4 – NOISE REDUCTION

Smoothing operations are used for reducing noise and other spurious effects that may be present as a result of sampling, quantization, transmission, or disturbances in the environment during image acquisition.

Noise is classified based on the distribution of its intensity. Some common types of noise are:

- *Uniform noise*: The noise values are distributed with equal probability within a finite range of values.
- *Gaussian noise*: The intensity values follow Gaussian distribution. It is a very good model for many kinds of sensor noise.
- *Salt-and-pepper noise*: The image contains random occurrences of both black and white intensity values. Salt-and-pepper noise belongs to the family of noise called outlier noise, because the noise values that are generated deviate far beyond the values that are normally expected.
- *Impulse or salt noise*: Contains only random occurrences of white intensity values.

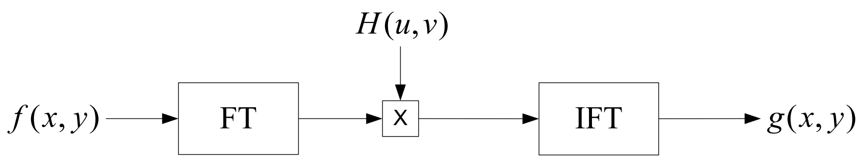
(a) contaminated with Gaussian noise

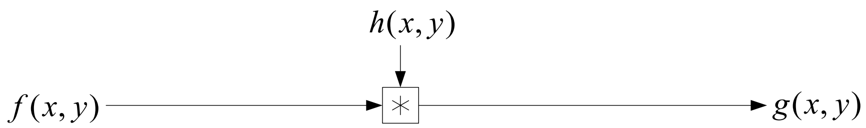
(b) contaminated with salt-and-pepper noise

Smoothing may be implemented as filtering in the frequency domain or as convolution in the spatial domain.

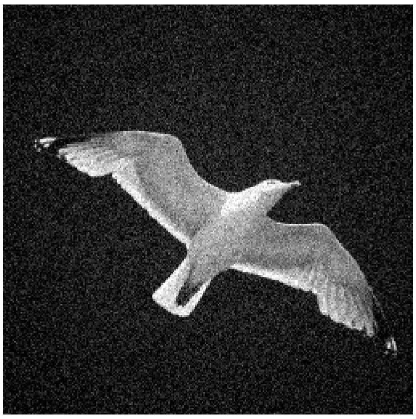
(a) *Frequency domain approach*: FT of image is multiplied by the filter transfer function. The IFT is performed on the product.

(b) *Spatial domain approach*: The image is convolved with the filter impulse response or a suitable mask.

