- Restoration is also an image enhancement technique, here we can perform image enhancement both in spatial and frequency domain.

A Model of the Image Degradation/Restoration Process

we have two stages in image degradation & restoration.

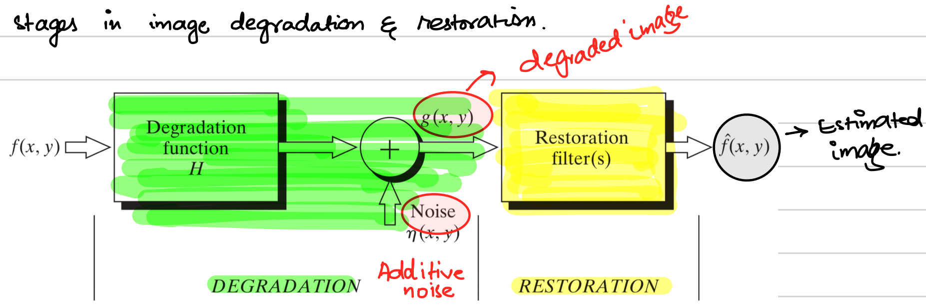


FIGURE 5.1 A model of the image degradation/restoration process.

Estimated image must be as close as original image. It is only possible if $g(x,y) \rightarrow$ degraded image has sufficient knowledge of the H (degradation function) as well as noise (η).

This can be mathematically represented as.

In Spatial domain:

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

In frequency domain:

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

- **Noise models:** The primary source of noise in digital image arise during the image acquisition and/or transmission. When we capture an image using digital camera or any other device, sometimes the noise is added to the acquired image and this is due to the various factors such as: performance of the sensors, light levels, sensor temperature, wrong adjustment of lens aperature. Images are also get corrupted during transmission due to the interference in the channel used for transmission. So, these noises should be eliminated or removed to improve the quality of the images.

Spatial and Frequency Properties of Noise

Some of the noises are directly present in the images which can be dealt in spatial domain & some of the noises are related to the frequencies. Such type of the image can be filtered using frequency domain filters.

For example AWGN, whenever we operate that in frequency domain, it remains constant & uniform and easy to eliminate. Depending upon type of noise, we choose the domain for noise removal.

There are various types of noise.

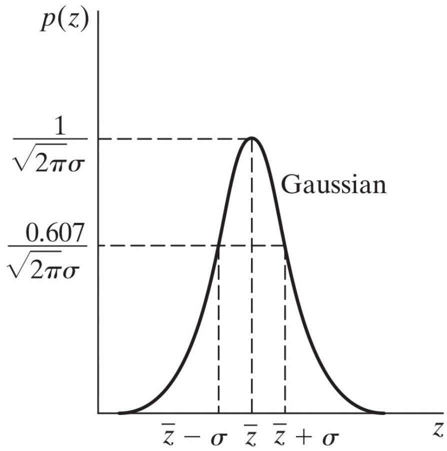
Noise Models: Important noise probability density functions:

(i) **Gaussian noise :-** The PDF of Gaussian noise is :

$$P(z) = \frac{1}{\sqrt{2\pi}\,\delta} e^{-(z-\bar{z})^2/2\delta^2}$$

$\therefore$ z = Intensity, $\bar{z}$ = Mean Value of z

, δ = S.D, δ^2 = variance

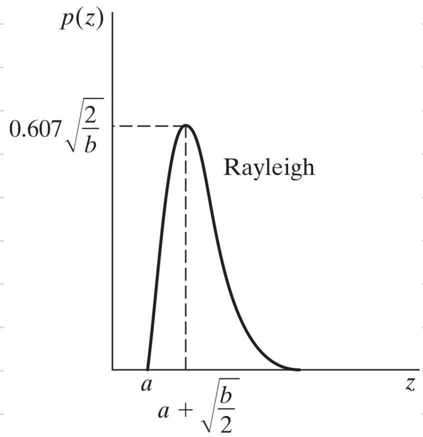


(ii) **Rayleigh noise:** The PDF of rayleigh noise is :

