Image Restoration-1

Recap

- Introduction to Image Compression
- Types of Data Redundancy
  - Coding redundancy
  - Spatial and Temporal Redundancy
  - Irrelevant Information
- Measuring Image Information
- Fidelity Criteria
- General Image Compression Model
  - Encoding/Compression Process
  - Decoding/Decompression Process
- Lossless Compression
  - Huffman Coding

Lecture Objectives

- The image degradation/restoration model
- Noise models
  - Important noise probability density functions
  - Periodic noise
  - Estimating noise parameters
- Restoration using spatial filters
  - Mean filters
  - Order-static filters
  - Adaptive filters

Key Stages in DIP

Image Acquisition → Image Enhancement → Image Restoration → Image Compression

Image Compression → Morphological Processing → Image Segmentation

Image Segmentation → Object Recognition → Representation & Description

Image Degradation and Restoration Model

Subjective/Objective Assessment

- **Subjective assessment:** making assumptions, making interpretations based on personal opinions without any verifiable facts.

- **Objective assessment:** making an unbiased, balanced observation based on facts which can be verified.

What is Image Restoration?

- Overall goal
  - Improve an image in a pre-defined sense
  - Bring the degraded image close to its original form

- **Overlap** with some **image enhancement** techniques
  - Image enhancement is a very subjective process
    - Contrast stretching --> producing a “pleasing” image to eye
  - Image Restoration is an objective process
    - Removal of image blur using mathematical functions

What is Image Restoration?

- Restoration attempts to *recover an image* that has been degraded by using a *priori knowledge* of the degradation phenomenon.

- Restoration techniques *model the degradation* and apply the inverse process in order to *recover the original image*.

Image Enhancement Vs. Image Restoration

	Image Enhancement		Image Restoration
1.	As the name suggests, in Image Enhancement, the original image is processed so that the resultant image is more suitable than the original for specific applications.		The aim of image restoration is to bring the image towards what it would have been if it had been recorded without degradation.

Image Enhancement Vs. Image Restoration

	Image Enhancement		**Image Restoration**
1.	As the name suggests, in Image Enhancement, the original image is processed so that the resultant image is more suitable than the original for specific applications.		The aim of image restoration is to bring the image towards what it would have been if it had been recorded without degradation.
2.	Image enhancement makes a picture look better, without regard to how it really truly should look.		Image restoration tries to fix the image to get back to the real, true image.

Image Enhancement Vs. Image Restoration

	Image Enhancement		Image Restoration
1.	As the name suggests, in Image Enhancement, the original image is processed so that the resultant image is more suitable than the original for specific applications.		The aim of image restoration is to bring the image towards what it would have been if it had been recorded without degradation.
2.	Image enhancement makes a picture look better, without regard to how it really truly should look.		Image restoration tries to fix the image to get back to the real, true image.
3.	Image enhancement means improving the image to show some hidden details.		Image restoration means improving the image to match the original image.

Image Enhancement Vs. Image Restoration

	Image Enhancement		Image Restoration
1.	As the name suggests, in Image Enhancement, the original image is processed so that the resultant image is more suitable than the original for specific applications.		The aim of image restoration is to bring the image towards what it would have been if it had been recorded without degradation.
2.	Image enhancement makes a picture look better, without regard to how it really truly should look.		Image restoration tries to fix the image to get back to the real, true image.
3.	Image enhancement means improving the image to show some hidden details.		Image restoration means improving the image to match the original image.
4.	Image enhancement is a purely subjective processing technique.		Image restoration is an objective process.

Image Enhancement Vs. Image Restoration

	Image Enhancement		Image Restoration
1.	As the name suggests, in Image Enhancement, the original image is processed so that the resultant image is more suitable than the original for specific applications.		The aim of image restoration is to bring the image towards what it would have been if it had been recorded without degradation.
2.	Image enhancement makes a picture look better, without regard to how it really truly should look.		Image restoration tries to fix the image to get back to the real, true image.
3.	Image enhancement means improving the image to show some hidden details.		Image restoration means improving the image to match the original image.
4.	Image enhancement is a purely subjective processing technique.		Image restoration is an objective process.
5.	Image enhancement is a cosmetic procedure i.e. it does not add any extra information to the original image. It merely improves the subjective quality of the images by work in with the existing data.		Restoration tries to reconstruct by using a priori knowledge of the degradation phenomena. Restoration hence deals with getting an optimal estimate of the desired result

Underlying Assumption

- **We have some information** about how the image has been degraded.
  - It's a loose statement

- How to quantify image degradation when we don't know how an image was degraded?
  - Model the process of degradation
  - By making certain assumptions about the *structure of signal and noise relationship*

- **How is noise distributed as a function of intensity?**

What is Image Degradation ?

- **Image degradation** is said to occur when an image undergoes loss of stored information, resulting in decreased visual quality.

- Some reasons for image degradation are:
  - Motion blur (Movements during the image capture process)
  - Noise
  - Out-of-focus optics, use of a wide-angle lens, atmospheric turbulence, or a short exposure time, which reduced the number of photons captured.
  - Scattered light distortion
  - Error in sensor operations
  - Error in digitization/conversion process
  - Error in processing algorithmic operations, etc.

What is Noise in Images ?

- **Noise** in an image is the presence of artifacts that do not originate from the original scene content.

- Noise in an image is partly due to **physical/electronic constraints** on the acquisition equipment, transmission lines, storage media etc..

- In any case, noise results in a degraded digital image.

- Noise from an image can be removed in:
  - Spatial domain
  - Frequency domain

- A restoration filter is used to remove noise in images.

What is Noise in Images ?

- The presence of noise in an image might be additive or multiplicative.

- In the Additive Noise Model, an additive noise signal is added to the original signal to produce a corrupted noisy signal that follows the following rule:

$$g(x, y) = f(x,y) + n(x, y)$$

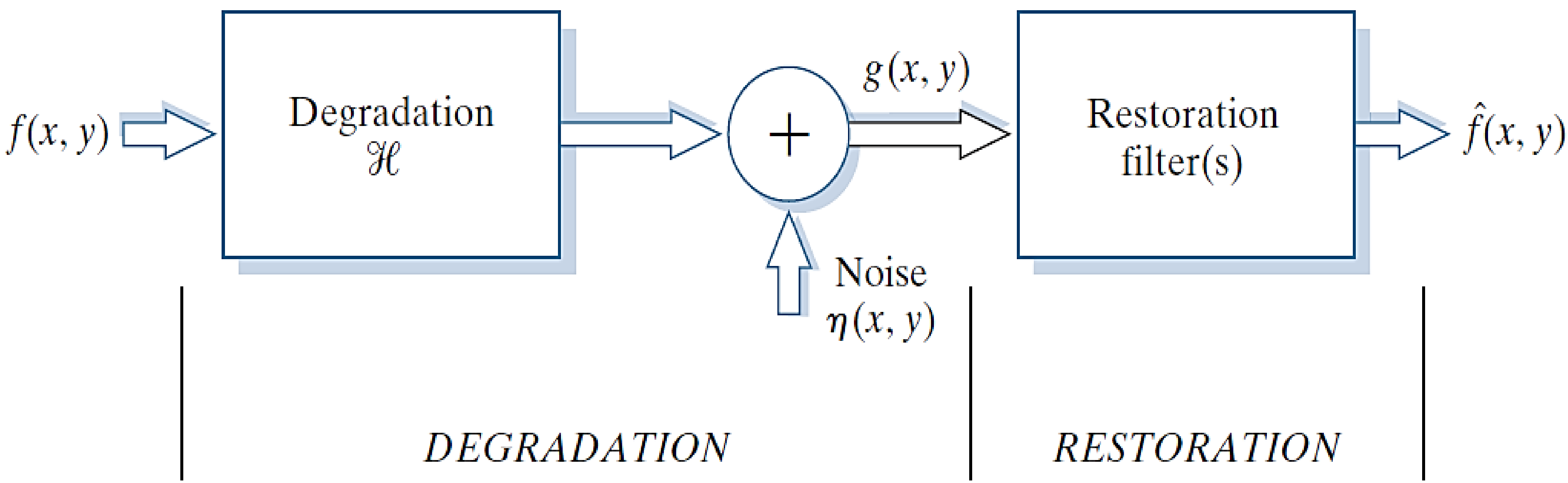
Here,

- $f(x,y)$ represents the original image intensity
- $n(x,y)$ represents the noise added to produce the corrupted image
- $g(x,y)$ degraded image

- Similarly, the Multiplicative Noise Model multiplies the original signal by the noise signal.

A Model of Image Degradation/Restoration

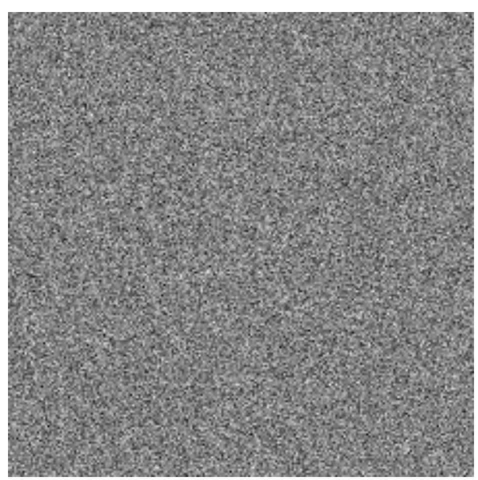
- In this course, we model *image degradation* as an operator ***H*** that, together with an *additive noise term* ***n(x, y)***, operates on an *input image* ***f*** **(*x, y*)** to produce a *degraded image* **g(*x, y*).**



- Given ***g*(*x, y*)**, some knowledge about ***H***, and some knowledge about the additive noise term ***n(x, y)***, the objective of restoration is to obtain an estimate $\hat{f}$**(*x, y*)** of the original image.

A Model of Image Degradation/Restoration

Original
***f* (*x*, *y*)**



Noise $\boldsymbol{n(x, y)}$

With noise

A Model of Image Degradation/Restoration

With noise

Inverse degradation

Restoration Filter

Restored $\hat{f}(x, y)$

Original $f\,(x, y)$

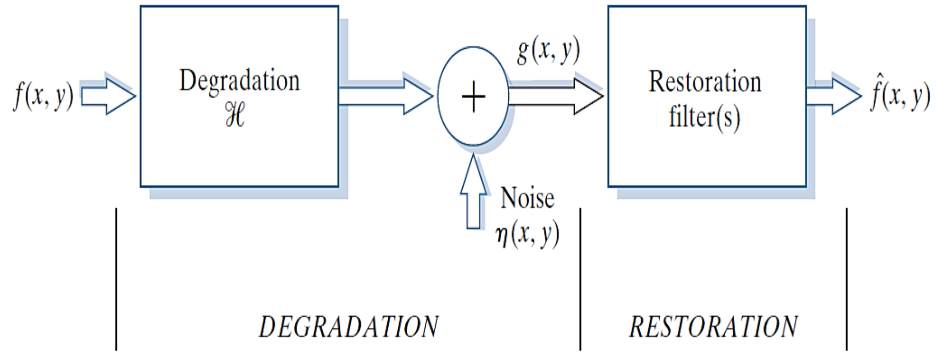
A Model of Image Degradation/Restoration

- **In spatial domain:**

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

Where,

- **g(x,y)**: degraded image
- **f(x,y)**: original image
- **h(x,y)**: degradation Filter
- **η(x,y)**: additive noise term



- **In frequency domain:**

$$G(u,v) = H(u,v).F(u,v) + \eta(u,v)$$

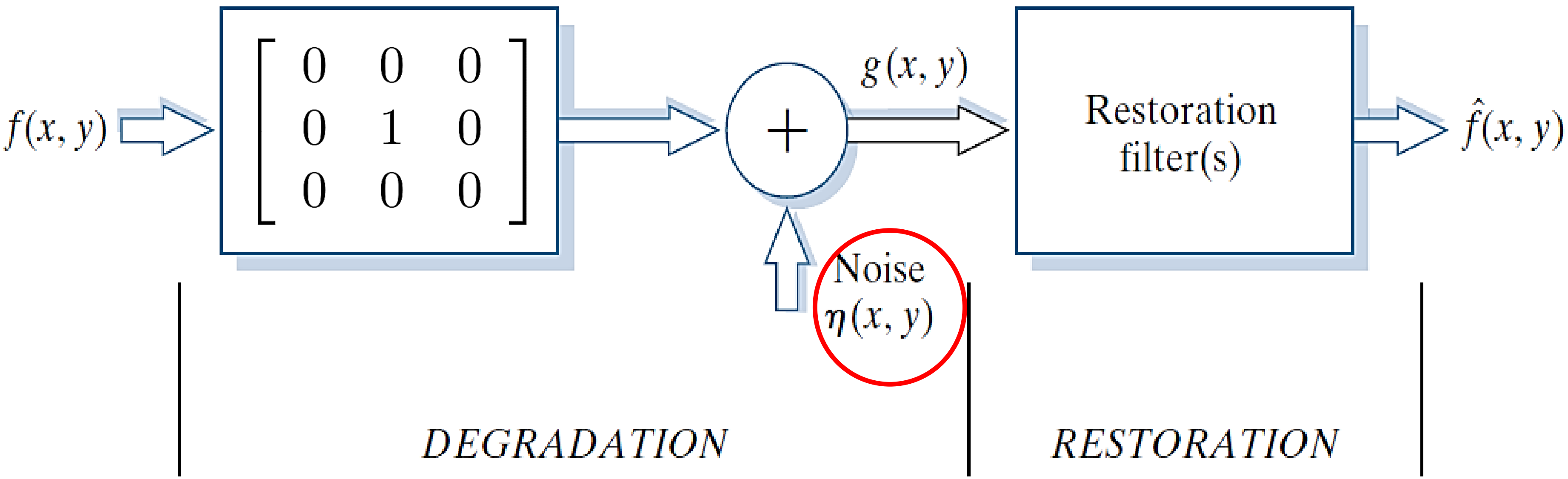
Noise Models

Assume H is the identity operator (kernel)

0	0	0
0	1	0
0	0	0

An Identity Kernel when applied to an image through convolution, **will have no effect on the resulting image**.