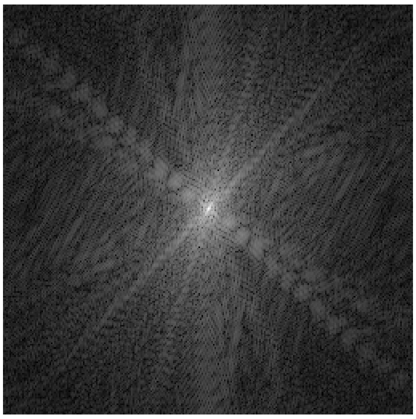




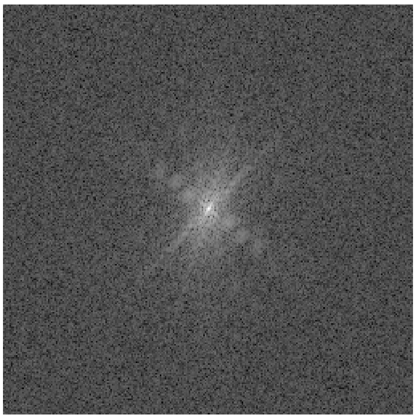
Original image



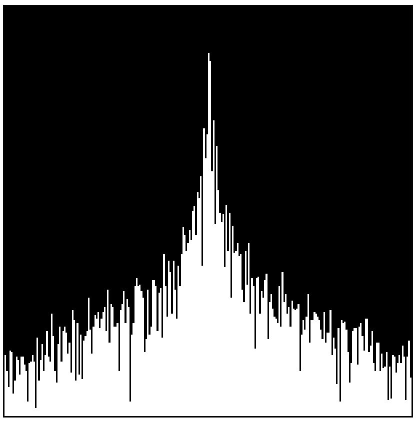
Noisy image



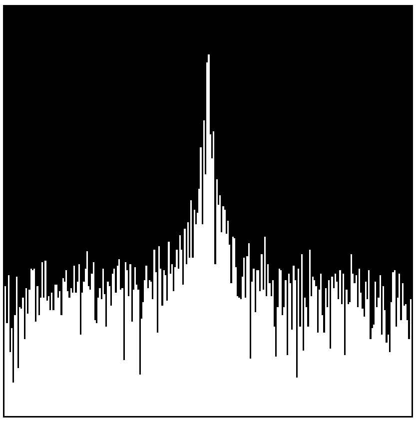
FT



FT



Profile



Profile

MEAN FILTERING

The filtering operation of an image $f(i,j)$ with the filter function (or mask) $h(i,j)$ of size $M \times M$ is written as

$$\begin{aligned} g(i,j) &= h(i,j) \star f(i,j) & (1) \\ &= \sum_{m,n=-M/2}^{m,n=M/2}\sum h(m,n) f(i-m, j-n) & (2) \end{aligned}$$

Mean filtering (or neighbourhood averaging) is achieved by setting all the filter coefficient values to 1/9. The filter weight of $\frac{1}{9}$ is introduced so that an intensity bias is not introduced into the processed image.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Other commonly used noise-cleaning masks are:

$$\frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Although noise is suppressed by mean filtering, the signal is also affected. This is manifested in the form of blurring.

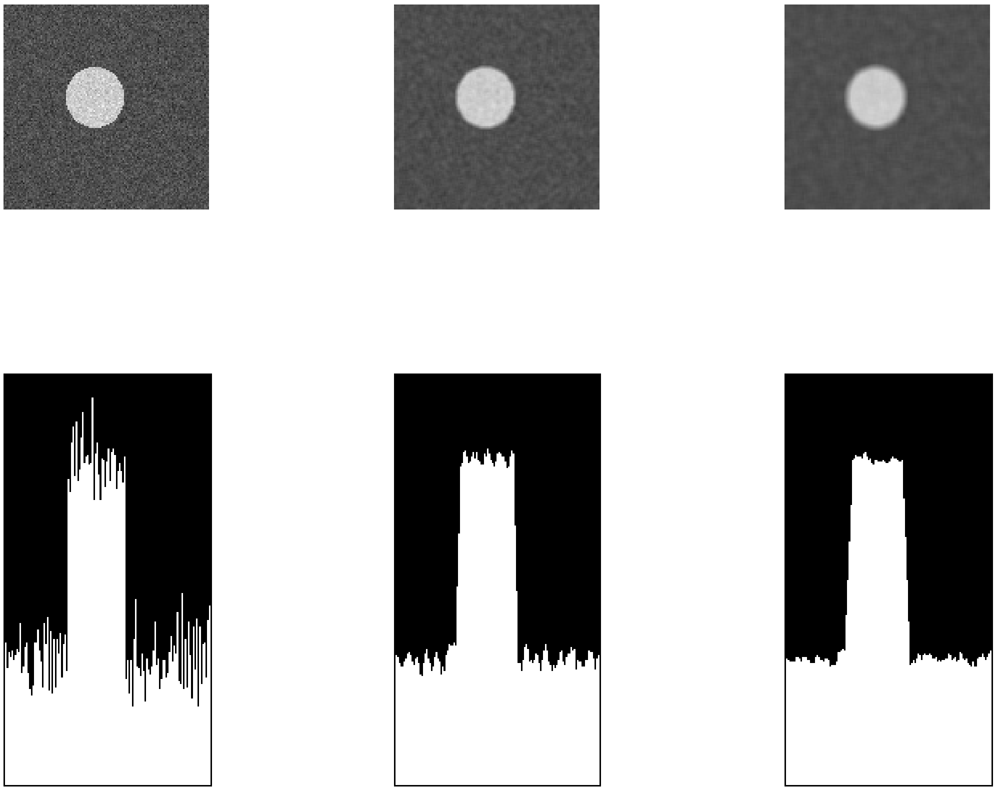
Masks bigger than 3×3 may be used. A larger convolution mask will result in greater noise reduction and also greater loss of image detail.