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

$$\bar{z} = a + \sqrt{\pi b/4}$$

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

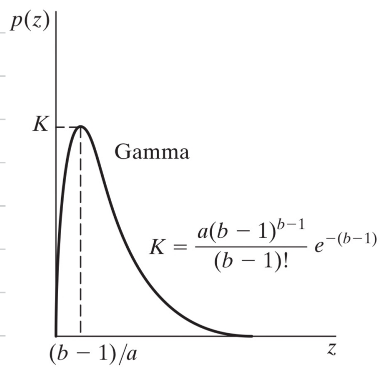


(iii) **Erlang (Gamma) noise:**

The PDF of Erlang noise is :

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

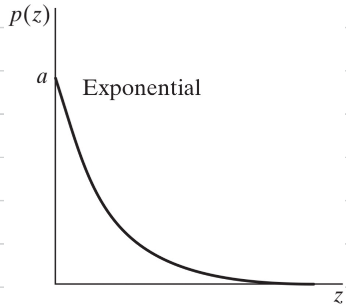
$$\bar{z} = b/a, \quad \delta^2 = b/a^2$$



(iv) **Exponential noise :** The PDF is :

$$P(z) = \begin{cases} a e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

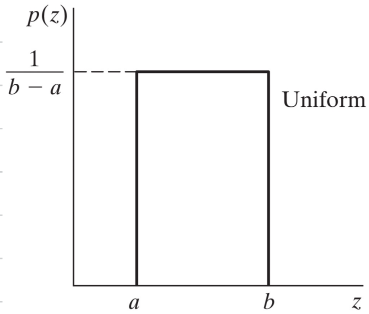
$$\bar{z} = 1/a, \quad \delta^2 = 1/a^2$$



(v) **Uniform noise :** The PDF is :

$$P(z) = \begin{cases} \frac{1}{b-a} & a \leq z \leq b \\ 0 & \text{else where} \end{cases}$$

$$\bar{z} = \frac{a+b}{2}, \quad \delta^2 = \frac{(b-a)^2}{12}$$



(vi) **Impulse noise :** (Salt & pepper noise)

The PDF is :

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

if $b > a$, then b is a light dot in an image.
if $b < a$, then b is a dark dot in an image.
if P_a or $P_b = 0$ the impulse noise is called unipolar.
if neither P_a or $P_b = 0$ then it is called "bipolar impulse noise".

