Neighborhood Processing

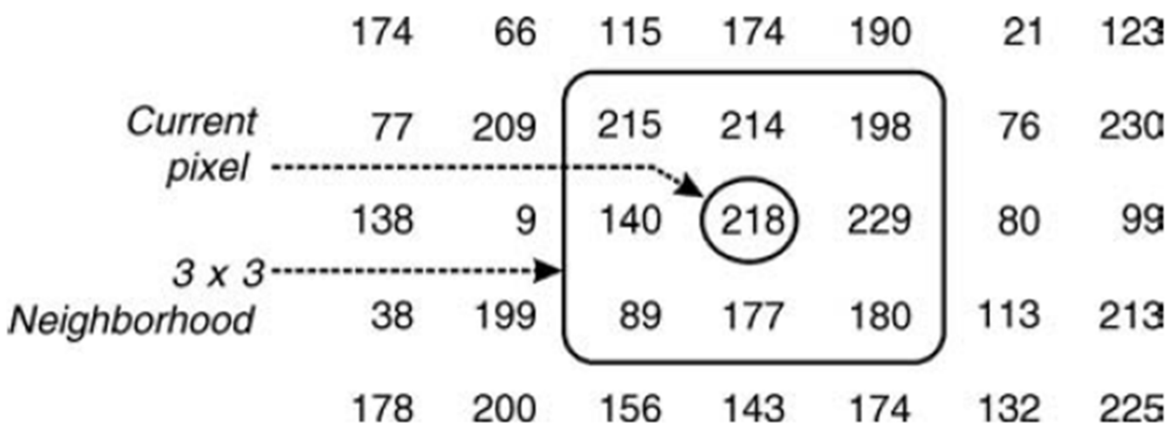
1. *Spatial domain* refers to the image plane itself, and methods in this category are based on direct manipulation of pixels in an image.
2. *For spatial filtering,* sometimes is referred to as *neighborhood processing,* or *spatial convolution.*

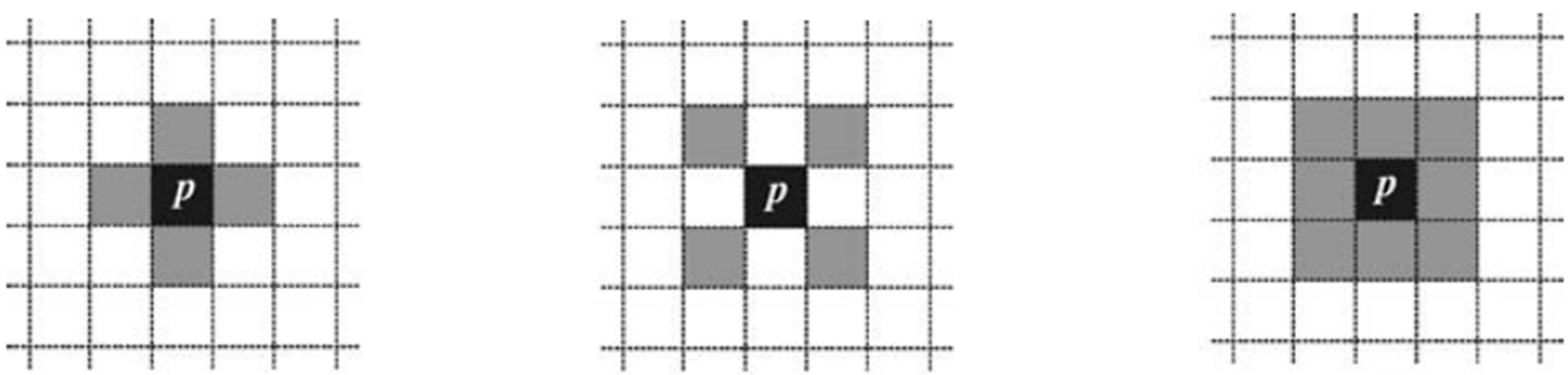
neighborhood processing

The process of moving the center point creates new neighborhoods, one for each pixel in the input image.