Filtering can also be done with a Gaussian mask, in which the mask coefficients are samples from a 2D Gaussian function.

$$h(x,y) = \exp\left[-\frac{(x^2+y^2)}{2\sigma^2}\right]$$

Example of a 5×5 Gaussian mask:

$$\frac{1}{121}\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 7 & 11 & 7 & 2 \\ 3 & 11 & 17 & 11 & 3 \\ 2 & 7 & 11 & 7 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$



Noisy image | Mean filtering (3×3) | Mean filtering (5×5)

Threshold Averaging

Standard mean filtering may be modified to reduce blurring. The output is not automatically set to the averaged value. Instead, the local mean value $m_l(x, y)$ is first compared with the original gray level $f(x, y)$; the output image is then

$$g(x,y) = \begin{cases} m_l(x,y) & \text{if } |f(x,y) - m_l(x,y)| < T \\ f(x,y) & \text{otherwise} \end{cases} \tag{3}$$

where T is a specified non-negative threshold. Blurring is reduced because we leave unchanged regions with large variations in gray level.

Example

(a)

21	19	17	23	3	0	1	0
20	20	21	20	1	1	1	2
22	18	21	22	0	2	1	3
17	21	19	20	3	0	1	1
20	19	23	18	1	2	1	0
20	18	19	20	1	3	1	2

(b)

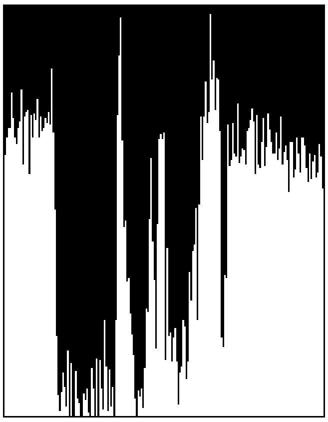
	20	20	14	8	1	1	
	20	20	14	8	1	1	
	20	20	14	8	1	1	
	19	20	14	8	1	1	

(c)

	20	20	20	1	1	1	
	20	20	22	0	1	1	
	20	20	20	3	1	1	
	19	20	18	1	1	1	

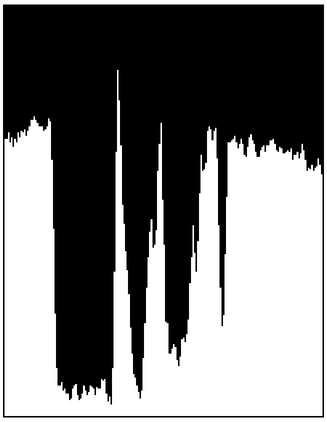
(a) Original (b) After mean filtering (c) After threshold averaging ($T = 4$)

Noisy image



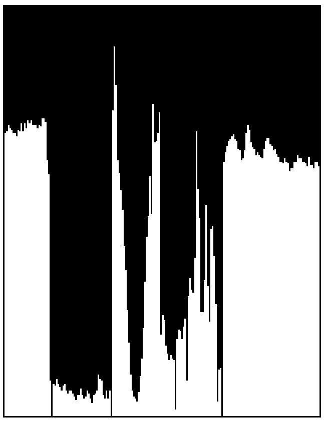
Profile

After mean filtering



Profile

After threshold averaging



Profile

MEDIAN FILTERING

To preserve edges and details, median filtering may be more appropriate than mean filtering. The gray level of each pixel is replaced by the median of the intensities in a predefined neighbourhood of this pixel. This technique is particularly useful for salt and pepper noise and impulse noise.

Given a set of N pixel intensities obtained over a local image region, S, denoted as z_i, $i = 1, 2, \ldots, N$, the ordering of these values in increasing value is $R(\mathbf{x}) = \{z_1, z_2, \ldots, z_N\}$, where $z_i \leq z_{i+1}$. The output is the median of the pixel intensities:

$$g(\mathbf{x}) = \mathrm{med}(R(\mathbf{x})). \tag{4}$$

In a 3×3 neighbourhood, the output is the 5th value in the ordered sample set.

Example:

75	99	36
38	49	10
19	98	22

The median is $g(\mathbf{x}) = 38$.

