Digital Image Processing

Lecture # 9
Image Restoration & Compression

WHAT IS IMAGE RESTORATION?

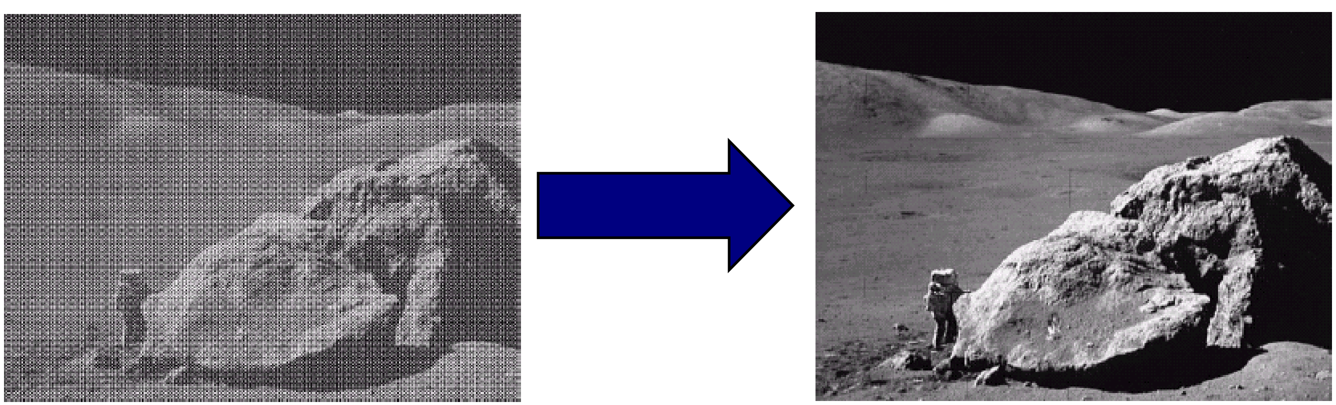
- The purpose of image restoration is to restore a degraded/distorted image to its original content and quality
- Restoration attempts to reconstruct or recover an image that has been degraded by using a priori knowledge of the degradation phenomenon
- Restoration techniques are oriented toward modeling the degradation and applying the inverse process in order to recover the original image
- Image enhancement is largely a subjective process, while image restoration is for the most part an objective process

WHAT IS IMAGE RESTORATION?

- **Image enhancement:** “improve” an image subjectively
- **Image restoration:** remove distortion from image in order to go back to the “original” -> objective process

WHAT IS IMAGE RESTORATION?

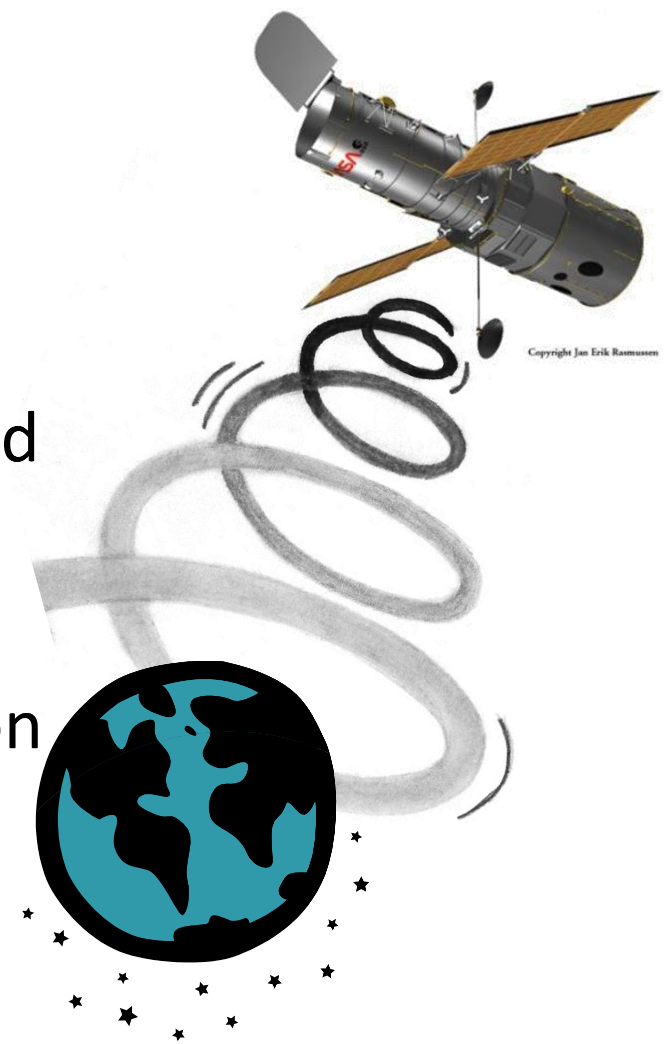
- Image restoration attempts to restore images that have been degraded
  - Identify the degradation process and attempt to reverse it
  - Similar to image enhancement, but more objective



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

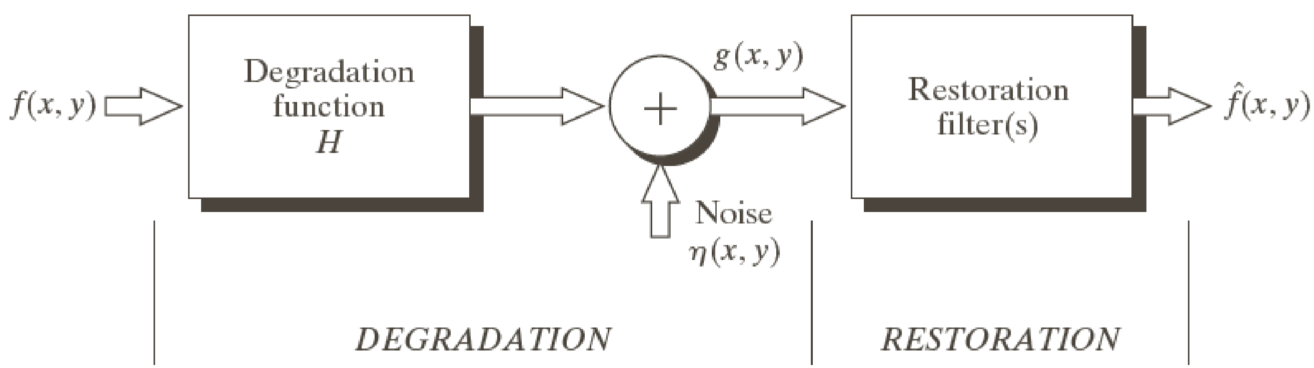
We can consider a noisy image to be modelled as follows:

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

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

If we can estimate the model the noise in an image is based on this will help us to figure out how to restore the image

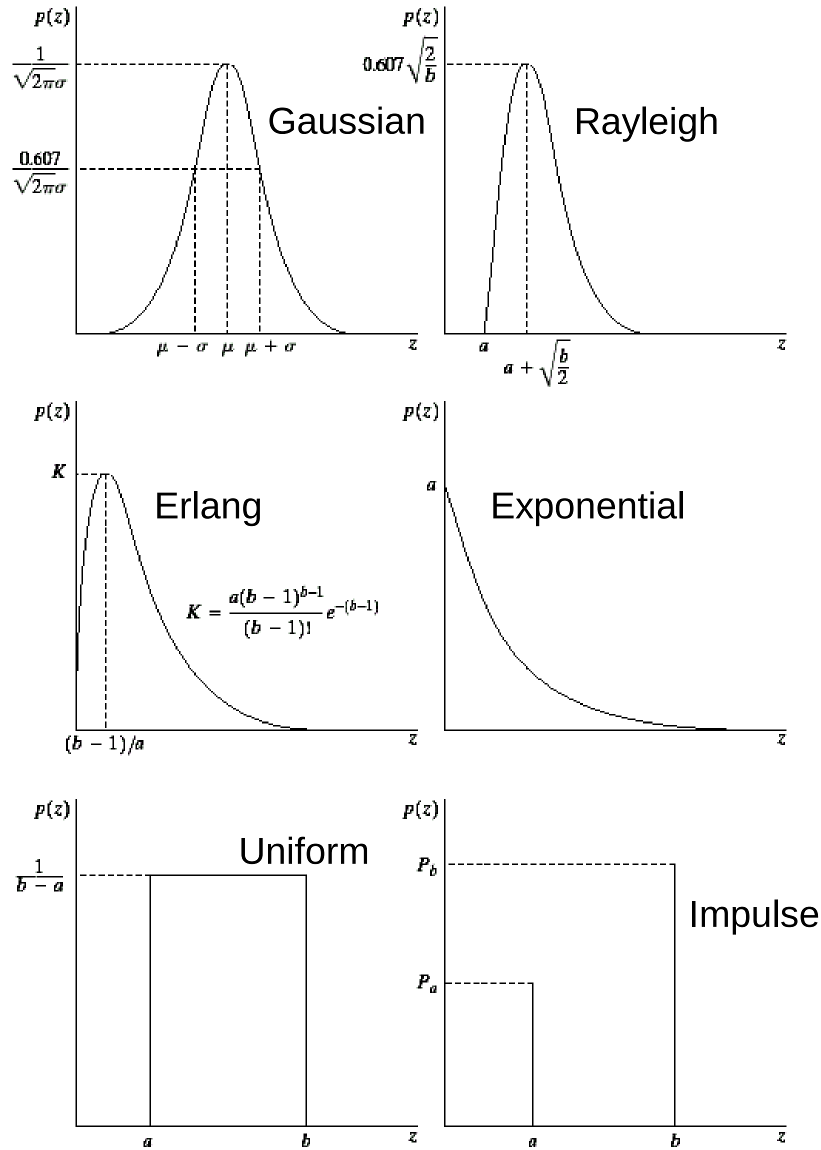
FIGURE 5.1 A model of the image degradation/restoration process.



Noise Models

There are many different models for the image noise term $\eta(x, y)$:

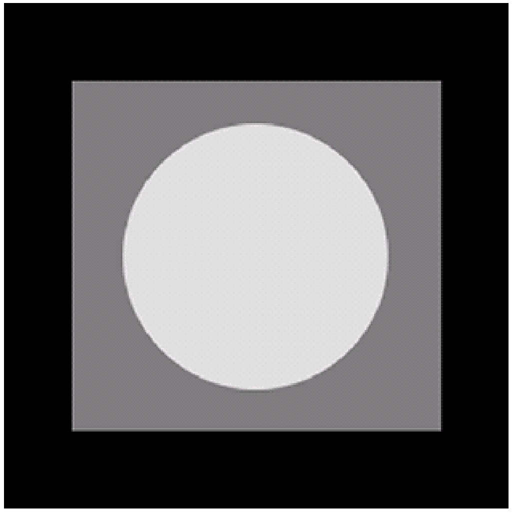
- Gaussian
  - Most common model
- Rayleigh
- Erlang
- Exponential
- Uniform
- Impulse
  - *Salt and pepper* noise



Noise Example

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

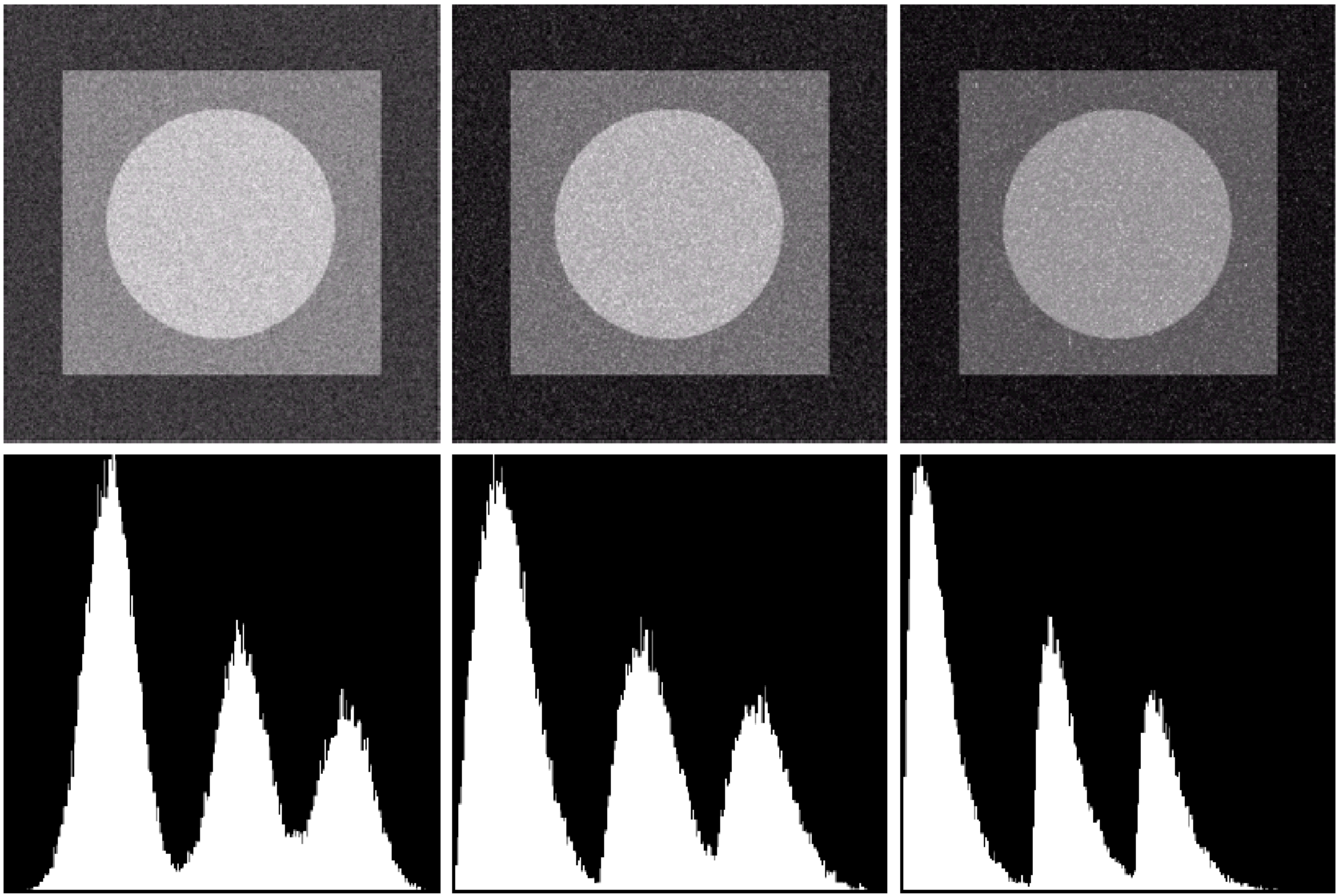
The following slides will show the result of adding noise based on various models to this image



