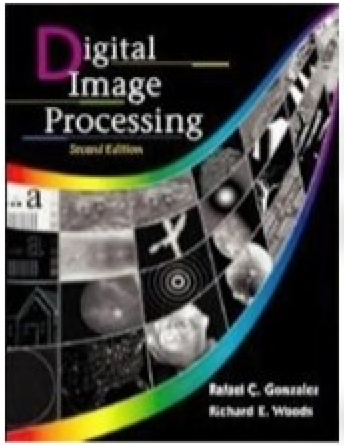
Chapter 4

Image Restoration

REFERENCES



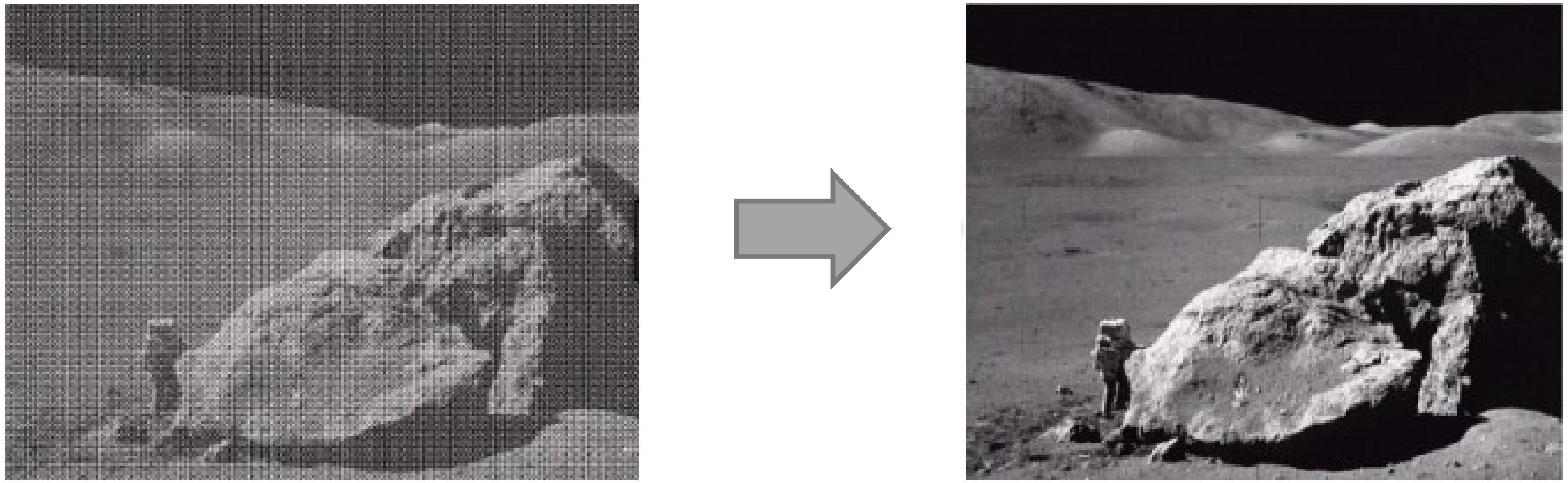
"Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002

Much of the material that follows is taken from this book

Slides by Brian Mac Namee
Brian.MacNamee@comp.dit.ie

Image Restoration

- Image restoration attempts to restore images that have been degraded
- Process to reconstruct or recover degraded image by a priori knowledge of degradation phenomena
- **Approach**
  - ✓Identify the degradation process and attempt to reverse it.
  - ✓Almost Similar to image enhancement, but more objective
- **Spatial Domain Restoration** : Applicable when degradations involves only additive noise
- **Frequency Domain Restoration**: Often used for the degradation like image blur



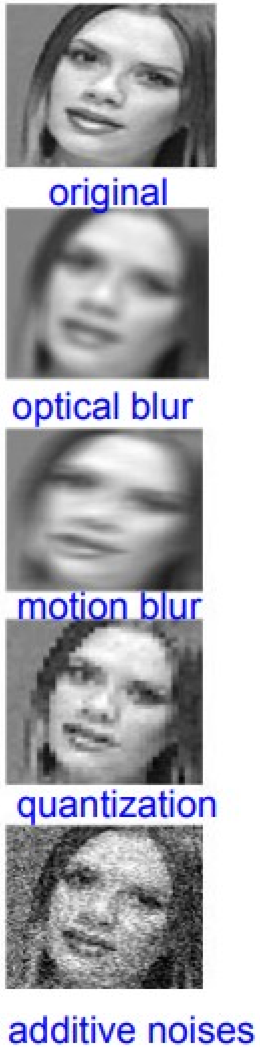
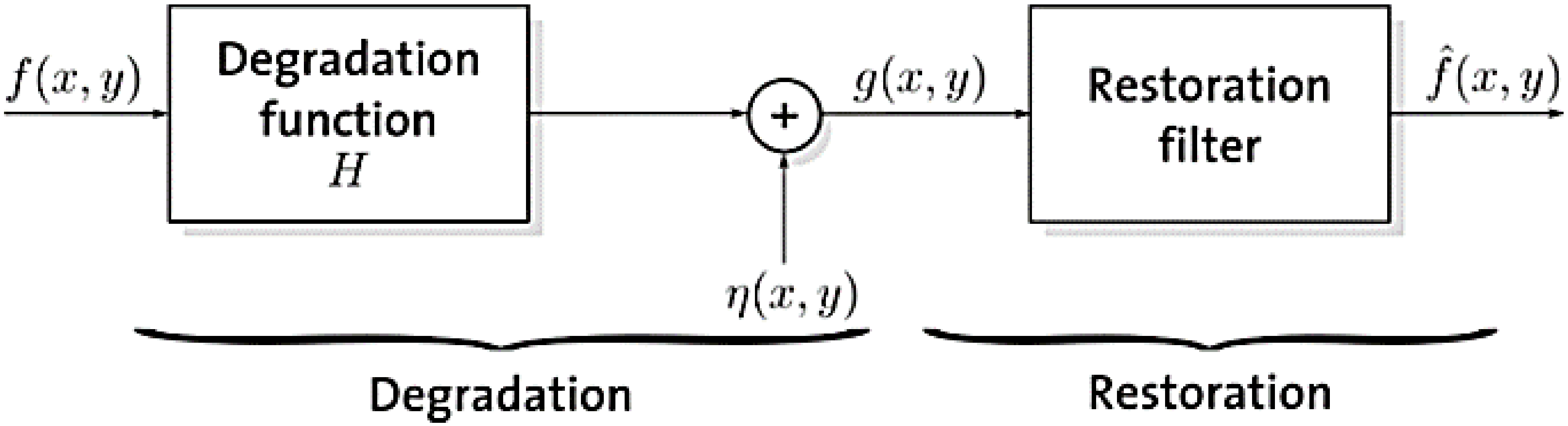


Image Restoration Vs Image Enhancement

Image Restoration	Image Enhancement
objective process	subjective process
formulate a criterion of goodness that will yield an optimal estimate of the desired result	involves heuristic procedures and designed to manipulate an image in order to satisfy the human visual system
Techniques include noise remove and deblurring (remove image blur)	Techniques include contrast stretching

Like enhancement techniques, restoration techniques can be performed in the spatial domain and frequency domain. For example, noise removal is applicable using spatial domain filters whereas deblurring is performed using frequency domain filters.

Image Degradation And Restoration Model



$g(x, y)$ = degraded image

$f(x, y)$ = input or original image

$\widehat{f}(x, y)$ = recovered or restored image

$\eta(x, y)$ = additive noise term

Image Degradation And Restoration Model

Degradation:

- Degradation function H
- Additive Noise
- Spatial Domain

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

- Frequency Domain

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

Restoration:

$$g(x,y) \Rightarrow \text{Restoration Filter} \Rightarrow \hat{f}(x,y)$$

Sources of Noise

Principal sources of noise in digital images

- Image Acquisition(digitization)
  - Imaging sensors can be affected by environmental conditions
  - Quality of sensor
- Transmission
  - Interferences can be added to an image during transmission
- Assumption of noise models
  - Noise is independent of spatial co-ordinates
  - Noise is uncorrelated with respect to the image itself

Noise

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

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

where

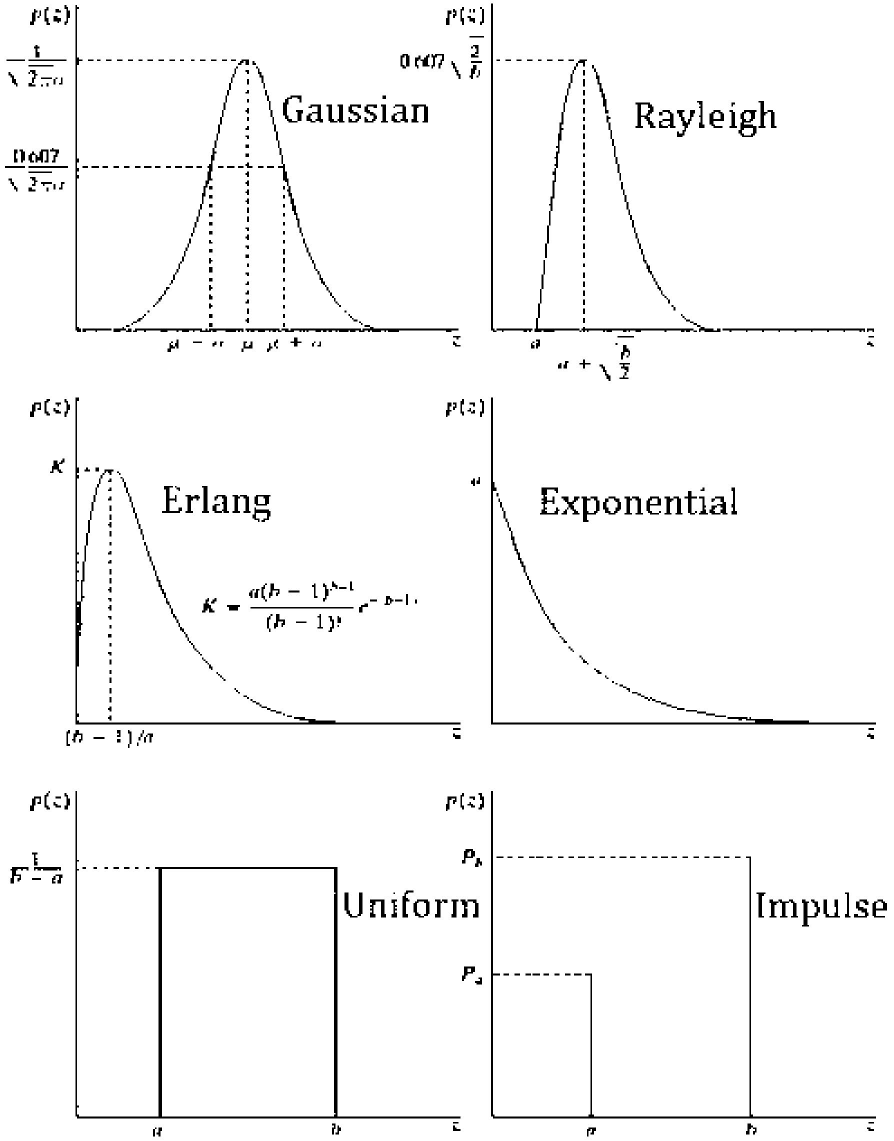
- $f(x, y)$ is the original image
- $\eta(x, y)$ is the noise term
- $g(x, y)$ is the resulting noisy pixel

If we can estimate the model, the noise in an image is based on, we can figure out how to restore the image

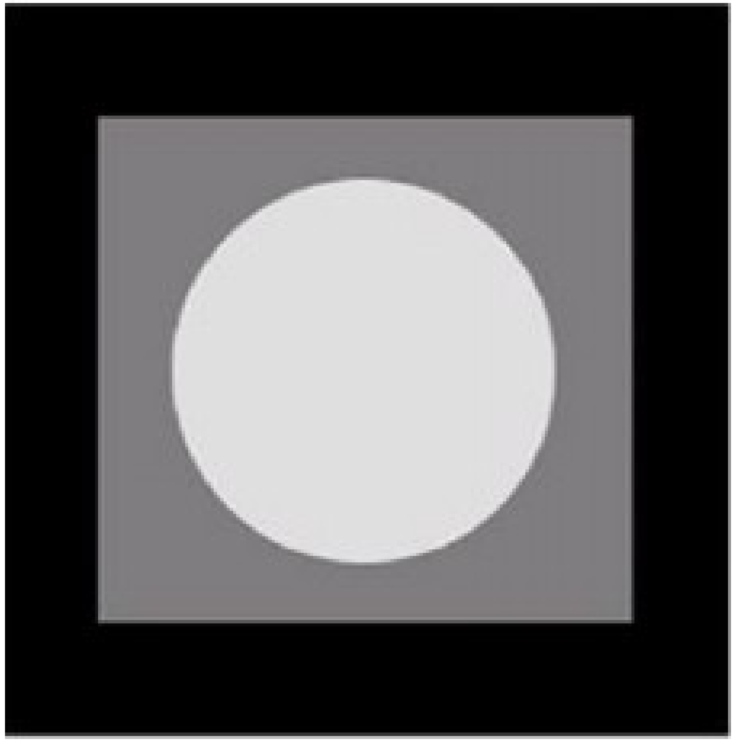
Noise Model

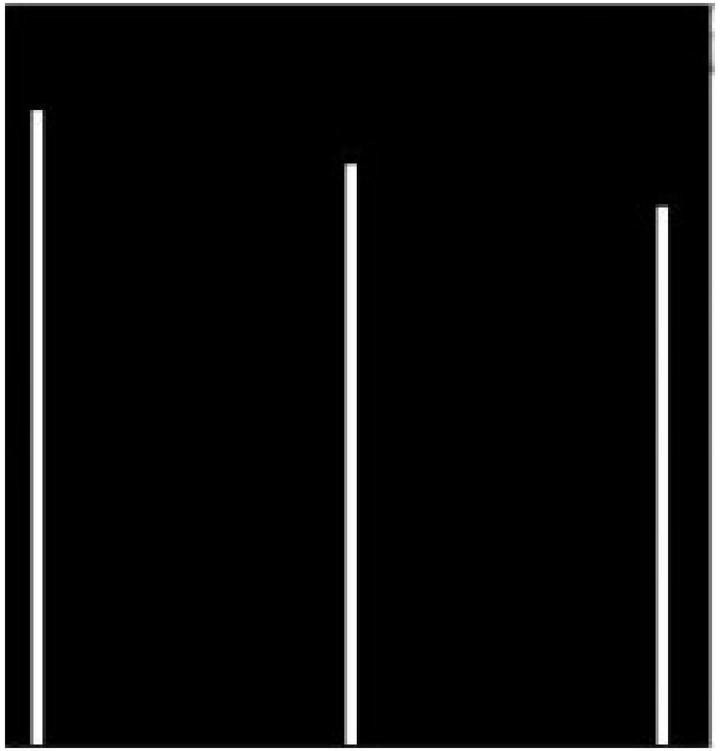
Different models for the image noise term η(x, y)

- ❑ Gaussian
- ❑ Rayleigh
- ❑ Erlang or Gamma
- ❑ Exponential
- ❑ Uniform
- ❑ Impulse