Since the median is the $(N+1)/2$ largest value ($N =$ odd), its search requires

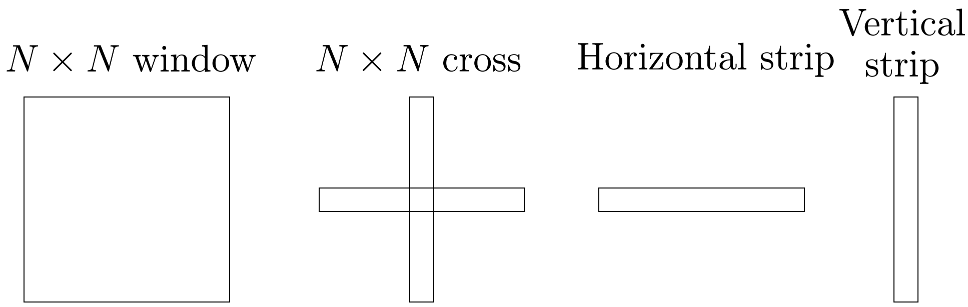
$$(N-1) + (N-2) + \cdots + (N-1)/2 = 3(N^2-1)/8 \text{ comparisons} \tag{5}$$

with the bubble sort algorithm. This number equals 30 for 3×3 windows and 224 for 5×5 windows. (More efficient algorithms are available.)

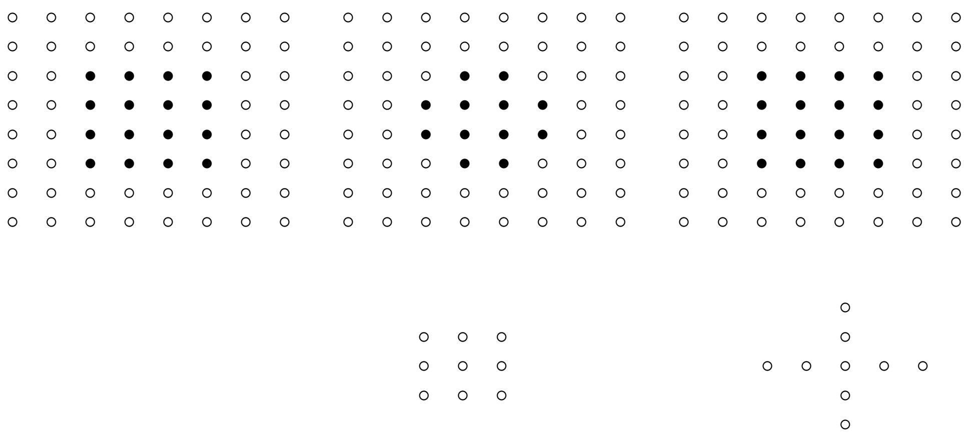
A method of reducing the computational load is to remove the appropriate values from the sorted list corresponding to the discarded pixels as the window is moved to a new location. The new pixel values are then inserted at the correct positions in the list and the median extracted.

Properties

- The window shape and size greatly determines the output. It may be chosen based on a *a priori* knowledge of the image noise characteristics.



- The shape chosen for the filter may affect the processing results. A 2D $L \times L$ filter results in more noise suppression than processing with a cross-shaped filter, but also results in greater signal suppression.



- The median filter reduces the variance of the intensities in the image.
- No new gray values are generated; binary images remain binary, the dynamic range of the filtered image cannot exceed that of the input image.
- In general, the median filter allows a great deal of high spatial frequency detail to pass while remaining very effective at removing noise.

With Gaussian noise

With salt & pepper noise

After mean filtering

After mean filtering

After median filtering

After median filtering

Non-linear Spatial Filtering

Non-linear filters based on order statistics require that all the pixels be first ordered from their minimum to their maximum values:

$$z_1 \leq z_2 \leq \ldots \leq z_{N-1} \leq z_N$$

The median filter is an example of such a filter; in this case the output is

$$\text{output} = z_{(N+1)/2} \qquad \text{for } N \text{ odd}$$

There are several filters in this family that have been found to be useful in noise reduction.

(a) Weighted median filter

One of the disadvantages of the median filter is that it removes one-pixel wide lines from an image. For example, applying the 3×3 median filter to this subimage removes the line at the centre:

0	0	0	0	0
0	0	0	0	0
5	5	5	5	5
0	0	0	0	0
0	0	0	0	0

This problem is overcome in the weighted median filter. This filter is similar to the median filter, but the filter mask now defines which pixels are to be used in the median calculation, and the value of the mask determines the number of times a pixel's gray value is repeated in the ordering of the pixels.