Image

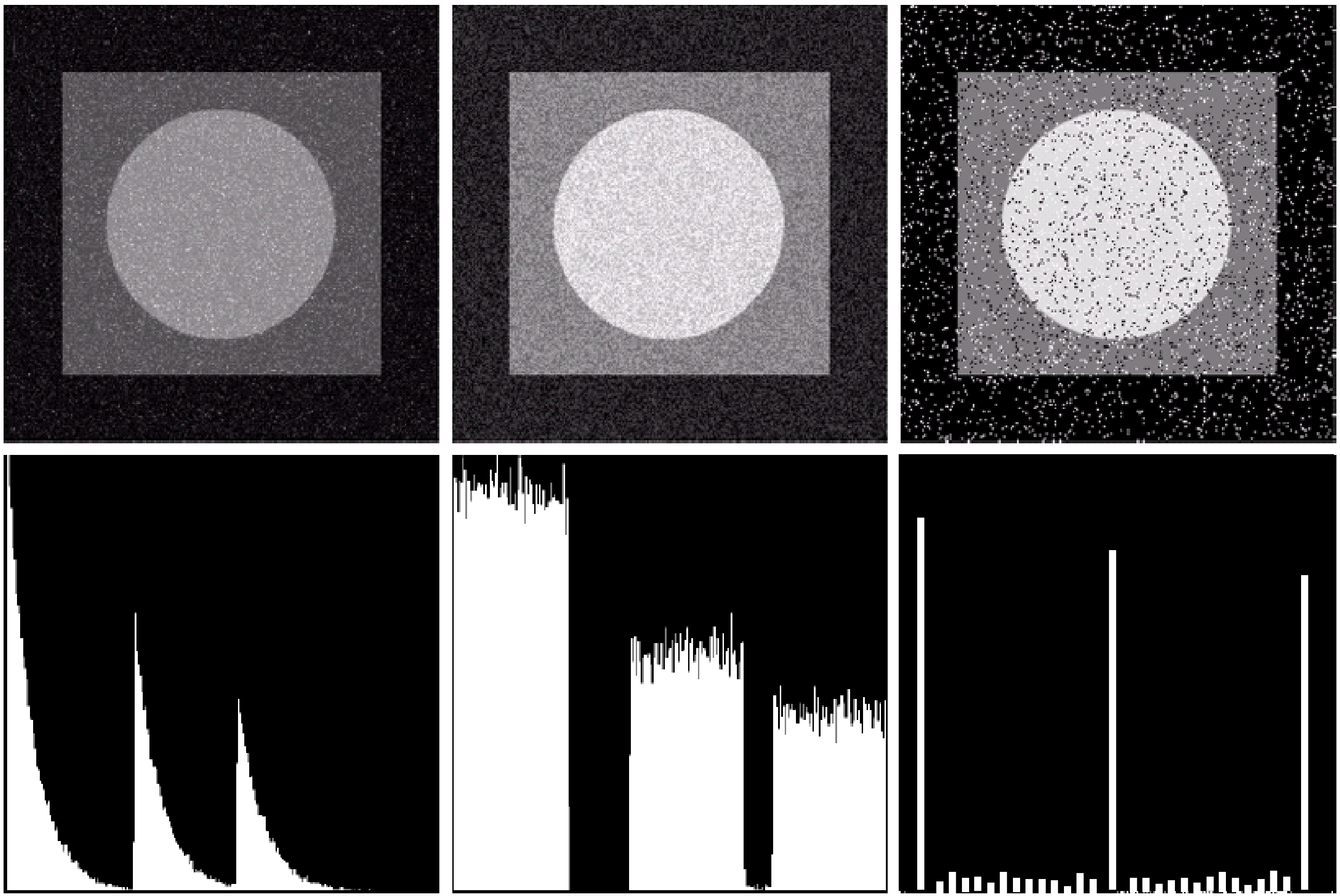
Histogram

Noise Example (cont…)



Gaussian Rayleigh Erlang

Noise Example (cont…)



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:

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

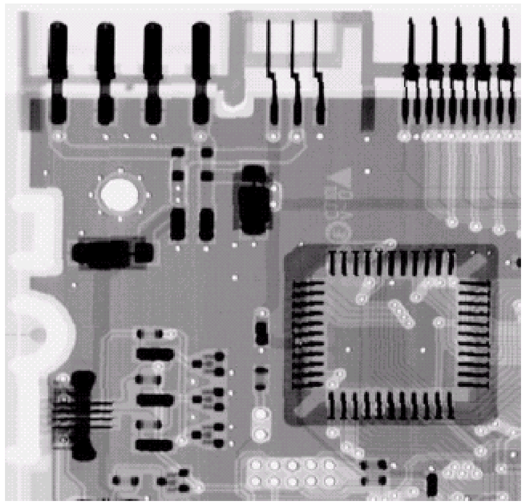
This is implemented as the simple smoothing filter