Linear Additive Noise and Degradation



$f(x, y)$: Input image
$H(x, y)$: Degradation filter
$g(x, y)$: Degraded image
$\eta(x, y)$: Noise
$\hat{f}(x, y)$: Restored Image

Noise Models

- Sources of noise
  - Image acquisition and transmission
    - Environmental factors (temperature, electro-mechanical failures)
  - Communication channel
    - Interference

Spatial and Frequency Properties of Noise

- Spatial properties of noise
  - Spatial characteristics of noise – pixel location
  - Correlation of noise with the input image - pixel value

- Frequency properties of noise
  - Refers to the frequency content of noise in the Fourier sense, not actual electromagnetic spectrum frequencies
  - E.g. When the Fourier spectrum of noise is constant, it is known as white noise
    - Physical properties of **white light** - contain all frequencies in equal proportion

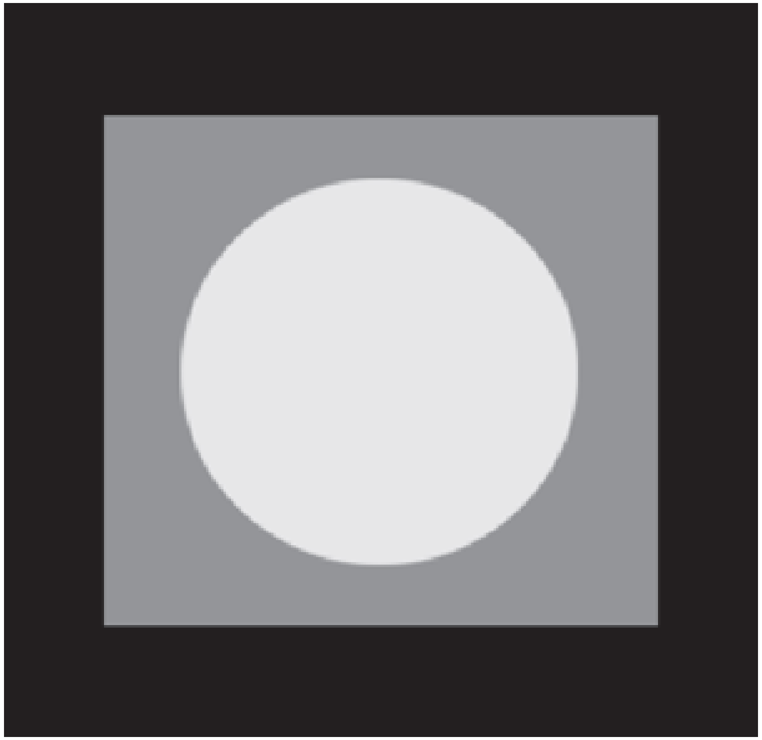
Modeling Noise as a Probability Density Function (PDF)

Modelling Noise Components

- **Spatial Noise** - independent of original image pixel location and pixel intensity.

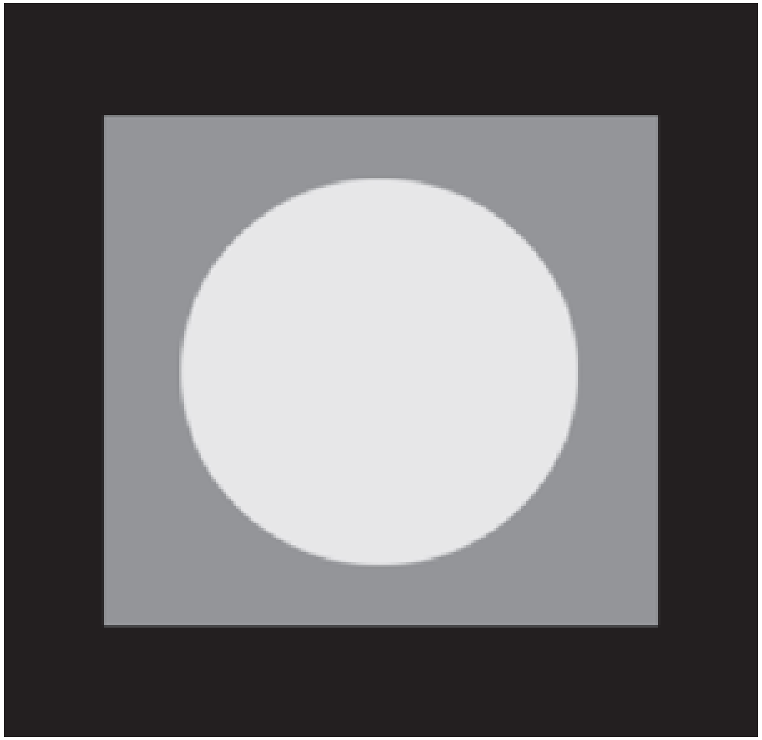
- The noise model:
  - Presents the statistical behavior of intensity values in the noise component $\eta(x,y)$.
  - The noise components may be considered as random variables, characterized by a probability density function (PDF).
  - The noise component of the noise model is also an image $\eta(x,y)$, of the same size as the input image.
  - We generate noise component (image) with a specified probability density function.

Visual Characteristics of Noise Models

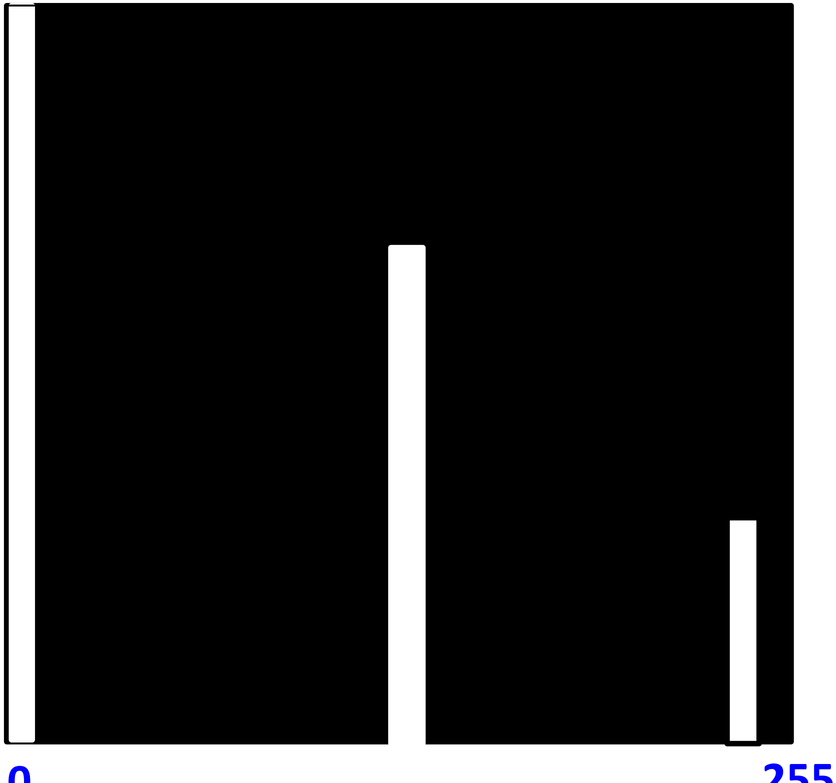


Input Image: constant areas that span the gray scale from black to near white in only three increments

Visual Characteristics of Noise Models



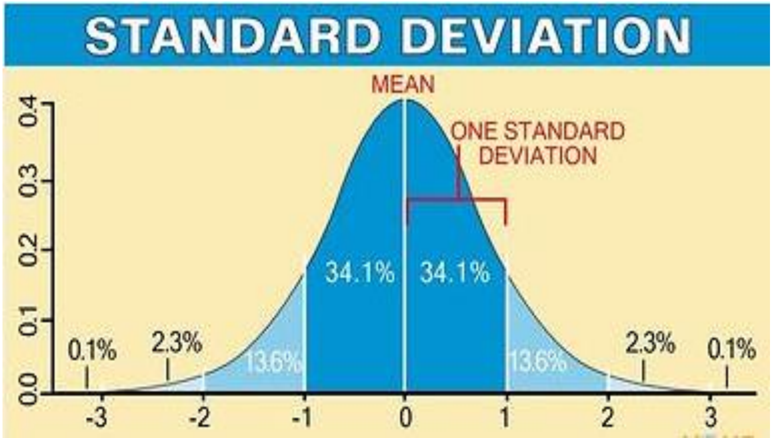
Input Image: constant areas that span the gray scale from black to near white in only three increments



Input Image histogram

What is Standard Deviation?

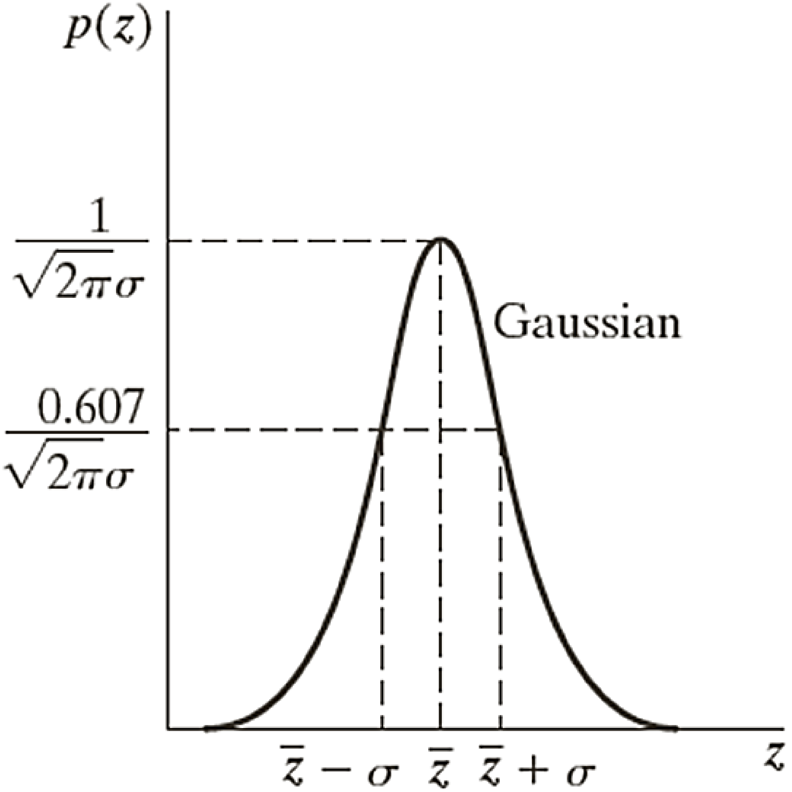
To compute standard deviation by hand: The standard deviation is simply the square root of the variance. This description is for computing population standard deviation. If sample standard deviation is needed, divide by n - 1 instead of n. Since standard deviation is the square root of the variance, we must first compute the variance.	
1. Find the mean.	$\overline{x}$
2. Subtract the mean from each data value and square each of these differences (*the squared differences*).	$(x-\overline{x})^2$
3. Find the average of the squared differences (add them and divide by the count of the data values). This will be the variance.	$\frac{1}{n}\sum_{i=1}^{n}(x-\overline{x})^2$ variance
4. Take the square root. This will be the population standard deviation. Round the answer according to the directions in the problem.	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x-\overline{x})^2}$ standard deviation



PDF of Gaussian Noise

- Gaussian (Normal) distribution
- $\approx 68\%$ values are between $(\bar{z}-\sigma,\ \bar{z}+\sigma)$ and 95% values are between $(\bar{z}-2\sigma, \bar{z}+2\sigma)$

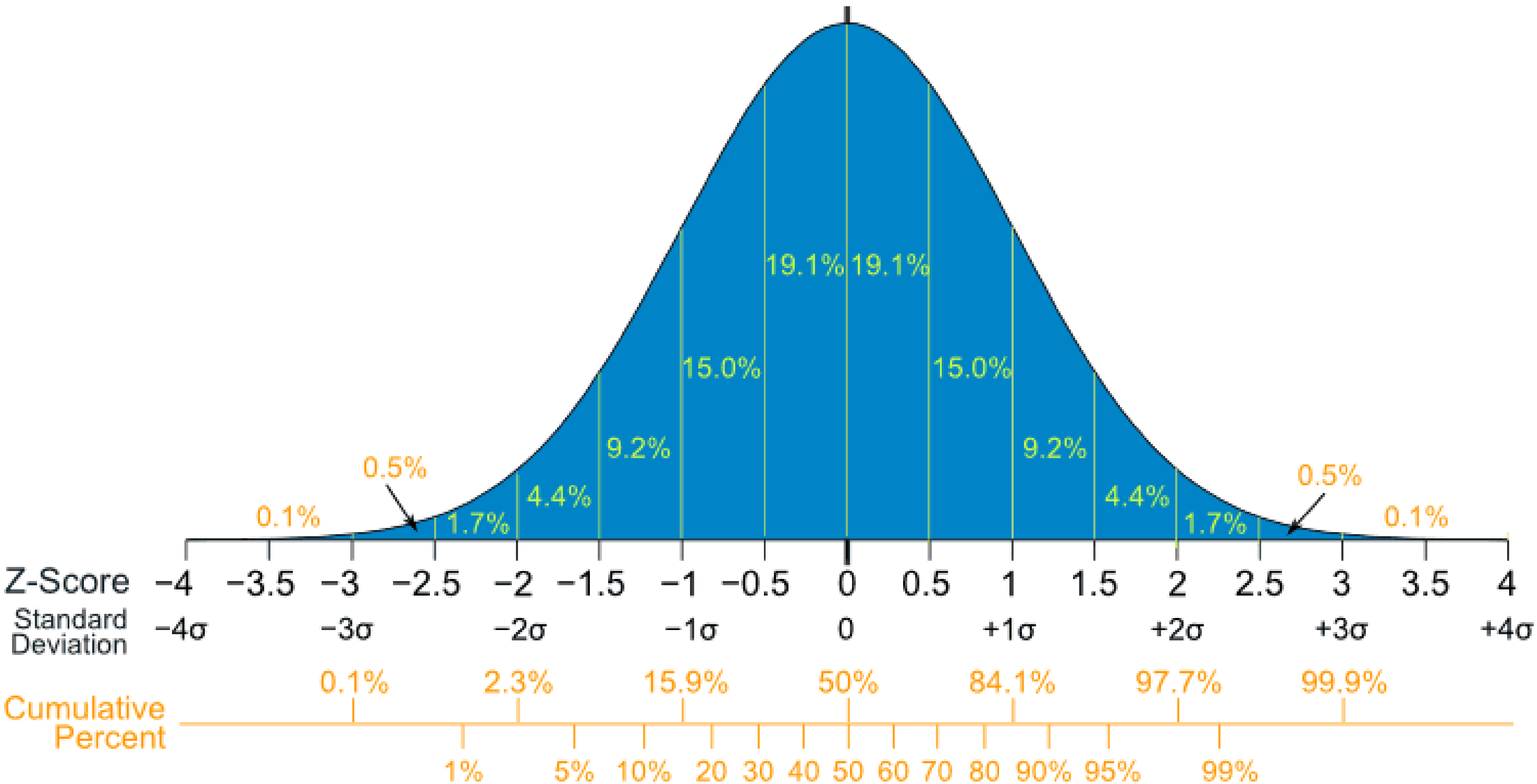
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}} \qquad -\infty < z < \infty$$