Examples of filter masks:

$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	Standard median-filter mask
$\begin{bmatrix} 1 & 1 & 1 \\ 3 & 3 & 3 \\ 1 & 1 & 1 \end{bmatrix}$	Preserves one-pixel wide horizontal lines
$\begin{bmatrix} 1 & 3 & 1 \\ 1 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix}$	Preserves one-pixel wide vertical lines
$\begin{bmatrix} 1 & 1 & 3 \\ 1 & 3 & 1 \\ 3 & 1 & 1 \end{bmatrix}$	Preserves one-pixel wide diagonal lines of slope 1
$\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix}$	Preserves one-pixel wide diagonal lines of slope -1
$\begin{bmatrix} 1 & 3 & 1 \\ 3 & 6 & 3 \\ 1 & 3 & 1 \end{bmatrix}$	Preserves one-pixel wide horizontal and vertical lines

Applying the second mask to the to the centre of the subimage above produces the following ordered set of 15 pixels:

$$\{0, 0, 0, 0, 0, 0, (5, 5, 5), (5, 5, 5), (5, 5, 5)\}$$

The median value is now 5. Thus, the horizontal line is preserved.

(b) Midpoint filter

This filter is defined as the average of the minimum and maximum gray levels of the ordered set of pixels that are included in the filter operation:

$$\text{output} = \frac{z_1 + z_N}{2}$$

The filter mask for this filter simply defines the pixels within the image that are to be included in the filter operation. This is the best filter to use to remove uniform-type noise from an image.

The midpoint filter should not be used with images that contain outlier noise such as salt-and-pepper noise.

(c) Maximum and minimum filters

The minimum filter is defined as the minimum gray level of all the pixels defined by the filter mask, i.e.,

$$\text{output} = z_1$$

If an image contains only salt noise, then the minimum filter removes this noise.

The maximum filter is defined as the maximum gray level of all the pixels defined by the filter mask, i.e.,

$$\text{output} = z_N$$

It is effective in smoothing an image containing only pepper noise.

Both the minimum and maximum filters are biased filters. The maximum filter increases the average brightness of the filtered image, while the minimum filter decreases it.


Image with salt noise


After mininum filtering

(d) Alpha-trimmed mean filter

This gives a mixture of the mean and median filters. It performs reasonably well in the presence of both Gaussian and outlier noise. The computation of the output requires the ordering of pixels that are defined by the filter mask. The output is

$$\text{output} = \frac{1}{N-2p} \sum_{i=p+1}^{N-p} z_i$$

where

$$p = 0, 1, 2, 3, \ldots, \frac{N-1}{2} \qquad N \text{ odd}$$

The value of p determines the type of filtering that is performed.

- For $p = 0$, the α-trimmed mean filter reduces to the mean filter.
- When p is set to its maximum value, the filter reduces to the median filter.

The filter is a mean or averaging filter that removes a selective number of pixels that are close in gray level to the maximum and minimum gray levels contained within the neighbourhood. If an image contains both outlier and Gausssian type noise, selecting p other than zero removes some of the outlier noise pixels in the calculation of the mean. For a given $p \neq 0$, there is is less blurring compared to an equivalent sized mean filter.

Examples:

Consider

20	20	8
21	19	12
19	22	10

The ordered list is $\{8, 10, 12, 19, 19, 20, 20, 21, 22\}$

For different values of p, the output is as follows:

p:	0	1	2	3	4
output:	16.8	17.3	18.0	19.3	19.0

Image with Gaussian and salt-and-pepper noise

Mean filter

Median filter

Alpha-trimmed mean filter ($N = 9$, $p = 2$)

(e) Modified trimmed mean filter

In an image containing both Gaussian and salt-and-pepper noise, the aim is to remove the outlier pixels followed by performing an averaging calculation to reduce the Gaussian noise present. A measure to determine if a pixel's gray level is an outlier is to use the standard deviation.

The modified trimmed mean filter first computes the median and standard deviation (σ) of the pixel gray levels in the defined neighbourhood. The median value is used as an estimate for the mean value. All the pixels that have gray levels outside the range

$$\text{median} \pm K \times \sigma$$

are removed, and the mean value of the remaining pixels are calculated. For $K = 0$, the modified trimmed mean filter becomes the median filter, and for $K = \infty$, it becomes the mean filter.