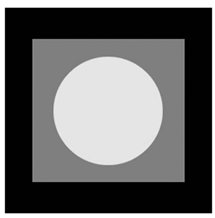


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

318 **Chapter 5** ■ Image Restoration and Reconstruction

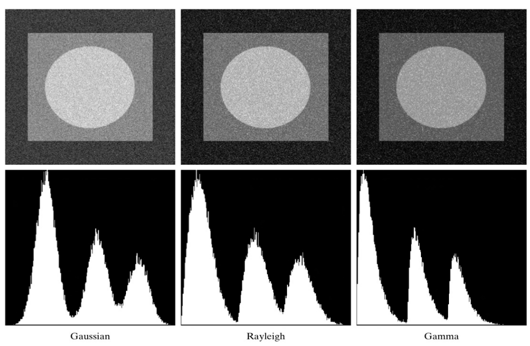


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

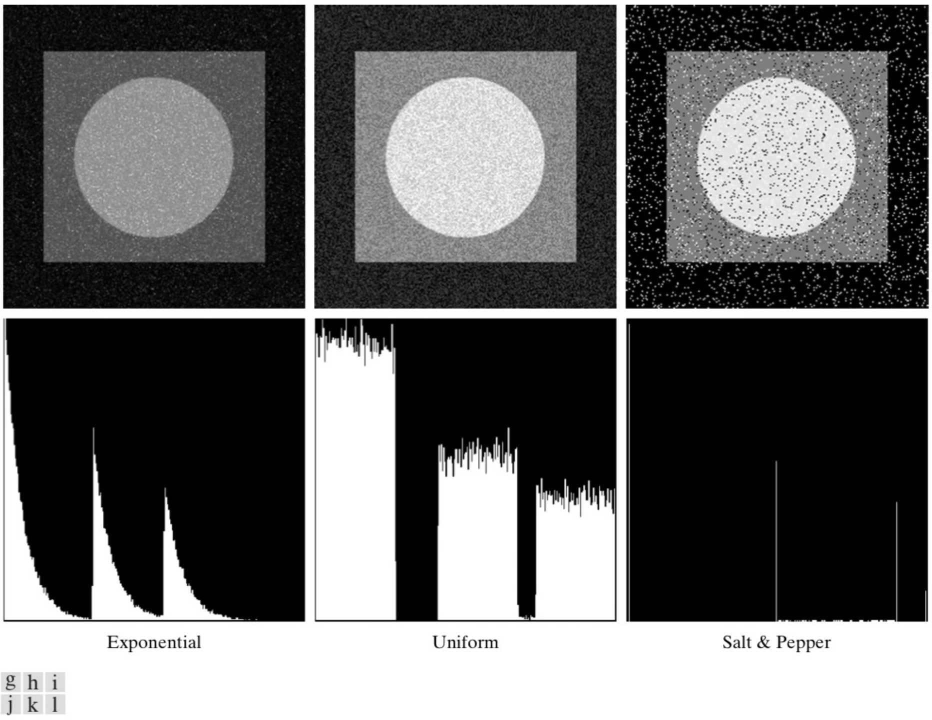


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt-and-pepper noise to the image in Fig. 5.3.

- **Periodic Noise :** Periodic noise arises typically from the electrical or electromechanical interference during image acquisition. This is the only type spatially dependent noise we will be considering. For example sine noise of varying frequencies. Such noise can be handle better in frequency domain because the sinusoidal signal in frequency domain is represented as the pair of conjugate impulses located at the conjugate frequencies of the sine wave.

- **Estimation of Noise parameters:**

Noises can be estimated with two parameters called mean & variance

$$\bar{z} = \sum_{i=0}^{L-1} z_i \, p_s(z_i)$$

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 \, p_s(z_i)$$

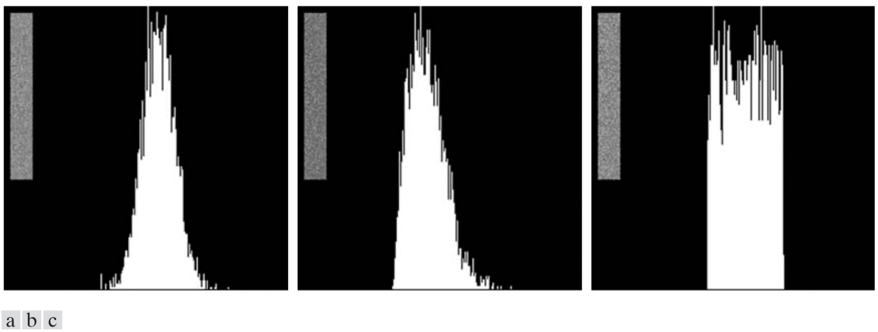


FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

Restoration in the presence of noise only – Spatial filtering

When the only degradation present in an image is noise then

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

and

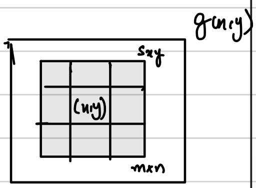
$$G(u,v) = F(u,v) + N(u,v)$$

we will use spatial filtering to reduce/remove the noise. This is quite challenging task as subtracting $\eta(x+y)$ from $f(x,y)$ will not estimate the $\hat{f}(x,y)$ as $\eta(x+y)$ is unknown.

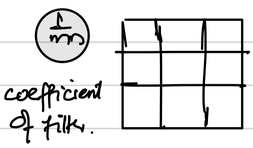
There are numerous filters we can use to restore/estimate the original image.

Mean Filters:

– **Arithmetic mean:** Suppose we have a degraded image $g(x,y)$, and we choose the neighborhood in this represented as S_{xy}. The response of the filter will be placed at particular (x,y) i.e., centre and the filter moves on overall image.



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



- Smoothen the noise
- Removes noise by blurring.

(ii) **Geometric mean:** Estimation of $\hat{f}(x,y)$ is done in similar fashion. The response of the filter can be computed using GM such as:

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

- Smoothing image better than AM filter.

but
- fine details can't be retained.