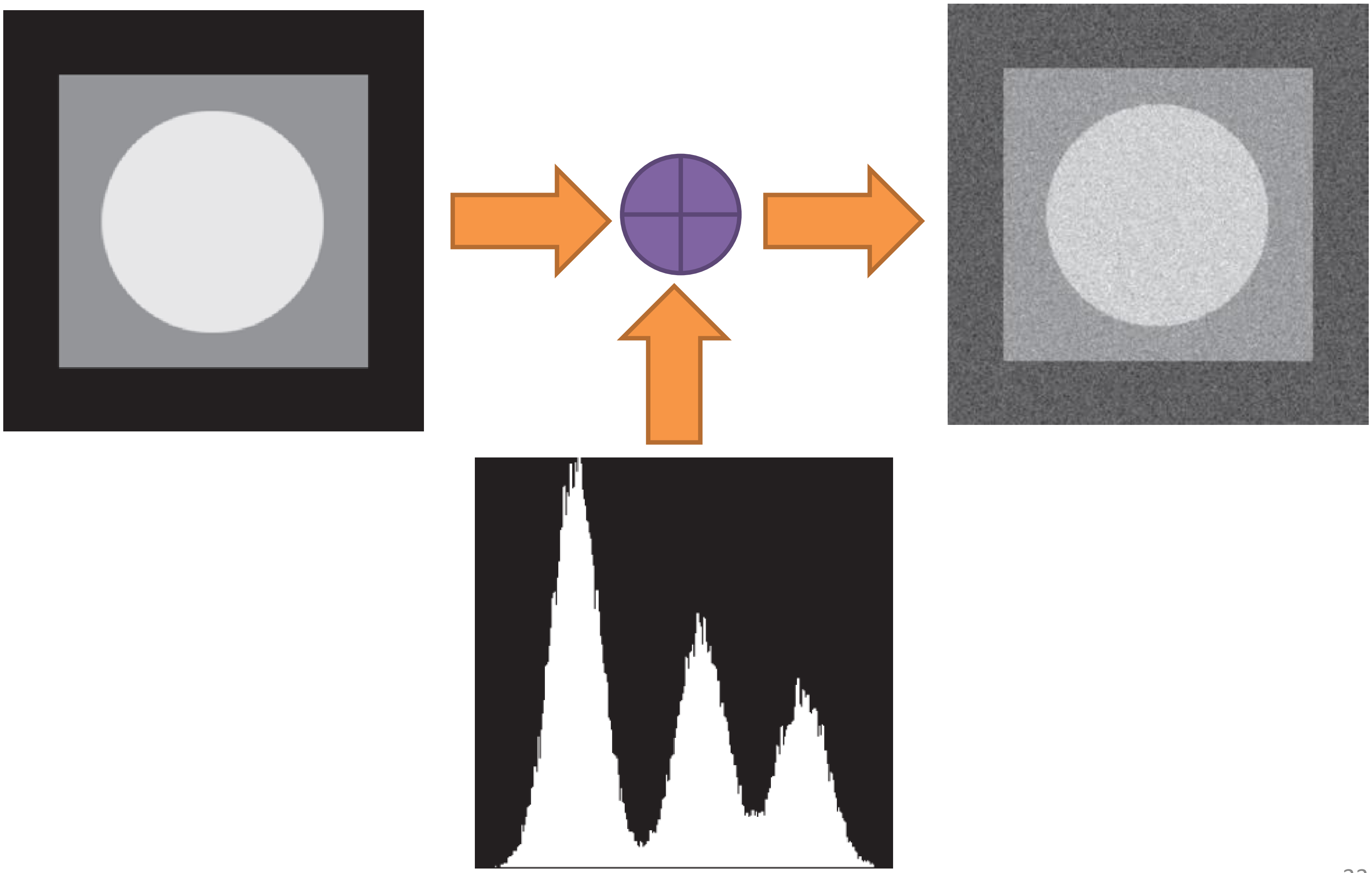
where,

- $\boldsymbol{z}$ represents intensity
- $\bar{\boldsymbol{z}}$ is the mean (average) value of z
- $\boldsymbol{\sigma}$ is its standard deviation

Standard Gaussian Distribution



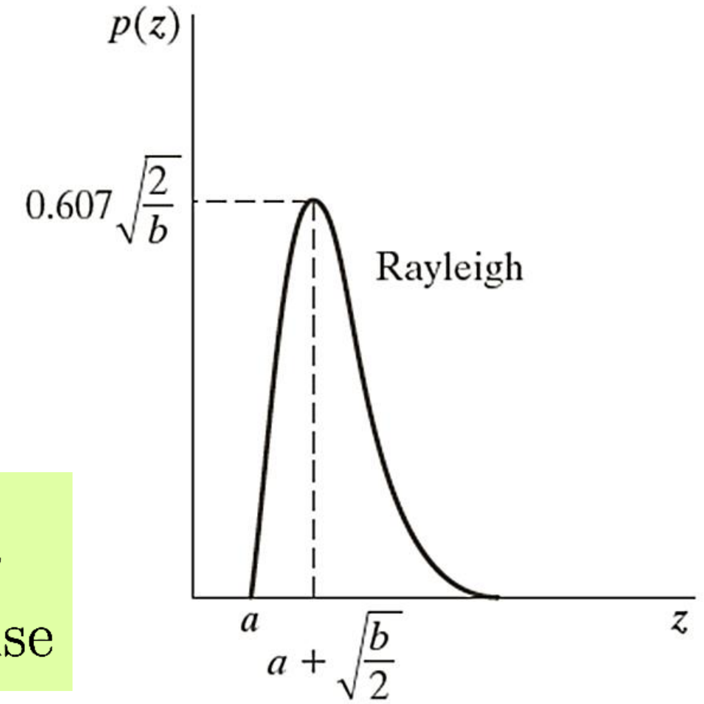
Effect of Gaussian Noise



PDF of Rayleigh Noise

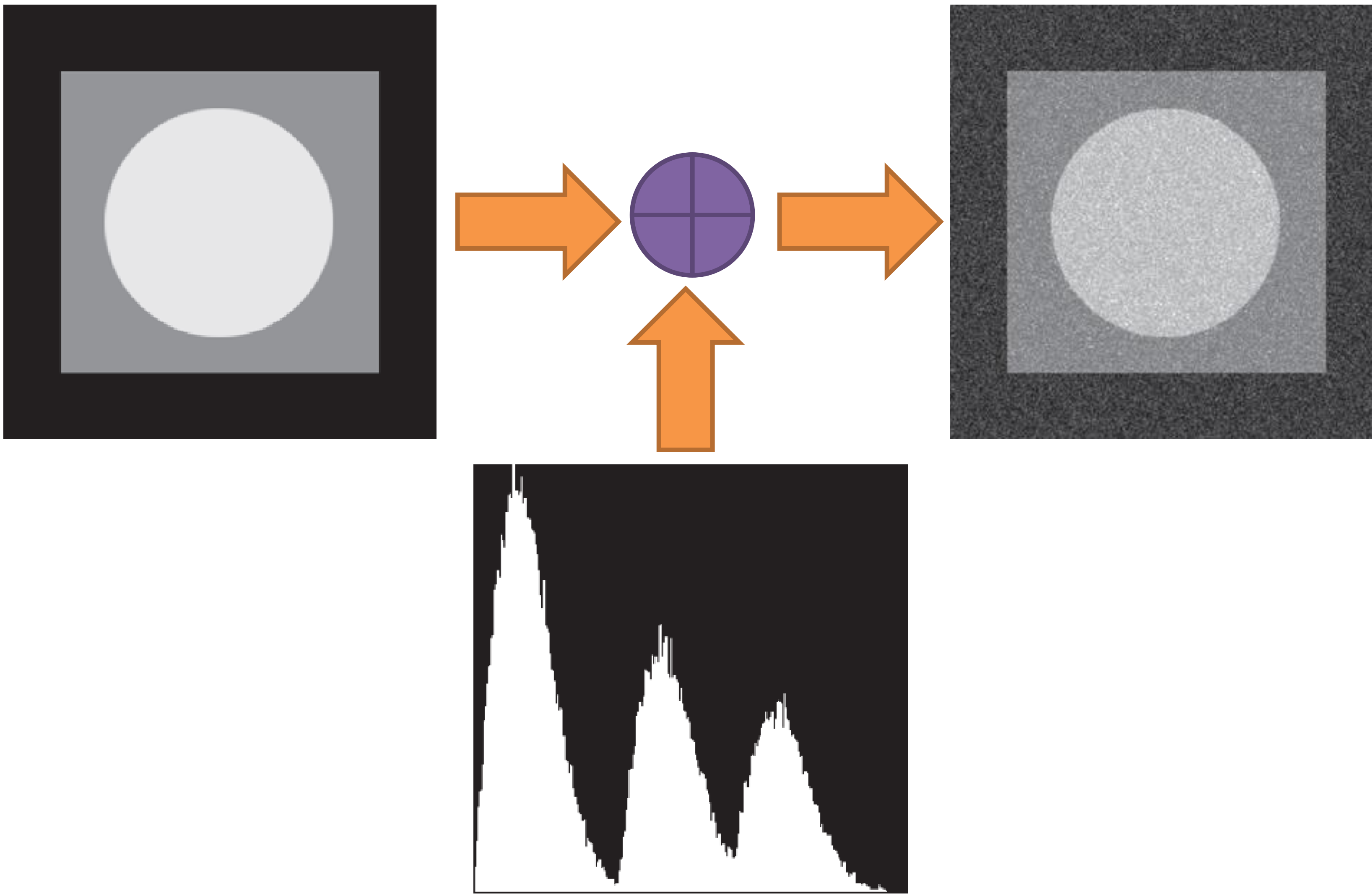
- Basic plot of the density is skewed to the right
- Useful for approximating skewed histograms

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{if } z \geq a \\ 0 & \text{otherwise} \end{cases}$$



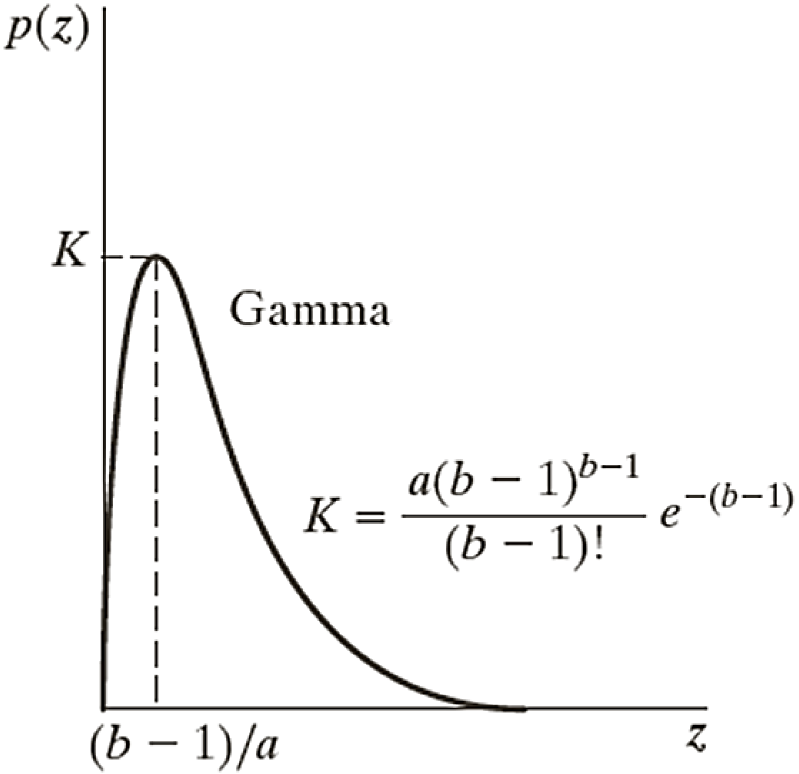
$$\bar{z} = a + \sqrt{\pi b/4} \quad and \quad \sigma^2 = \frac{b(4-\pi)}{4}$$