Example:

46	51	22
48	95	20
55	50	17

ordered list: $\{17, 20, 22, 46, 48, 50, 51, 55, 95\}$

median: 48

σ : 22.6

K	*Range*	*Selected gray levels*	*Output*
0	$[48]$	48	48
1	$[25.4, 70.6]$	$46, 48, 50, 51, 55$	50
2	$[2.8, 93.2]$	$17, 20, 22, 46, 48, 50, 51, 55$	38.6

Adaptive Filtering

An adaptive filter alters its basic behaviour as the image is processed according to the local image characteristics at that point. The typical criteria used to determine filter behaviour are usually measured by the local gray-level statistics.

An example of an adaptive filter is the minimum mean-square error (MMSE) filter, which works best with Gaussian noise:

$$\begin{aligned} g(x,y) &= f(x,y) - \frac{\sigma_\eta^2}{\sigma_l^2}[f(x,y) - m_l(x,y)] && (6) \\ &= \left(1 - \frac{\sigma_\eta^2}{\sigma_l^2}\right) f(x,y) + \frac{\sigma_\eta^2}{\sigma_l^2} m_l(x,y) && (7) \end{aligned}$$

σ_η^2 = noise variance
σ_l^2 = local variance (in the window under consideration)
m_l = local mean (in the window under consideration)

- With no noise in the image, $\sigma_\eta^2 = 0$, and $g(x,y) = f(x,y)$, i.e., no filtering takes place.
- In background regions of the image, where the gray values are fairly constant in the uncorrupted image, $\sigma_\eta^2 = \sigma_l^2$, and the equation reduces to the mean filter.
- In areas of the image where the local variance is much greater than the noise variance, the filter returns a value close to the unfiltered image data. This is desired since the high local variance implies high detail (edges), and an adaptive filter tries to preserve the original image detail.

In general, the MMSE filter returns a value that consists of the unfiltered image data $f(x,y)$ with some of the original value subtracted out and some of the local mean added. The MMSE filter adapts itself to the local image statistics, preserving image details while removing noise. The window size and noise variance to be used has to be specified by the user.

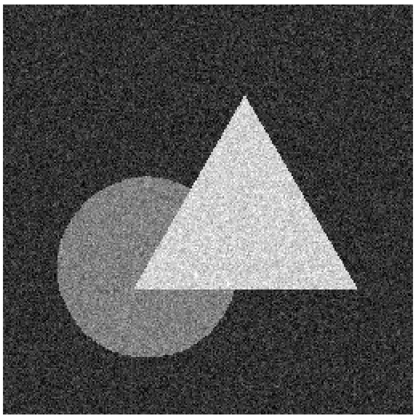
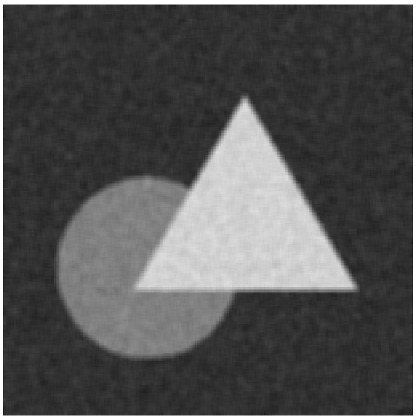
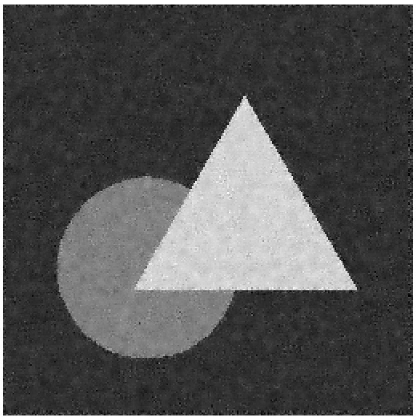


Image with Gaussian noise



Mean filtering



MMSE filter

Image Averaging

Consider a noisy image $f_\eta(x,y)$ formed by the addition of noise $\eta(x,y)$ to an uncorrupted image $f(x,y)$, where the noise is uncorrelated and has zero mean:

$$f_\eta(x,y) = f(x,y) + \eta(x,y). \tag{8}$$

The image formed by averaging K different noisy images is

$$\begin{aligned}\bar{f}_\eta(x,y) &= \frac{1}{K}\sum_{t=1}^{K} f_{\eta,t}(x,y) \\ &= \frac{1}{K}\sum_t f(x,y) + \frac{1}{K}\sum_t \eta_t(x,y) \\ &= f(x,y) + \frac{1}{K}\sum_t \eta_t(x,y)\end{aligned}$$