Blurs the image to remove noise

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

Noise Removal Examples

Original Image



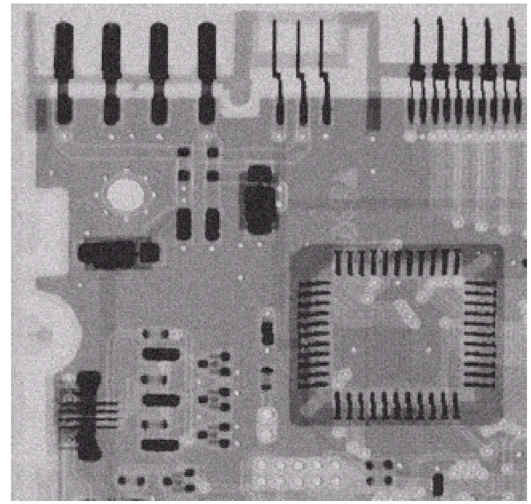
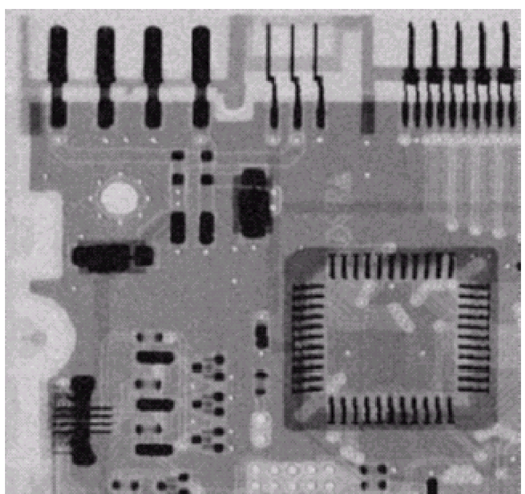
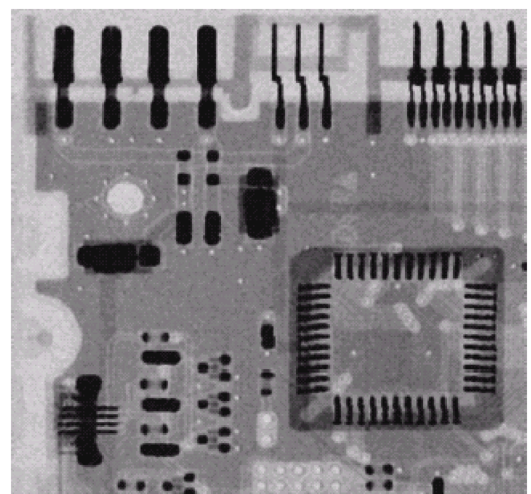


Image Corrupted By Gaussian Noise

After A 3*3 Arithmetic Mean Filter





After A 3*3 Geometric Mean Filter

Mean Filters

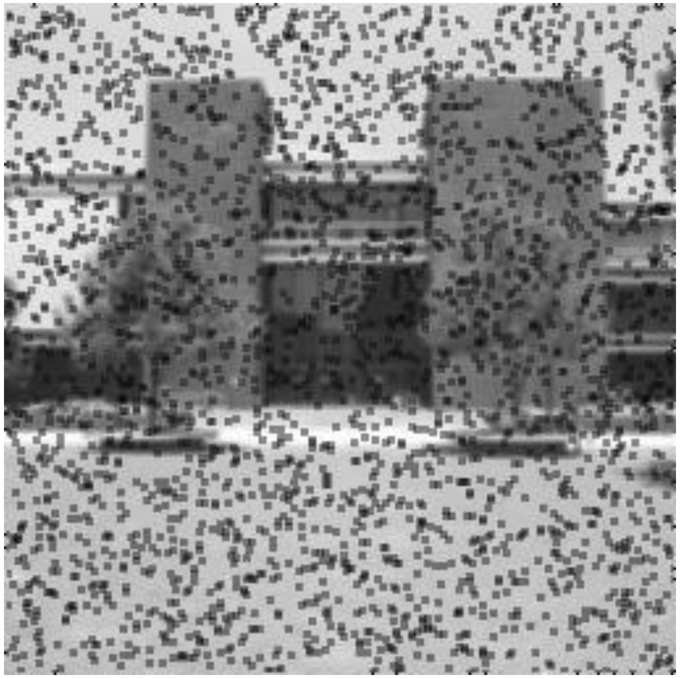
- Geometric mean filter:

$$\text{Geometric Mean} = \prod_{(r,c)\in w} \left[d(r,c)\right]^{\frac{1}{N^2}}$$

  - Works best with gaussian noise.
  - Retains detail better than arithmetic mean filter.
  - Ineffective in the presence of pepper noise (if very low values present in the window, the equation will return a very small number).

Mean Filters

Image with pepper noise
Probability = .04



Result of geometric filter
Mask size = 3

Mean Filters

Image with salt noise
Probability=.04

Result of geometric filter
Mask size = 3

Mean Filters

- Harmonic mean filter:

$$\text{Harmonic Mean} = \frac{N^2}{\displaystyle\sum_{(r,c)\in w} \frac{1}{d(r,c)}}$$

  - Works with gaussian noise.
  - Retains detail better than arithmetic mean filter.
  - Works well with salt noise.

Mean Filters

Image with pepper noise
Probability = .04

Result of harmonic filter
Mask size = 3

Mean Filters

Image with salt noise
Probability=.04

Result of harmonic filter
Mask size = 3

Mean Filters

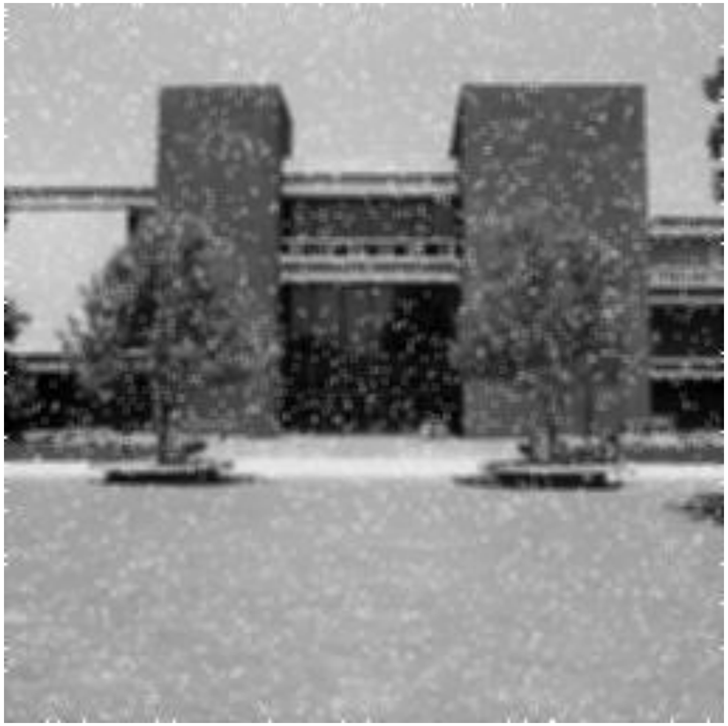
- Contra-harmonic mean filter:

$$\text{Contra-Harmonic Mean} = \frac{\sum_{(r,c)\in w} d(r,c)^{R+1}}{\sum_{(r,c)\in w} d(r,c)^{R}}$$