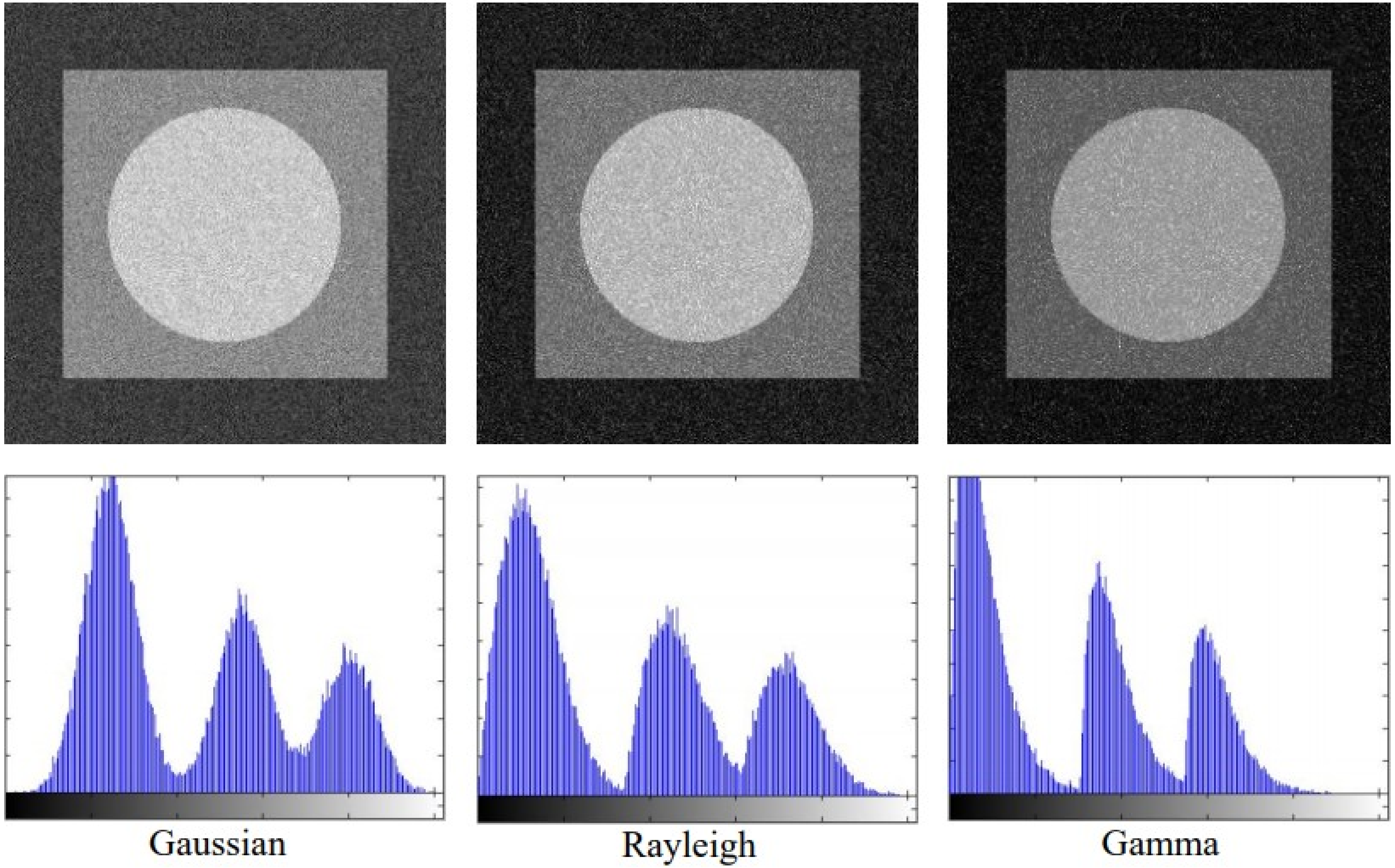


Noise Model Example

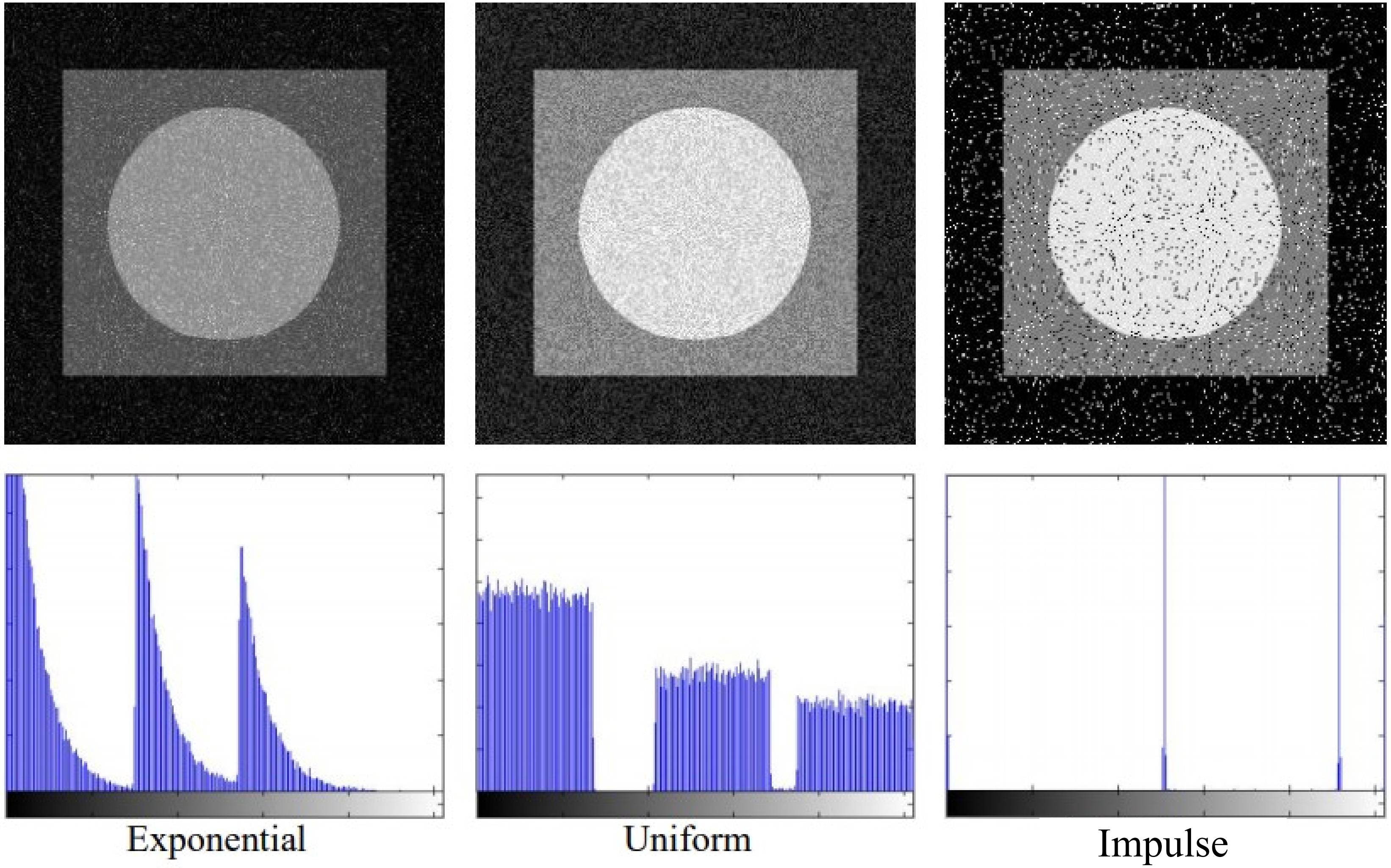




Noise Model Example



Noise Model Example



Pepper and Salt Noise

Salt-and-pepper noise is a form of noise sometimes seen on images. It is also known as impulse noise. This noise can be caused by sharp and sudden disturbances in the image signal. It presents itself as sparsely occurring white and black pixels.

Application of Noise Models

- Gaussian → Electronic Circuit Noise and Sensor Noise due to poor illumination or high temperature
- Rayleigh → Characterize noise phenomena in ranging image
- Exponential and Gamma → Laser Imaging
- Uniform → The least descriptive of practical situations
- Impulse → Occurs when faulty switching take place during imaging