Effect of Rayleigh Noise



Erlang (Gamma) Noise

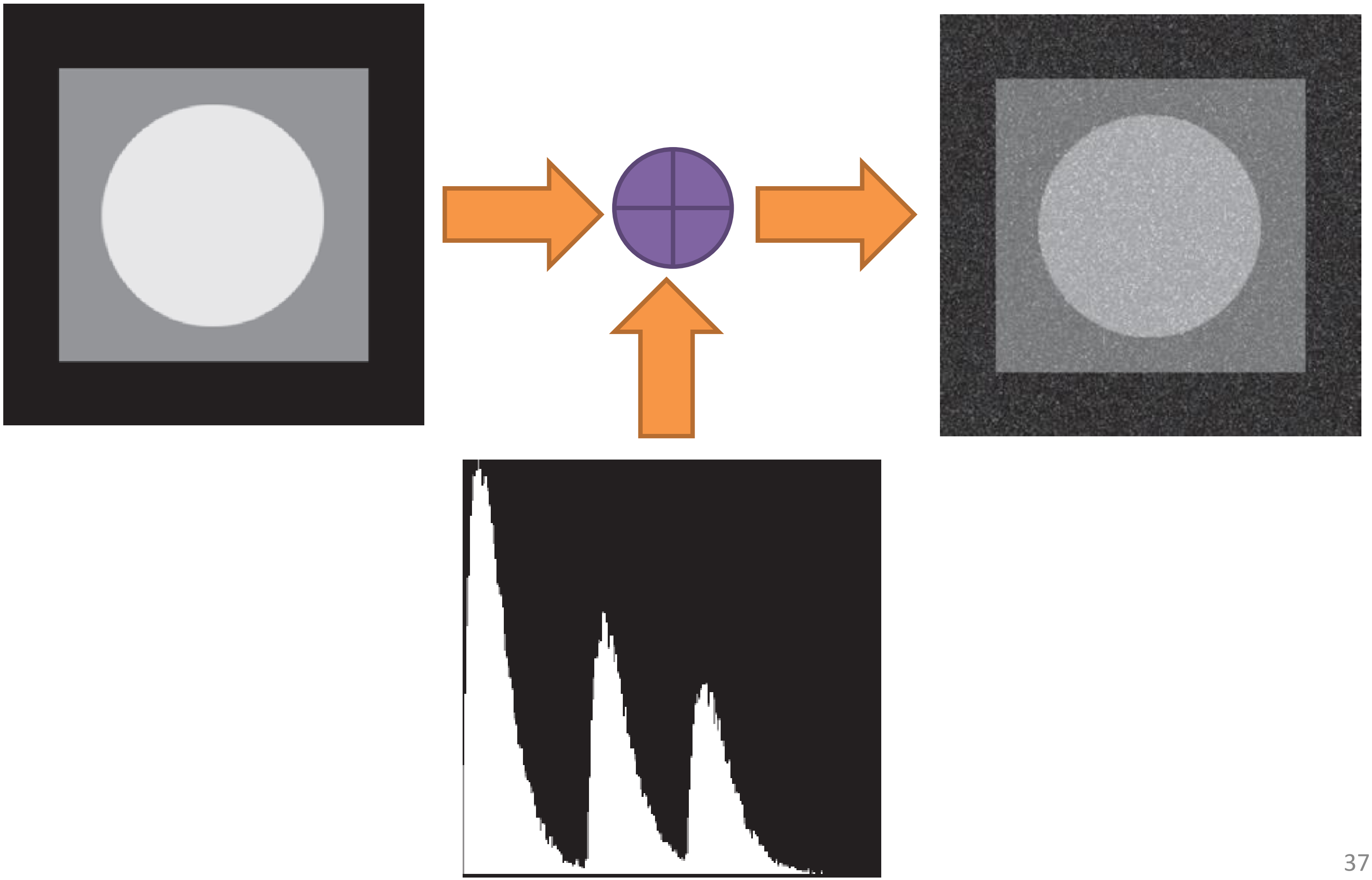
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



$$\bar{z} = \frac{b}{a} \quad and \quad \sigma^2 = \frac{b}{a^2}$$

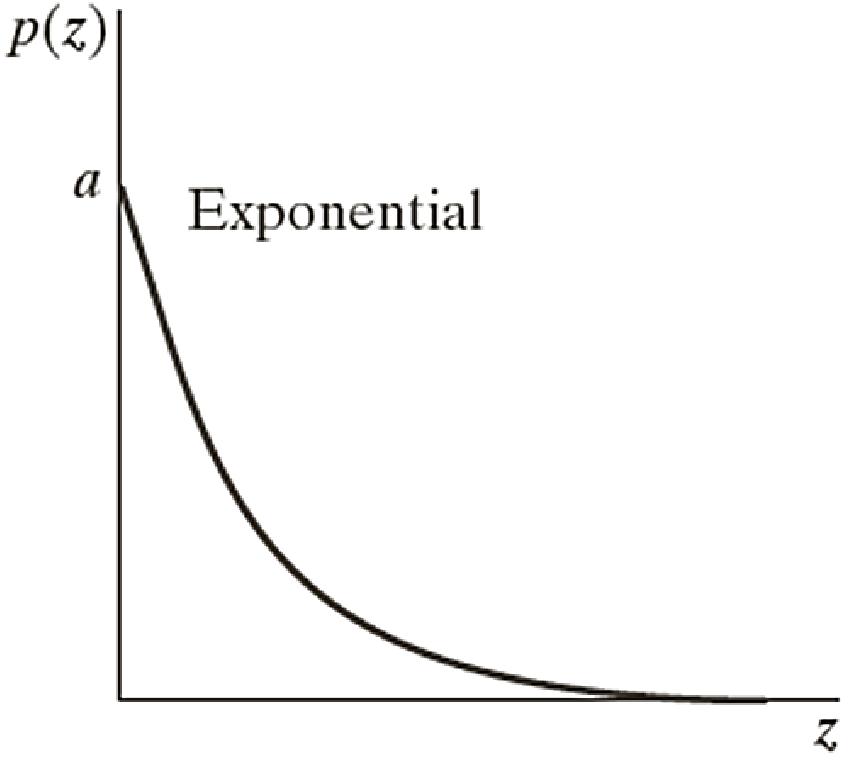
Parameters are such that ***a* > *b***, *b* is a **positive integer**, and "**!**" indicates factorial

Effect of Erlang (Gamma) Noise



PDF of Exponential Noise

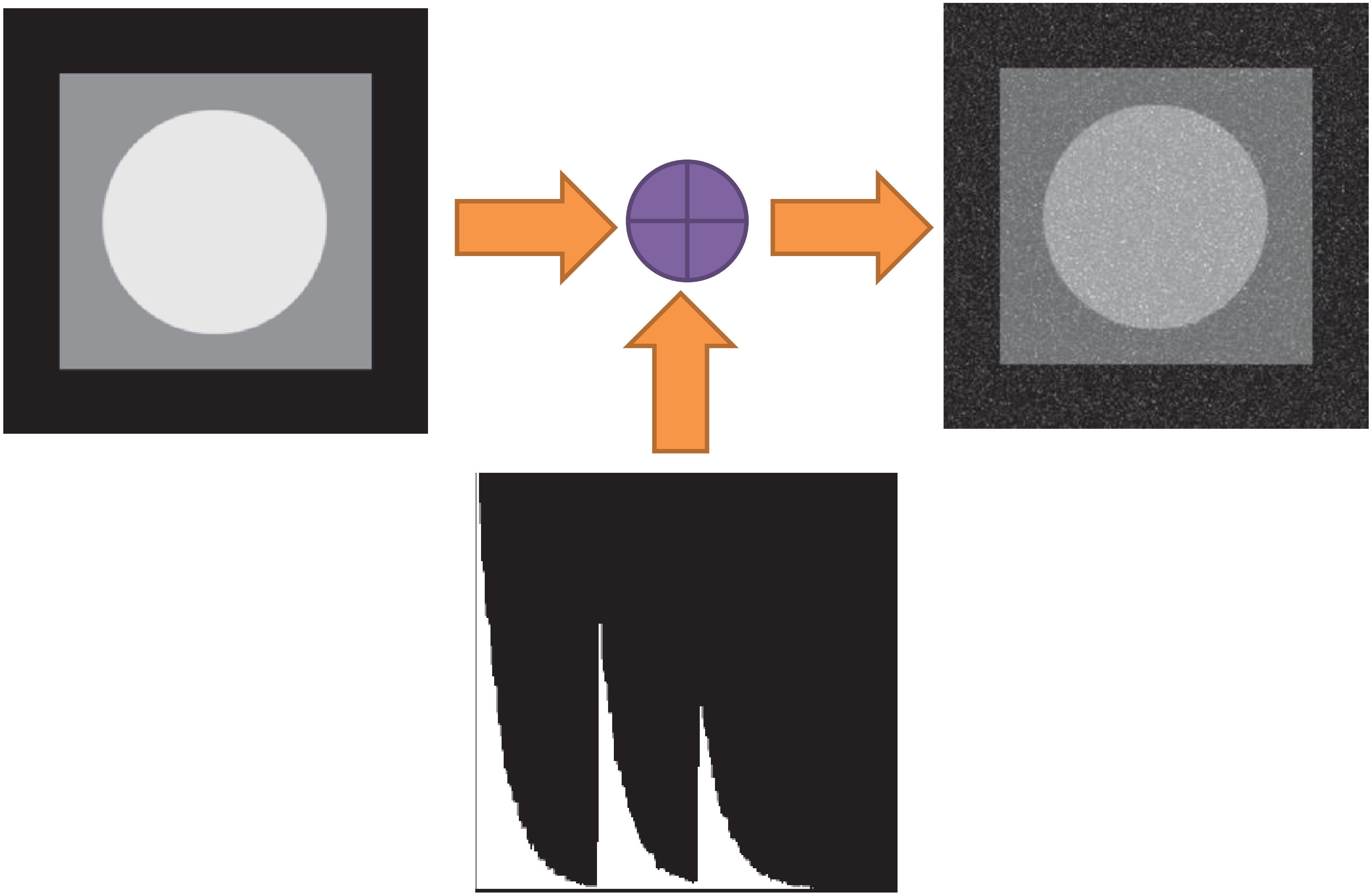
$$p(z) = \begin{cases} ae^{-az} & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



$$\bar{z} = \frac{1}{a}, \qquad and \qquad \sigma^2 = \frac{1}{a^2}$$

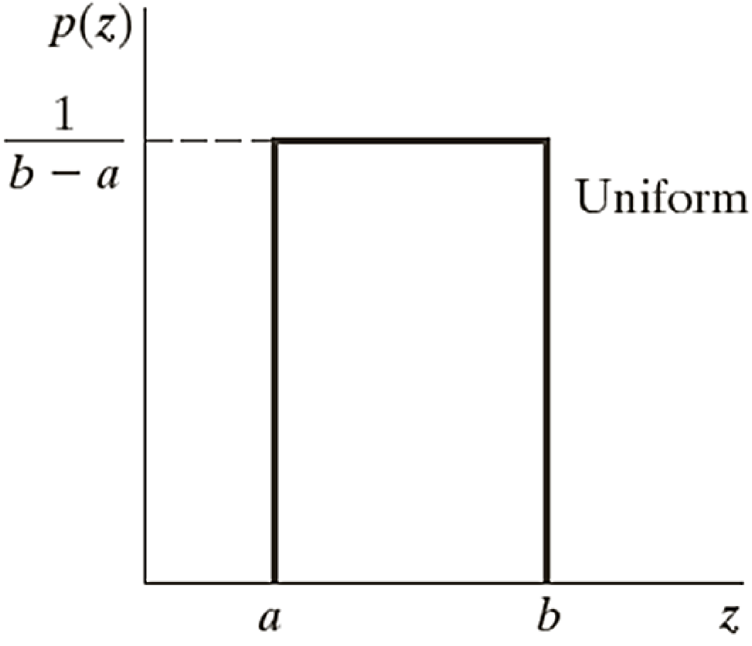
This PDF is a special case of the *Erlang* PDF with ***b* = 1**

Effect of Exponential Noise



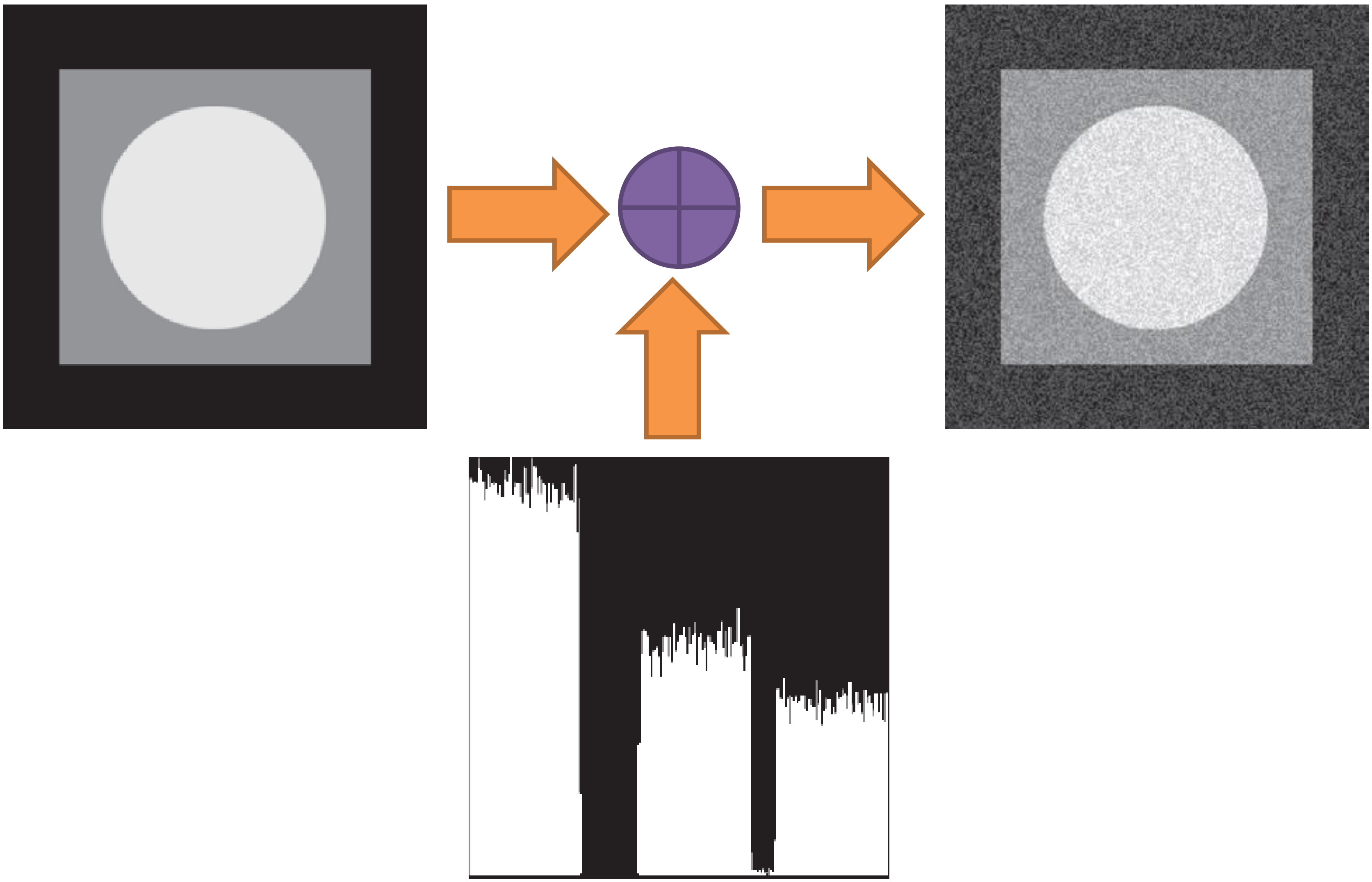
PDF of Uniform Noise

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$



$$\bar{z} = \frac{a+b}{2}, \qquad and \qquad \sigma^2 = \frac{(b-a)^2}{12}$$

Effect of Uniform Noise

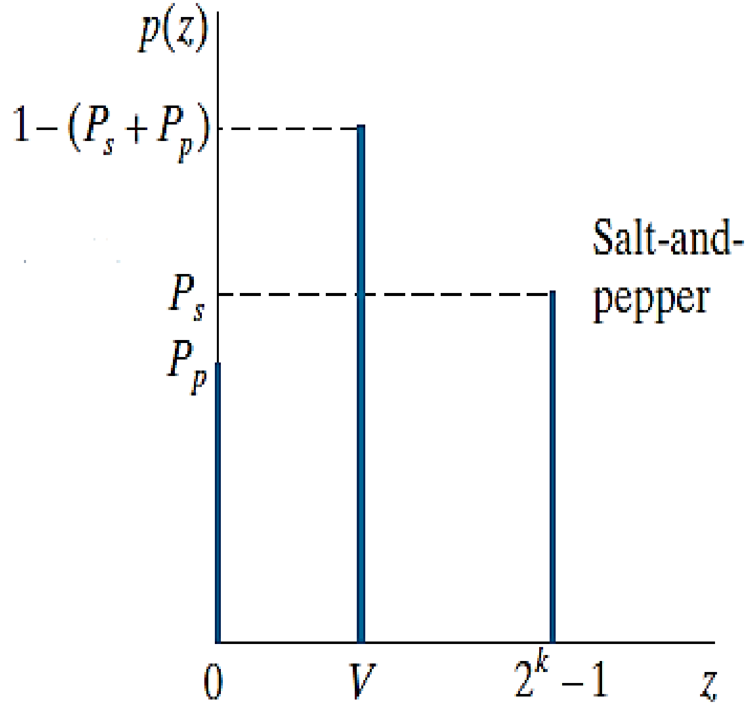


PDF of Impulse (Salt-and-pepper) Noise

$$p(z) = \begin{cases} P_s & \text{for } z = 2^k - 1 \\ P_p & \text{for } z = 0 \\ 1 - (P_s + P_p) & \text{for } z = V \end{cases}$$

$$\bar{z} = (0)P_p + K(1 - P_s - P_p) + (2^k - 1)P_s$$

$$\sigma^2 = (0 - \bar{z})^2 P_p + (K - \bar{z})^2 (1 - P_s - P_p) + (2^k - 1)^2 P_s$$



