Mathematical Morphology

Jesus J. Caban

Outline

- Assignment #1
- Reading list
- Mathematical Morphology
- Image Restoration

Mathematical Morphology

- Morphological image processing (or *morphology*) describes a range of image processing techniques that deal with the shape (or morphology) of features in an image
- Often used to design tools/methods for extracting image components
- Morphological operations can be used to
  - remove imperfections in the image masks
  - provide information on the form and structure of the image
- Note: most of the images in this lecture are binary images

Mathematical Morphology: Example

Original Image

Image after morphological processing

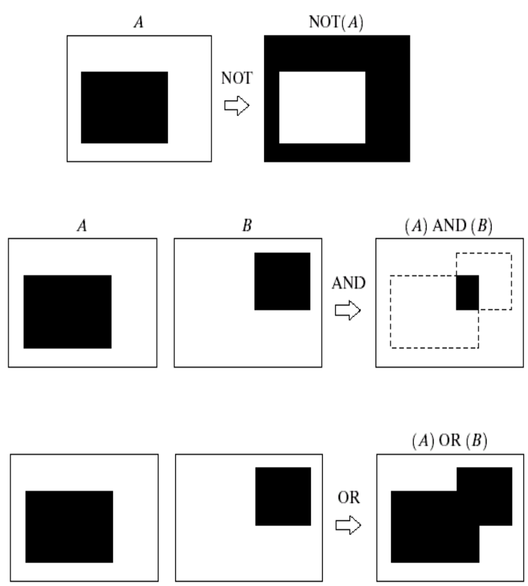
Set Theory

- A is a subset of B: every element of A is an element of another set B

$$A \subseteq B$$

- Union $C = A \cup B$
- Intersection $C = A \cap B$
- Mutually Exclusive $A \cap B = \varnothing$

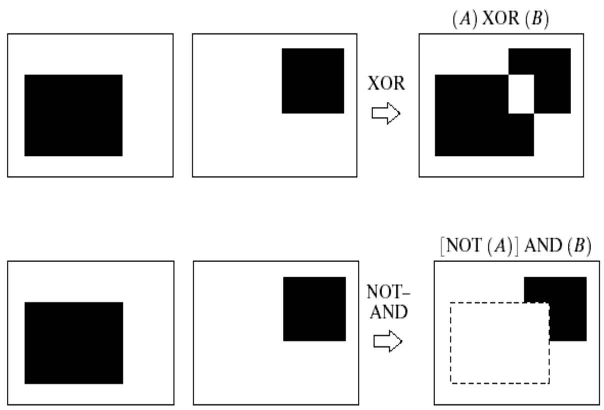
Logical Operations



$A \cap B$

$A \cup B$

Logical Operations

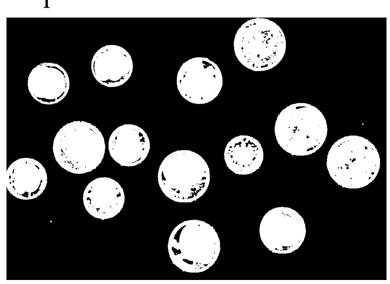


$B - A$

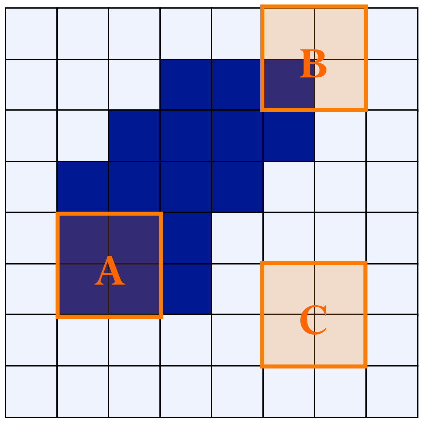
Erosion and Dilation

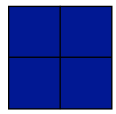
- There are two primary morphological operations:
  1. Erosion
  2. Dilation

- The combination of these two operations forms other morphological operations.



Structuring Elements, Hits & Fits





Structuring Element

Fit: All *on pixels* in the structuring element cover *on pixels* in the image

Hit: Any *on pixel* in the structuring element covers an *on pixel* in the image

All morphological processing operations are based on these simple ideas

1. Erosion

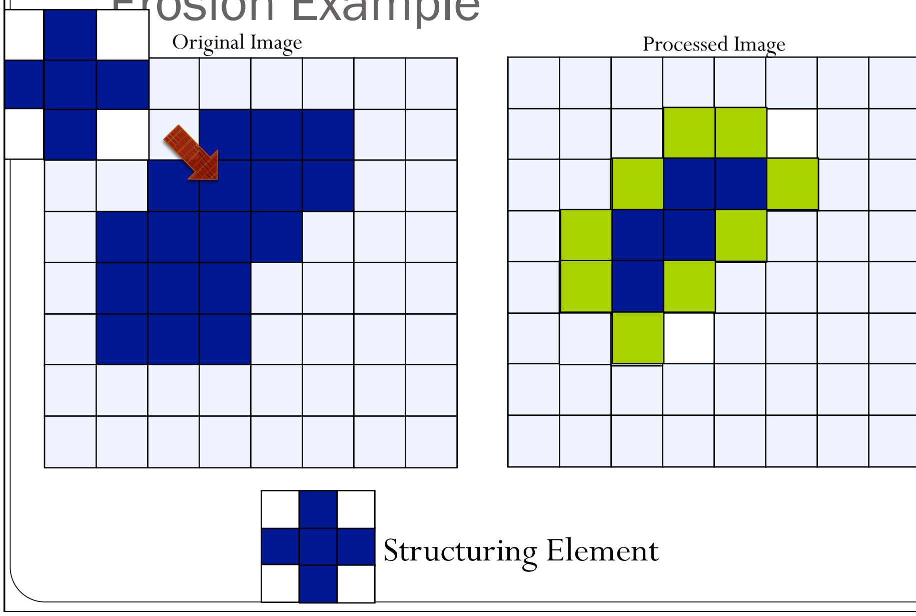
$$A \ominus B = \{ \ z \mid (B)_z \subseteq A \}$$

- **Definition:** The erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

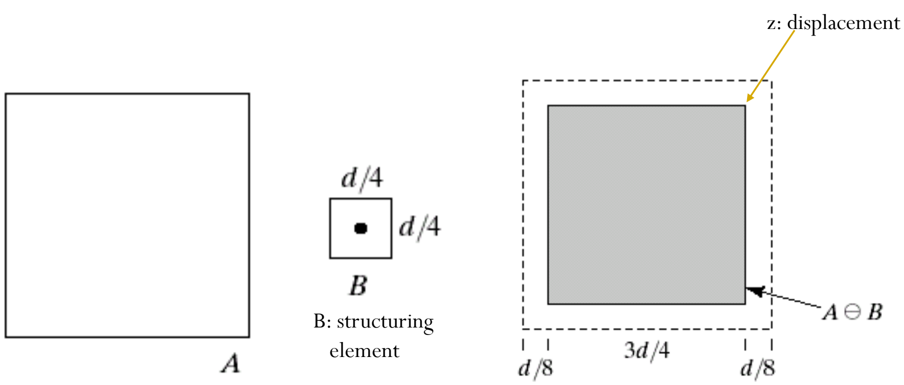
- The structuring element s is positioned with its origin at *(x, y)* and the new pixel value is determined using the rule:

$$g(x,y) = \begin{cases} 1 \text{ if } s \text{ fits } f \\ 0 \text{ otherwise} \end{cases}$$