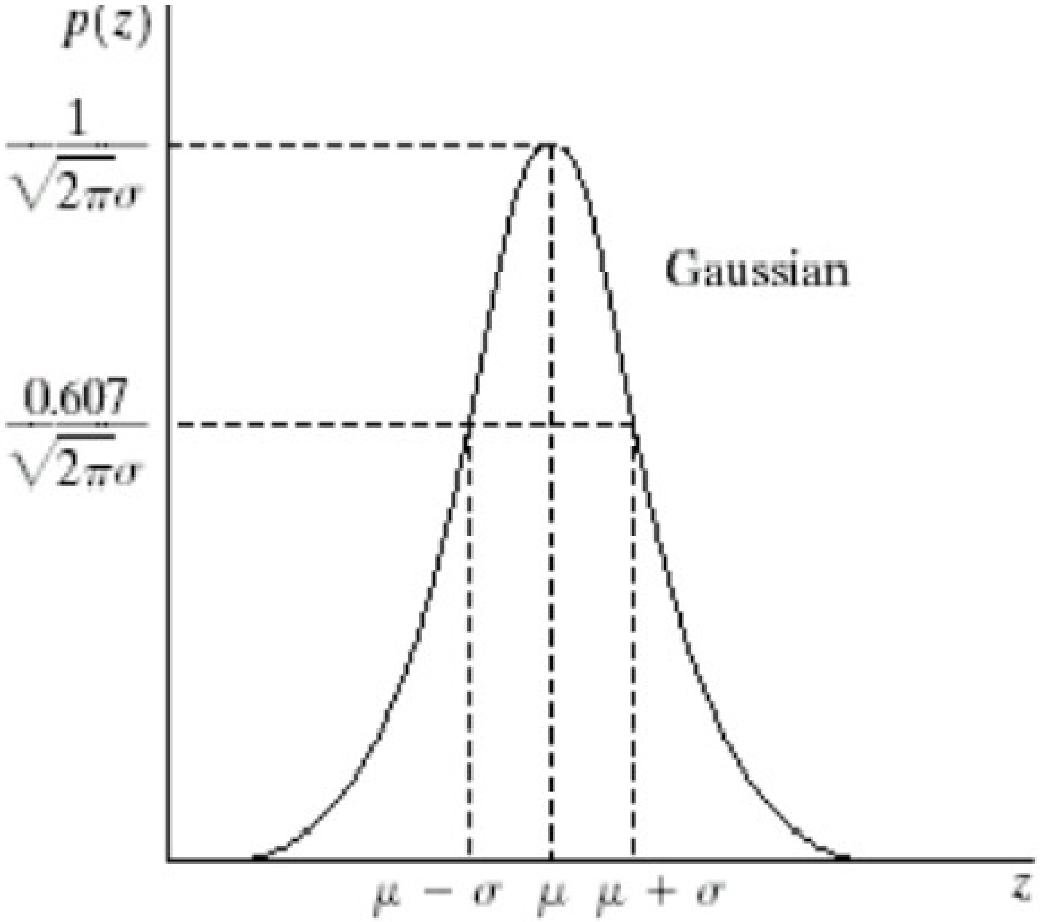
Gaussian Noise

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

z: gray level

μ: mean of random variable z

σ^2: variance of z

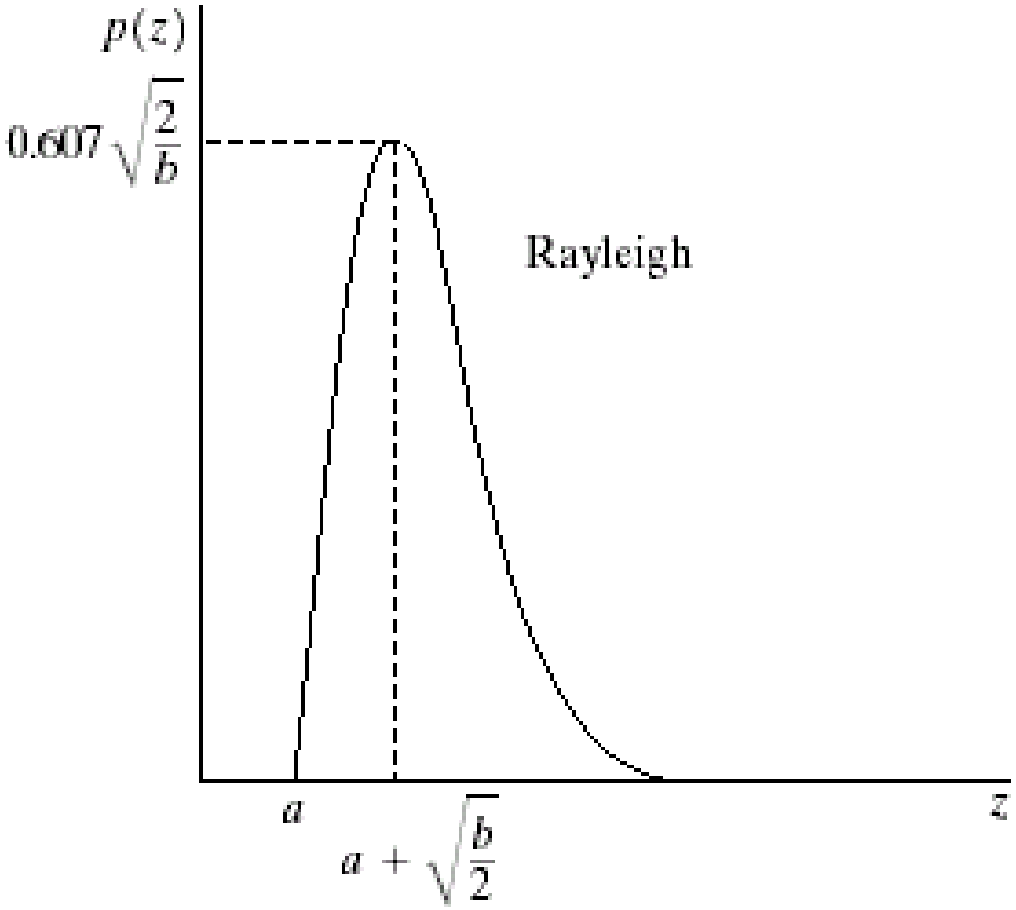


Rayleigh Noise

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

$$\text{mean}: \mu = a + \sqrt{\frac{\pi b}{4}}$$

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



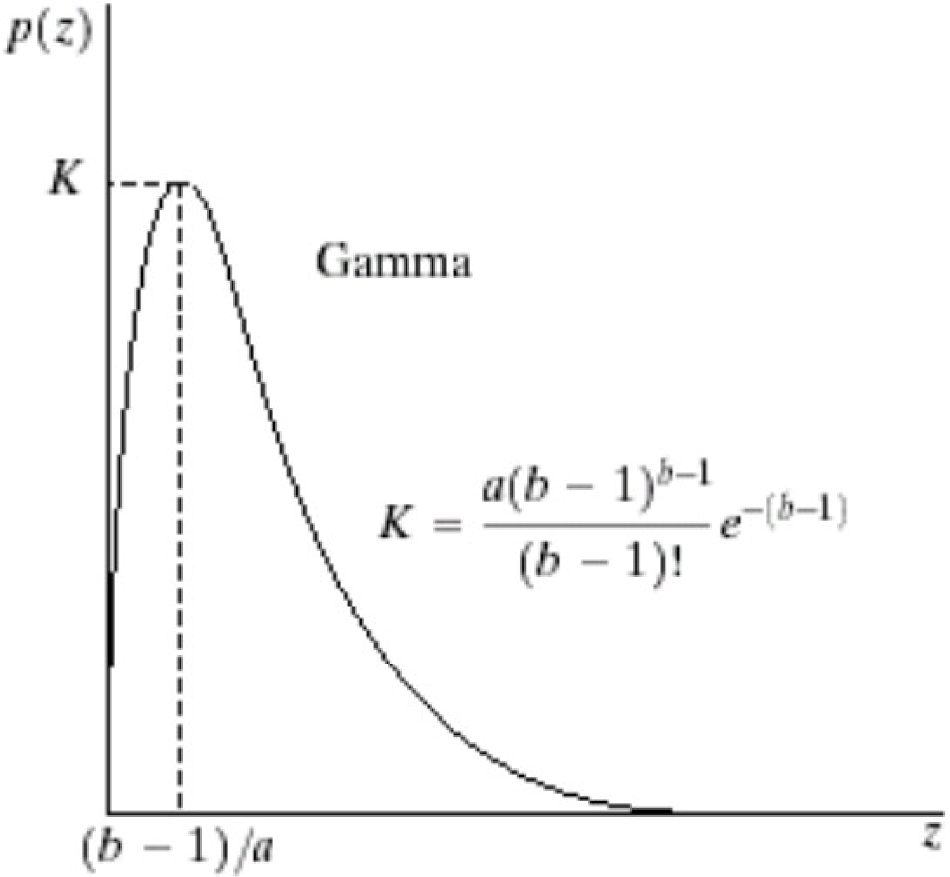
Erlang(Gamma) Noise

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

$$a > 0, b \in I+$$

$$\text{mean}: \mu = \frac{b}{a}$$

$$\text{variance}: \sigma^2 = \frac{b}{a^2}$$



Exponential Noise

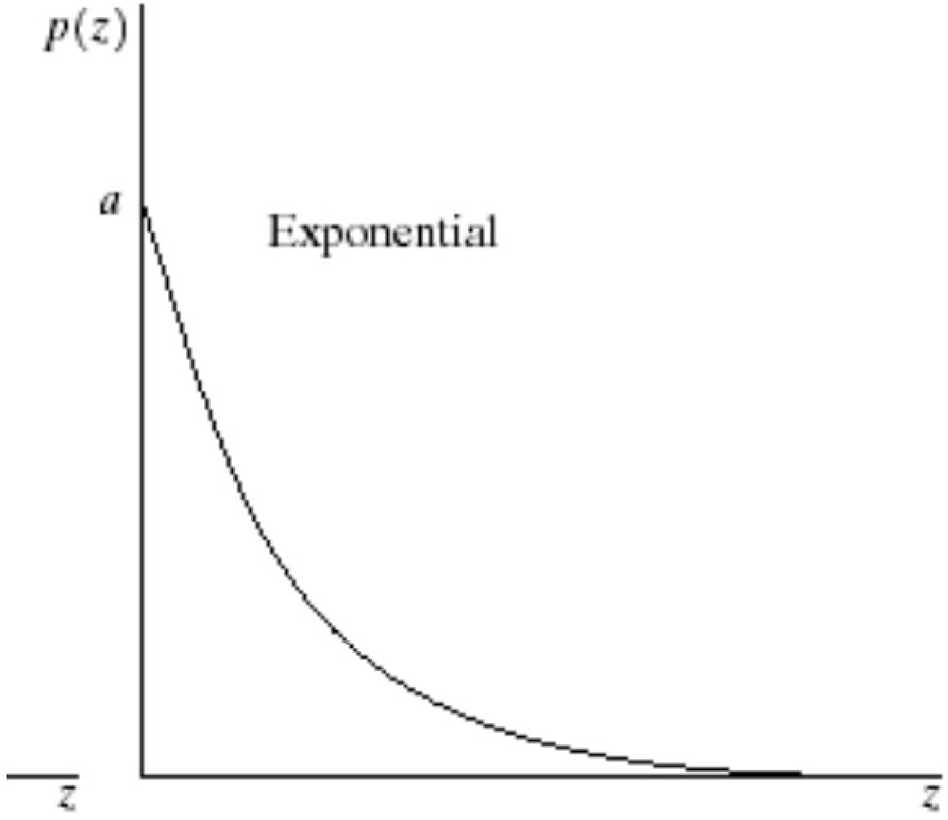
Special Case of Erlang when b=1

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

Where $a > 0,$

$$\text{mean}: \mu = \frac{1}{a}$$

$$\text{variance}: \sigma^2 = \frac{1}{a^2}$$

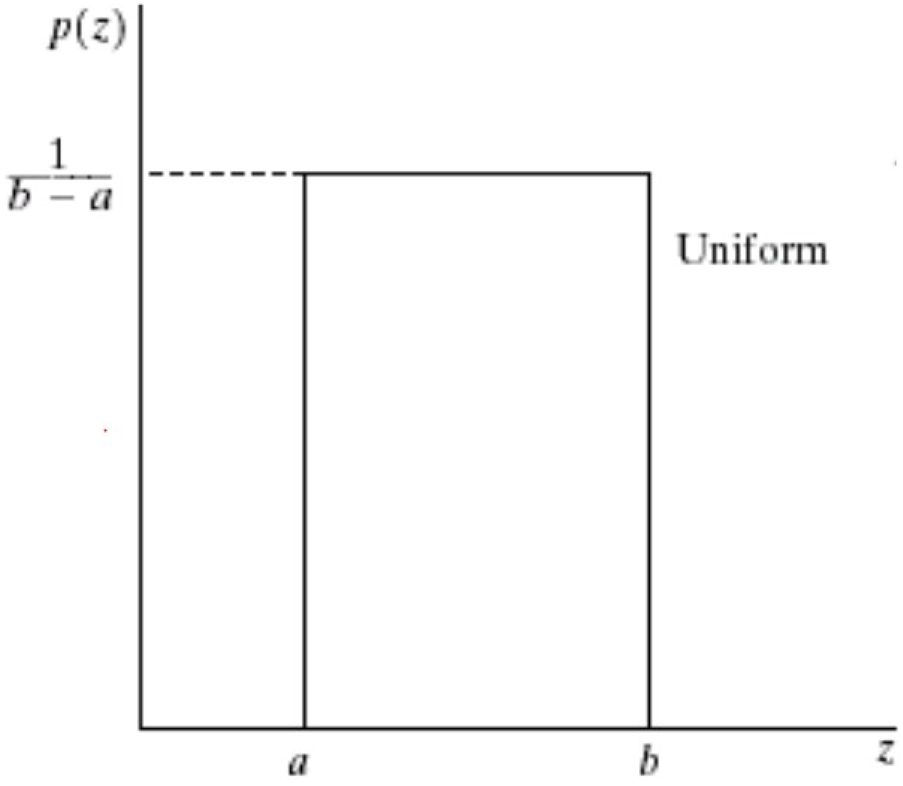


Uniform Noise

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

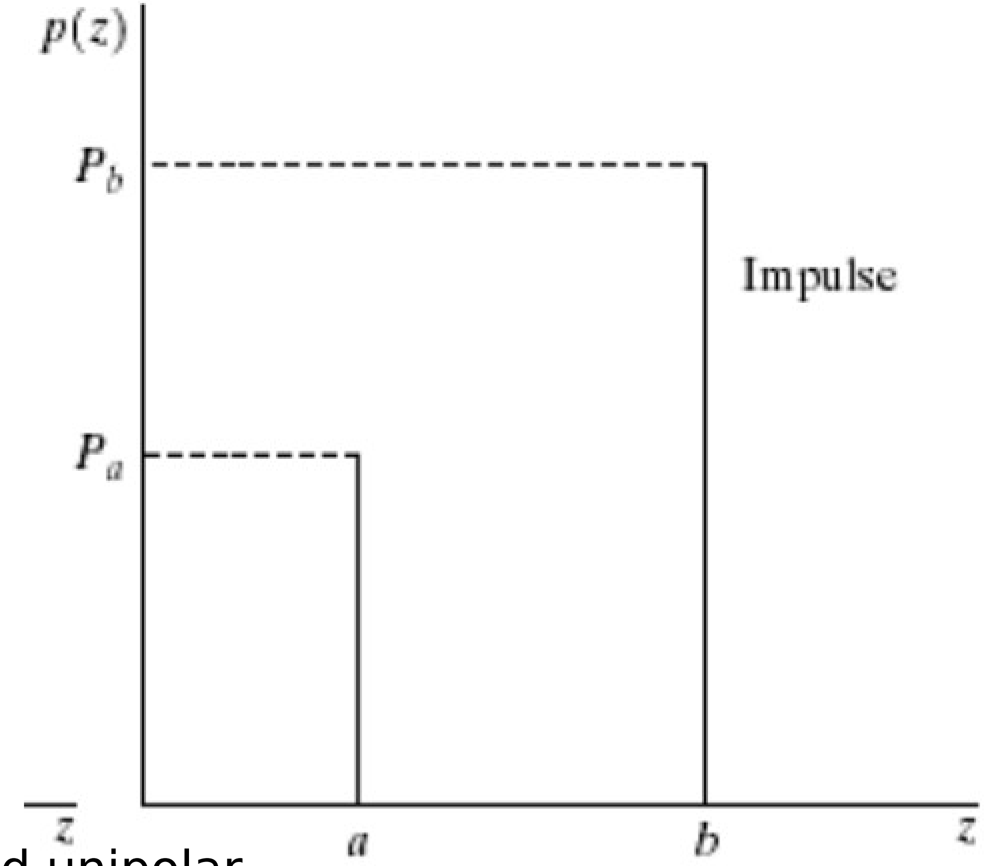
The mean and variance are given by

$$\mu = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}$$



Impulse Noise

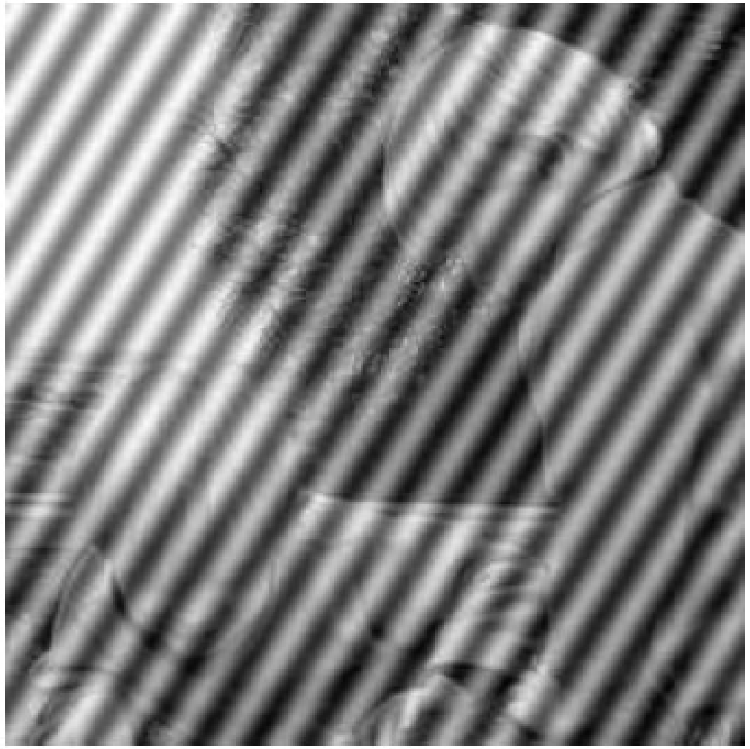
$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$



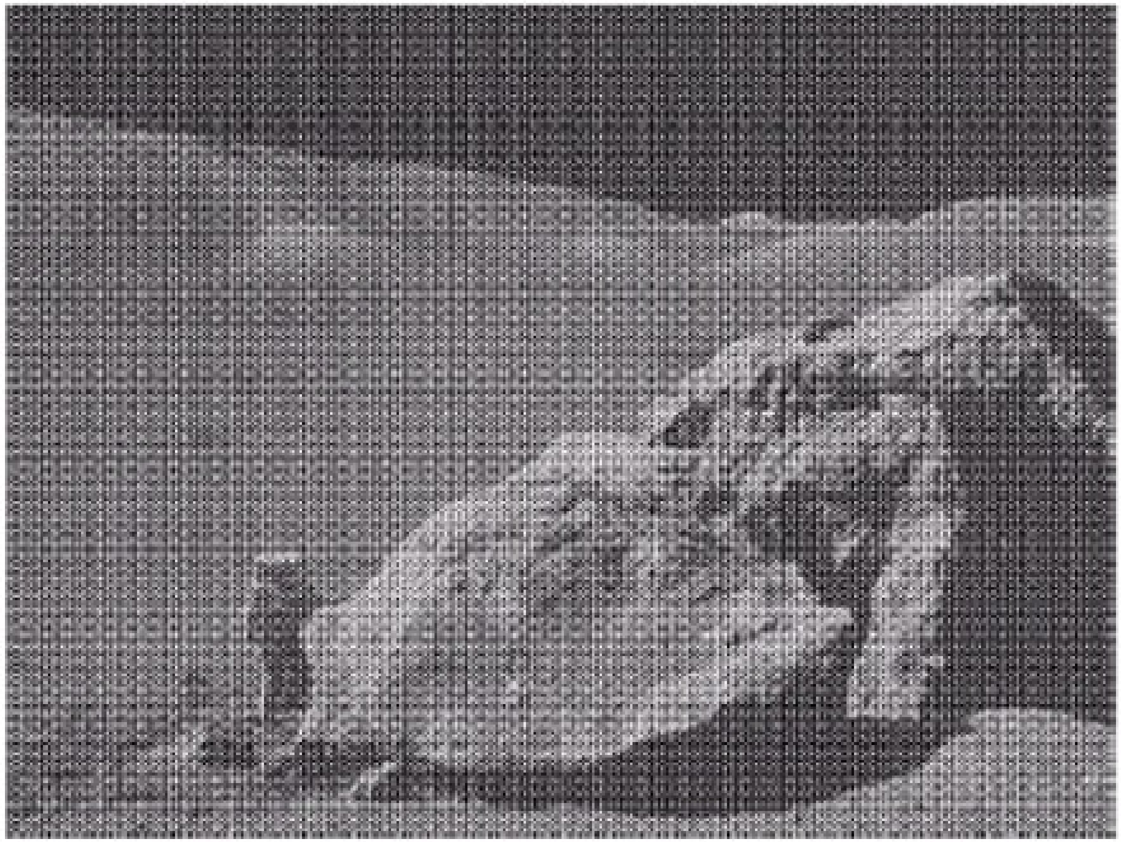
If either Pa or Pb is zero, the impulse noise is called unipolar
a and b usually are extreme values because impulse corruption is usually large compared with the strength of the image signal

Periodic Noise

- Arises typically from electrical or electromechanical interference during image acquisition
- It can be observed by visual inspection both in the spatial domain and frequency domain
- The only spatially dependent noise will be considered



Periodic Noise



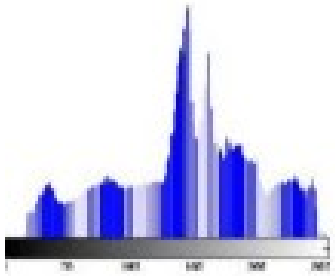
Estimation of Noise Parameter

- **Inspection of Fourier spectrum** – Periodic Noise
- **If Imaging system is available**, study characteristics of system noise by acquiring a set of images of flat environment under uniform illumination (Constant Background)
- **If only images are available** , estimate noise pdf from small patches of reasonably constant gray level
- Once the PDF is determined, estimate model parameters like mean and variable

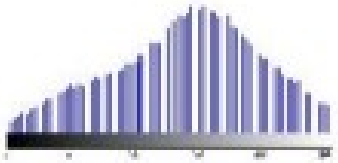
Estimation of Noise Parameter

Original Image

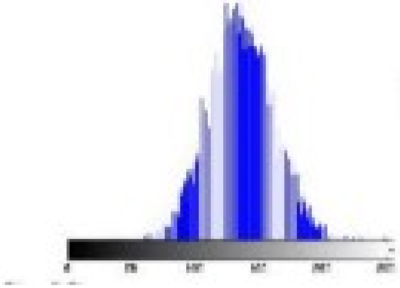
Noisy Image (Rectangle Indicates the selected Region)



Histogram of Original Image



Histogram of Noisy Image



Histogram of Selected Region

The histogram of selected Region Indicates there is gaussian type of Noise

Image Restoration Filters
Spatial and Adaptive

- **Mean filters**
  - Arithmetic mean filter
  - Geometric mean filter
  - Harmonic mean filter
  - Contra-harmonic mean filter
- **Order statistics filters**
  - Median filter
  - Max and min filters
  - Mid-point filter
  - alpha-trimmed filter
- **Adaptive filters**
  - Adaptive median filter

- Arithmetic Mean Filter

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

This is implemented as the simple smoothing filter Blurs the image to remove noise.