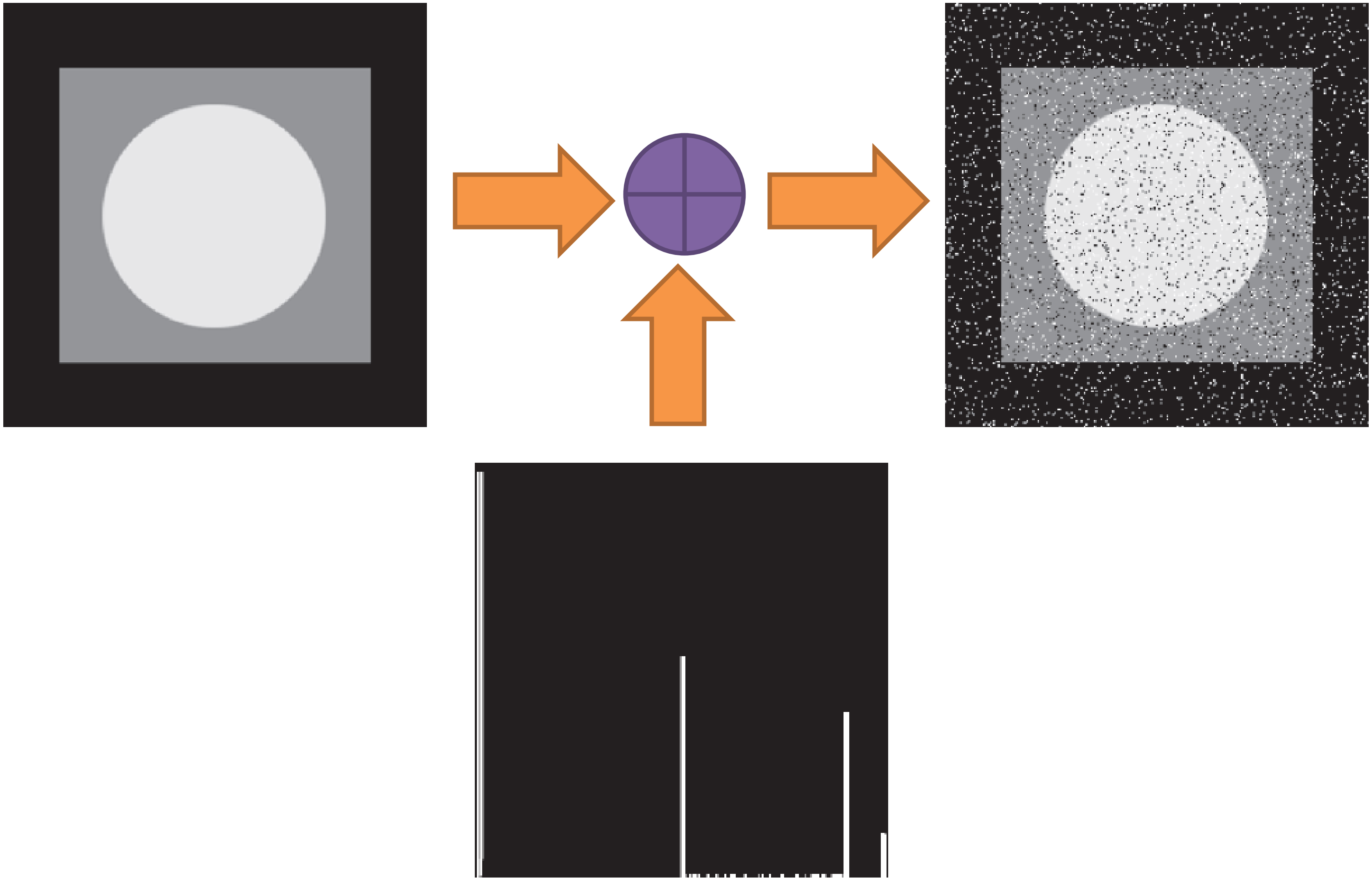
- ***P_s*** is the *salt pixels* and ***P_p*** is the *pepper pixels*.
- ***k*** is the *number of bits* used to represent the *intensity values* in a digital image.
- ***V*** is any integer value in the range **$0 < V < 2^k - 1$**.

How Salt-and-Pepper Noise is Added to an Image?

- Let **η(x,y)** denote a *salt-and-pepper noise* image. Given an *image* **f(x,y)** of the **same size** as **η(x,y)**, we **corrupt** it with salt-and-pepper noise as follows:
  - Assign a **0** to all locations in ***f*** wherever a **0** occurs in **η**.
  - Assign a value of $\mathbf{2^k - 1}$ to all location in ***f*** wherever $\mathbf{2^k - 1}$ occurs in **η**.
  - Leave unchanged all location in ***f*** where ***V*** occurs in **η**.

- Salt-and-pepper noise is also termed as:
  - *bipolar impulse noise*
  - *Unipolar impulse noise* if either P_s or P_P is 0
  - *data-drop-out noise*
  - *spike noise*

Effect of Salt-and-Pepper Noise



Applications of Noise Models

- Gaussian – modelling electronic circuit noise, sensor noise due to heat/poor illumination
- Rayleigh – modelling noise in range imaging (distance fields)
- Exponential, Gamma – modelling noise in laser imaging
- Impulse – modelling noise in faulty switching in images
- Uniform – modelling noise in random number generation, simulation

Periodic Noise

Periodic Noise

- Periodic noise is a type of *spatially dependent noise.*
- It typically arises from electrical or electromechanical interference during image acquisition.
- Periodic noise can be reduced significantly via frequency domain filtering.

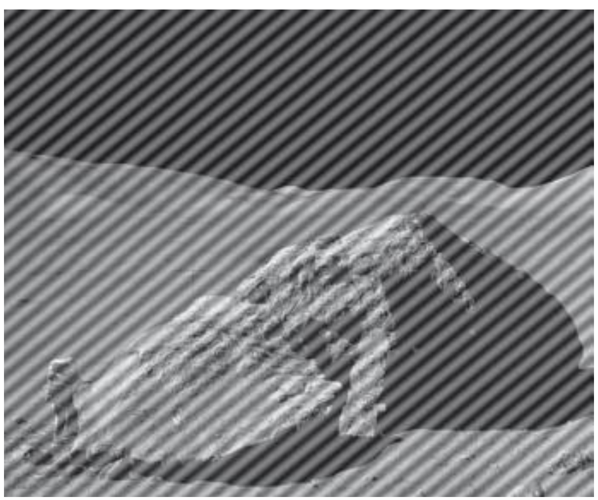
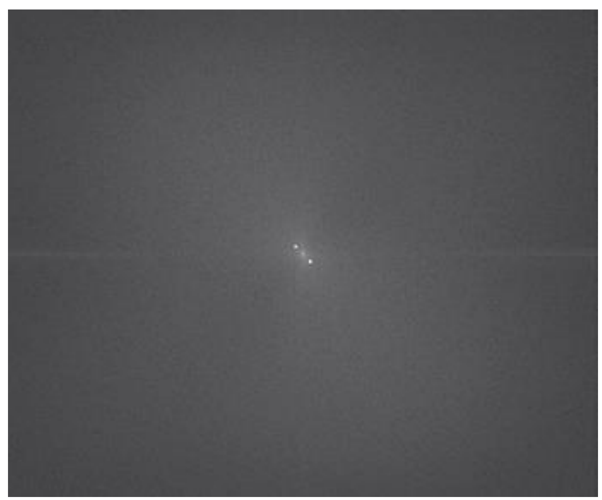


Image corrupted by additive periodic sinusoidal noise



Spectrum showing two conjugate impulses caused by the sine wave

Eliminating or reducing these impulses in the frequency domain will eliminate or reduce the sinusoidal noise in the spatial domain.

Estimating Noise Parameters

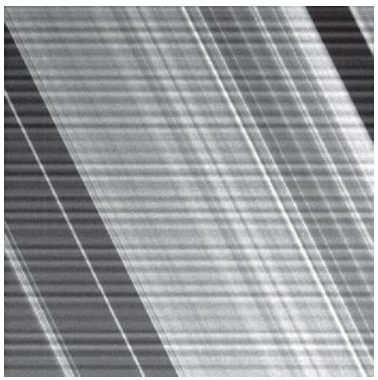
Estimating Noise Parameters

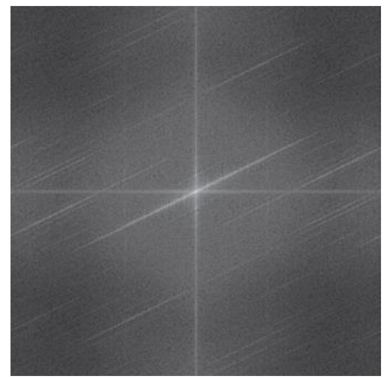
1. The parameters of ***periodic noise*** typically are estimated by inspection of the Fourier spectrum.

Notch Filtering Example

obtaining image of interference pattern

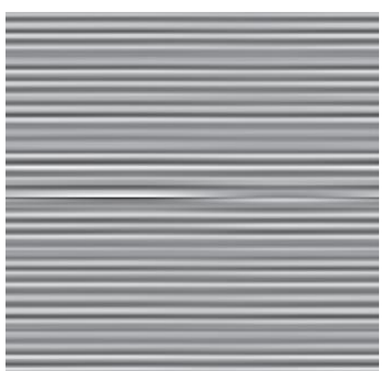
Image of Saturn rings showing periodic interference **f(x,y)**





F(u,v)

A vertical notch pass filter transfer function

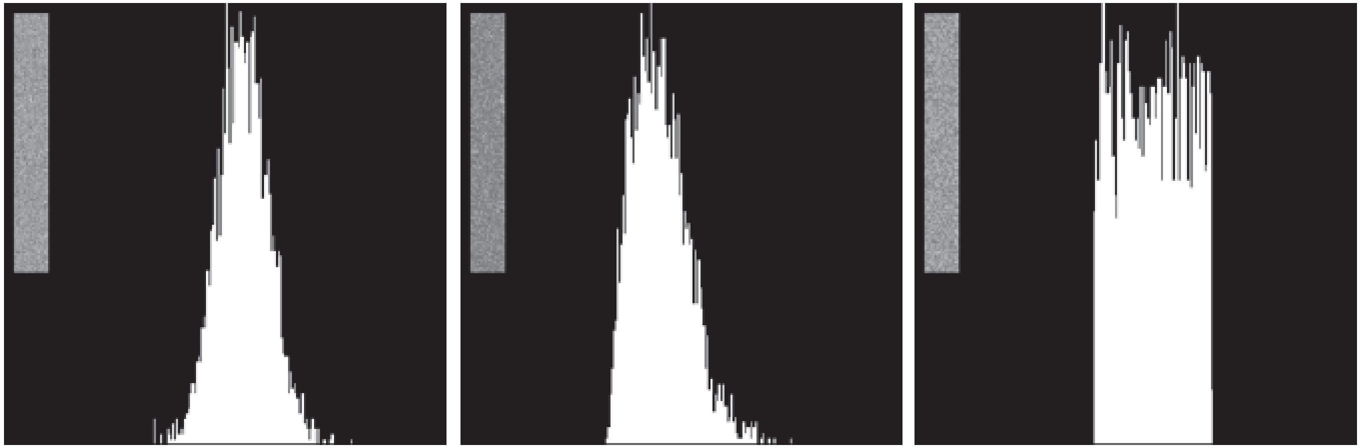


Interference pattern obtained by computing IDFT of filter result

Estimating Noise Parameters

1. The parameters of ***non-periodic noise PDFs*** may be known partially from sensor specifications, but it is often necessary to estimate them for a particular imaging arrangement.
   – By imaging a solid gray board that is illuminated uniformly and then using spatial filters to estimate the noise pattern.

3. When only the images ***already generated*** by a sensor are available, it is often possible to estimate the parameters of the PDF from small patches of reasonably constant background intensity.