(iii) **Harmonic mean filter:**

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

- Best for salt noise
- failed for pepper noise
- Good for Gaussian noise

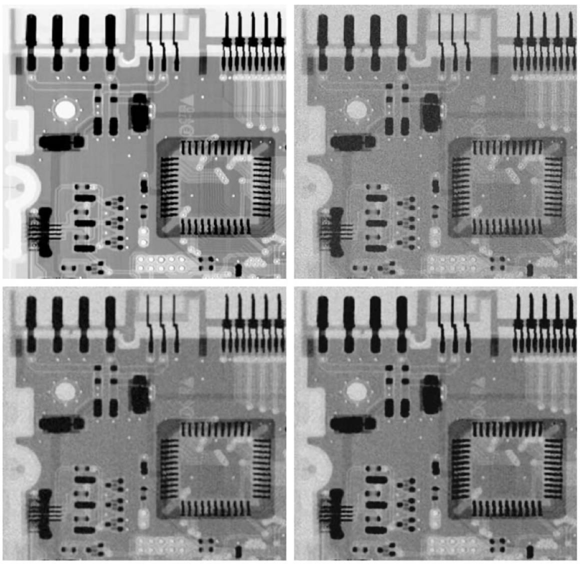
(iv) Contraharmonic mean filter:

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

∴ Q is the order of the filter.

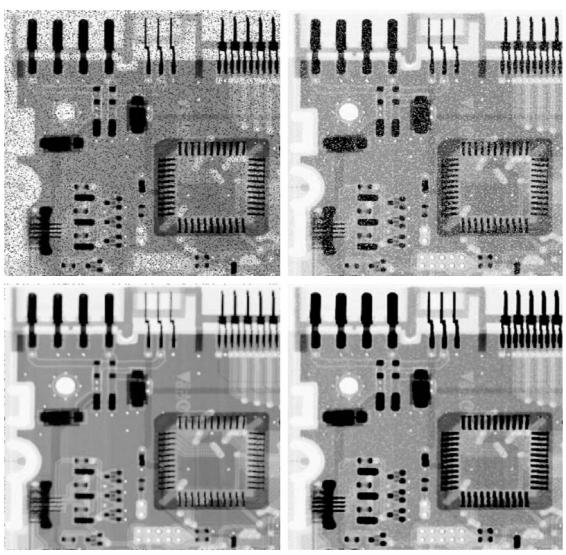
- Best for both salt & pepper noise.

Q = +ve removes pepper noise.
Q = −ve " salt "
Q = 0 → AM
Q = −1 → HM



a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)



a b
c d

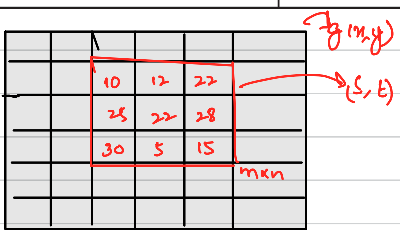
FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3 × 3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

- **Order-statistic filters:** These are spatial filters, where the response of these types of filters depends on the ordering/ranking the pixel values of the neighbourhood in an image.

(i) Median filter

Best known order-statistic filter

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}} \{g(s,t)\}$$ → 50 percentile



- Reduce impulse noises

median
[5, 10, 12, 15, 22, 22, 25, 28, 30]

100 percentile.
0 "

(ii) Max & min filter :

100th percentile $\rightarrow$ $\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$ $\therefore$ select the max intensity values

- Reduce the pepper noise
- identify the brightest pixel in an image.

0th percentile $\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$ $\therefore$ select the min intensity pixel value.

- identify the darkest pixel in an image
- reduce the salt noise

(iii) Mid point filter : Use to identify the mid intensity values b/w max & min.

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

- Reduce the Gaussian noise & uniform noise.

(iv) Alpha-trimmed mean filter: initially we remove $d/2$ lowest pixels and $d/2$ highest intensity pixels and we left with $(mn-d)$ pixels represented as $g_r(s,t)$

$\therefore$ So $g_r(s,t) = (mn-d)$

The Alpha trimmed mean filter now consider the remaining pixels $g_r(s,t)$ & takes the average of these pixels.