  - Works for salt OR pepper noise, depending on the filter order R.
  - Negative R → Eliminate salt-type noise.
  - Positive R → Eliminate pepper-type noise.

Mean Filters

Image with salt noise
Probability = .04



Result of contra-harmonic filter
Mask size = 3; order = 0

Mean Filters

Result of contra-harmonic filter
Mask size = 3; order = -1

Result of contra-harmonic filter
Mask size = 3; order = -5

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

Examples

- A 4x4 grayscale image is given by

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

impulse?

impulse?

1) Filter the image with a 3x3 median filter, after **zero-padding**

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

$\xrightarrow{\text{zero-padding}}$

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

$\xrightarrow{\text{median filtering}}$

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

Examples

2) Filter the image with a 3x3 median filter, after **replicate-padding** at the image borders

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

replicate
-padding

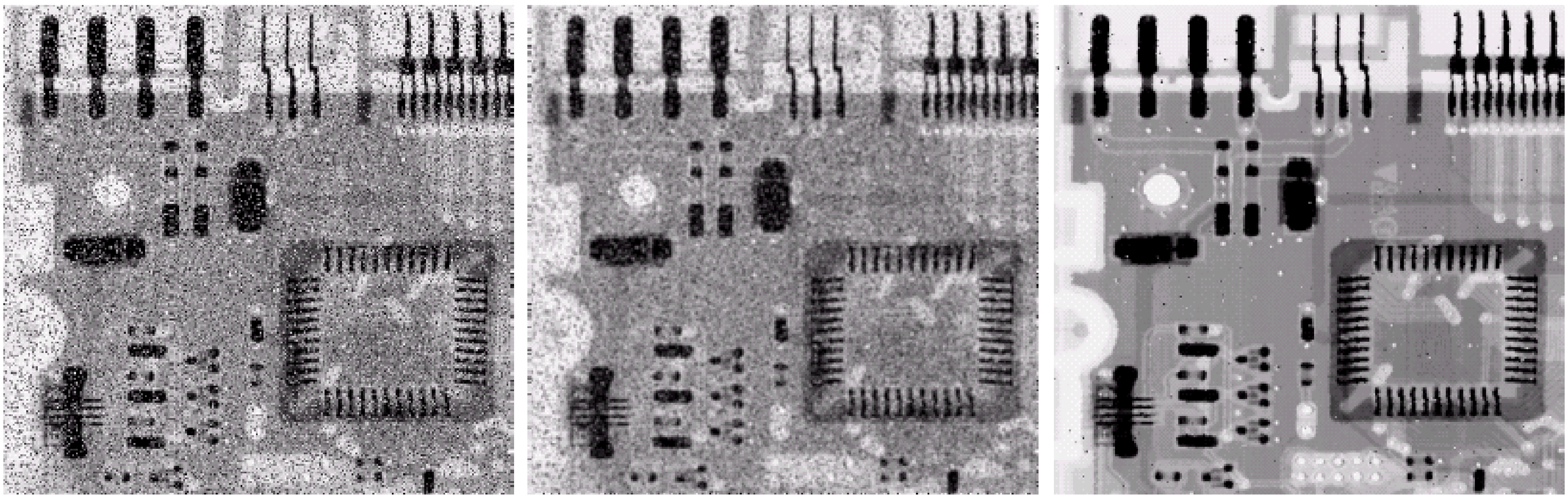
5	5	6	7	8	8
5	5	6	7	8	8
0	0	6	7	8	8
5	5	6	15	8	8
5	5	6	7	8	8
5	5	6	7	8	8

median
filtering

5	6	7	8
5	6	7	8
5	6	7	8
5	6	7	8

impulse cleaned!

Median Filtering

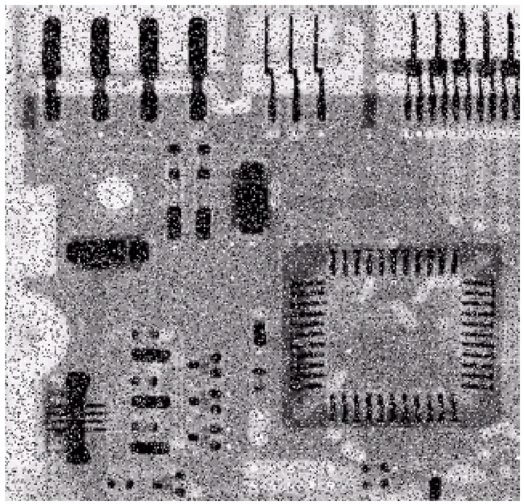


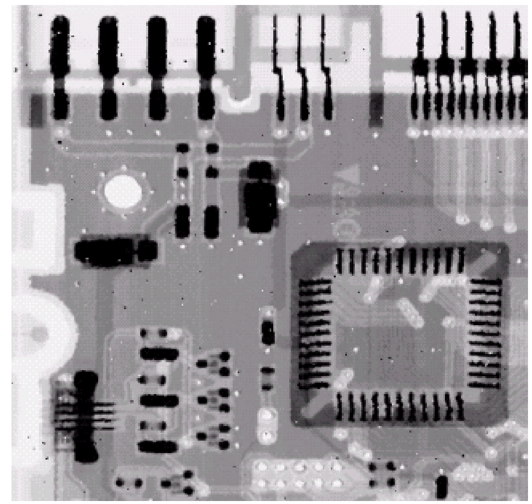
a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Noise Removal Examples