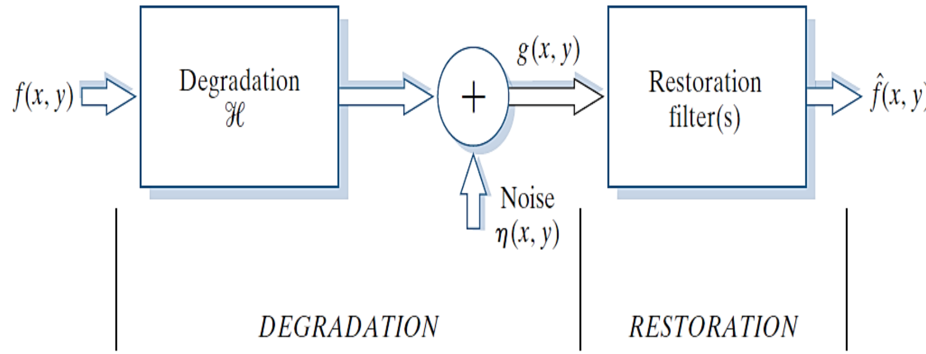
Restoration Filters with Noise Only in Spatial Domain

Presence of only Additive Noise

- If the only degradation present is the additive noise **η(x,y)**, the noise model becomes:

In spatial domain

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



In frequency domain

$$G(u,v)=F(u,v)+N(u,v)$$

- Spatial filtering is the method of choice in spatial domain for estimating ***f*(*x*, *y*)** [i.e., *denoising* image *g*(*x*, *y*)] in situations when only additive random noise is present.

Mean Filters

Arithmetic Mean Filter
(same as the Box filter)

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

$S_{x,y}$

10	4	25	0	24
15	20	21	21	23
0	25	75	24	24
25	29	67	25	27
28	30	25	75	21

Where ,

- S_{xy} represent the set of coordinates in a *neighborhood* of size $m \times n$, centered on point (x, y)
- r and c are the *row* and *column* coordinates of the pixels contained in the neighborhood S_{xy}
- $\hat{f}$ is the *restored* image

- Performs the average of local neighborhood.
- Results in a blurred version of the original image.

Geometric Mean Filter

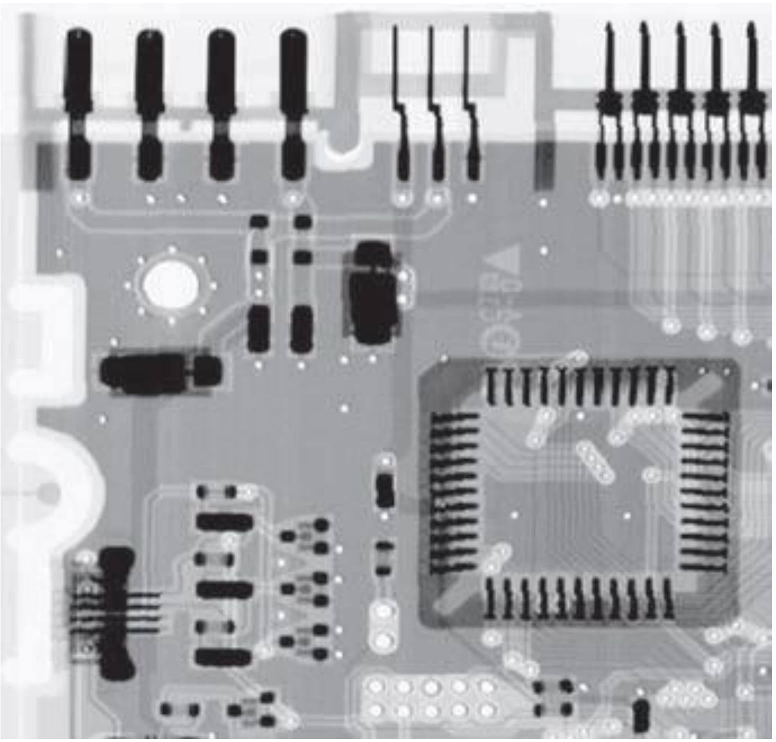
$$\hat{f}(x,y) = \left[\prod_{(r,c)\in S_{xy}} g(r,c) \right]^{\frac{1}{mn}}$$

Where ,

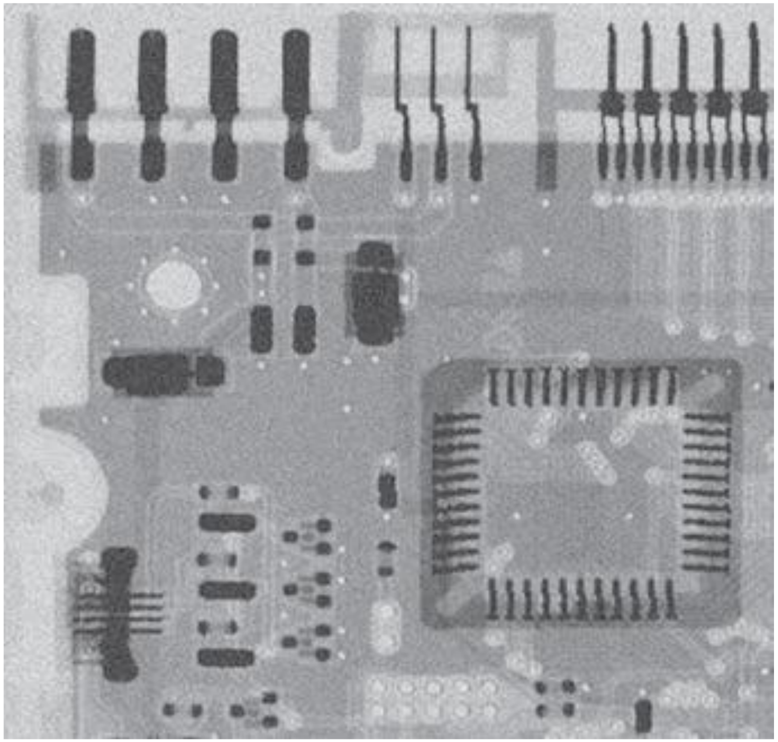
- Π indicates multiplication

- Achieves smoothing comparable to the arithmetic mean filter.
- Looses less image detail.

Arithmetic/Geometric Filter - Example

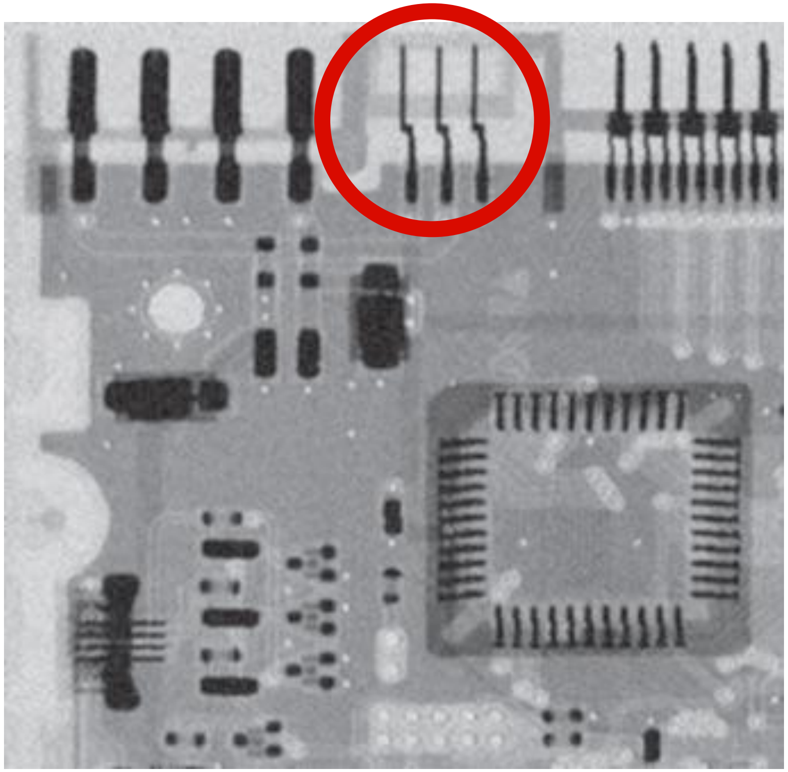


Original Image

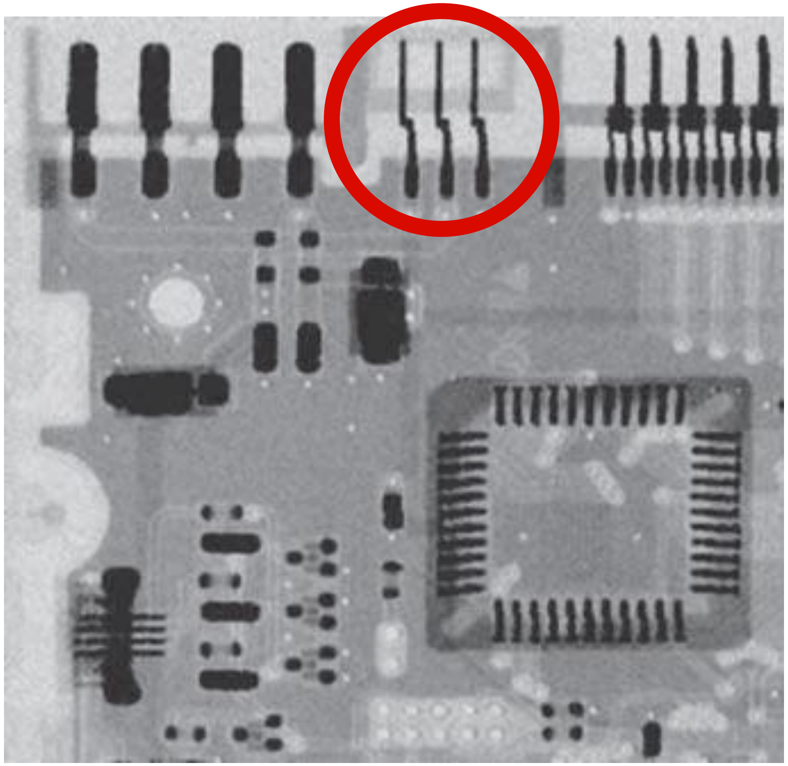


Additive Gaussian noise applied (mean=0, σ=400)

Arithmetic/Geometric Filter - Example



3×3 arithmetic mean Filter



3×3 geometric mean filter

Harmonic Mean Filters

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

- Works well for salt noise, but fails for pepper noise.

- Works well for Gaussian noise.

Contraharmonic Mean Filter

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

- **Q**: order of the filter

- Works well to reduce or virtually eliminate the effects of salt-and-pepper noise
  - **Q>0**, it eliminates pepper noise
  - **Q<0**, it eliminates salt noise
  - **Q=0**, arithmetic mean filter
  - **Q=-1**, harmonic mean filter

Contraharmonic Filter - Example

Image
+
Pepper Noise
(p=0.1)

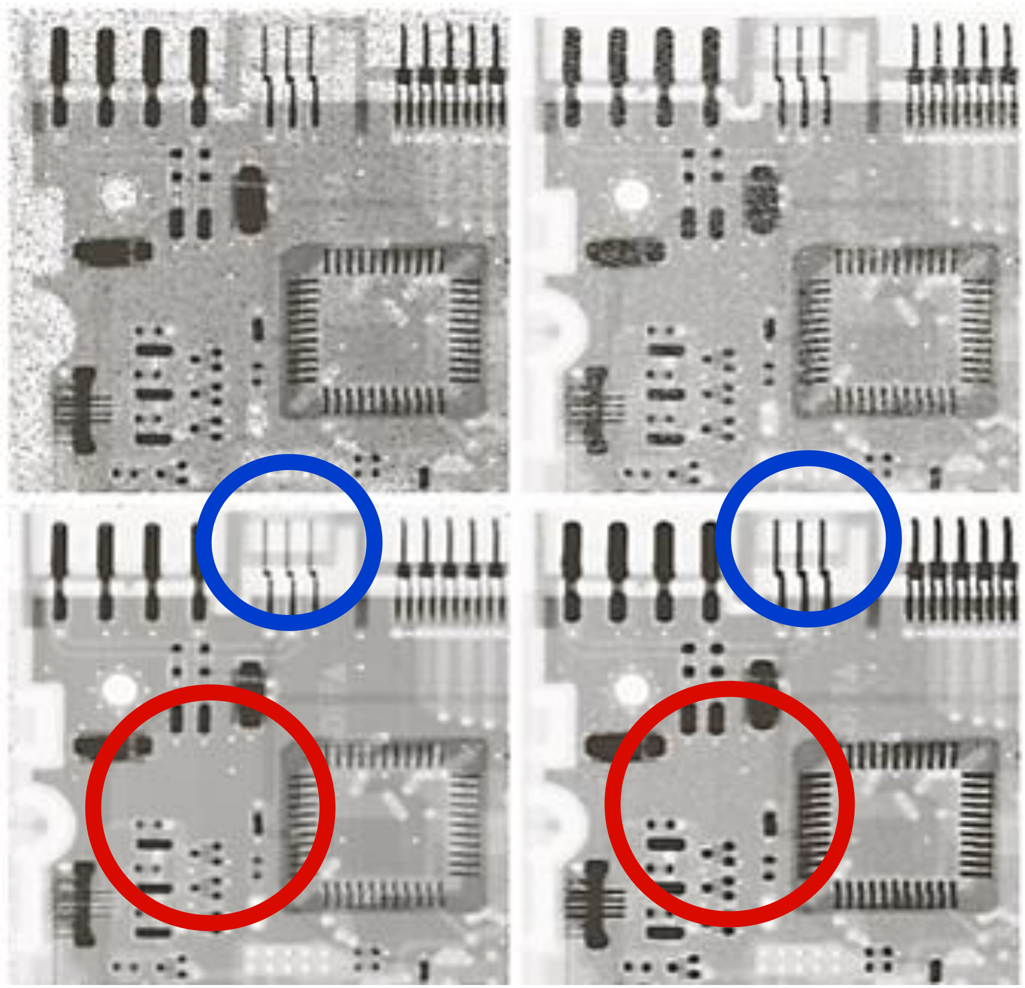


Image
+
Salt Noise
(p=0.1)