Thus the expected value of $\bar{f}_\eta(x,y)$ is

$$E\{\bar{f}_\eta(x,y)\} = f(x,y). \tag{9}$$

The variance of $\bar{f}_\eta(x,y)$ is

$$\sigma^2_{\bar{f}_\eta}(x,y) = \frac{1}{K}\sigma^2_\eta(x,y) \tag{10}$$

where $\sigma^2_\eta(x,y)$ is the variance of $\eta(x,y)$.

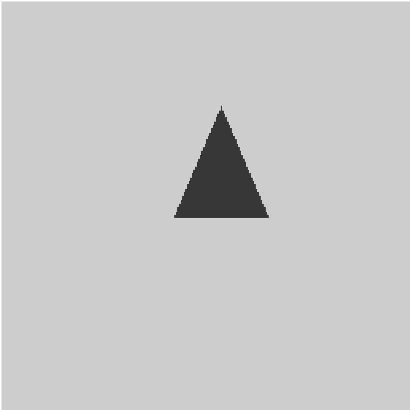
The standard deviation at any point in the averaged image is

$$\sigma_{\bar{f}_\eta(x,y)} = \frac{1}{\sqrt{K}}\sigma_\eta(x,y). \tag{11}$$

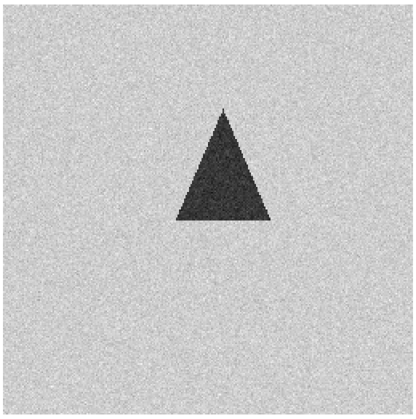
We can conclude that as K increases

1. $\bar{f}_\eta(x,y)$ approaches the uncorrupted image $f(x,y)$, and
2. the variability of the pixel values decreases.

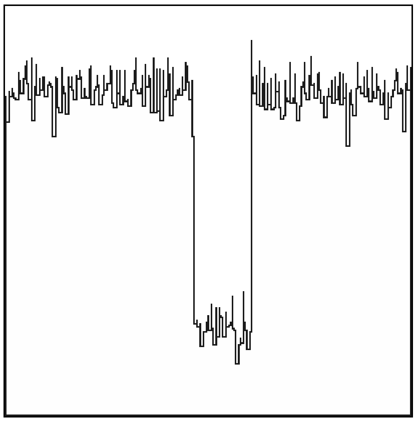
This operation is often considered an automatic image acquisition pre-processing operation. With appropriate hardware, an entire image addition can be done in one frame interval.



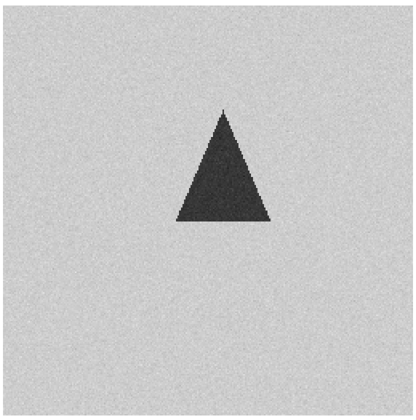
Noise-free image



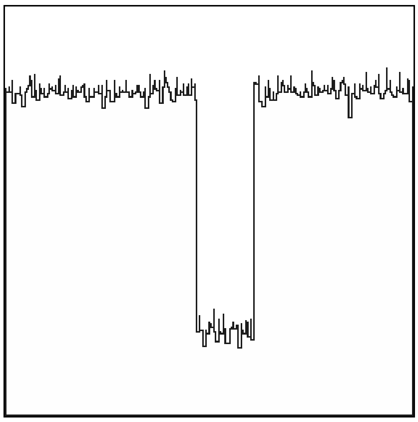
Noisy image



Profile



After averaging 4 frames



Profile