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

Image Restoration Filters
Mean Filter

Arithmetic Mean Filter

Original Image

54	52	57	55	56	52	51
50	49	51	50	52	53	58
51	204	52	52	0	57	60
48	50	51	49	53	59	63
49	51	52	55	58	64	67
50	54	57	60	63	67	70
51	55	59	62	65	69	72

Image $f(x, y)$

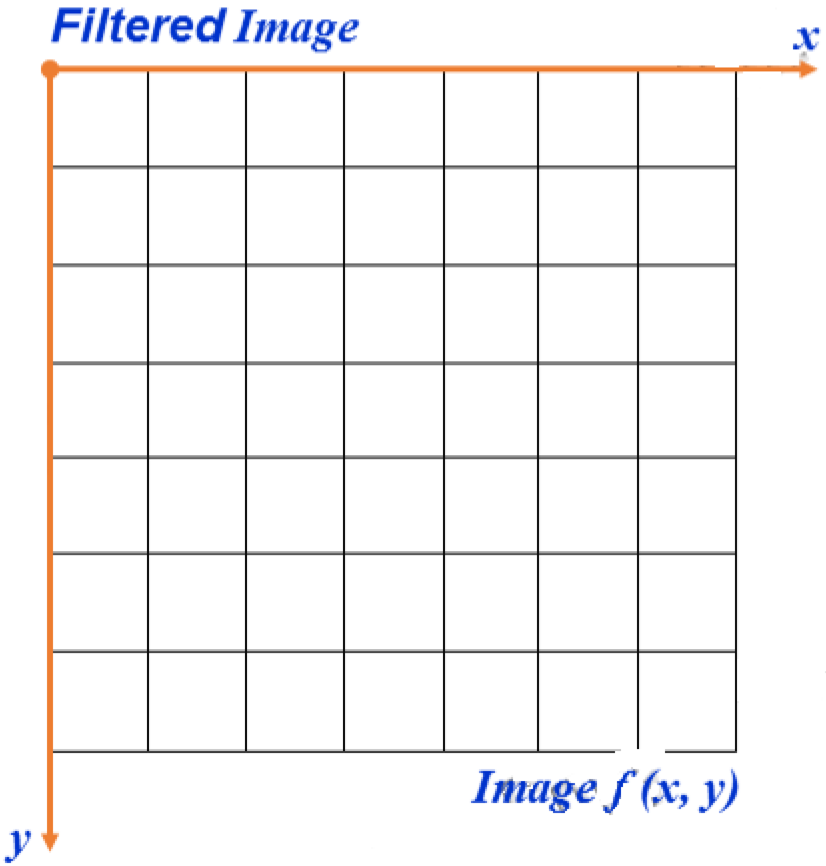


Image Restoration Filters
Mean Filter

Geometric Mean Filter

Geometric Mean Formula

$$\bar{X}_{geom} = \sqrt[n]{\prod_{i=1}^{n} x_i} = \sqrt[n]{x_1 . x_1 .... x_n}$$

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

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

- Harmonic Mean Filter

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

Satisfactory result in other kinds of noise such as Gaussian noise

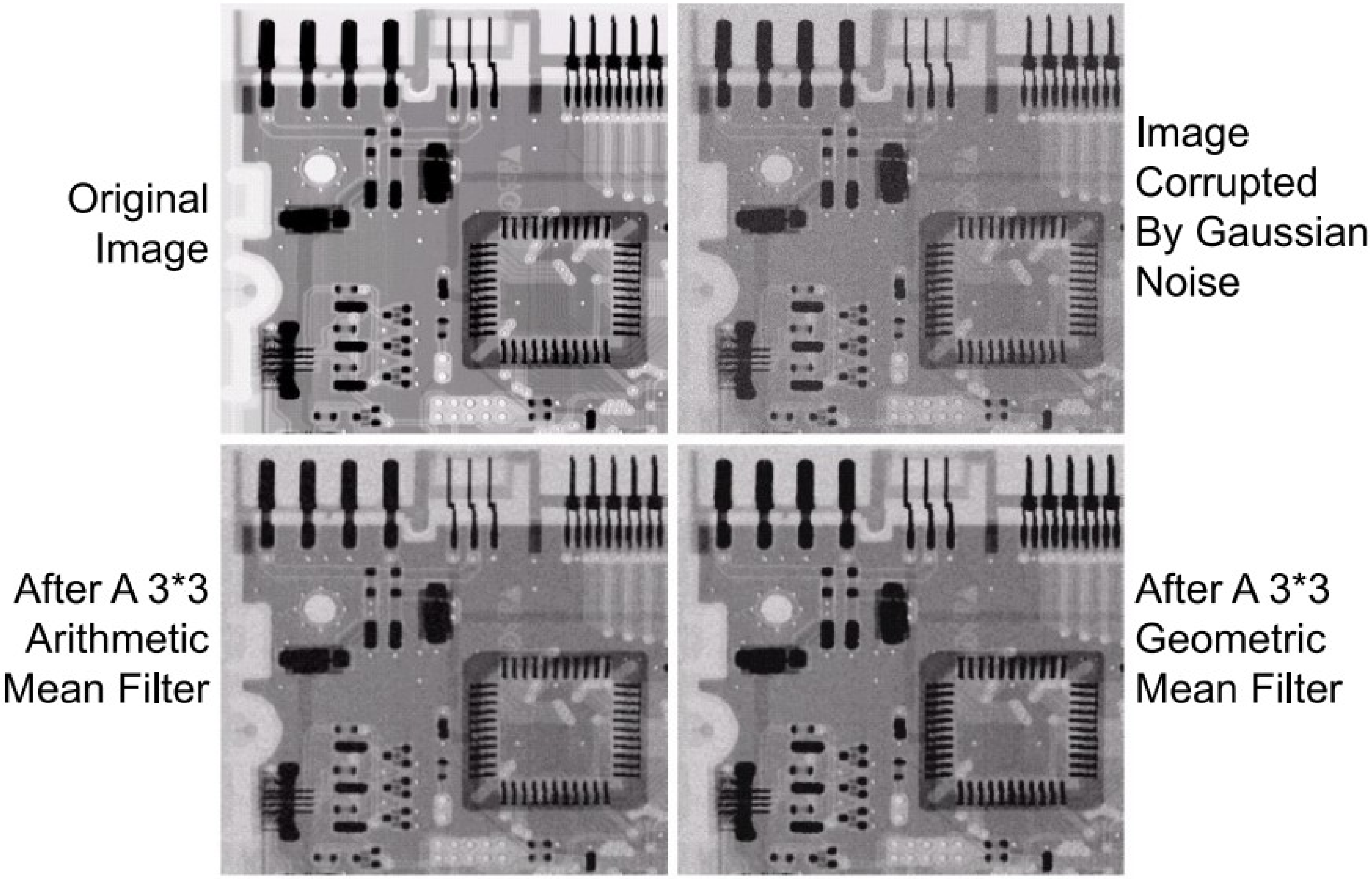
❑ Contra Harmonic Mean Filter

- Positive Q removes pepper noise
- Negative Q removes salt noise

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

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

Image Restoration Filters
Mean 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

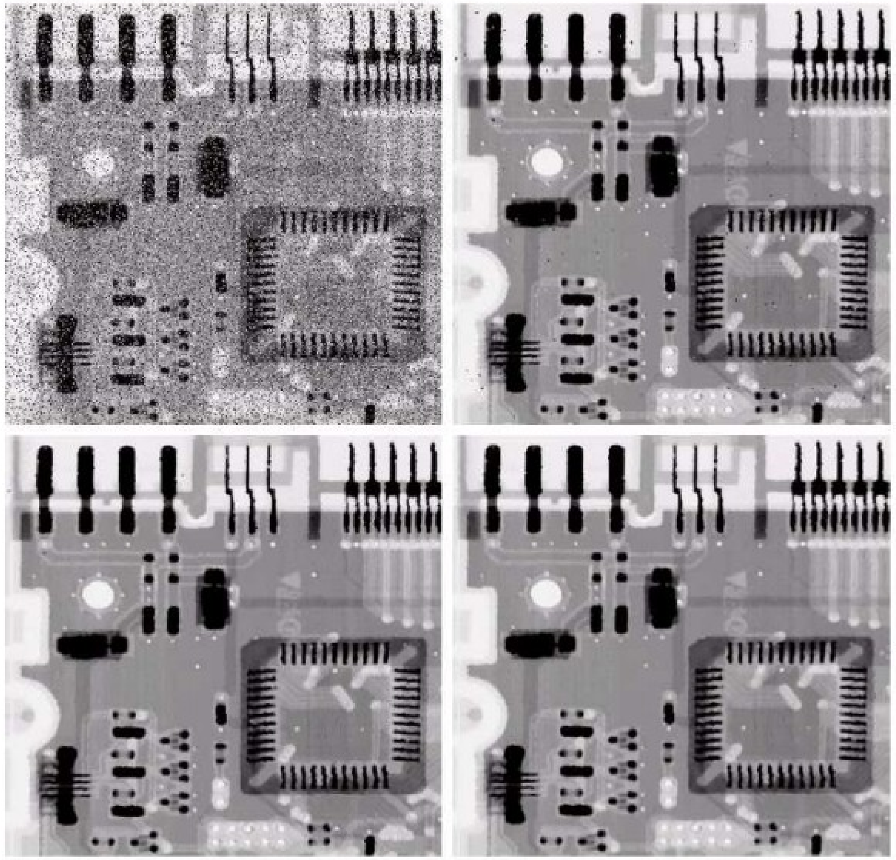
Best result for removing salt and pepper noise.

Image Restoration Filters
Order statistics filters

❑ Median Filter

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

Image Restoration Filters
Order statistics filters

❑ Median Filter

5	6	7	8
0	6	7	8
5	6	15	8
5	6	7	8

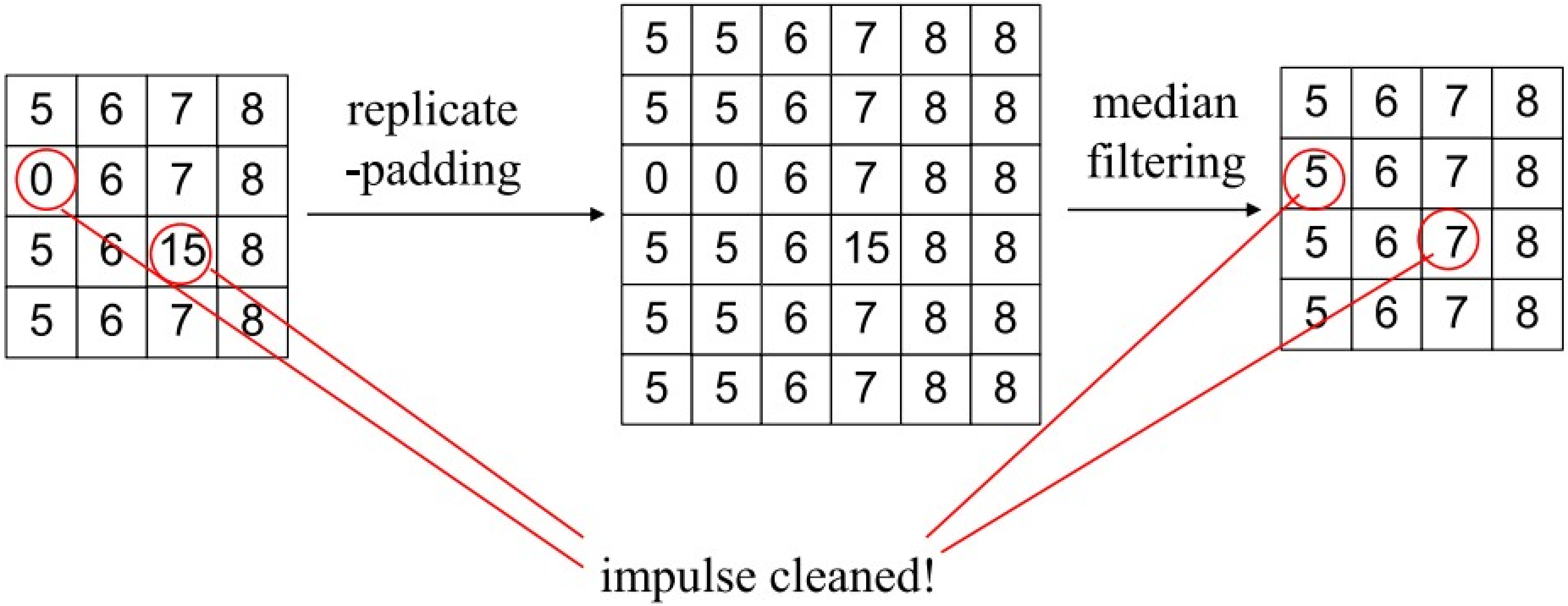
zero-padding →

0	0	0	0	0	0
0	5	6	7	8	0
0	0	6	7	8	0
0	5	6	15	8	0
0	5	6	7	8	0
0	0	0	0	0	0

median filtering →

0	5	6	0
5	6	7	7
5	6	7	7
0	5	6	0

❑ Median Filter



- Min and Max Filter

Max Filter:

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

Min Filter:

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

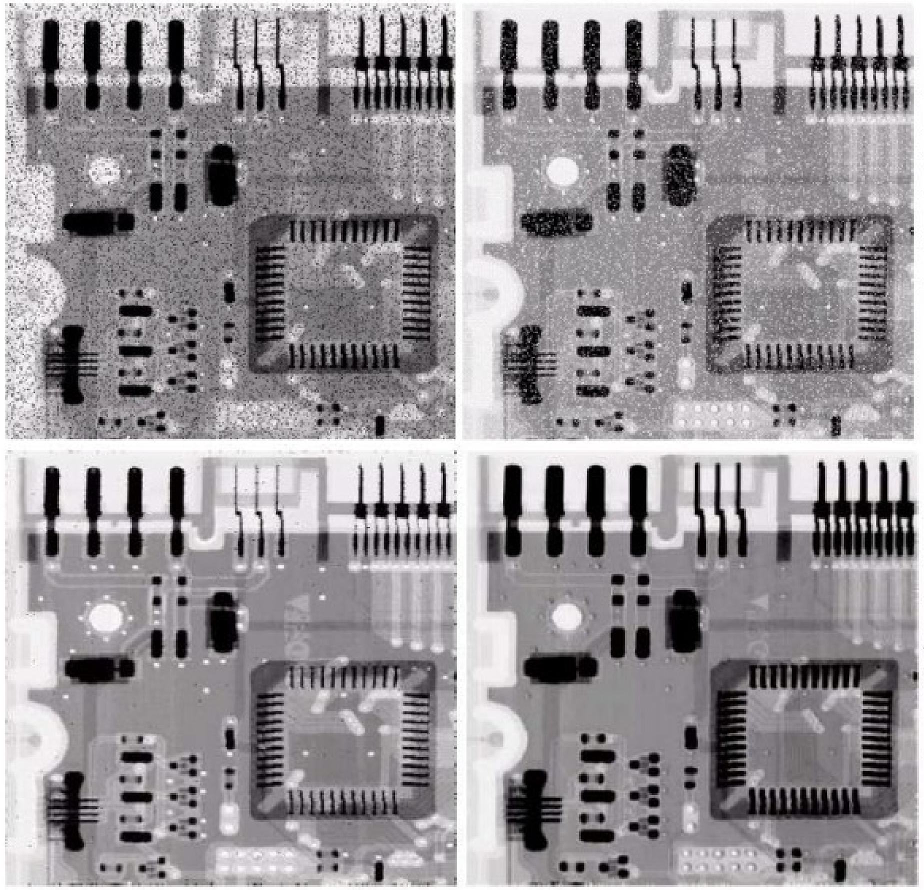
Max filter is good for pepper noise and min is good for salt noise

Image Restoration Filters
Order statistics filters

❑ Min and Max Filter

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

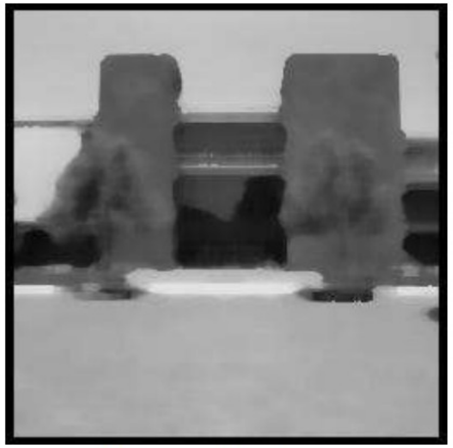
Image Restoration Filters
Order statistics filters

Min Filter

Image with salt noise
Probability = .04

Result of minimum filtering
Mask 3 x 3

Minimum filtering
Mask 5 x 5



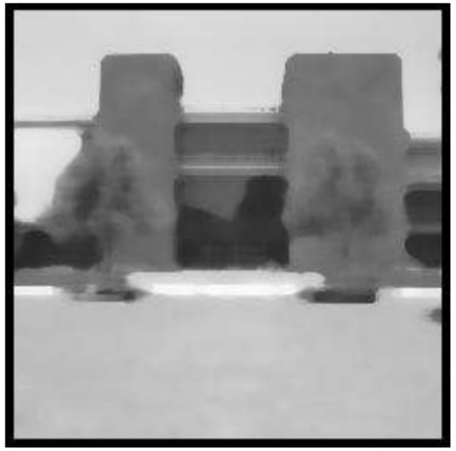
Minimum filtering
Mask 9 x 9

Max Filter

Image with pepper noise
Probability = .04

Maximum filtering
Mask 3 x 3

Maximum filtering
Mask 5 x 5



Maximum filtering
Mask 9 x 9

- 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

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

Here deleted the *d/2* lowest and *d/2* highest grey levels, so $g_r(s, t)$ represents the remaining *mn - d* pixels

Image Restoration Filters
Order statistics filters

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

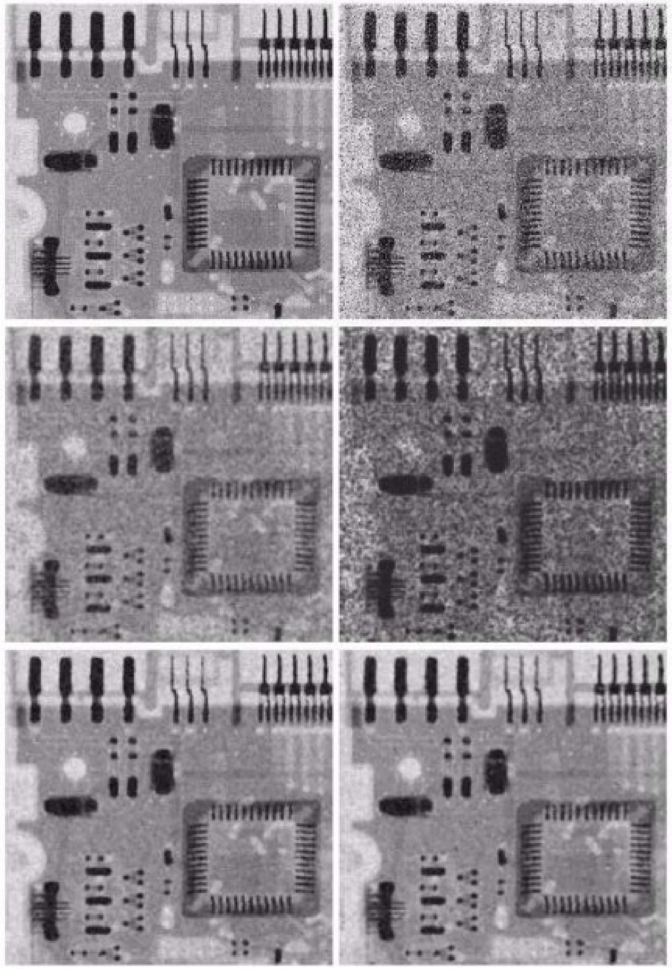


Image Restoration Filters
Adaptive filters

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 behavior of adaptive filters changes depending on the characteristics of the image inside the filter region

❑ Adaptive Median Filtering

The median filter performs relatively well on the impulse noise as long as the spatial density of the impulse noise is not large.

The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise

The key insight in the adaptive median filter is that the filter size changes depending on the characteristics of the image

Image Restoration Filters
Adaptive filters

Adaptive Median Filtering

- Two levels of operations
  - Level A:
    - $A1 = Z_{med} - Z_{min}$
    - $A2 = Z_{med} - Z_{max}$
    - If $A1 > 0$ AND $A2 < 0$, Go to level B
      else increase the window size by 2
    - If window size <= S_{max} repeat level A
      else output Z_{xy}

Used to test whether Z_{med} is part of s-and-p noise. If yes, window size is increased

  - Level B:
    - $B1 = Z_{xy} - Z_{min}$
    - $B2 = Z_{xy} - Z_{max}$
    - If $B1 > 0$ AND $B2 < 0$, output Z_{xy}
      else output Z_{med}

Used to test whether Z_{xy} is part of s-and-p noise. If yes, apply regular median filtering

- z_{min} = minimum grey level in S_{xy}
- z_{max} = maximum grey level in S_{xy}
- z_{med} = median of grey levels in S_{xy}
- z_{xy} = grey level at coordinates (x, y)
- S_{max} =maximum allowed size of S_{xy}

Adaptive Median Filtering

1. to remove salt-and-pepper (impulse) noise.
2. to provide smoothing of other noise that may not be impulsive.
3. to reduce distortion, such as excessive thinning or thickening of object boundaries.

Periodic Noise Reduction by Frequency Domain Filtering

- Lowpass and high pass filters for image enhancement have been studied
- Band reject, bandpass, and notch filters as tools for periodic noise reduction or removal

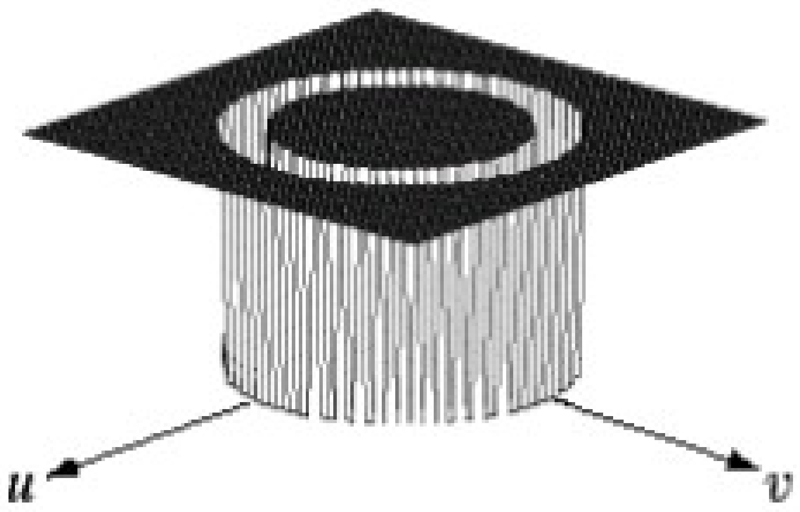
Band Reject Filters

- Band reject filters remove or attenuate a band of frequencies about the origin of the Fourier transform.
- Similar to those LPFs and HPFs, we can construct ideal, Butterworth, and Gaussian band reject filters

Band Reject Filters

- Ideal Filter

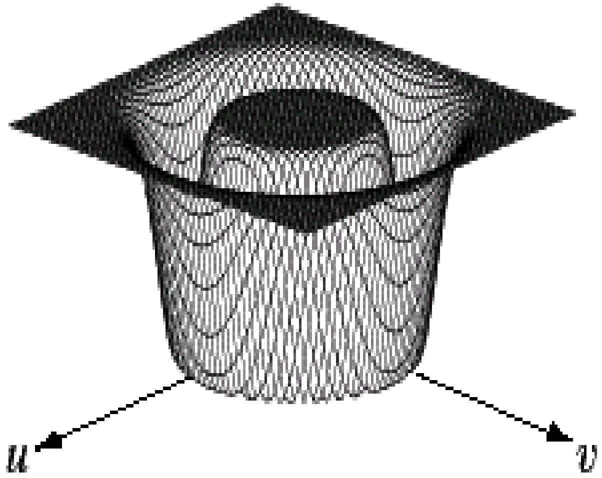
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \frac{W}{2} \end{cases}$$



Band Reject Filters

- Gaussian Filter

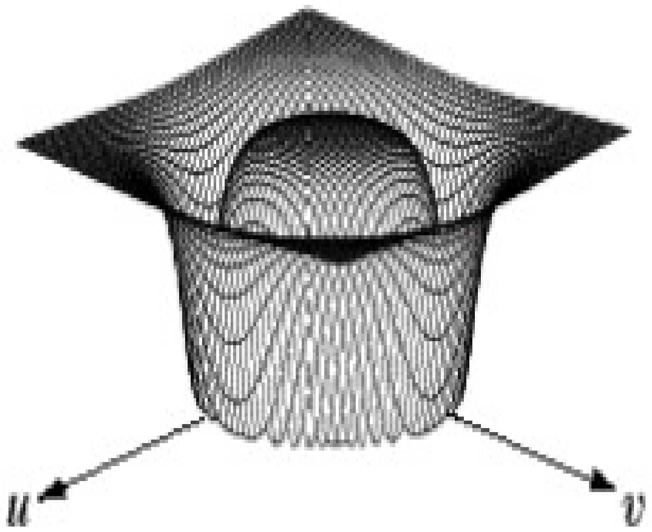
$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)W}\right]}$$



Band Reject Filters

- Butterworth Filter

$$H(u,v) = \frac{1}{1+[D_0 / D(u,v)]^{2n}}$$



Band Pass Filters

- Accepts particular region of frequency of an image only
- Can be obtained from the equations for band reject filter as follows

$$H_{bp}(u,v) = 1 - H_{br}(u,v)$$

Image Compression

Image Compression

Three levels of JPG compression. The left-most image is the original. The middle image offers a medium compression, which may not be immediately obvious to the naked eye without closer inspection. The right-most image is maximally compressed.

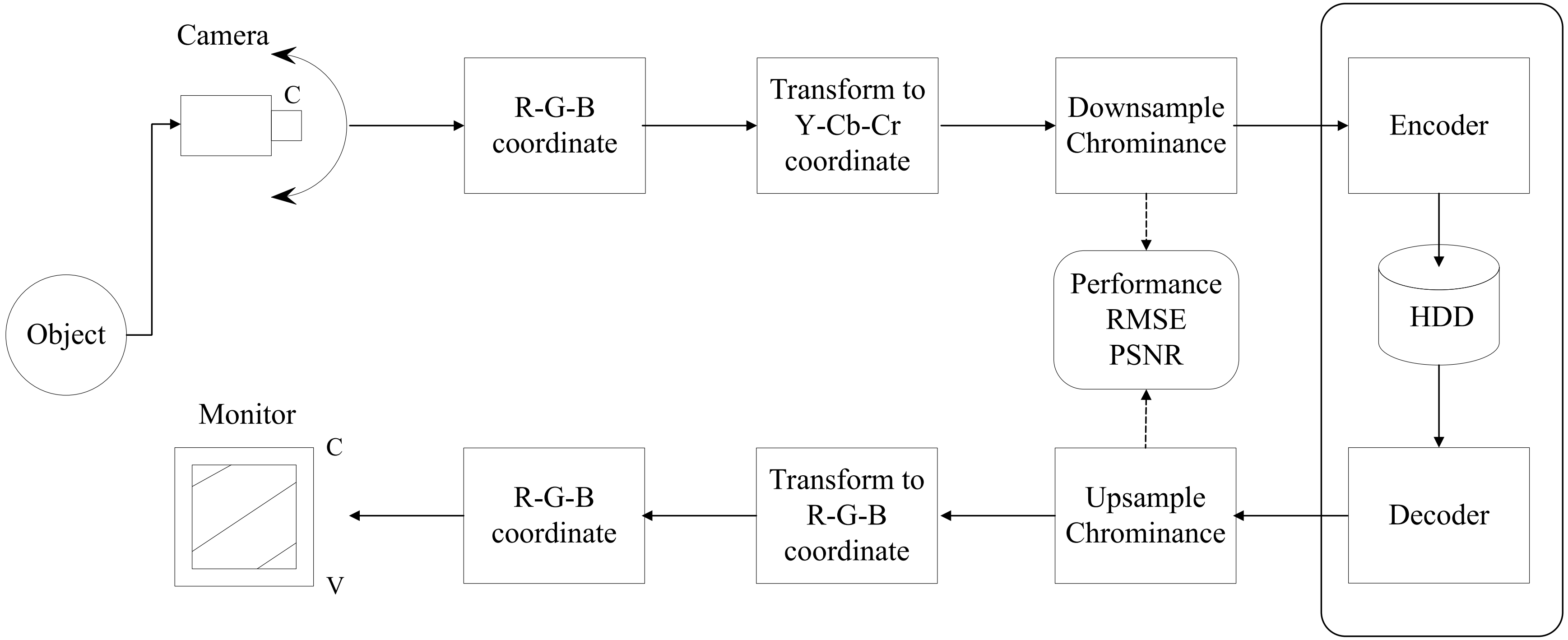
Why Image Compression

Image compression is the process of reducing total number of bits required to represent the image