Q = 1.5

Q = -1.5

Sensitivity of the Contraharmonic Filter to Q

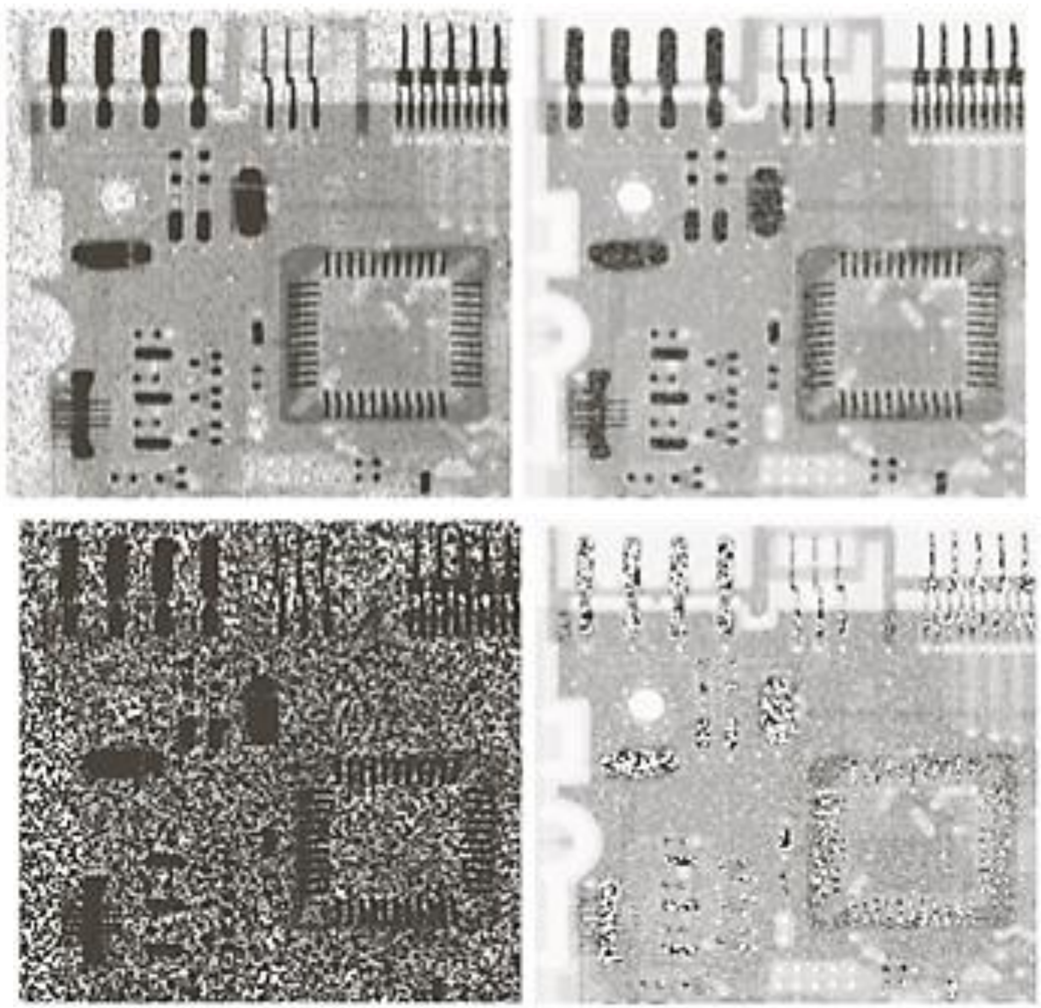


Image
+
Pepper Noise
(p=0.1)

Image
+
Salt Noise
(p=0.1)

Q = -1.5

Q = 1.5

Order-statistic Filters

Order-statistic Filters

- Order-statistic filters are spatial filters whose response is based on *ordering (ranking) the values of the pixels* contained in the *neighborhood* encompassed by the filter.

- The *ranking result* determines the *response* of the filter.

Order-statistic Filters

- Median filter - choose the median of the local neighborhood.

$$\hat{f}(x,y) = \underset{(r,c)\in S_{xy}}{\text{median}}\{g(r,c)\}$$

- Max filter - choose the maximum of the local neighborhood.
  - Used to find areas of maximum intensity in the image
  - Helpful in removing pepper noise (darker pixels)

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

- Min filter - choose minimum of the local neighborhood.
  - Used to find areas of minimum intensity in the image
  - Helpful in removing salt noise (lighter pixels)

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

Order-statistic Filters

- Midpoint filter - computes the midpoint between the maximum and minimum values in the area encompassed by the filter
  - Works well for randomly distributed noise, such as Gaussian or uniform noise.

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

Order-statistic Filters

- Alpha-trimmed filter:
  - Suppose that we delete the ***d/2*** *lowest* and the ***d/2*** *highest* intensity values of ***g*(*r*,*c*)** in the neighborhood S_{xy}
  - Let $g_R(r, c)$ represent the remaining ***mn − d*** pixels in S_{xy}
  - A filter formed by **averaging** these remaining pixels is called an *alpha-trimmed mean filter*.

$$\hat{f}(x,y) = \frac{1}{mn - d} \sum_{(r,c)\in S_{xy}} g_R(r,c)$$

- Works well for multiple types of noise, such as the combination of salt-and-pepper and Gaussian noise.

Median Filter Application

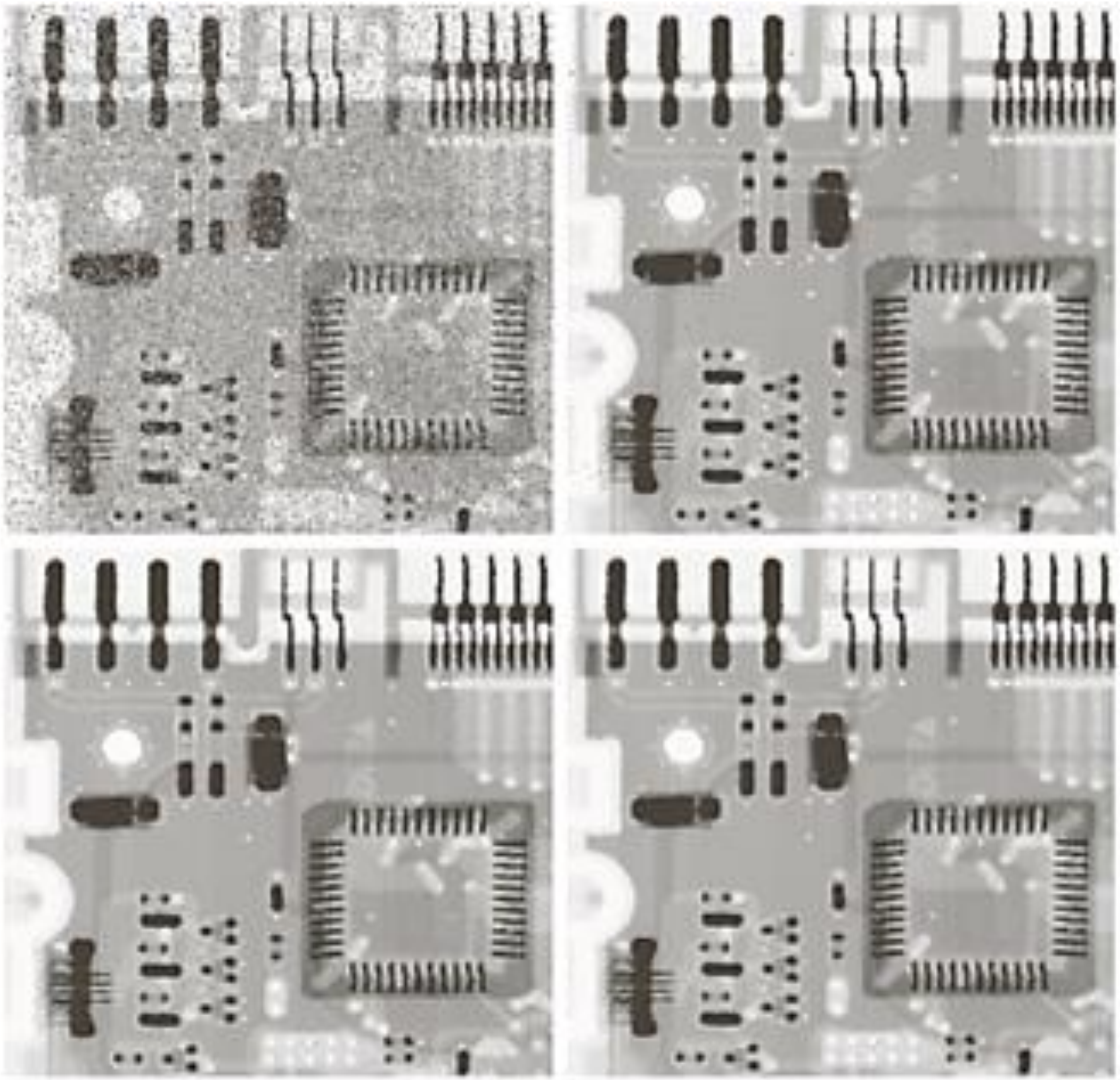


Image
+
Pepper Noise
(p=0.1)

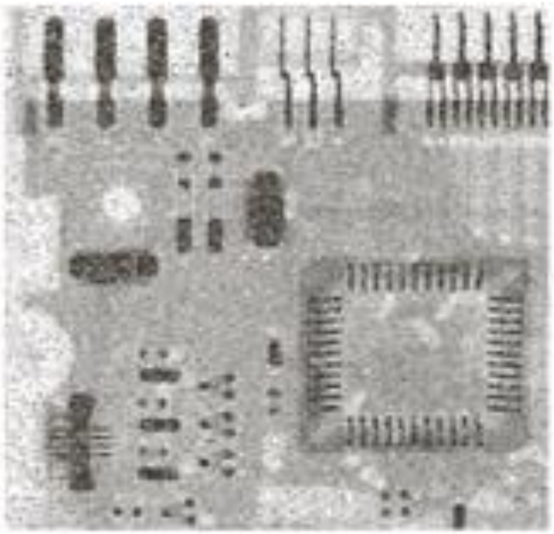
Median
Filter
1-pass

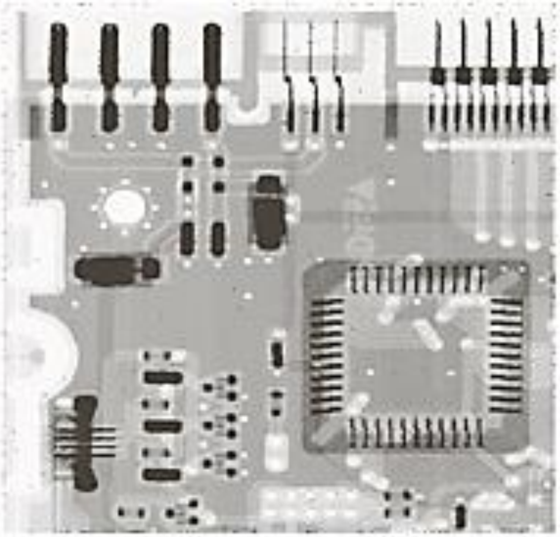
Median
Filter
2-pass

Median
Filter
3-pass

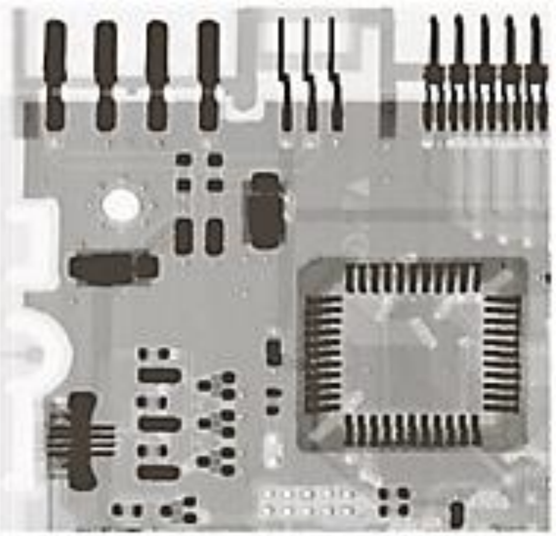
Max-Min Filtering

Image
+
Pepper Noise
(p=0.1)





3x3 Max Filter



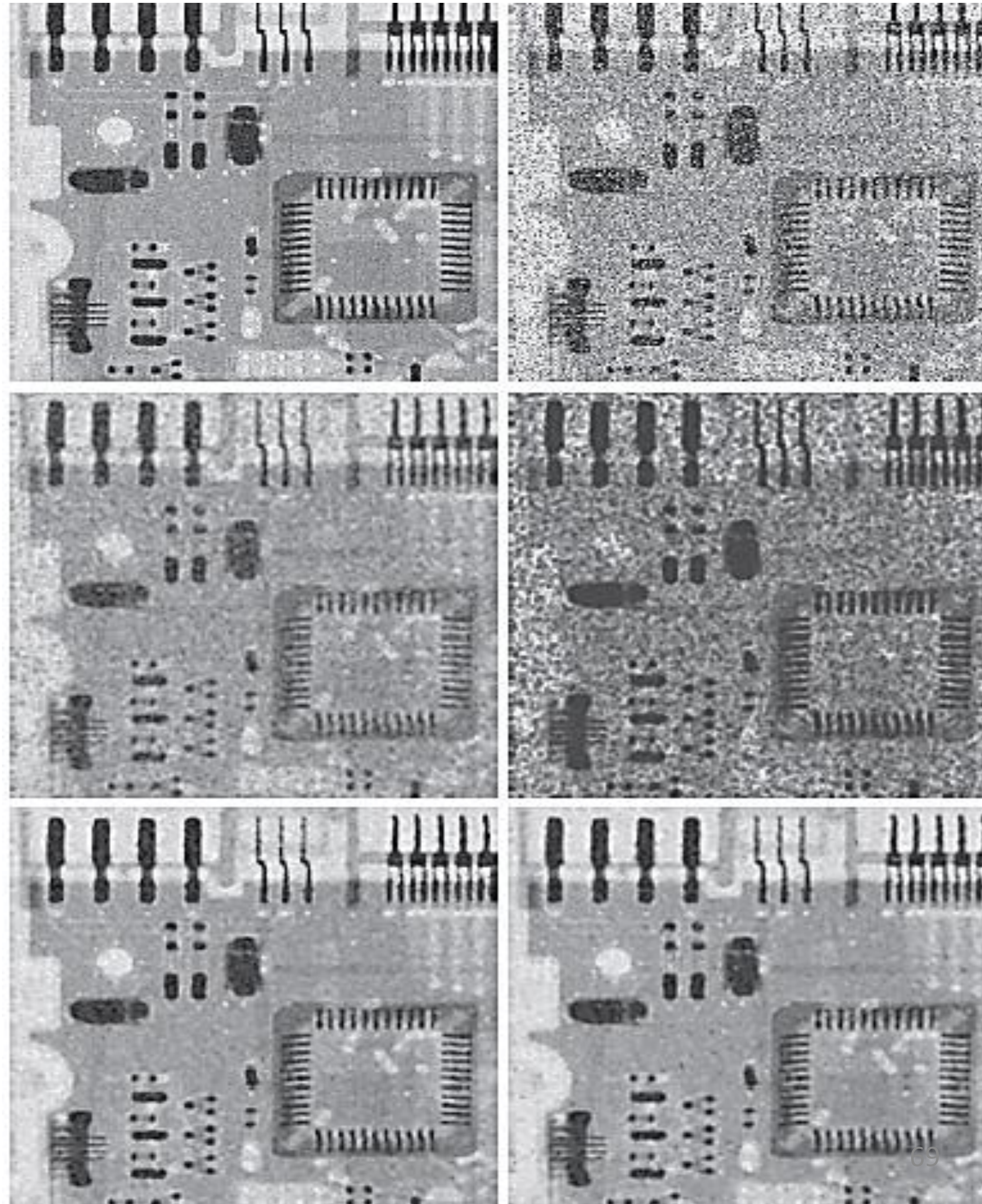
3x3 Min Filter

- Max filter removes some dark pixels from the borders of the dark objects.
- Min filter removes some white points around the border of light objects.