Erosion Example



Erosion



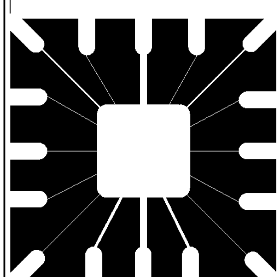
Erosion Example 1

Original image

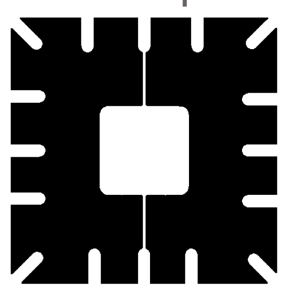
Erosion by 3*3 square structuring element

Erosion by 5*5 square structuring element

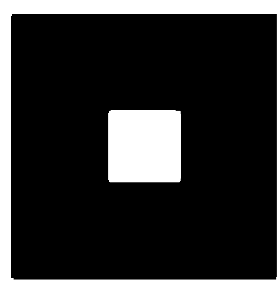
Erosion Example 2



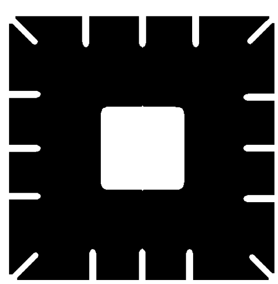
Original image



After erosion with a disc of radius 5



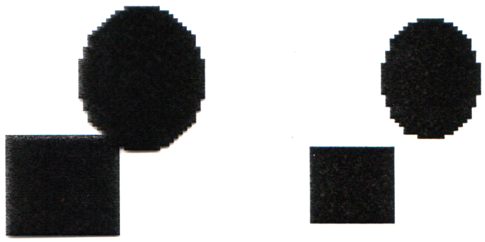
After erosion with a disc of radius 20



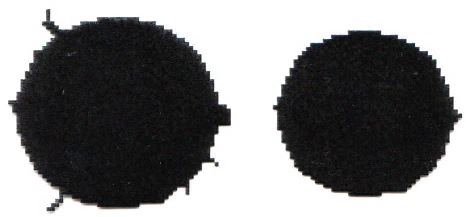
After erosion with a disc of radius 10

Erosion Example 3

Erosion can split apart joined objects and remove artifacts

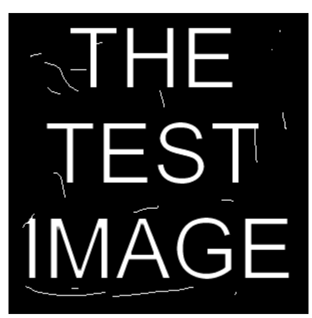


Erosion can split apart



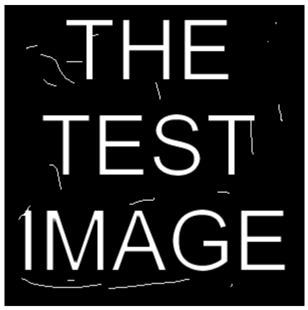
Erosion: Example 4

Original image



Eroded image

Erosion



Eroded once

Eroded twice

Dilation

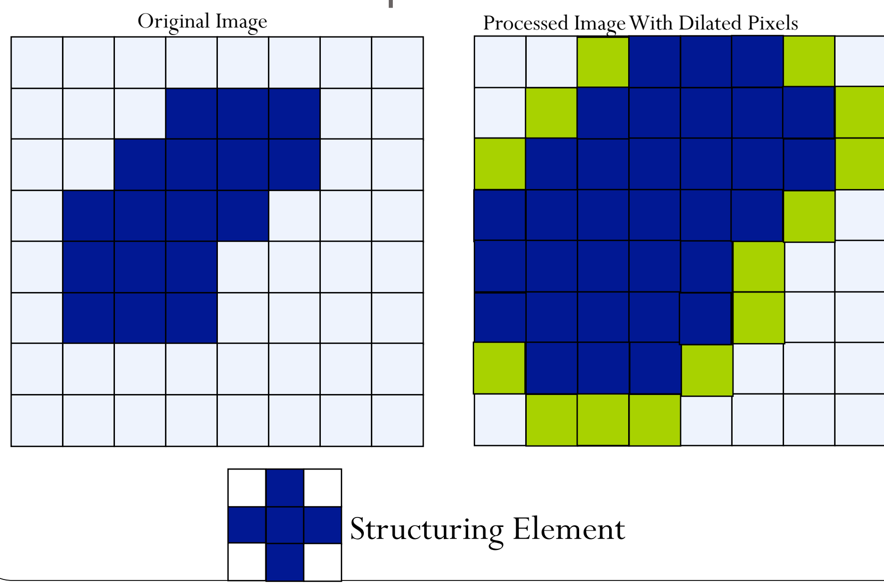
$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \varnothing \right\}$$

- **Definition:** The dilation of A by B is the set of all displacements z such that B' and A overlap by at least one element.

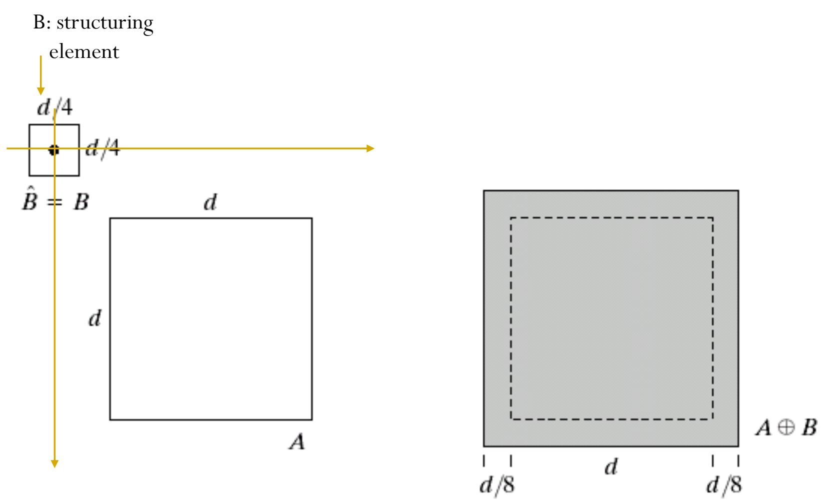
- The structuring element s is positioned with its origin at *(x, y)* and the new pixel value is determined using the rule:

$$g(x,y) = \begin{cases} 1 \text{ if } s \text{ hits } f \\ 0 \text{ otherwise} \end{cases}$$

Dilation Example



Dilation



Dilation Example 1

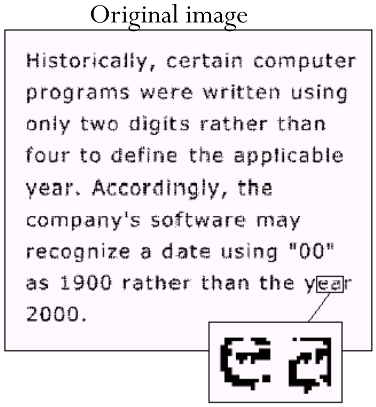
Original image

Dilation by 3*3 square structuring element

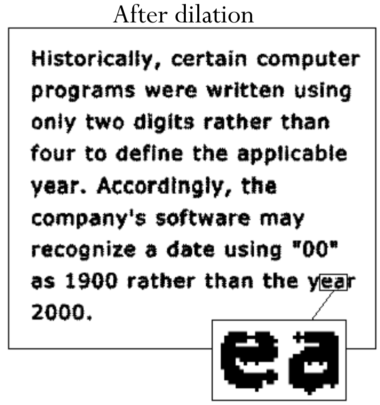
Dilation by 5*5 square structuring element