A 90 minutes color movie, each second playing 24 frames when is digitized with each frame of 512*512 pixels, each pixel having three components R,G,B (8 Bits each) has total bytes :

$$90 * 60 * 24 * 3 * 512 * 512 = 97200 \text{ MB}$$

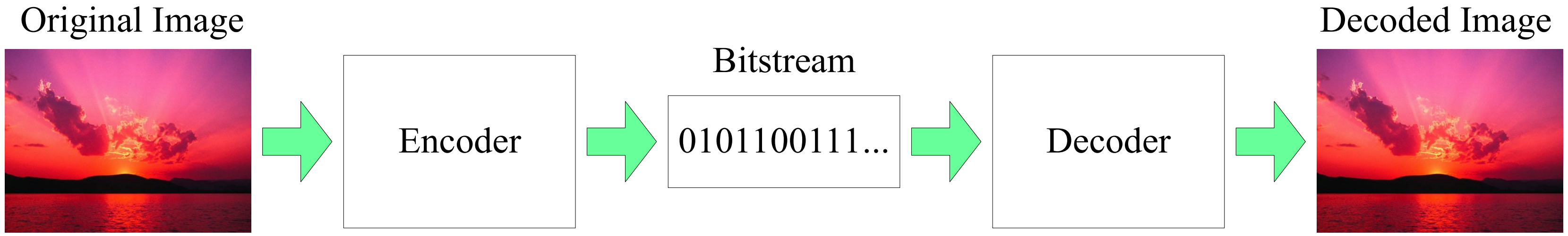
Image Compression Fundamentals



‘Y’ weighted sum of gamma-compressed is more sensitive to the human eye, it needs to be more correct and ‘Cb’ and ‘Cr’ is less sensitive to the human eye. Therefore it needs not to be more accurate. When in JPEG compression, it uses these sensitivities of the human eye and eliminate the unnecessary details of the image.

Flow of Image Compression

- The image file is converted into a series of binary data, which is called the bit-stream
- The decoder receives the encoded bit-stream and decodes it to reconstruct the image
- The total data quantity of the bit-stream is less than the total data quantity of the original image



Measure to evaluate the performance of image compression

Root Mean square error:

$$RMSE=\sqrt{\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y)-f'(x,y)]^2}{MN}}$$

Peak signal to noise ratio:

$$PSNR=20log10\frac{255}{MSE}$$

Compression Ratio:

$$Cr=\frac{n1}{n2}$$

Where n_1 is the data rate of original image

n_2 is the data rate of the encoded bit-stream

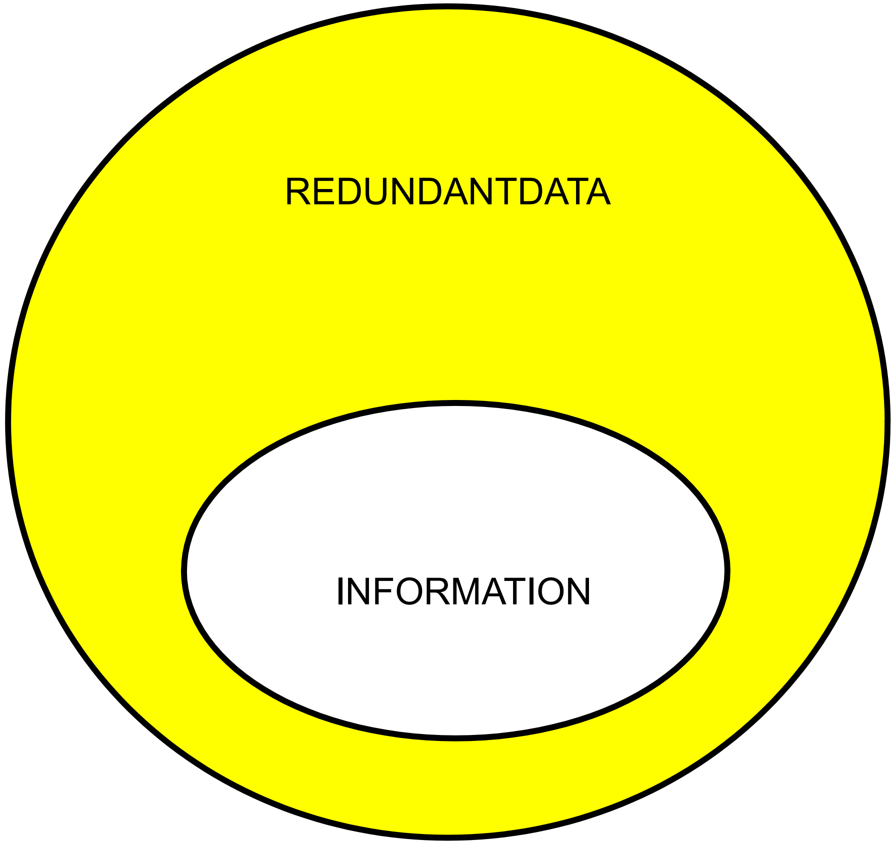
Application of Compressions

- Objective : To reduce the amount of data required to represent an image
- Applications :
  - Video Coding
  - Progressive transmission of Images(Internet/WWW)
  - Remote Sensing through satellite
  - Military Communication

Data Redundancy

- The wasted space consumed by storage media to store image information in a digital image
- Three Types
  - Psycho Visual Redundancy (Associated to Human Perception)
  - Coding Redundancy (associated with the representation of information)
  - Pixel Redundancy (Associated with pixel values)

Data Redundancy



Coding Redundancy

- A code is a set of symbols used to represent a body of information or a set of events.
- Each piece of information is assigned a code word.
- The number of symbols in the code comprise its length.
- Length redundancy happens when the code has more bits than required to represent the information.
- It happens when the coding scheme does not make use of the non-uniformity of intensities probabilities.

Coding Redundancy

Let us assume, that a discrete random variable r_k in the interval [0,1] represent the gray level of an image:

$$p_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \cdots, L-1$$

If the number of bits used to represent each value of r_k is $L(r_k)$, then the average number of bits required to represent each pixel:

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

The total number bits required to code an MxN image:

$$M.N.L_{avg}$$

Coding Redundancy

- To achieve less average length of bits per pixel of the image.
- Assigns short descriptions to the more frequent outcomes and long descriptions to the less frequent outcomes

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

L_{avg} 3 bits/symbol

L_{avg} 2.7 bits/symbol

Entropy

Given a source of statistically independent random events from a discrete set of possible events $\{a_1, a_2, a_3, a_4 \ldots . a_n\}$ with associated probabilities $\{p(a_1), p(a_2), p(a_3) \ldots\}$, the average information per source output is called entropy of the source.

$$\text{Entropy(H)} = -\sum_{1}^{N} P_i \log_x P_i$$

Variable Length Coding

The simplest approach to error-free image compression is to reduce only redundant codes. Coding redundancy normally is present in any natural binary encoding of the gray levels in an image. It can be eliminated by coding the gray levels.

To do so requires construction of a variable-length code that assigns the shortest possible code words to the most probable gray levels

Huffman Coding

1. Order the symbols according to the probabilities
   - Alphabet set: $S_1, S_2, \ldots, S_N$
   - Probabilities: $P_1, P_2, \ldots, P_N$
   - The symbols are arranged so that $P_1 \geqq P_2 \geqq \ldots \geqq P_N$
2. Apply a contraction process to the two symbols with the smallest probabilities. Replace the last two symbols S_N and S_{N-1} to form a new symbol H_{N-1} that has the probabilities $P_1 + P_2$.

   The new set of symbols has $_{N-1}$ members: $S_1, S_2, \ldots, S_{N-2}, H_{N-1}$
3. Repeat the step 2 until the final set has only one member.
4. The codeword for each symbol S_i is obtained by traversing the binary tree from its root to the leaf node corresponding to S_i

Huffman Coding

Source generates the symbol s1, s2, s3, s4 and s5 with probability 0.25,0.25,0.2, 0.15 and 0.15 respectively

Generate the code word for each symbol using Huffman Coding . Also efficiency of the system

X	Probability					Codeword	Codeword length
s1	0.25 (01)	0.3 (00)	0.45 (1)	0.55 (0)	1	01	2
s2	0.25 (10)	0.25 (01)	0.3 (00)	0.45 (1)		10	2
s3	0.2 (11)	0.25 (10)	0.25 (01)			11	2
s4	0.15 (000)	0.2 (11)				000	3
s5	0.15 (001)					001	3

L_{avg}=

= 0.25*2+0.25*2+0.2*2+0.15*3+0.15*3

= 2.33 bits per symbol

Entropy(H)=

= - [0.25 * -2 + 0.25 * -2 + 0.2*-1.74 + 0.15*-1.74 + 0.15*-1.74]

= 1.87

Efficiency(η)= $\frac{Entropy}{L_{avg}}$ *** 100 %**

=80.25 bits per symbol

Compression Ratio(n1/n2)

= 3 / 2.33

Huffman Coding

Draw the Huffman tree and Calculate the efficiency from below image

Gray Level	0	1	2	3	4	5	6	7
No. of Pixel	400	1350	659	2034	816	2560	250	1500

Ans: 99.11%

Interpixel Redundancy

The value of certain pixel in the image can be reasonably predicted from the values of its neighbor in the image.
Thus, the values of the individual pixels carries relatively small amount of information and much more information about the pixel value that can be inferred on the basis of its neighbor's value.

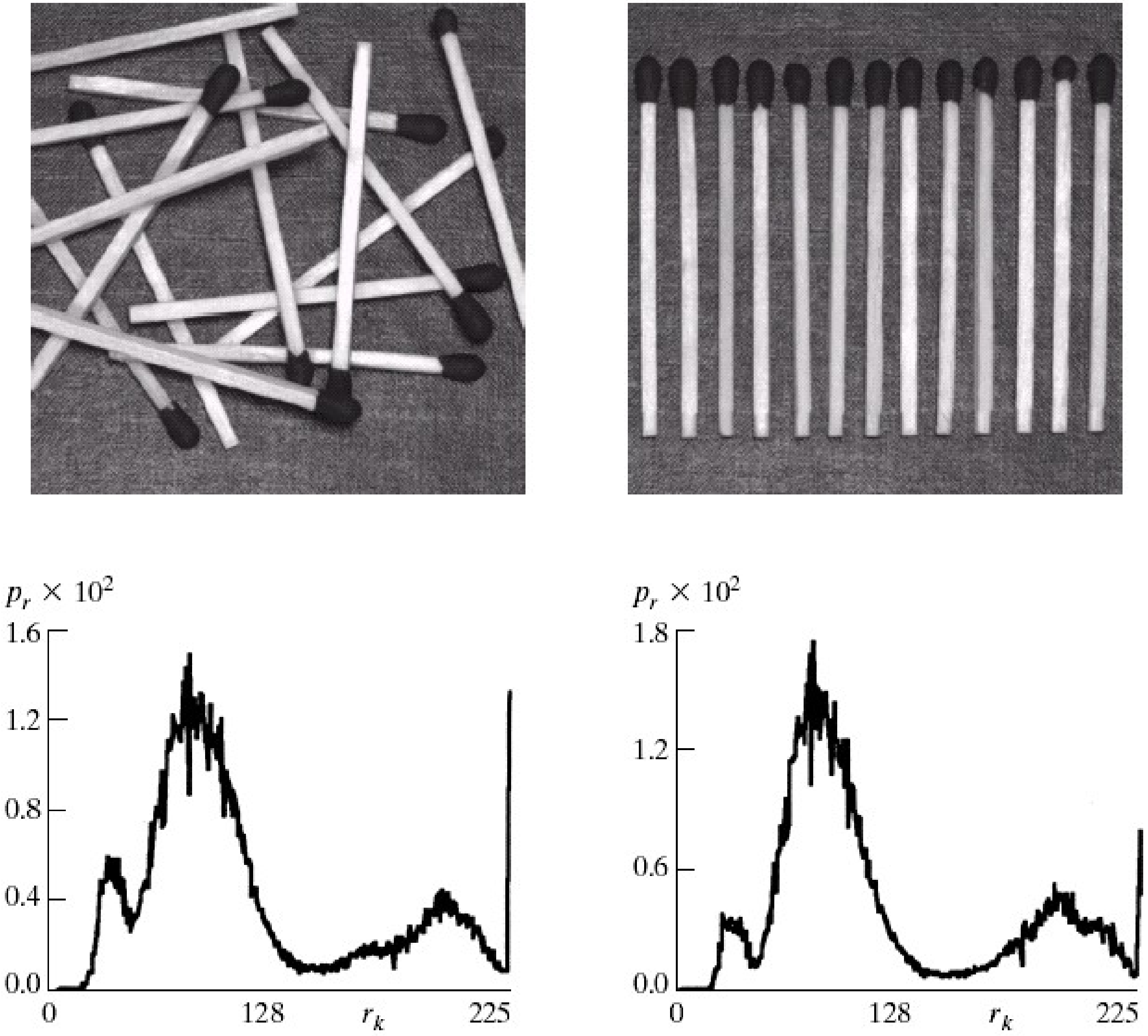
These dependencies between pixel value in the image is called inter pixel redundancy.

Interpixel Redundancy

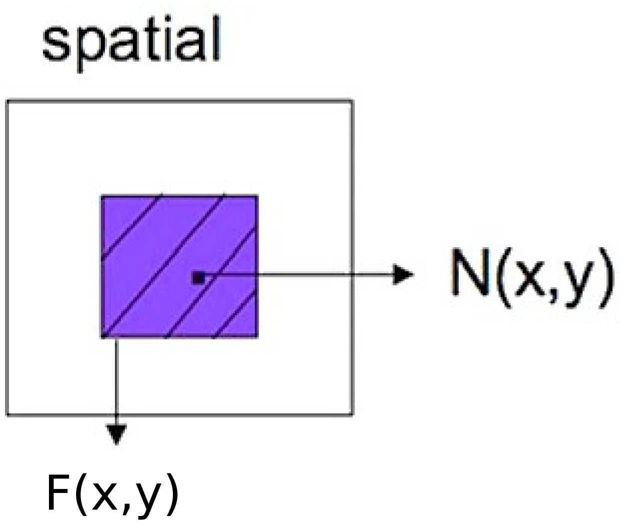
These images have virtually identical histograms. Note also that both histograms are trimodal, indicating the presence of three dominant ranges of gray-level values.