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

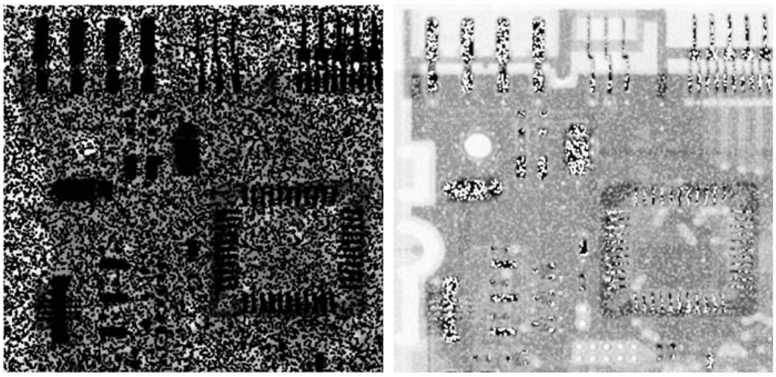
The value of d is from 0 to $mn-1$

if $d=0$, $\rightarrow$ AM filter
if $d=mn-1$, $\rightarrow$ median filter.

- reduce salt & pepper noise
- reduce Gaussian noise

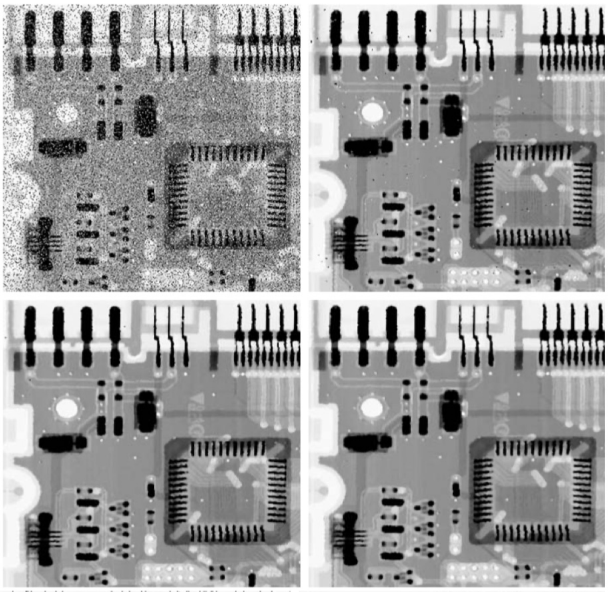
a b

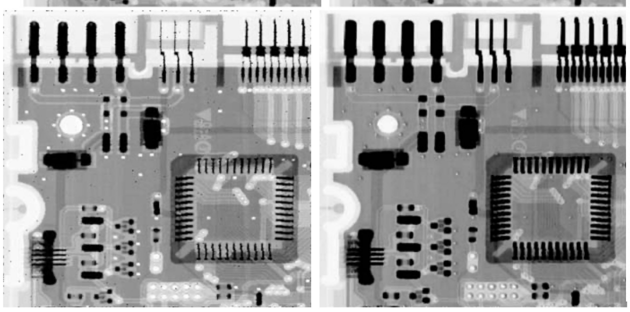
FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.



a b
c d

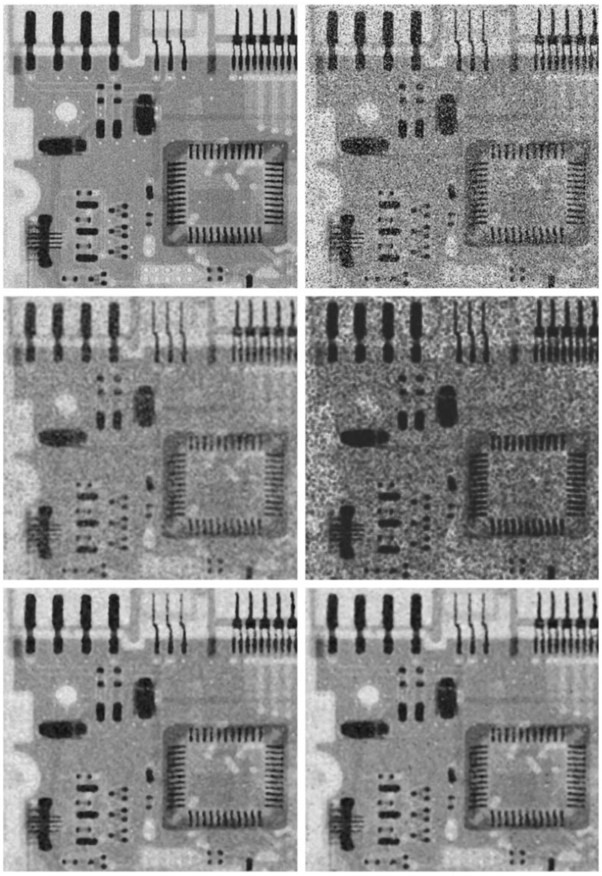
FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3 × 3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.