Dilation Example 2

Original image

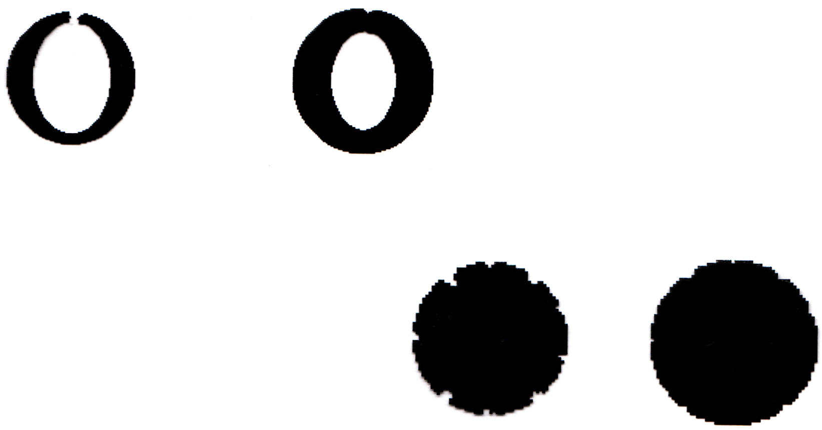


After dilation



Dilation Example 3

Dilation can repair breaks or intrusions



Useful for?

- Erosion
  - removal of structures of certain shape and size, given by element
- Dilation
  - filling of holes of certain shape and size, given by element

Combining erosion and dilation

- What about if we want:
  - remove structures / fill holes
  - without affecting remaining parts

- SOLUTION:
  - combine erosion and dilation
  - using same element

Erosion : eliminating irrelevant detail

Compound Operations

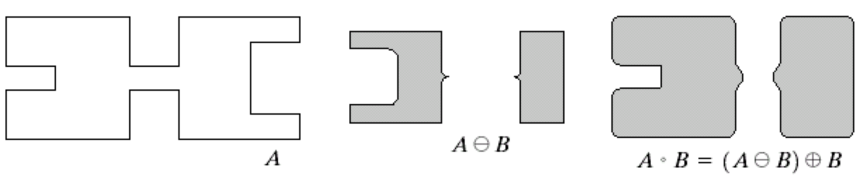
- More interesting morphological operations can be performed by performing combinations of erosions and dilations
- The most widely used of these *compound operations* are:
  - Opening
  - Closing

Opening

Opening : smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions

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

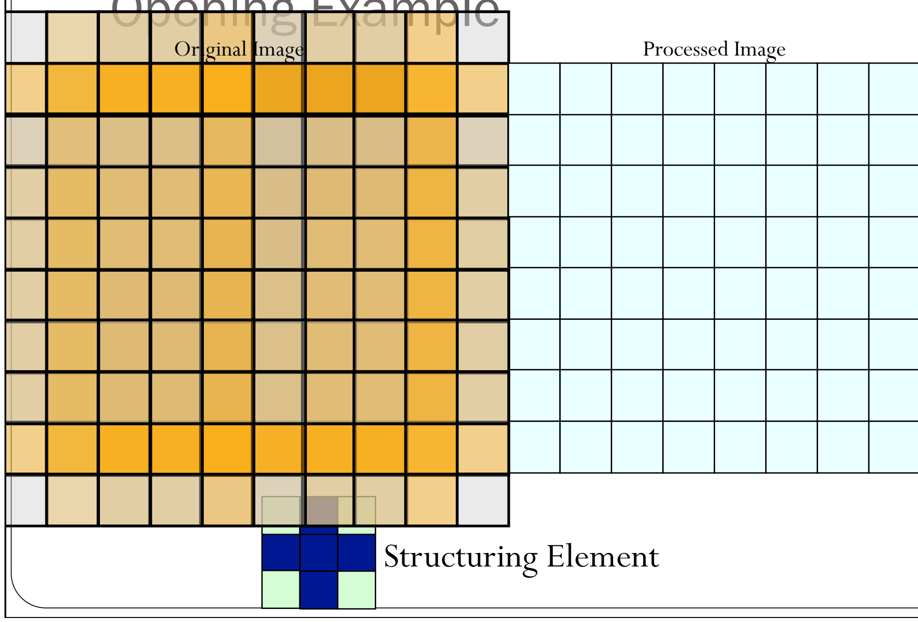
Simply an erosion followed by a dilation



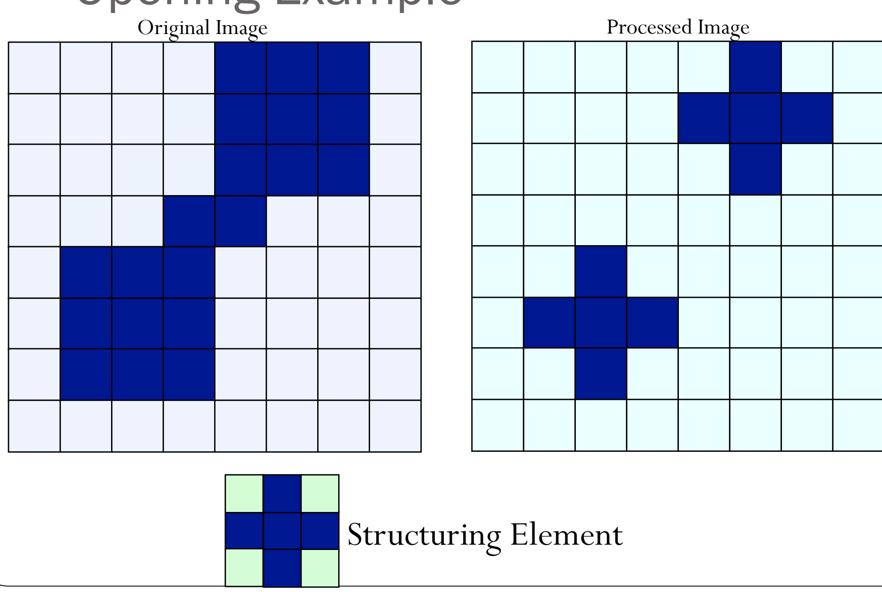
Original shape | After erosion | After dilation (opening)

Note a disc shaped structuring element is used

Opening Example



Opening Example

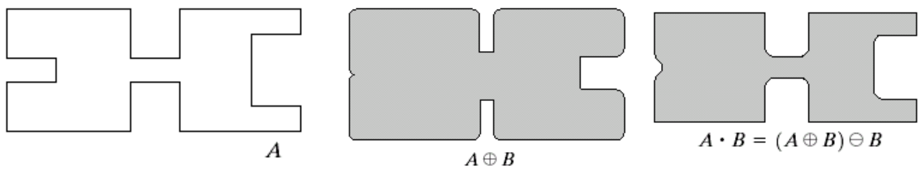


Closing

Closing : smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour

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

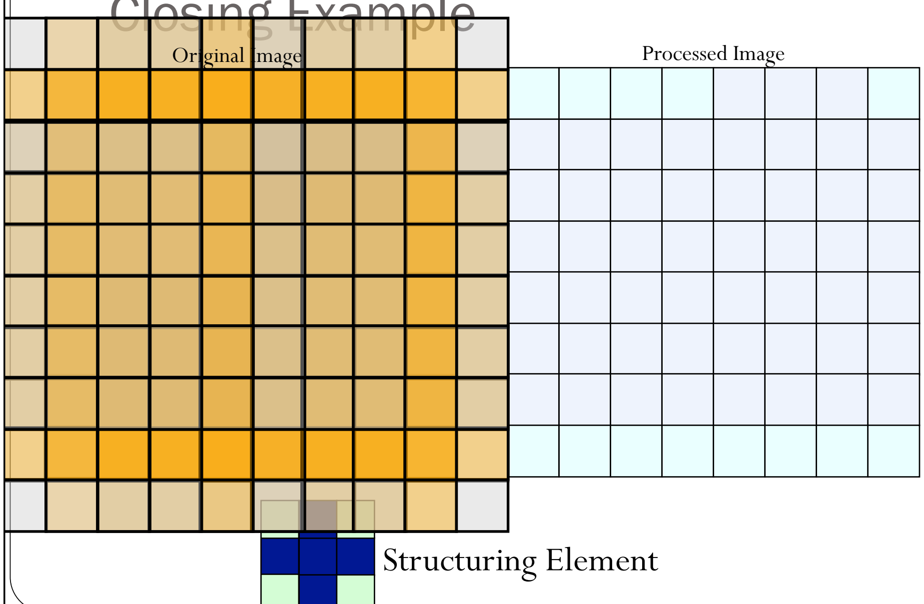
Simply a dilation followed by an erosion