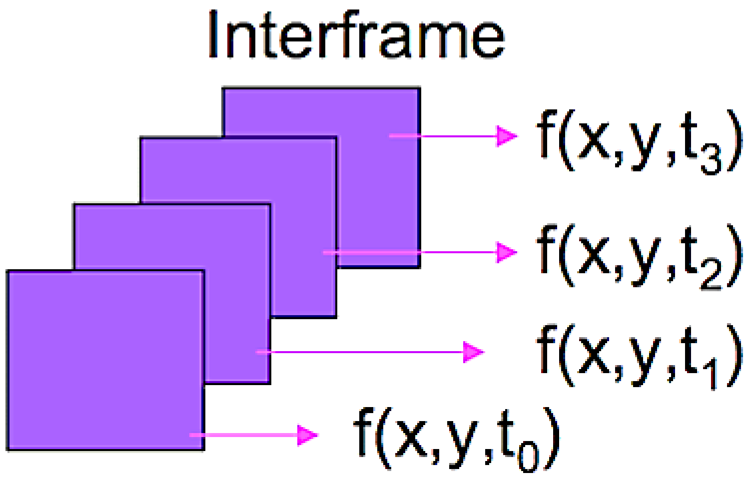
The codes used to represent the gray levels of each image have nothing to do between pixels.

These result from the structural or geometric relationships between the objects / Neighbor in the image



Interpixel Redundancy





f(x,y) depends on f(x',y'), (x',y') Ɛ Nxy
Nxy is neighborhood of pixels around (x,y).

f(x,y,ti) i = 0,1,2,3 .... are related to each. This can be exploited for video compression.

Run Length Coding

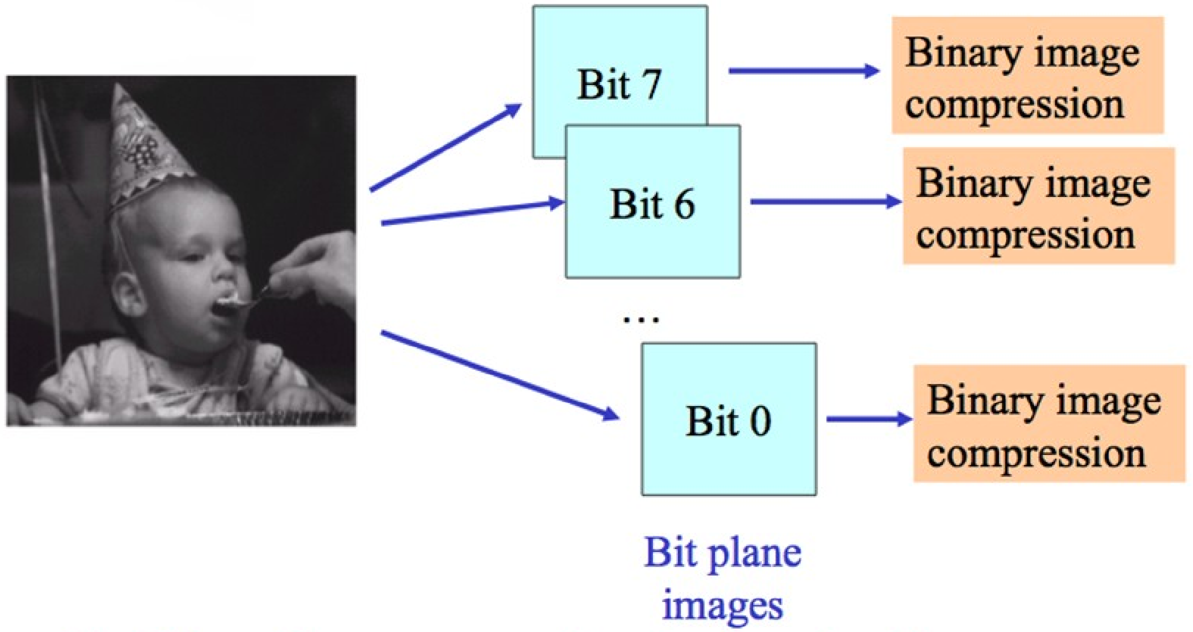
- Run-length encoding (RLE) is a form of data compression in which runs of data (sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count *(length of run, value)*, rather than as the original run.
- This is most efficient on data that contains many such runs, for example simple graphic images such as icons, line drawing etc.
- For files that do not have many runs, RLE could increase the file size

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW

12W1B12W3B24W1B14W

Bit Plane Coding

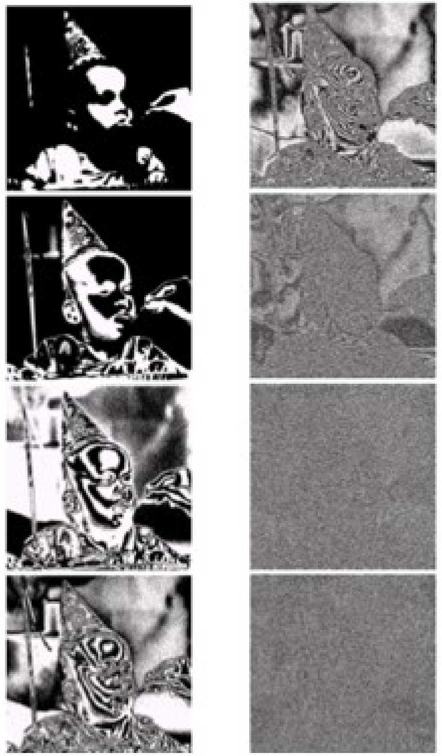
An effective technique for reducing an image's interpixel redundancies is to process the image's bit planes individually. It is based on the concept of decomposing a multilevel (monochrome or color) image into a series of binary images and compressing each binary image via one of several well-known binary compression methods



Example of binary image compression: Run length coding

Bit Plane Coding

Original gray scale image



Gray code:

$$g_i = a_i \otimes a_{i+1}$$

for $0 \le i \le 6$

and

$$g_7 = a_7$$

a_i= Original bit planes

$\otimes$ = XOR

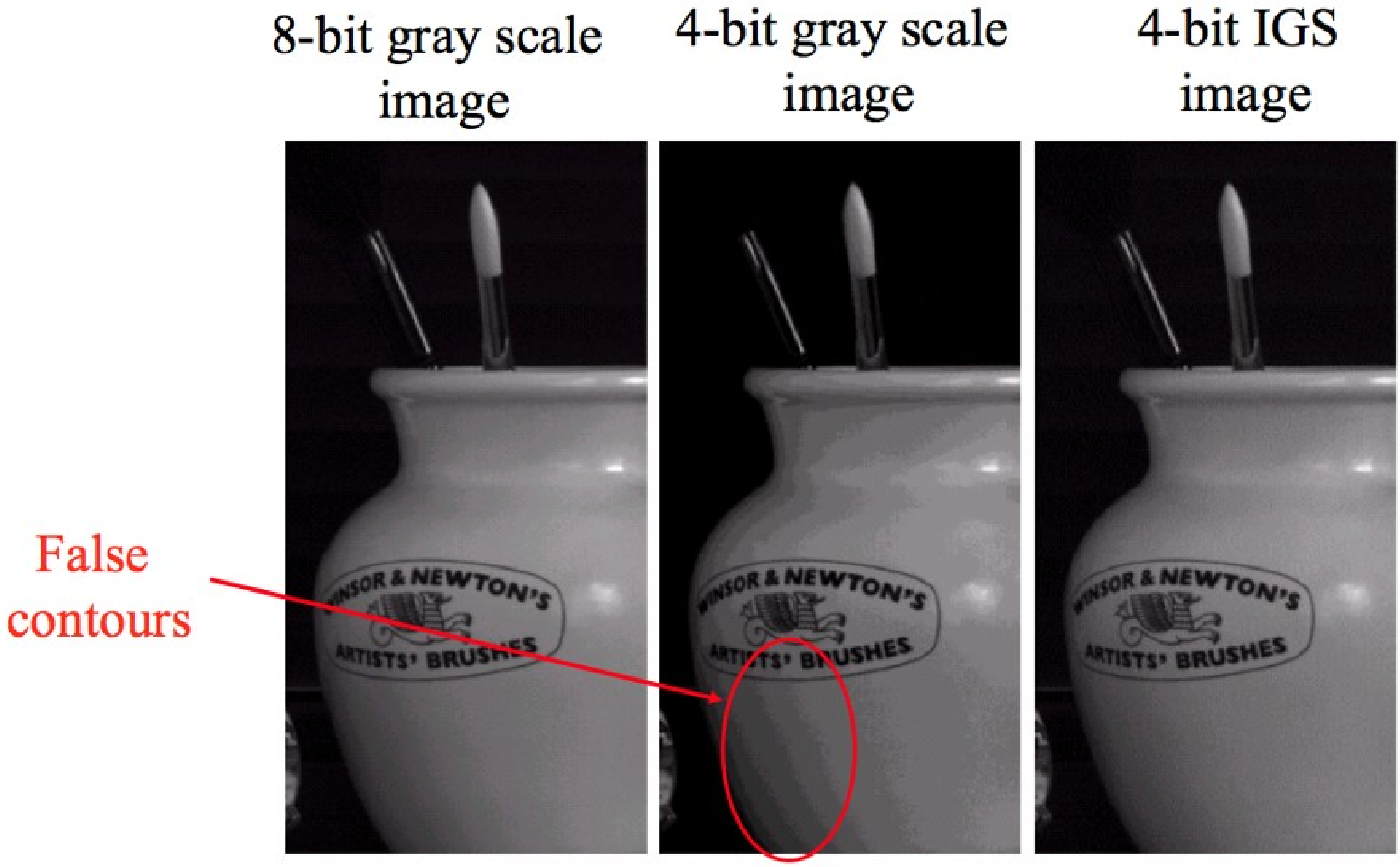
Psychovisual Redundancy

Human eye does not respond with equal sensitivity to all visual information. Some information is visually more important than others.

Information which is not visually important is called Psychovisual Redundancy.

Psychovisual redundancies exist in all images and are eliminated without hampering the subjective quality of image

Psychovisual Redundancy



(Improved Gray Scale)IGS Quantization

Simply reducing the quantization i.e. number of bits of representation, compress the image but also produces false contouring. IGS coding reduces the quantization but also reduces false contouring. Steps involved in IGS coding are

a. Lower 4-bits on the preceding modified pixel are added to the present pixel.
b. The new 4 MSB's of the present pixel are taken as the I.G.S code.
c. Repeat step 1 and 2 after moving on to new pixel.
d. If the MSB is 1111, then add 0000 instead of the 4 LSB's of the previous sum.

IGS Quantization

Construct the IGS code for the given grey level data set {100,110,124,124,130,110,200,210}

From the grey level values it is clear that we could require 8-bits for their representation. Let I be the first pixel.

We start by adding 0000 to the 1st pixel

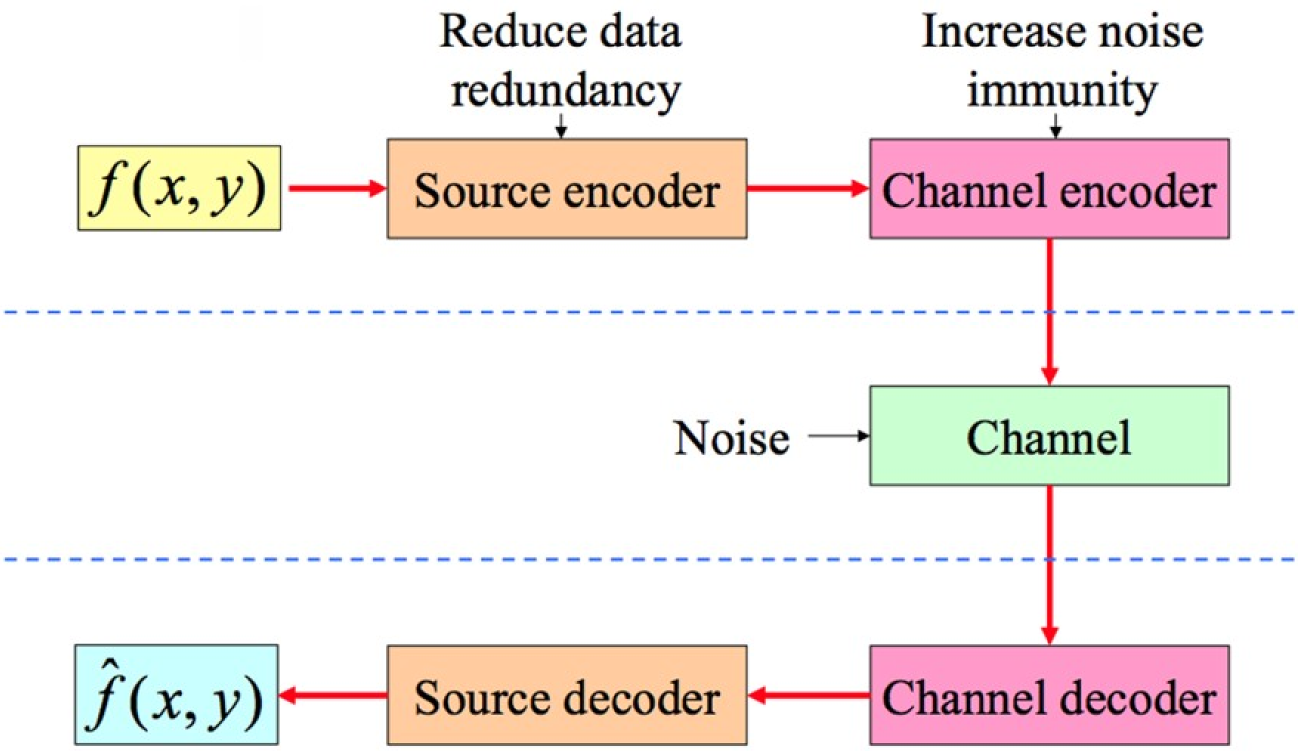
		Grey level		SUM		IGS Code
i - 1				0000	0000	–
i	100	0110	0100	0110	0100	0110
i + 1	110	0110	1110	0111	0010	0111
i + 2	124	0111	1100	0111	1110	0111
i + 3	124	0111	1100	1000	1010	1000
i + 4	130	1000	0010	1000	1100	1000
i + 5	110	0110	1110	0111	1010	0111
i + 6	200	1100	1000	1101	0010	1101
i + 7	210	1101	0010	1101	0100	1101

Image Compression Models

A compression system consists of two distinct structural blocks:

- an encoder
- a decoder.

An input image f(x, y) is fed into the encoder, which creates a set of symbols from the input data. After transmission over the channel, the encoded representation is fed to the decoder, where a reconstructed output image f^(x, y) is generated.