Spatial filtering - Example

a b
c d
e f

(a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. (c)-(f) Image (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; (f) alpha-trimmed mean filter, with $d = 6$.



Adaptive Filters

Adaptive Filters

- The filters discussed thus far are applied to an image **without regard** for *how image characteristics vary from one point to another* in the neighborhood $\boldsymbol{S_{x,y}}$.

- Adaptive filters
  - A class of filters whose behavior in a local neighborhood depends on the statistical characteristics of the neighborhood.
  - Are capable of performing superior to that of the filters discussed thus far.
  - The price paid is the increase in filter complexity.

- Adaptive, local noise reduction filter
- Adaptive median filter

Adaptive, Local Noise Reduction Filter

- Considers the statistical characteristics of a *local region* in an image.
  - Mean gives a measure of ***average intensity*** in the *local region*
  - Variance gives a measure of ***image contrast*** in the *local region*

- *Let* $\boldsymbol{S_{xy}}$,be the neighborhood centered on coordinates (x, y) on which the filter operates. The **response of the filter** at (x, y) is to be based on the following quantities:
  - $\boldsymbol{g(x, y)}$ is the value of the noisy image at (x, y)
  - $\boldsymbol{\sigma^2}$ **η** is the variance of the noise in the image
  - $\bar{z}_{S_{xy}}$ is the local average intensity of the pixels in S_{xy}
  - $\boldsymbol{\sigma^2} S_{xy}$ is the local variance of the intensities of pixels in S_{xy}

10	4	25	0	24
15	20	21	21	23
0	25	75	24	24
25	29	67	25	27
28	30	25	75	21

$S_{x,y}$

Adaptive, Local Noise Reduction Filter - Conditions of Applicability

1. If the *variance of the noise* $\sigma^2_\eta = 0$, the filter should return simply the value of ***g*** at **(*x*, *y*)**.
   – This is the trivial, zero-noise case in which *g* is equal to *f* at (*x*, *y*).

2. If the *local variance* $\sigma^2_{S_{xy}}$ is **high** relative to σ^2_η, the filter should return a value close to ***g*** at **(*x*, *y*)**.
   – A high local variance typically is associated with edges, and these should be preserved.

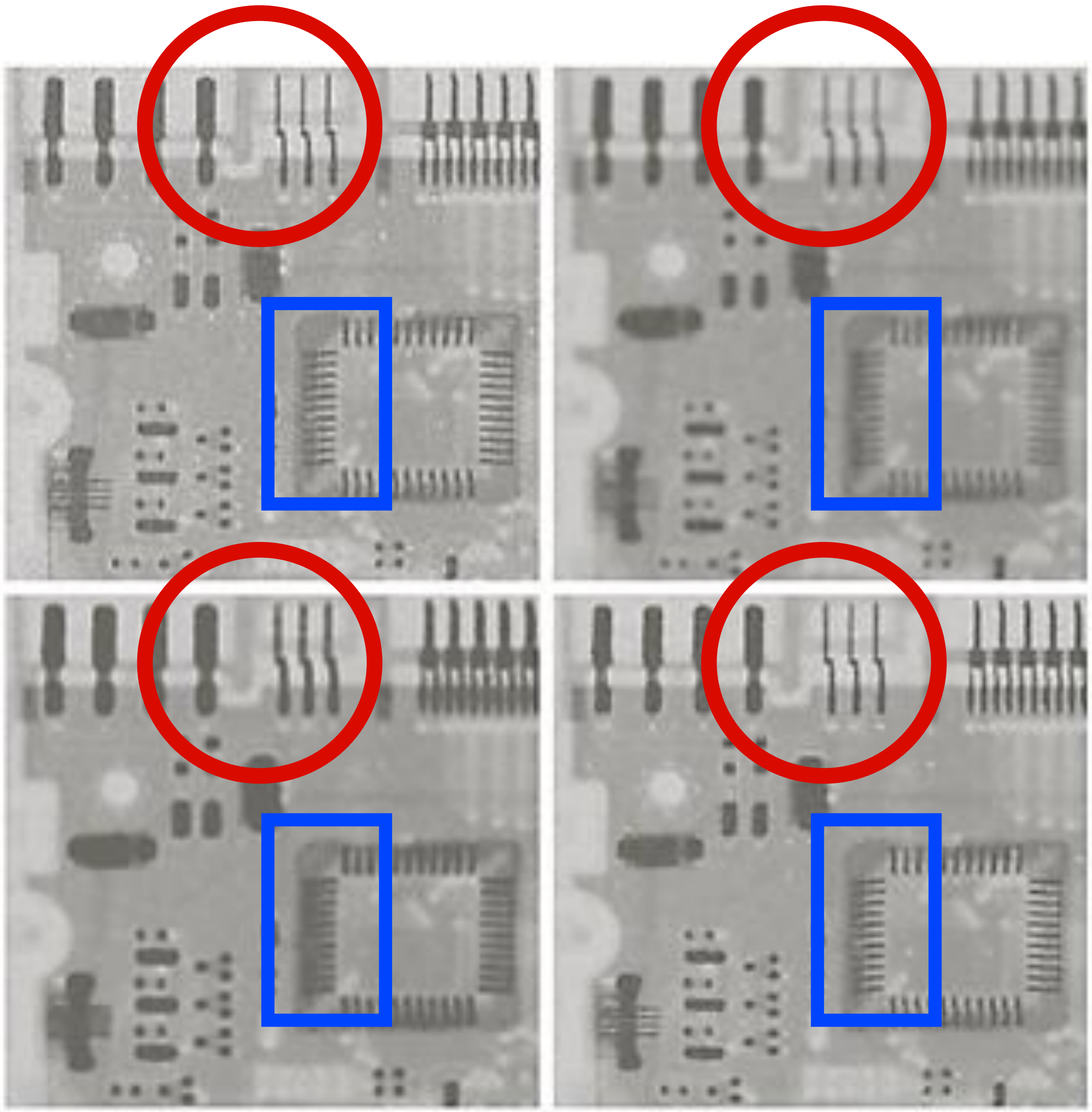
3. If the *two variances* are **equal**, we want the filter to return the arithmetic mean value of the pixels in S_{xy}.
   – This condition occurs when the local area has the same properties as the overall image, and local noise is to be reduced by averaging.

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_{S_{xy}}^2}\left[g(x,y) - \bar{z}_{S_{xy}}\right]$$

Adaptive, Local Noise Reduction Filter

Example

Image
+
Gaussian
Noise
(mean : zero
variance: 1000)



7x7 Arithmetic Mean Filtering

7x7 Geometric Mean Filtering

7x7 Adaptive Mean Filtering **(variance: 1000)**

Adaptive Median Filter

- *Traditional median filter* performs well for lower values of impulse noise (typically ≤ **0.2**)

- **Adaptive Median Filter**
  - Performs well for typically greater values of impulse noise
  - It is defined over a local neighborhood $\boldsymbol{S_{xy}}$ of pixel (x,y)
  - It changes the size of the local neighborhood **dynamically** while filtering
  - Output is still just **one value** that replaces the pixel at (x,y)
  - Preserves detail while simultaneously smoothing non-impulse noise
  - Performs significantly better than the traditional mean filter

Adaptive Median Filter

Procedure

$S_{x,y}$

10	4	25	0	24
15	20	21	21	23
0	25	75	24	24
25	29	67	25	27
28	30	25	75	21

$z_{\min}$ = minimum intensity value in S_{xy}

$z_{\max}$ = maximum intensity value in S_{xy}

z_{med} = median of intensity values in S_{xy}

z_{xy} = intensity at coordinates (x, y)

$S_{\max}$ = maximum allowed size of S_{xy}

Adaptive Median Filter

Procedure

- The adaptive median-filtering algorithm uses **two processing levels.**

Level A: If $z_{\min} < z_{\text{med}} < z_{\max}$, go to Level B
Else, increase the size of S_{xy}
If $S_{xy} \leq S_{\max}$, repeat level A
Else, output z_{med}.

Level B: If $z_{\min} < z_{xy} < z_{\max}$, output z_{xy}
Else output z_{med}.

$z_{\min}$ = minimum intensity value in S_{xy}
$z_{\max}$ = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity at coordinates (x, y)
$S_{\max}$ = maximum allowed size of S_{xy}

Why does Adaptive Median Filter work?

Level-A determines whether the output of the median filter **Z_{med}**, is an impulse or not.

- If **Z_{med}** lies in the range **(Z_{min}, Z_{max})**, then it cannot be an impulse.
- In this case, goto Level-B and check if pixel at **Z(x,y)** is itself impulse noise.

Level A : If $z_{\text{min}} < z_{\text{med}} < z_{\text{max}}$, go to Level B
Else, increase the size of S_{xy}
If $S_{xy} \leq S_{\text{max}}$, repeat level A
Else, output z_{med}.

Level B : If $z_{\text{min}} < z_{xy} < z_{\text{max}}$, output z_{xy}
Else output z_{med}.

z_{min} = minimum intensity value in S_{xy}
z_{max} = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity at coordinates (x, y)
S_{max} = maximum allowed size of S_{xy}

Note that there is no guarantee that this value is not an impulse.

Why does Adaptive Median Filter work?

In **Level-B**:

- ❑ If **Z(x,y)** is in the range **(Z_{min}, Z_{max})**, output the unchanged pixel value (resulting in minimal distortion).
- ❑ If **Z(x,y)** is not in the range **(Z_{min}, Z_{max})**, it is an extreme value, so output the median value.

Level A :	If $z_{\min} < z_{\text{med}} < z_{\max}$, go to Level B
	Else, increase the size of S_{xy}
	If $S_{xy} \le S_{\max}$, repeat level A
	Else, output z_{med}.
Level B :	If $z_{\min} < z_{xy} < z_{\max}$, output z_{xy}
	Else output z_{med}.

$z_{\min}$ = minimum intensity value in S_{xy}
$z_{\max}$ = maximum intensity value in S_{xy}
z_{med} = median of intensity values in S_{xy}
z_{xy} = intensity at coordinates (x, y)
$S_{\max}$ = maximum allowed size of S_{xy}

Why does Adaptive Median Filter work?

- This algorithm has **three main objectives** to fulfill:
  - to remove salt-and-pepper noise (impulse noise)
  - to provide smoothing of other types of noise
  - to reduce distortion (excessive thickening and thinning) of object boundaries

- $\mathbf{Z_{min}}$ and $\mathbf{Z_{max}}$ act as impulse-like components for the *neighborhood*, instead of being the **max** and **min** for the *entire image*.

Adaptive Median Filter

Example

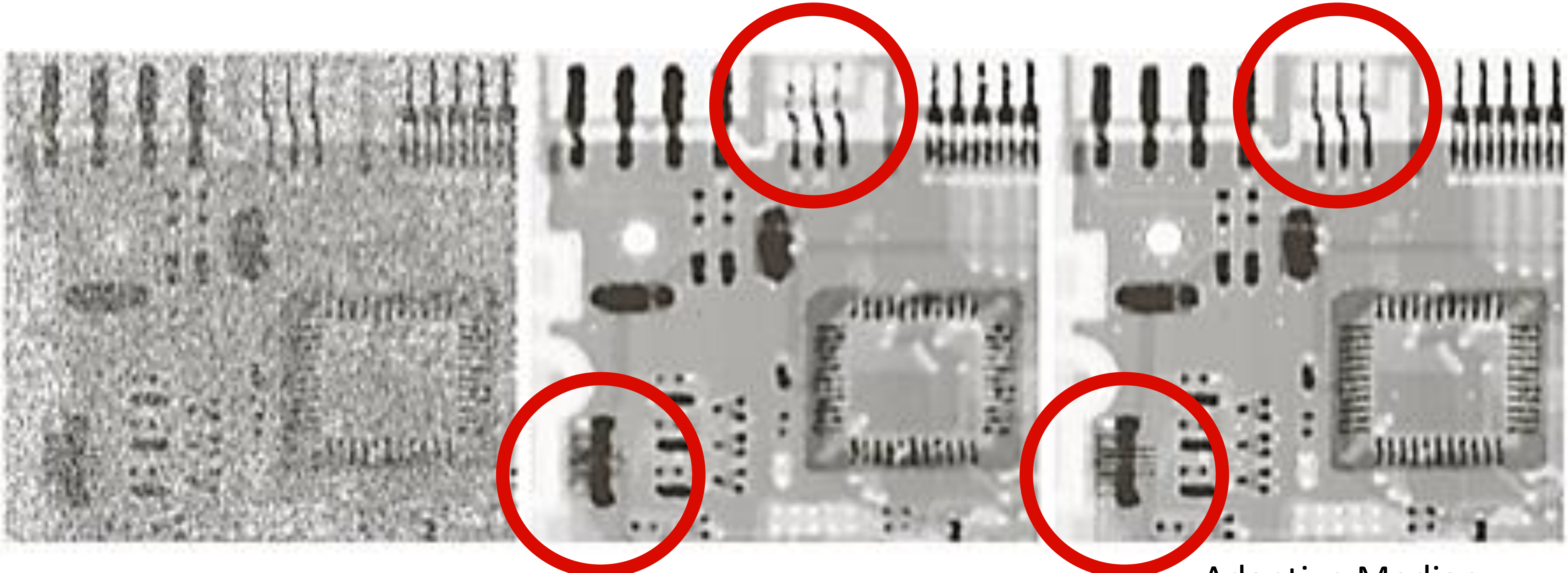


Image + S-P Noise
Ps = Pp = 0.25

7x7 Median Filter

Adaptive Median Filter with $\mathbf{S_{max} = 7}$

Next Lecture

- Periodic noise reduction using frequency domain filtering
  - Notch filtering
  - Optimum notch filtering (self study)
- Linear, position-invariant degradations
- Estimating degradation function (H)