Original shape — After dilation — After erosion (closing)

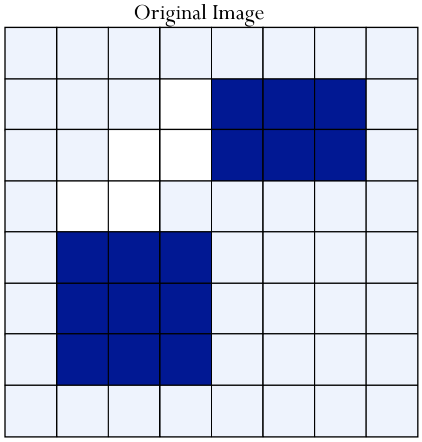
Note a disc shaped structuring element is used

Closing Example

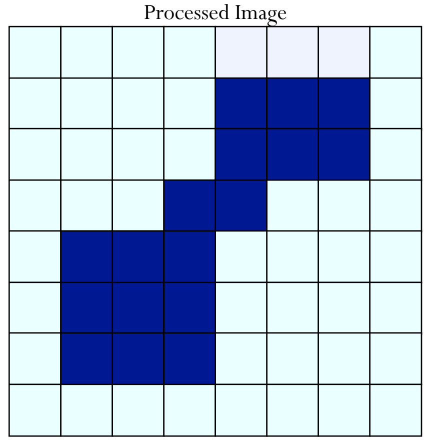


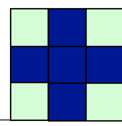
Closing Example

Original Image



Processed Image





Structuring Element

Closing Example

Original Image

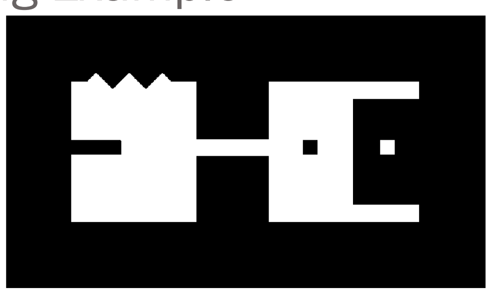
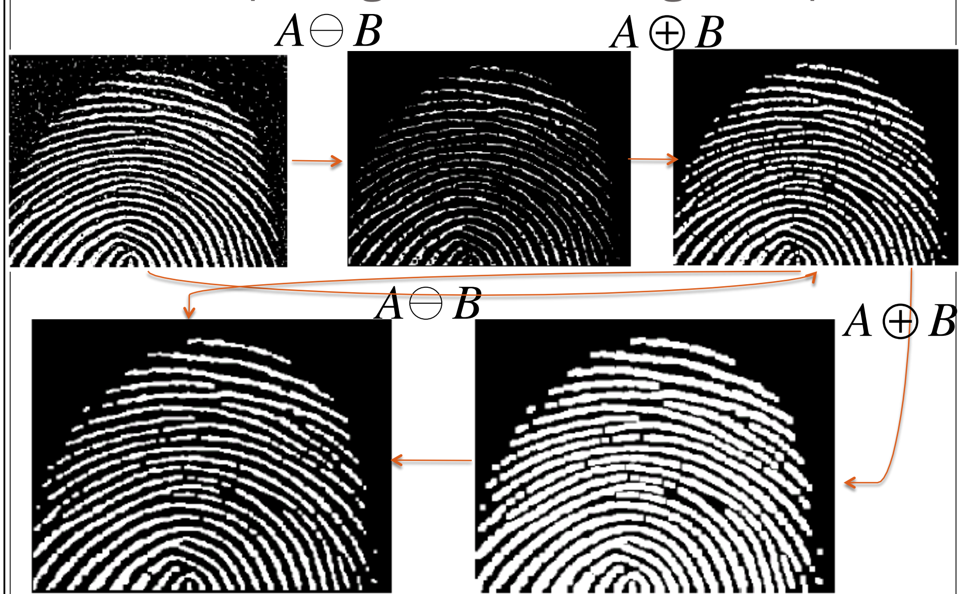
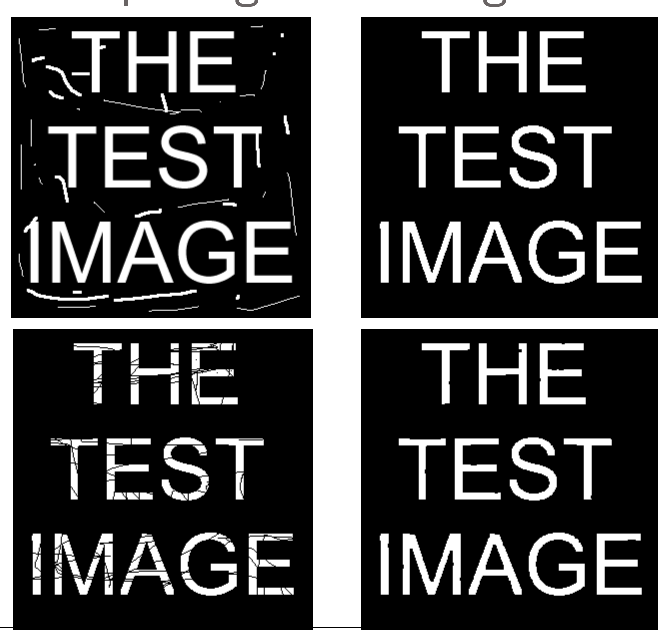


Image After Closing

Morphological Processing Example



Opening and Closing



OPENING: The original image eroded twice and dilated twice (opened). Most noise is removed

CLOSING: The original image dilated and then eroded. Most holes are filled.

Uses of opening & closing

A

opening of A
→removal of small protrusions, thin connections, …

closing of A
→ removal of holes

Morphological Algorithms

- Using the simple technique we have looked at so far we can begin to consider some more interesting morphological algorithms

- We will look at:
  - Boundary extraction
  - Region filling

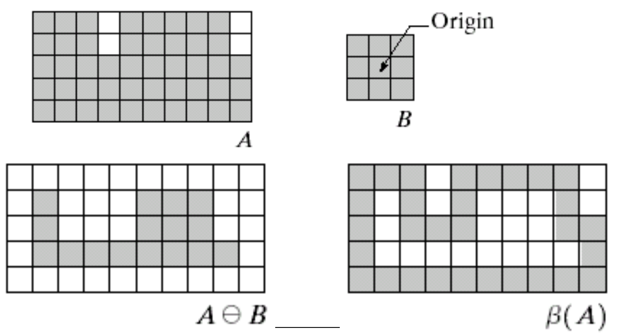
- There are lots of others as well though:
  - Extraction of connected components
  - Thinning/thickening
  - Skeleton

Boundary Extraction

Extracting the boundary (or outline) of an object is often extremely useful

The boundary can be given simply as

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



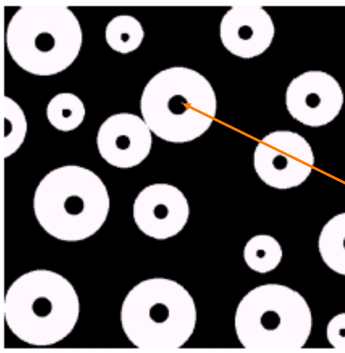
Boundary Extraction Example

A simple image and the result of performing boundary extraction using a square 3*3 structuring element

Original Image   Extracted Boundary

Region Filling

Given a pixel inside a boundary, *region filling* attempts to fill that boundary with object pixels (1s)



Given a point inside here, can we fill the whole circle?