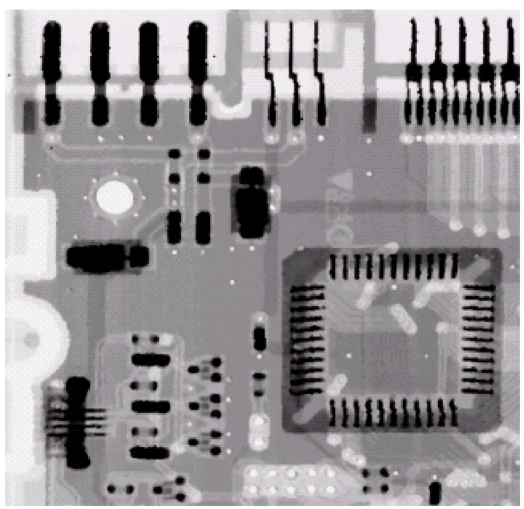
Image Corrupted By Salt And Pepper Noise

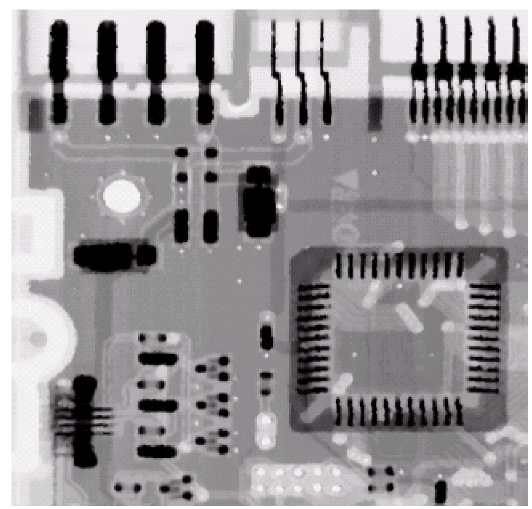




Result of 1 Pass With A 3*3 Median Filter

Result of 2 Passes With A 3*3 Median Filter





Result of 3 Passes With A 3*3 Median Filter

Max and Min Filter

Max Filter:

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

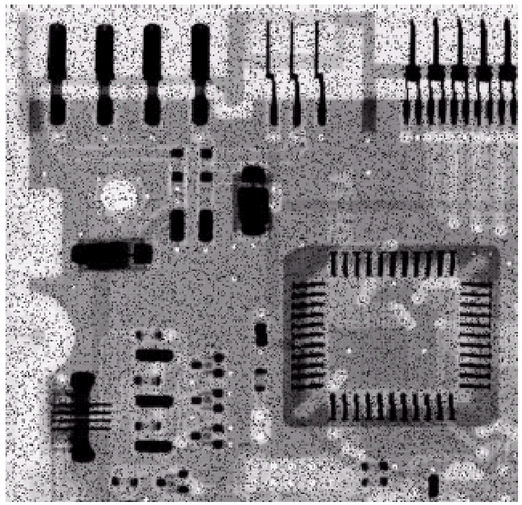
Min Filter:

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

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

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



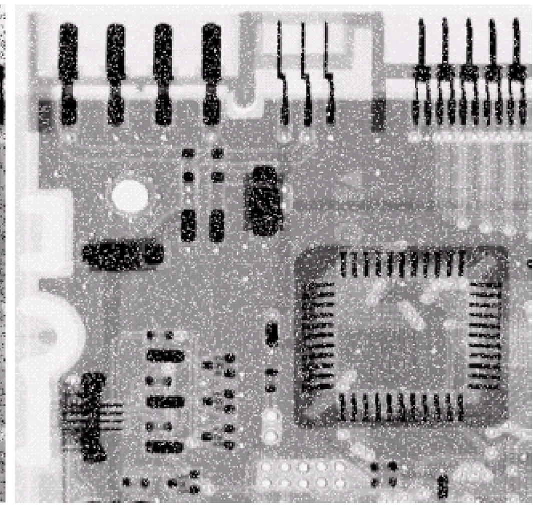
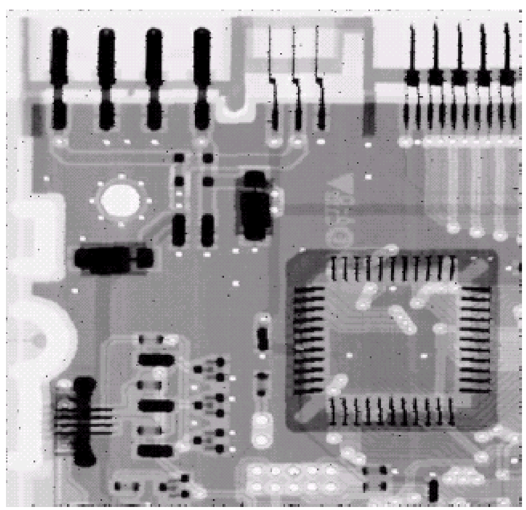
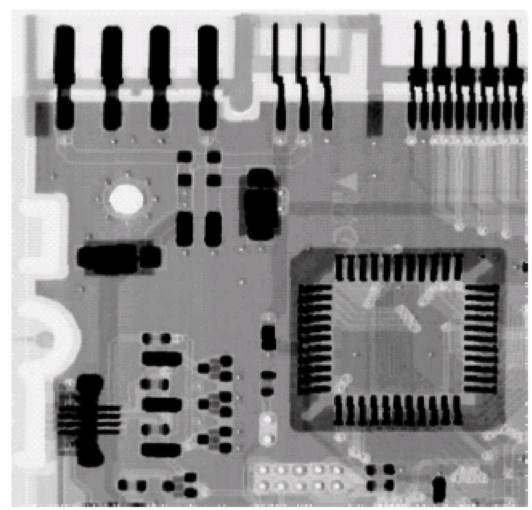


Image Corrupted By Salt Noise

Result Of Filtering Above With A 3*3 Max Filter





Result Of Filtering Above With A 3*3 Min Filter

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

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

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
- If d =0, the filter is reduced to arithmetic mean
- If $d = mn - 1$, the filter become median filter
- For other values, the filter is useful in situation involving multiple types of noise
  - Combination of salt-and-pepper and Gaussian noise

Noise Removal Examples (cont…)

