,\ k = 1,2,3,\ldots$$

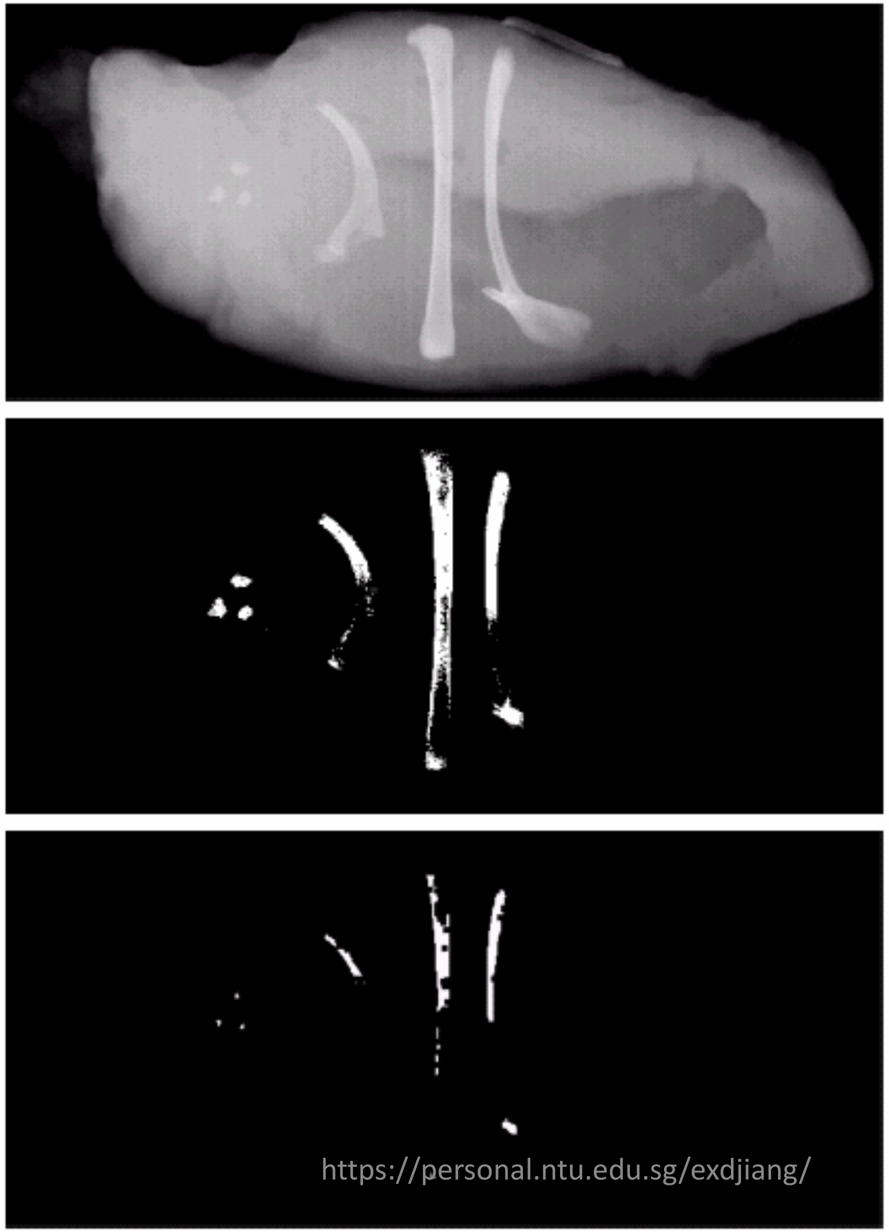


(a) Set A showing initial point p (all shaded points are valued 1, but are shown different from p to indicate that they have not yet been found by the algorithm). (b) Structuring element. (c) Result of first iterative step. (d) Result of second step. (e) Final result.

4 Morphology –Algorithms and Applications

a
b
c d

(a) X-ray image of chicken filet with bone fragments. (b) Thresholded image. (c) Image eroded with a 5×5 structuring element of 1's. (d) Number of pixels in the connected components of (c). (Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, www.ntbxray.com.)



Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

4 Morphology –Algorithms and Applications

- Denoising:

$$\left(A \circ B\right) \bullet B$$

- Or

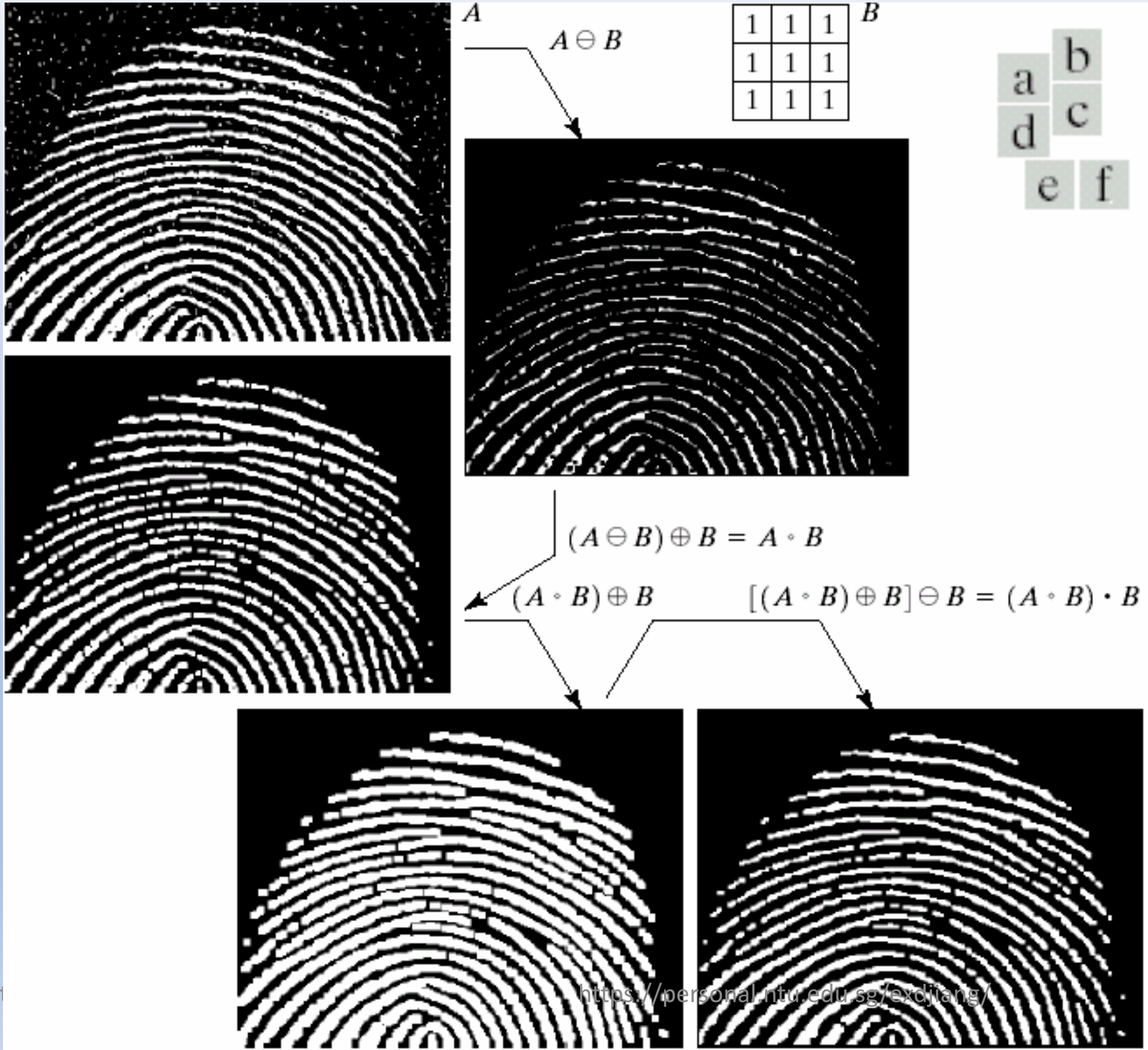
$$\left(A \bullet B\right) \circ B$$

Can be used to eliminate noise and its effect on the object.

➢ Noise pixels outside the object area are removed by opening with B while noise pixels inside the object area are removed by closing with B.

See example in the next slide

4 Morphology –Algorithms and Applications



(a) Noisy image.
(c) Eroded image.
(d) Opening of A.
(d) Dilation of the opening.
(e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

4 Morphology –Summary

Operation	Equation	Comments (The Roman numerals refer to the structuring elements shown in Fig. 9.26).
Translation	$(A)_z = \{w \mid w = a + z, \quad \text{for } a \in A\}$	Translates the origin of A to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \quad \text{for } b \in B\}$	Reflects all elements of B about the origin of this set.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points that belong to A but not to B.
Dilation	$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$	"Expands" the boundary of A. (I)
Erosion	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	"Contracts" the boundary of A. (I)

4 Morphology –Summary

Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)
Closing	$A \bullet B = (A \oplus B) \ominus B$	Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)
Boundary extraction	$\beta(A) = A - (A \ominus B)$	Set of points on the boundary of set A. (I)
Region filling	$X_k = (X_{k-1} \oplus B) \cap A^c; X_0 = p$ and $k = 1, 2, 3, \ldots$	Fills a region in A, given a point p in the region. (II)
Connected components	$X_k = (X_{k-1} \oplus B) \cap A; X_0 = p$ and $k = 1, 2, 3, \ldots$	Finds a connected component Y in A, given a point p in Y. (I)