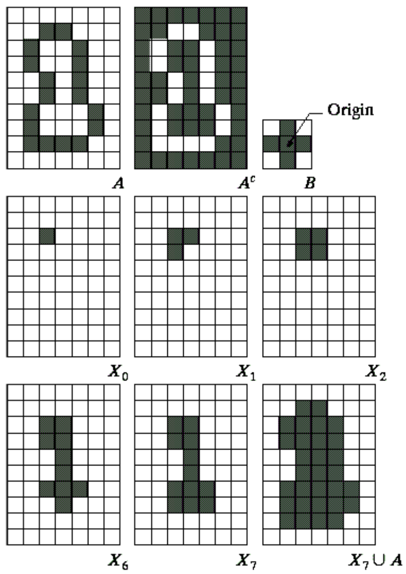
Region Filling

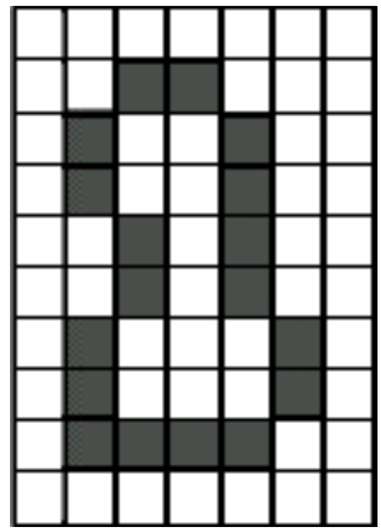
The key equation for region filling is

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3.....$$

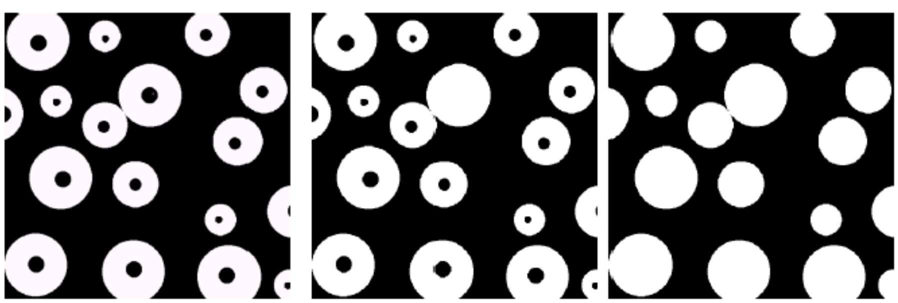
1. Where X_0 is simply the starting point inside the boundary, B is a simple structuring element and A^c is the complement of A
2. This equation is applied repeatedly until X_k is equal to X_{k-1}
3. Finally the result is unioned with the original boundary

Region Filling Step By Step



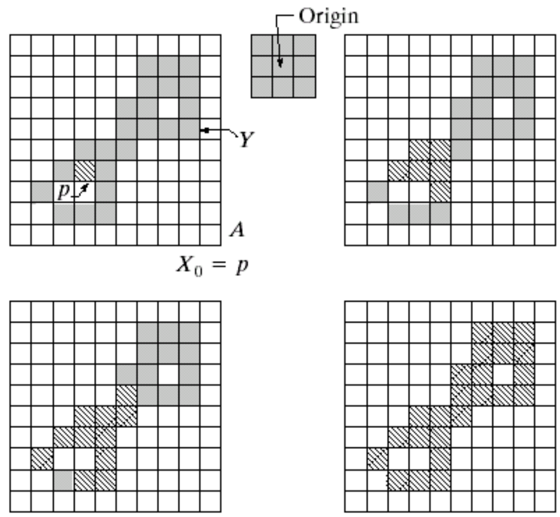


Region Filling Example



Original Image One Region Filled All Regions Filled

Extraction of connected components



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

Summary

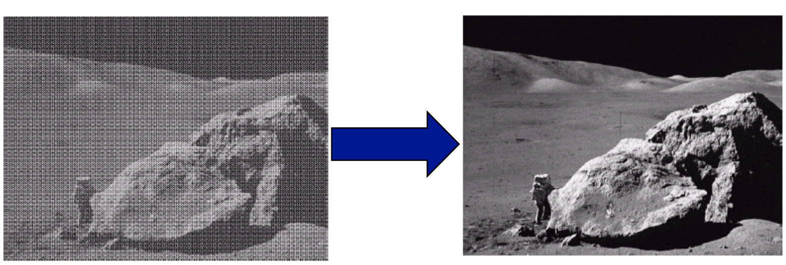
- The purpose of morphological processing is primarily to remove imperfections added during segmentation
- The basic operations are *erosion* and *dilation*
- Using the basic operations we can perform *opening* and *closing*
- More advanced morphological operation can then be implemented using combinations of all of these

Image Restoration: Noise Removal

What is Image Restoration?

Image restoration attempts to restore images that have been degraded

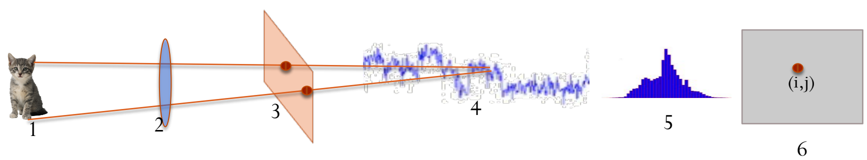
- Identify the degradation process and attempt to reverse it
- Similar to image enhancement, but more objective



Noise and Images

The sources of noise in digital images arise during image acquisition (digitization) and transmission

- Imaging sensors can be affected by ambient conditions
- Interference can be added to an image during transmission



Noise Model

We can consider a noisy image to be modelled as follows:

$$g(x,y) = f(x,y) + \eta(x,y)$$

where $f(x, y)$ is the original image pixel,

$\eta(x, y)$ is the noise term

$g(x, y)$ is the resulting noisy pixel

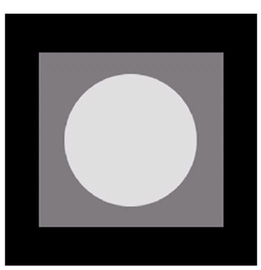
If we can estimate the noise model, we can figure out how to restore an image

Noise Models

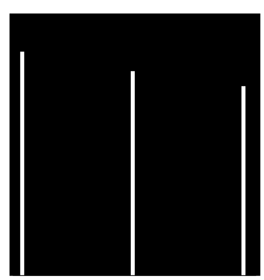
- There are many different models for the image noise term $\eta(x, y)$:
  - Gaussian
    - Most common model
  - Rayleigh
  - Erlang
  - Exponential
  - Uniform
  - Impulse
    - *Salt and pepper* noise