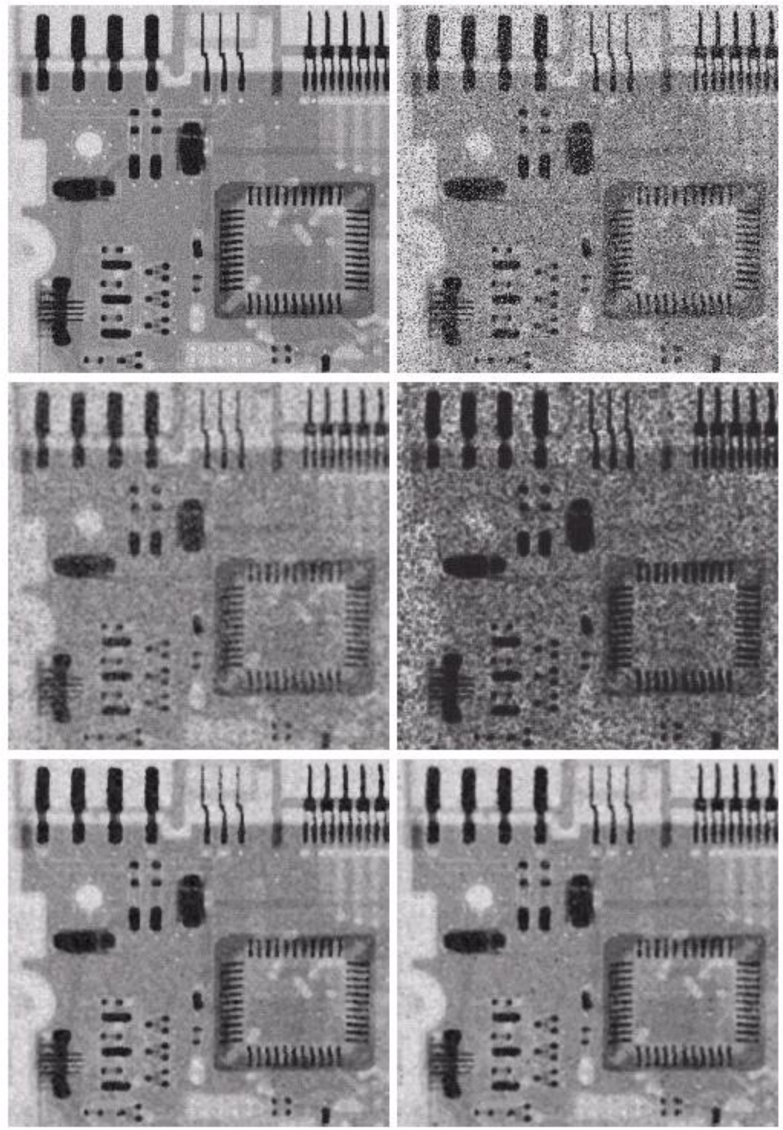


Image Corrupted By Uniform Noise

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

Order Filters

- Order filters can also be defined to select a specific pixel rank within the ordered set.
  - For example, we may find the second highest value is the better choice than the maximum value for certain pepper noise.
  - This type of ordered selection is application specific.
- Minimum filter tend to darken the image and maximum filter tend to brighten the image.

Order Filters

- Midpoint filter:
  - Average of the maximum and minimum within the window.
  - Useful for removing gaussian and uniform noise.

$$\text{Midpoint} = \frac{I_1 + I_{N^2}}{2}$$

Order Filters

Image with gaussian noise.
Variance = 300, mean = 0

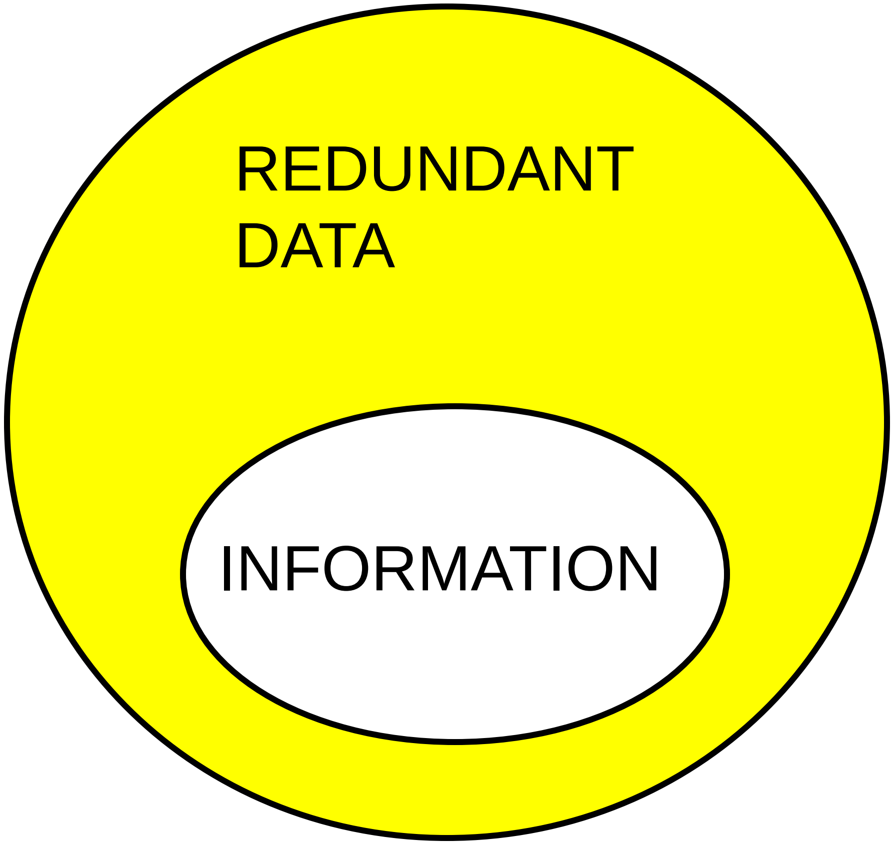
Result of midpoint filter
Mask size = 3

IMAGE COMPRESSION

IMAGE COMPRESSION

- Addresses the problem of reducing the amount of data required to represent a digital image
- The underlying basis of the reduction process is the removal of redundant data

Information vs Data



DATA = INFORMATION + REDUNDANT DATA

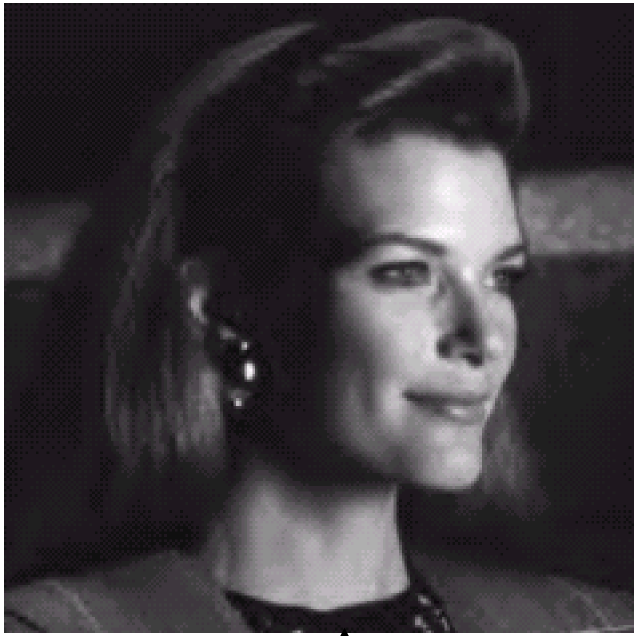
IMAGE COMPRESSION: CODING AND DECODING

original image
262144 Bytes

image encoder

compressed bit stream
00111000001001101…
(2428 Bytes)

image decoder



compression ratio (CR) = 108:1

LOSSY VS LOSSLESS COMPRESSION

- Lossless (Information preserving)
  - Images can be compressed and restored without any loss of information.
  - Application: Medical images
- Lossy
  - Perfect recovery is not possible but provides a large data compression.
  - Example: TV signals, teleconferencing