In general, f^(x, y) may or may not be an exact replica of f(x, y). If it is, the system is error free or information preserving; if not, some level of distortion is present in the reconstructed image

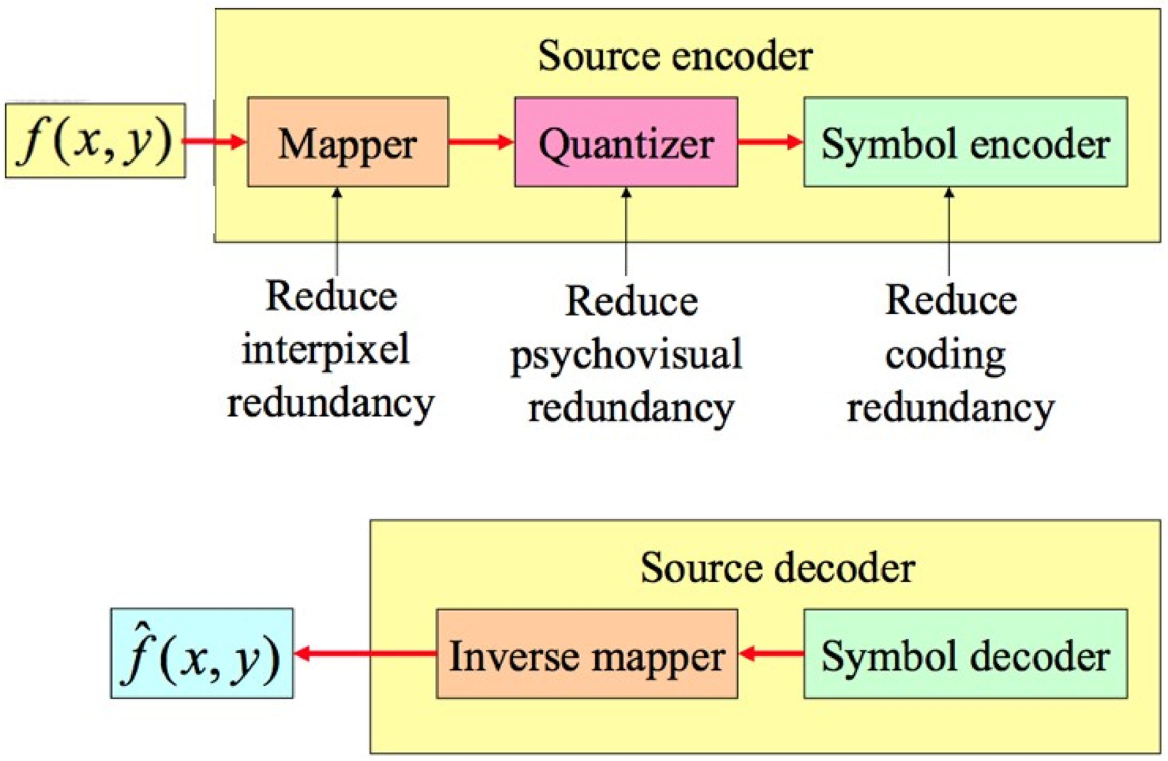
The encoder is made up of a source encoder, which removes input redundancies, and a channel encoder, which increases the noise immunity of the source encoder's output. The decoder includes a channel decoder followed by a source decoder

Encoder / Decoder Models

The mapper transforms the image into an array of coefficients, making its interpixel redundancies more accessible for compression in later stages of the encoding process

The quantizer reduces the accuracy of the mapper's output in accordance with some preestablished fidelity criterion. This stage reduces the psychovisual redundancies of the input image

Encoder assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancy.



Fidelity Criteria

The error between two functions is given by: $$e(x,y)=\hat{f}(x,y)-f(x,y)$$

So, the total error between the two images is: $$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)-f(x,y)$$

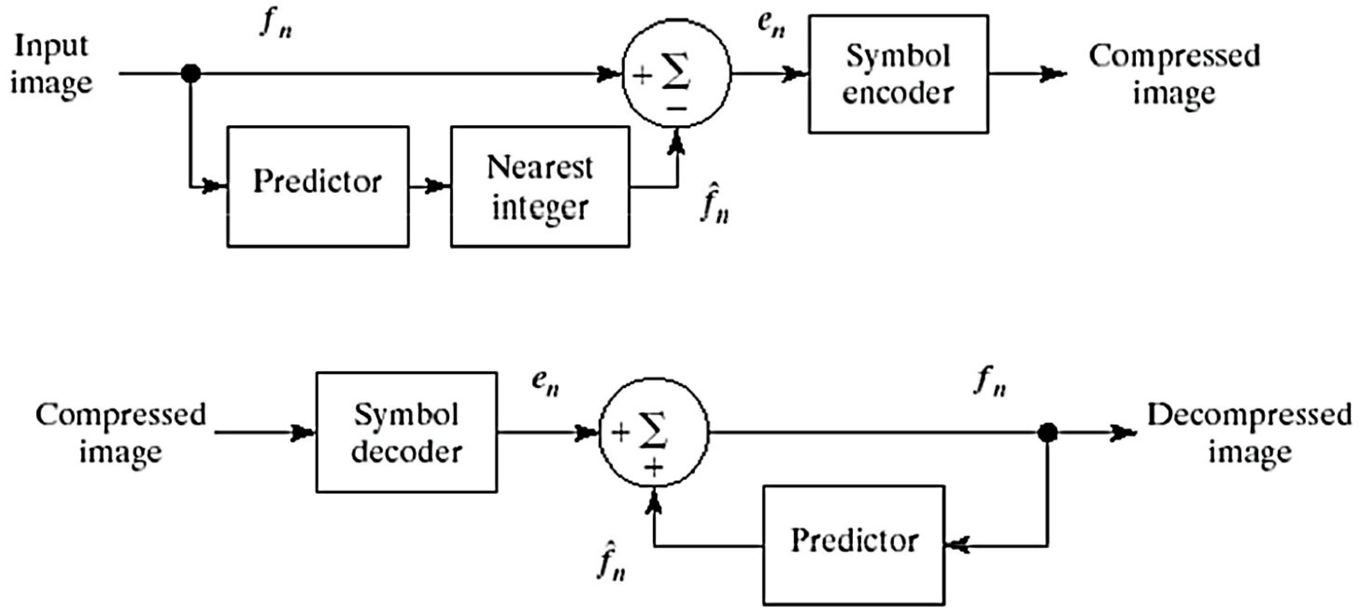
The root-mean-square error averaged over the whole image is: $$e_{rms}=\frac{1}{MN}\sqrt{[\hat{f}(x,y)-f(x,y)]^2}$$

A closely related objective fidelity criterion is the mean square signal to noise ratio of the compressed-decompressed image : $$SNR_{rms}=\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y)-f(x,y)]^2}$$

Error-Free Compression (Lossless Predictive Coding)

- The error-free compression approach does not require decomposition of an image into a collection of bit planes.
- The approach, commonly referred to as lossless predictive coding, is based on eliminating the interpixel redundancies of closely spaced pixels by extracting and coding only the new information in each pixel.
- The new information of a pixel is defined as the difference between the actual and predicted value of that pixel

Error-Free Compression (Lossless Predictive Coding)



Prediction error:

$e_n = f_n - \hat{f}_n$

e_n is coded using a variable length code

$f_n = e_n + \hat{f}_n$

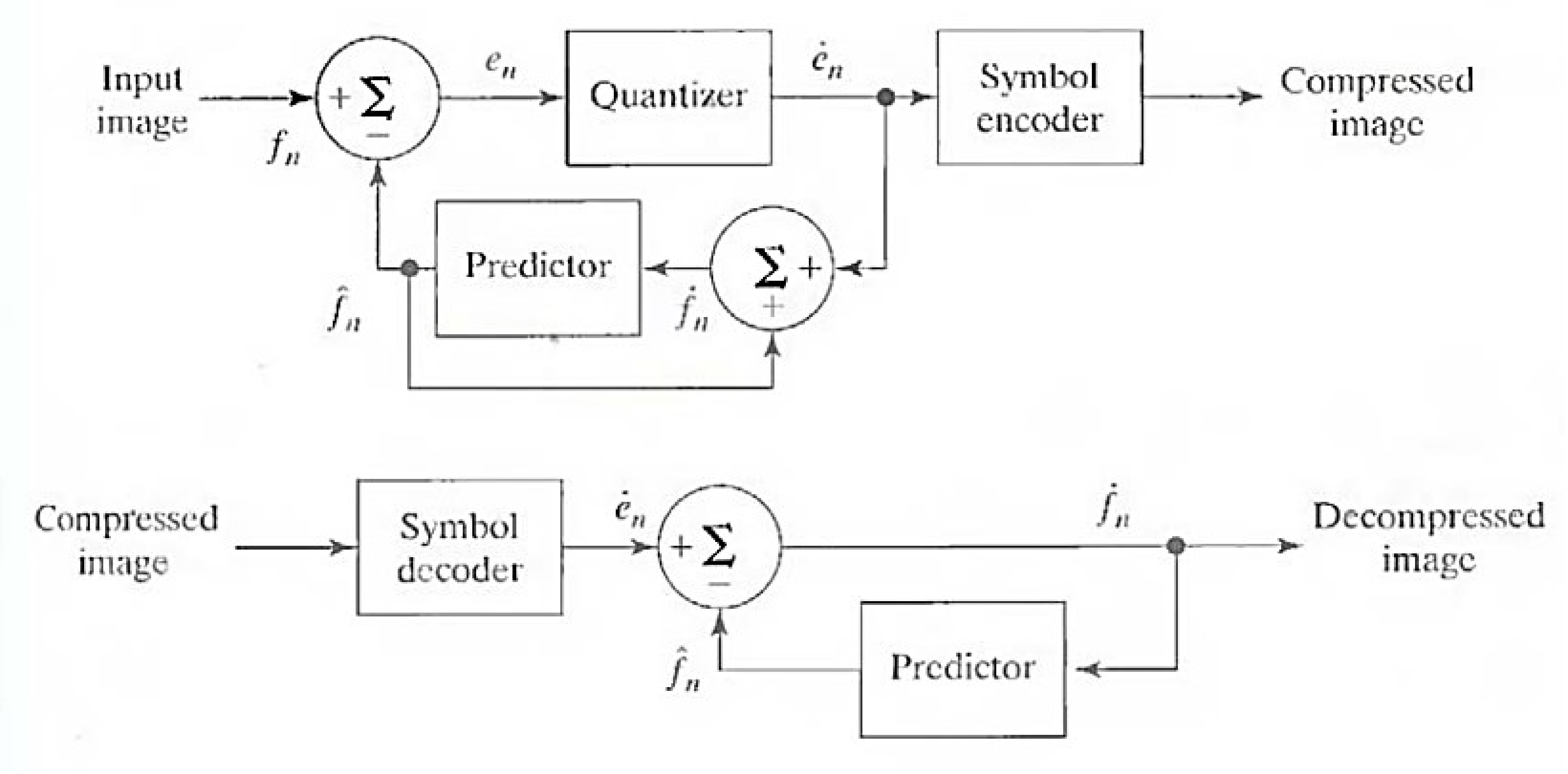
Figure shows the basic components of a lossless predictive coding system. The system consists of an encoder and a decoder, each containing an identical predictor.

As each successive pixel of the input image, denoted f_n, is introduced to the encoder, the predictor generates the anticipated value of that pixel based on some number of past inputs. The output of the predictor is then rounded to the nearest integer, denoted $\hat{f}_n$ and used to form the difference or prediction error which is coded using a variable-length code (by the symbol encoder) to generate the next element of the compressed data stream

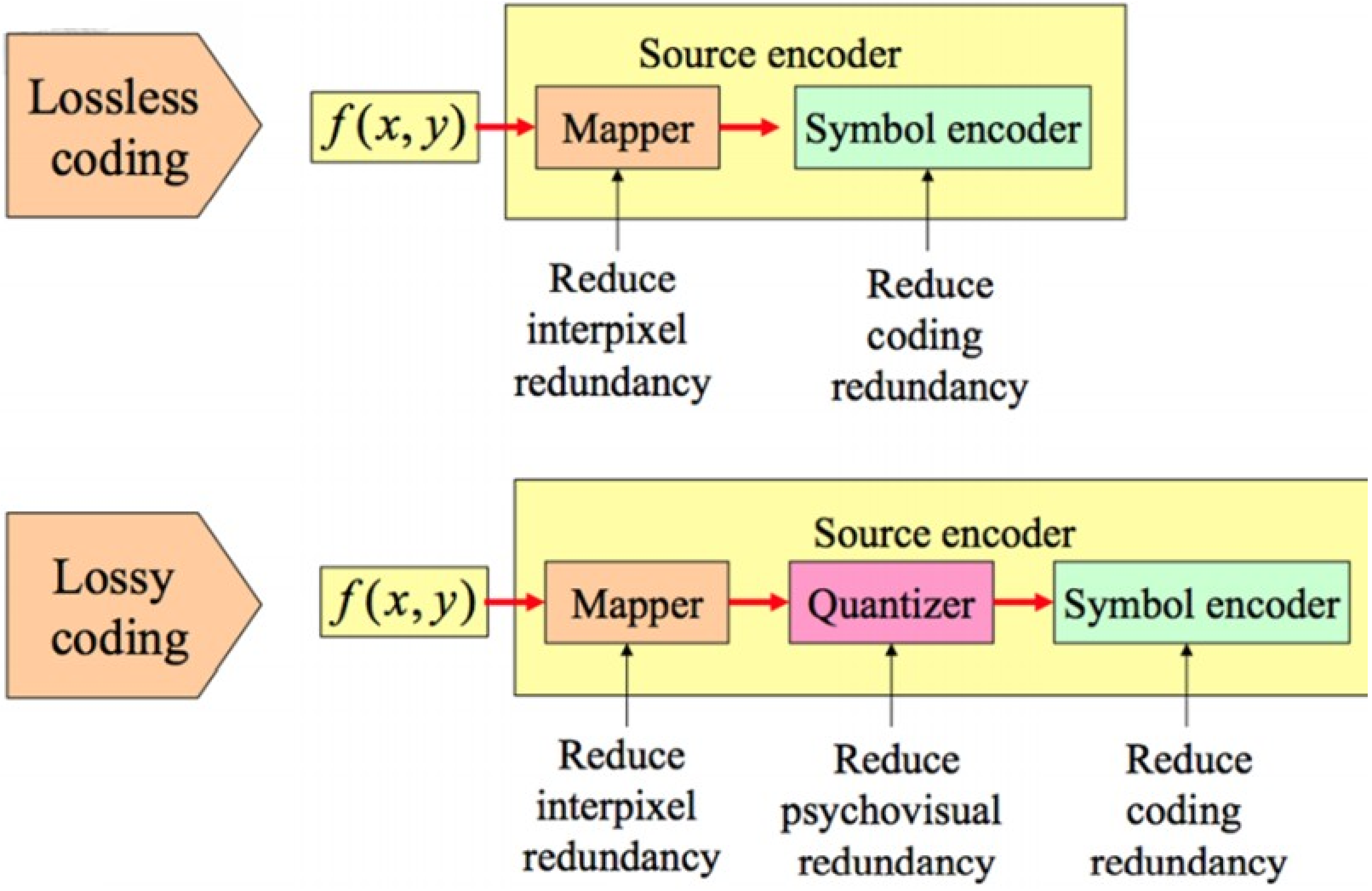
Lossy Predictive Coding

- In this type of coding, we add a quantizer to the lossless predictive model and examine the resulting trade-off between reconstruction accuracy and compression performance.
- The quantizer is inserted between the symbol encoder and the point at which the prediction error is formed.
- It maps the prediction error into a limited range of outputs, denoted $\dot{e}_n$ which establish the amount of compression and distortion associated with lossy predictive coding

Lossy Predictive Coding



Lossless Vs Lossy



Transform Coding

- All the predictive coding techniques operate directly on the pixels of an image and thus are spatial domain methods.
- In transform coding, a reversible, linear transform (such as the Fourier transform) is used to map the image into a set of transform coefficients, which are then quantized and coded.
- For most natural images, a significant number of the coefficients have small magnitudes and can be quantized (or discarded entirely) with little image distortion.
- A variety of transformations, including the discrete Fourier transform (DFT), can be used to transform the image data.

Transform Coding