Noise Example

The test pattern to the right is ideal for demonstrating the addition of noise



Image

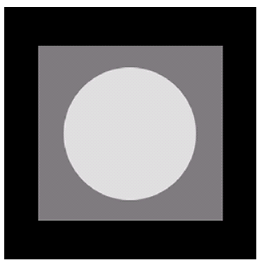


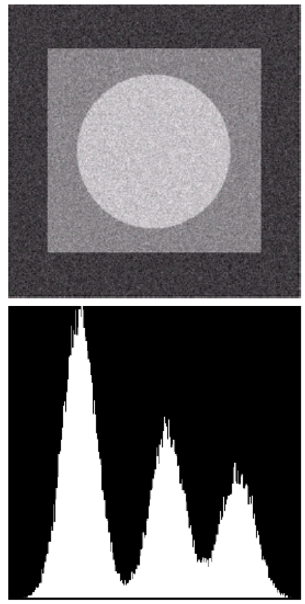
Histogram

Gaussian Noise

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$

z represents gray level
μ is the mean of average value of z
σ is the std deviation
σ^2 is the variance





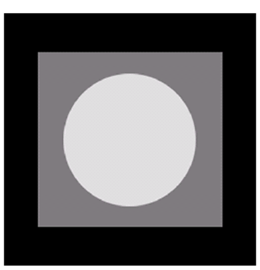
Gaussian

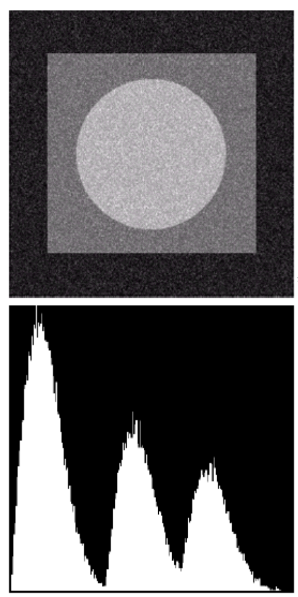
Rayleigh Noise

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for z >= a} \\ 0 & \text{for z < a} \end{cases}$$

$\mu = a + \sqrt{\pi b/4}$ b: scale parameter

$$\sigma^2 = \frac{b(4-\pi)}{4}$$

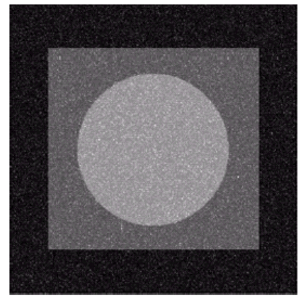


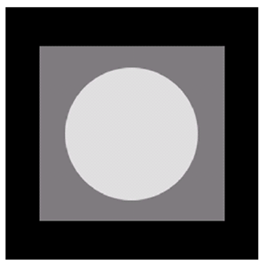


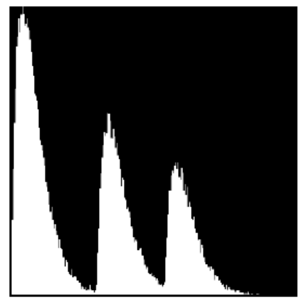
Rayleigh

Gamma Noise

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for z >= 0} \\ 0 & \text{for z < 0} \end{cases}$$



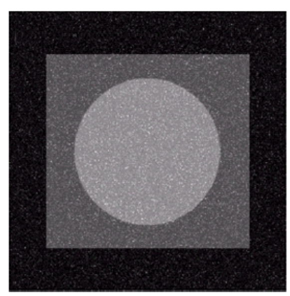


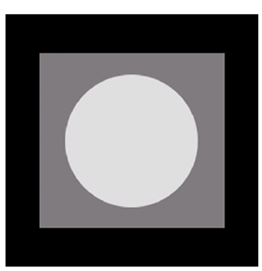


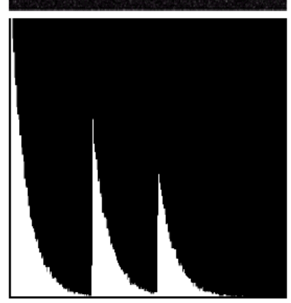
Gamma

Exponential Noise

$$p(z) = \begin{cases} ae^{-az} & \text{for z >= 0} \\ 0 & \text{for z < 0} \end{cases}$$



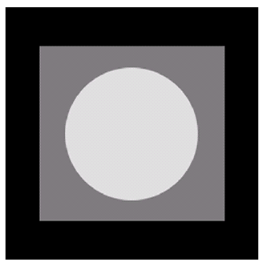


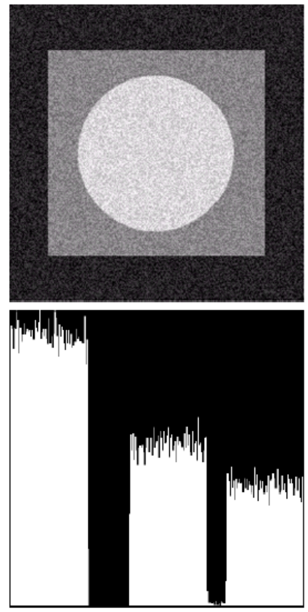


Exponential

Uniform Noise

$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{For } a <= z <= b \\ 0 & \text{otherwise} \end{cases}$$

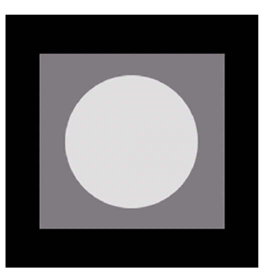


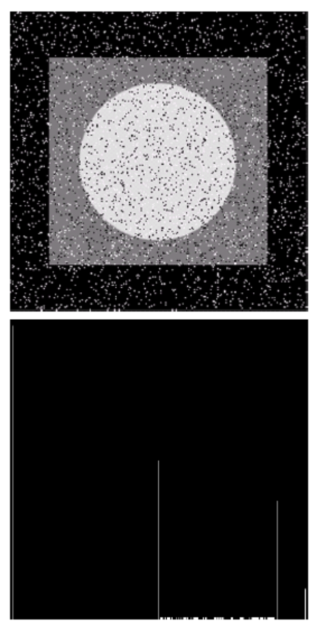


Uniform

Impulse Noise

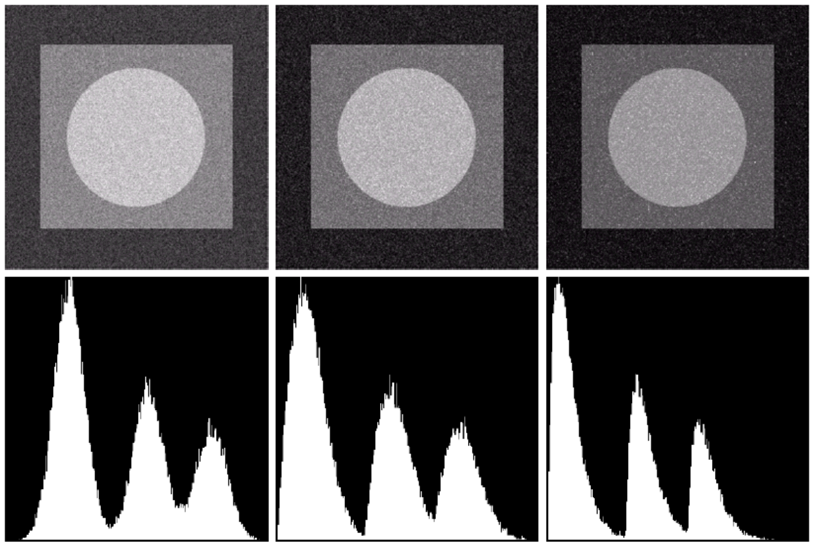
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$