FUNDAMENTALS

- Raw image: A set of n1 bits
- Compressed image: A set of n2 bits.
- Compression ratio: $$C_R = \frac{n_1}{n_2}$$
- Relative Data Redundancy of first set: $$R_D = 1 - \frac{1}{C_R}$$
- Example: *n1*= 100KB and *n2* = 10Kb, then *CR* = 10 , and *RD* = 90%
- Special cases:
  - n1>>n2 → CR ≈ ∞, RD ≈ 1
  - n1 ≈ n2 → CR ≈ 1, RD ≈ 0

IMAGE COMPRESSION MODEL

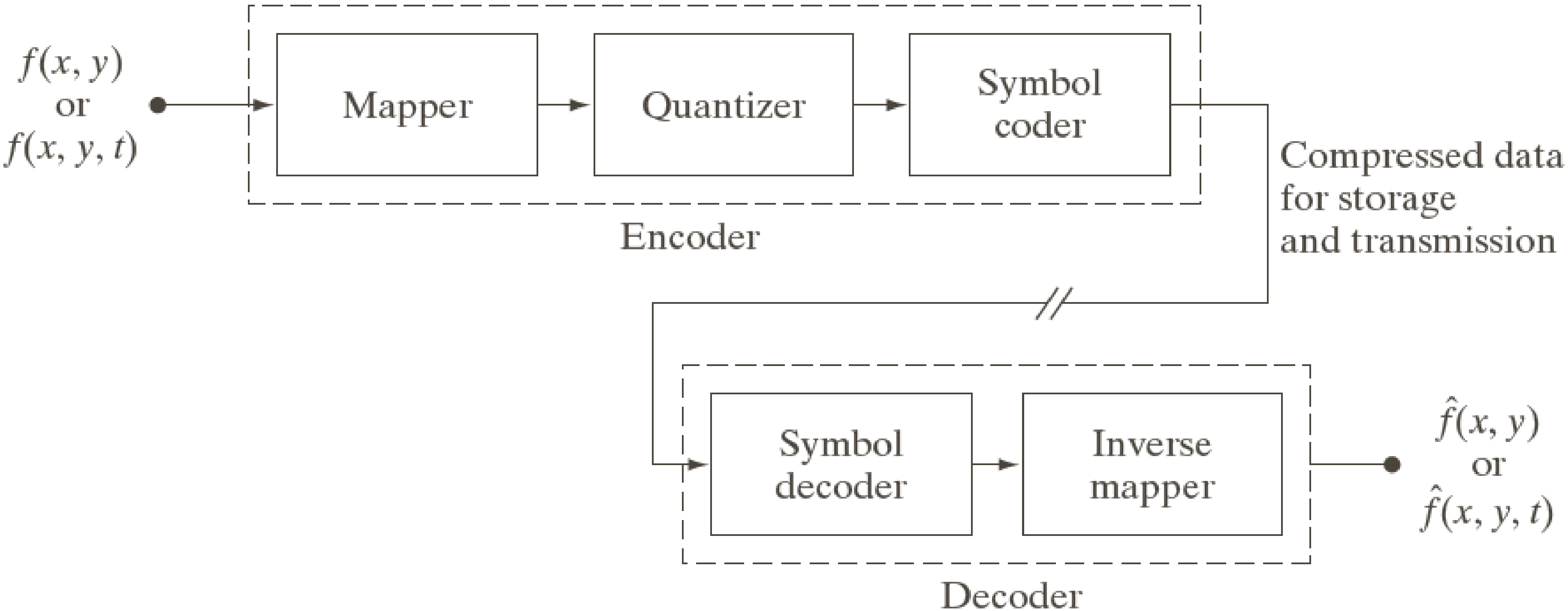


FIGURE 8.5
Functional block diagram of a general image compression system.

DATA REDUNDANCY

- Three basic data redundancies:
  - Coding redundancy
  - Spatial and Temporal redundancy
  - Irrelevant Information

a b c

FIGURE 8.1 Computer generated $256 \times 256 \times 8$ bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

CODING REDUNDANCY

- Type of coding (# of bits for each gray level)
- Image histogram:
  - *rk*: Represents the gray levels of an image
  - *pr*(*rk*): Probability of occurrence of *rk*

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

  - *l(rk)*: Number of bits used to represent each *rk* (after compression)
  - Lavg: Average # of bits required to represent each pixel:

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

CODING REDUNDANCY

- It makes sense to assign fewer bits to those rk for which pr(rk) are large in order to reduce the sum.
- This achieves data compression and results in a variable length code.
- More probable gray levels will have fewer # of bits.
- Basic type is variable length coding

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

VARIABLE LENGTH CODING

r_k	$p_r(r_k)$	code1	$l_1(r_k)$	code2	$l_2(r_k)$
$r_0=0$	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

VARIABLE LENGTH CODING

- Computing L_{avg}

$$L_{avg} = \sum_{k=0}^{7} l_2(r_k)\, p_r(r_k)$$

=2(0.19)+2(0.05)+2(0.21)+3(0.16)+4(0.08)+5(0.06)+6(0.03)+6(0.02)

=2.7 bits

- C_R=3/2.7=1.11
- R_D=1-1/1.11=0.099=9.9%

Readings from Book (3rd Edn.)

- Image Restoration (Chapter-5)
- Reading Assignment
  - Adaptive Filters
  - Adaptive Median Filtering

Acknowledgements

Material in these slides has been taken from, the following resources

- Digital Image Processing", Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002
- Peters, Richard Alan, II, Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008
- Brian Mac Namee, Digitial Image Processing, School of Computing, Dublin Institute of Technology
- Computer Vision for Computer Graphics, Mark Borg