a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image (b) filtered with a 5 × 5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

- ADAPTIVE FILTERS: These filters changes the behavior depending on the statistical behavior of the neighbourhood in an image. They are very superior than the other filter we discussed but they are more complex.

Adaptive filters

Adaptive local noise reduction filter

Adaptive median filter

(i) **Adaptive local noise reduction filter:** we perform the filtering on the local area i.e., neighbourhood S_{xy}. The R.V are identified by the statistical measures. The most popular statistical measures are mean & variance. The mean calculates the average of the intensities present in the neighbourhood and the variance is used to identify the contrast of the neighbourhood. The response of this kind of filter depends on the 4 quantities.

(a) $g(x,y)$ is the value of the noisy image at (x,y).

(b) $\delta_\eta^2 \rightarrow$ variance of noise added with $f(x,y)$ to form $g(x,y)$ $\eta(x+y)$

(c) $m_L \rightarrow$ local mean of all the pixels in S_{xy}

(d) $\delta_L^2 \rightarrow$ local variance of the pixels in S_{xy}.

Conditions:

(i) if $\delta_\eta^2 = 0$, means no noise $f(x,y) = g(x,y)$

(ii) if $\delta_L^2 > \delta_\eta^2$, $f(x,y)$ is very closer to $g(x,y)$

(iii) If $\delta_L^2 = \delta_\eta^2$, The response of this filter is same as AM filter.

The ALNF can be represented as:

$$\hat{f}(x,y) = g(x,y) - \frac{\delta_\eta^2}{\delta_L^2}\{g(x,y) - m_L\}$$

(ii) **ADAPTIVE MEDIAN FILTER:**

Median filter works well only when the impulse noise is very less. i.e., P_a & P_b less than 0.2. This disadvantage can be overcome by using Adaptive median filter. This filter even works for the larger values of impulses. It also preserves the details while smoothing non-impulse noise.

- Let Z_{min} = min value of gray level in S_{xy}.

  Z_{max} = max " " " " "

  Z_{med} = med " " " " "

  S_{max} = max allowed size of S_{xy}.

  Z_{xy} = gray value at coordinates (x,y)

The Algorithm works in two stages.

Stage A:

$A1 = Z_{med} - Z_{min}$

$A2 = Z_{med} - Z_{max}$

If $A_1 > 0$ & $A_2 < 0$, Go to stage B else increase the window size.
if window size $\leq S_{max}$, repeat stage A.
else output is Z_{xy}.

Stage B:

$B1 = Z_{xy} - Z_{min}$

$B2 = Z_{xy} - Z_{max}$

If $B1 > 0$ & $B2 < 0$ then output Z_{xy}.
else the output is Z_{med}.

This algorithm is used to

- remove salt & pepper noise
- provide smoothing

Stage A determines if the output of the median filter Z_{med} is an impulse or not (black or white). If it is not an impulse, we go to stage B. If it is an impulse, the window size is increased until it reaches S_{max} or Z_{med} is not an impulse.

Note: There is no guarantee that Z_{med} will not be an impulse. The smaller the density of the noise is, the larger the support S_{max}, & we expect not to have an impulse.

Stage B determines if the pixel value at (x,y), that is Z_{xy}, is an impulse or not (black or white).
If it is not an impulse, the algorithm output the unchanged pixel value Z_{xy}.
If it is an impulse, the algorithm outputs the median Z_{med}.