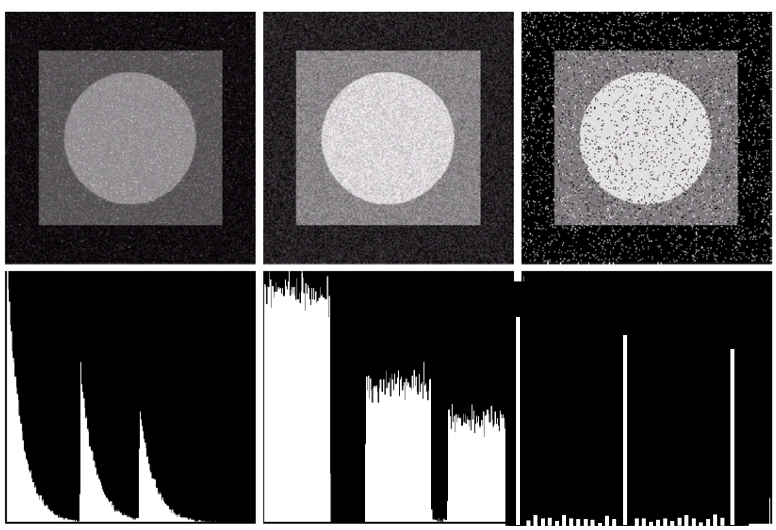
Salt & Pepper

Noise Example



Gaussian Rayleigh Erlang

Noise Example



Exponential Uniform Impulse

Filtering to Remove Noise

We can use spatial filters of different kinds to remove different kinds of noise

The *arithmetic mean* filter is a very simple one and is calculated as follows:

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

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

This is implemented as the simple smoothing filter

Blurs the image to remove noise

Other Means

There are different kinds of mean filters all of which exhibit slightly different behaviour:

- Geometric Mean
- Harmonic Mean
- Contraharmonic Mean

Geometric Mean

There are other variants on the mean which can give different performance

Geometric Mean:

$$\hat{f}(x,y) = \left[\prod_{(s,t) \in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail

Harmonic Mean

Harmonic Mean:

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

Works well for salt noise, but fails for pepper noise

Also does well for other kinds of noise such as Gaussian noise

Contraharmonic Mean

Contraharmonic Mean:

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Q is the *order* of the filter and adjusting its value changes the filter's behaviour

Positive values of Q eliminate pepper noise

Negative values of Q eliminate salt noise

Noise Removal Examples

Original Image

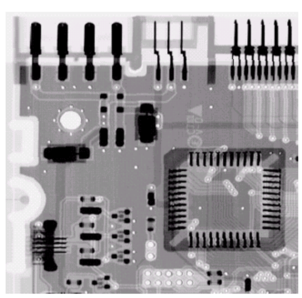
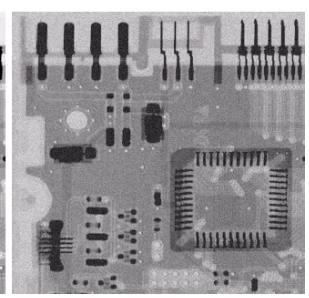
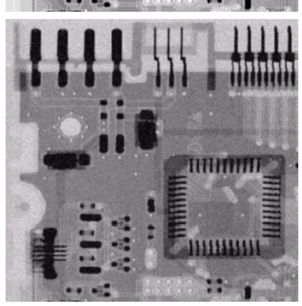


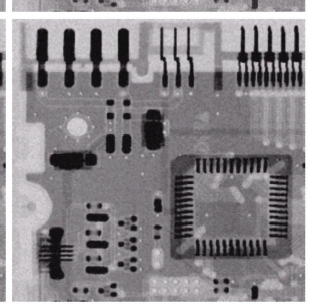
Image Corrupted By Gaussian Noise



After A 3*3 Arithmetic Mean Filter



After A 3*3 Geometric Mean Filter



Noise Removal Examples

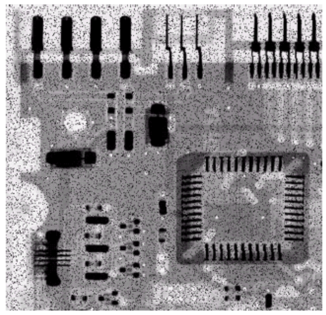
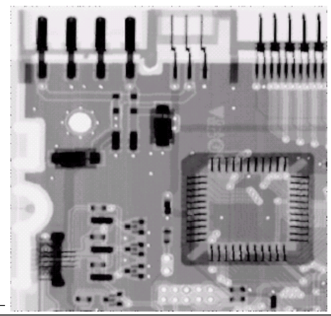


Image Corrupted By Pepper Noise



Result of Filtering Above With 3*3 Contraharmonic Q=1.5

Noise Removal Examples

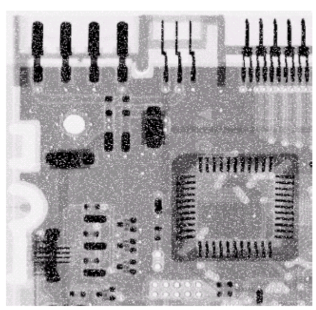
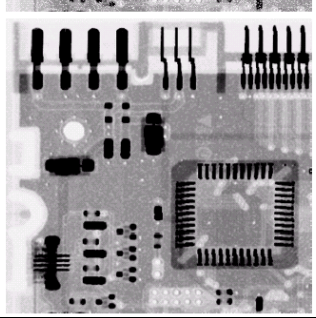


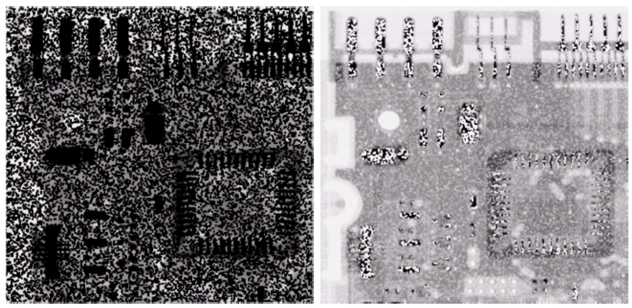
Image Corrupted By Salt Noise



Result of Filtering Above With 3*3 Contraharmonic Q=-1.5

Contraharmonic Filter: Limitations

Choosing the wrong value for Q when using the contraharmonic filter can have drastic results



Mean filters to Remove Noise

Different kinds of mean filters

- Arithmetic Mean
- Geometric Mean
- Harmonic Mean
- Contraharmonic Mean

Order Statistics Filters

Spatial filters that are based on ordering the pixel values that make up the neighbourhood operated on by the filter

Useful spatial filters include

- Median filter
- Max and min filter
- Midpoint filter
- Alpha trimmed mean filter

Median Filter

Median Filter:

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

Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters

Particularly good when salt and pepper noise is present

Max and Min Filter

Max Filter:

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

Min Filter:

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

Max filter is good for pepper noise and min is good for salt noise

Midpoint Filter

Midpoint Filter:

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

Good for random Gaussian and uniform noise

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

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

We can delete the $d/2$ lowest and $d/2$ highest grey levels

So $g_r(s, t)$ represents the remaining $mn - d$ pixels

Noise Removal Examples

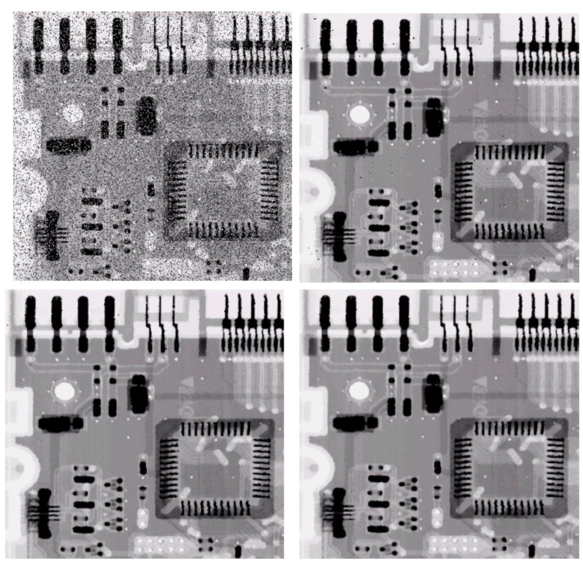


Image Corrupted By Salt And Pepper Noise

Result of 1 Pass With A 3*3 Median Filter

Result of 2 Passes With A 3*3 Median Filter

Result of 3 Passes With A 3*3 Median Filter

Noise Removal Examples

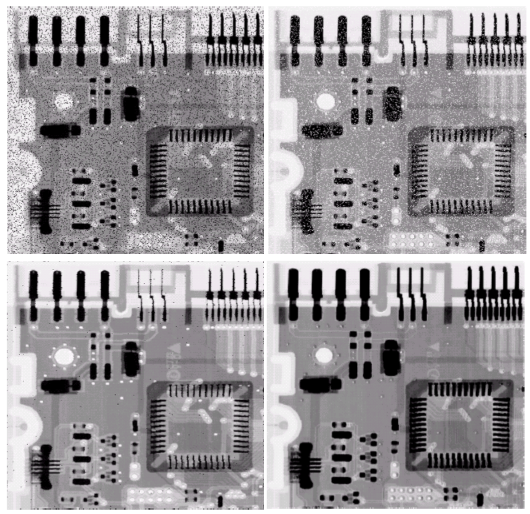


Image Corrupted By Pepper Noise

Image Corrupted By Salt Noise

Result Of Filtering Above With A 3*3 Max Filter

Result Of Filtering Above With A 3*3 Min Filter

Noise Removal Examples (cont…)

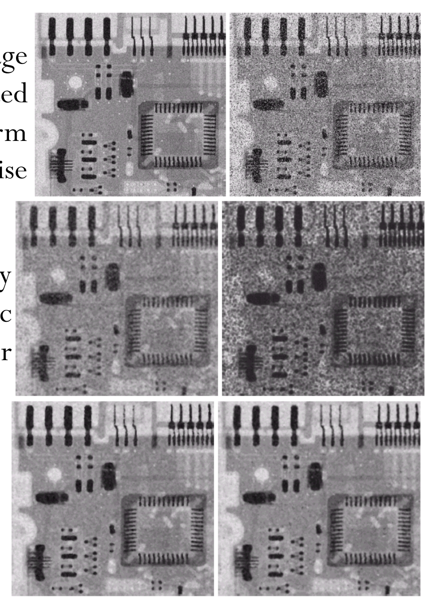


Image Corrupted By Uniform Noise

Image Further Corrupted By Salt and Pepper Noise

Filtered By 5*5 Arithmetic Mean Filter

Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Median Filter

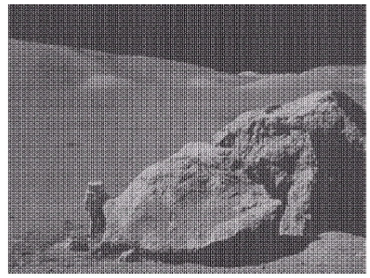
Filtered By 5*5 Alpha-Trimmed Mean Filter

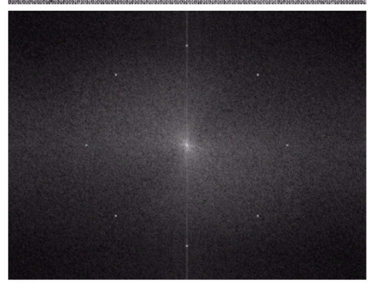
Periodic Noise

Typically arises due to electrical or electromagnetic interference

Gives rise to regular noise patterns in an image

Frequency domain techniques in the Fourier domain are most effective at removing periodic noise





Band Reject Filters

Removing periodic noise form an image involves removing a particular range of frequencies from that image

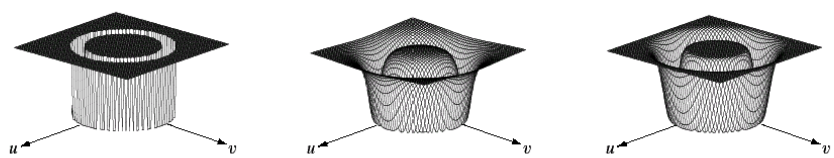
Band reject filters can be used for this purpose

An ideal band reject filter is given as follows:

$$H(u,v) = \begin{cases} 1 & if\ D(u,v) < D_0 - \frac{W}{2} \\ 0 & if\ D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & if\ D(u,v) > D_0 + \frac{W}{2} \end{cases}$$

Band Reject Filters (cont…)

The ideal band reject filter is shown below, along with Butterworth and Gaussian versions of the filter



Ideal Band Reject Filter

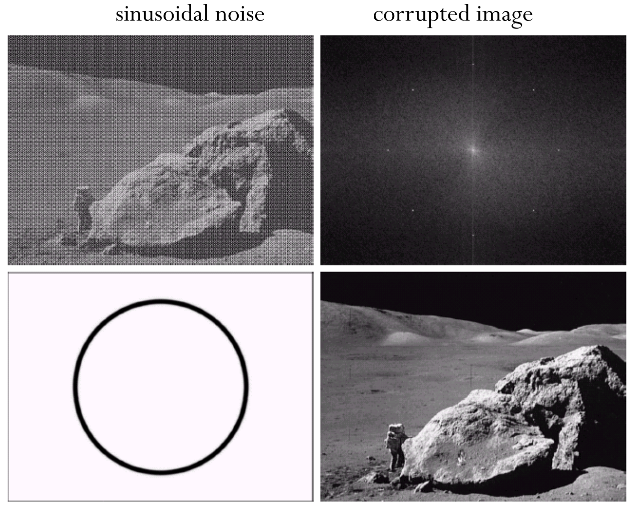
Butterworth Band Reject Filter (of order 1)

Gaussian Band Reject Filter

Band Reject Filter Example

Image corrupted by sinusoidal noise

Fourier spectrum of corrupted image



Butterworth band reject filter

Filtered image

Adaptive Filters

The filters discussed so far are applied to an entire image without any regard for how image characteristics vary from one point to another

The behaviour of **adaptive filters** changes depending on the characteristics of the image inside the filter region

Adaptive Filtering Example

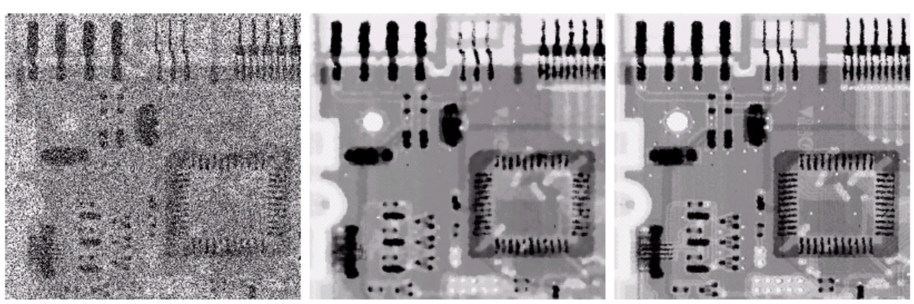


Image corrupted by salt and pepper noise with probabilities $P_a = P_b = 0.25$

Result of filtering with a 7 * 7 median filter

Result of adaptive median filtering with i = 7

Summary

- Morphological Image Analysis
- Image restoration for noise removal

Acknowledgements

- Some of the images and diagrams have been taken from the Gonzalez et al, "Digital Image Processing" book.