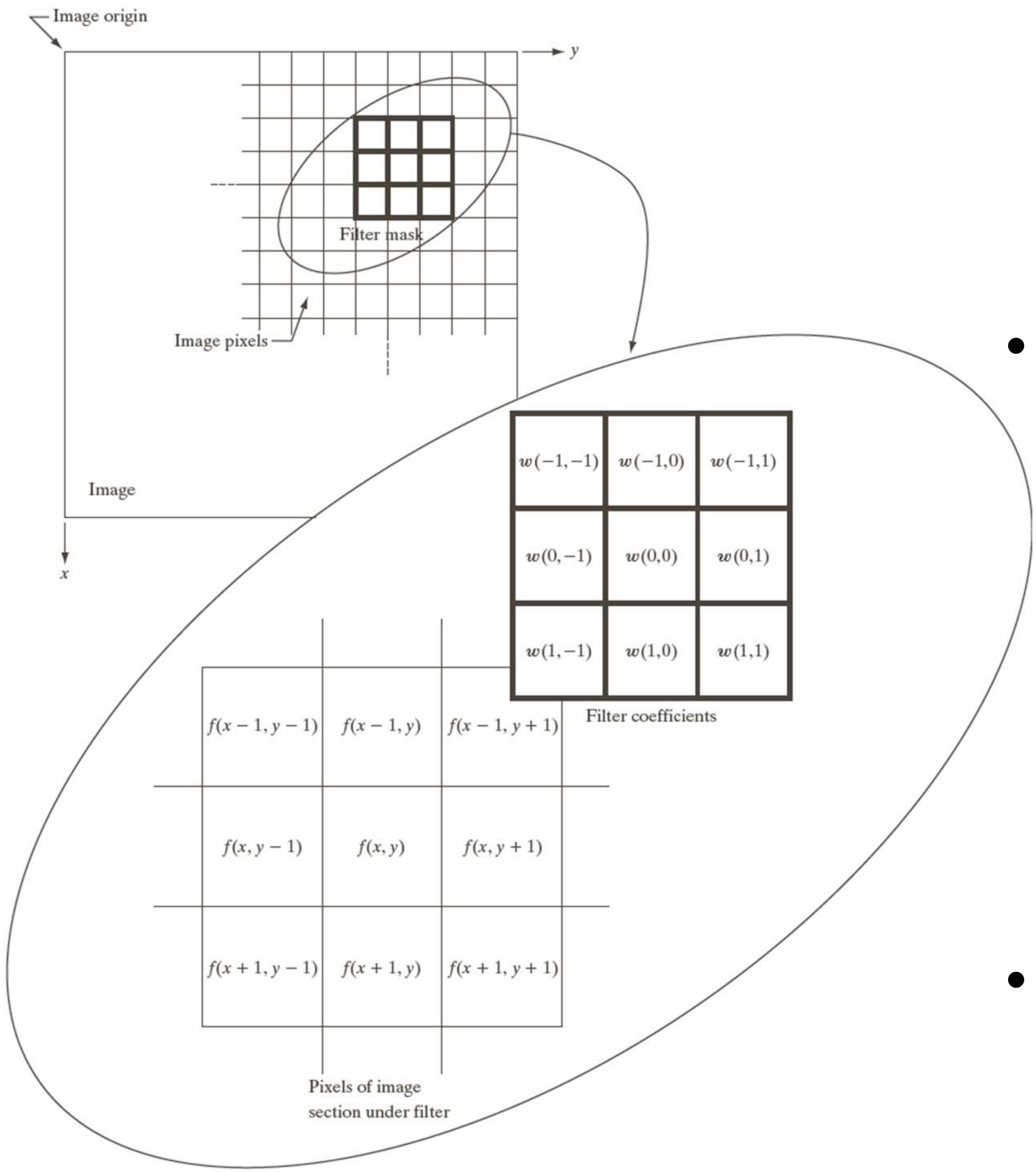
- The two principal terms used to identify this operation are ***neighborhood processing*** and ***spatial filtering***.
- If the computation performed on the pixels of the neighborhoods are linear,
  - the operation is called ***linear spatial filtering*** (the term ***spatial convolution*** also used);
  - otherwise, it is called ***nonlinear spatial filtering***.

BASIC TERMINOLOGY





(a)4-neighborhood; (b) diagonal neighborhood; (c) 8-neighborhood.



- The process consists simply of moving the center of the filter mask *w* from point to point in an image, *f*.
- At each point *(x, y),* the response of the filter at that point is the sum of products of the filter coefficients and the corresponding neighborhood pixels in the area spanned by the filter mask.
- For a mask of size $m \times n$, we assume typically that $m = 2a + 1$ and $n = 2b + 1$, where a and b are nonnegative integers.

Spatial Filtering

Spatial filtering is a neighborhood processing consisting of:

1. defining a center point, (x, y);
2. performing an operation that involves only the pixels in a predefined neighborhood of the center point;
3. letting the result of that operation be the "response" of the process at that point;
4. repeating the process for every point in the image.

Filtering

- ***Linear Filters***: Output pixel is computed as a **sum of products** of the pixel values and mask coefficients in the pixel's neighborhood in the original image. Ex: mean filter.
- ***Nonlinear Filters***: Output pixel is selected from an ordered (ranked) sequence of pixel values in the pixel's neighborhood in the original image. Ex: median filter.

The coefficients are arranged as a matrix, called a *filter, mask, filter mask, kernel, template,* or *window,* with the first three terms being the most prevalent. For reasons that will become obvious shortly, the terms *convolution filter, mask,* or *kernel,* also are used.

CONVOLUTION

- **In one-dimensional**

$$A * B = \sum_{j=-\infty}^{\infty} A(j) \cdot B(x-j)$$

- Ex: Let A = {0, 1, 2, 3, 2, 1, 0}, and B = {1, 3,−1}.

1. A(x=1)

A		0	1	2	3	2	1	0
B	−1	3	1					
$A * B$		1						

(0 × (−1)) + (0 × 3) + (1 × 1) = 1

CONVOLUTION (cont.)

2. A(x=2)

A	0	1	2	3	2	1	0
B	-1	3	1				
$A * B$	1	5					

$(0 \times (-1)) + (1 \times 3) + (2 \times 1) = 5$

⋮

7. A(x=7)

A	0	1	2	3	2	1	0	
B						-1	3	1
$A * B$	1	5	8	8	4	1	-1	

$(1 \times (-1)) + (0 \times 3) + (0 \times 1) = -1$

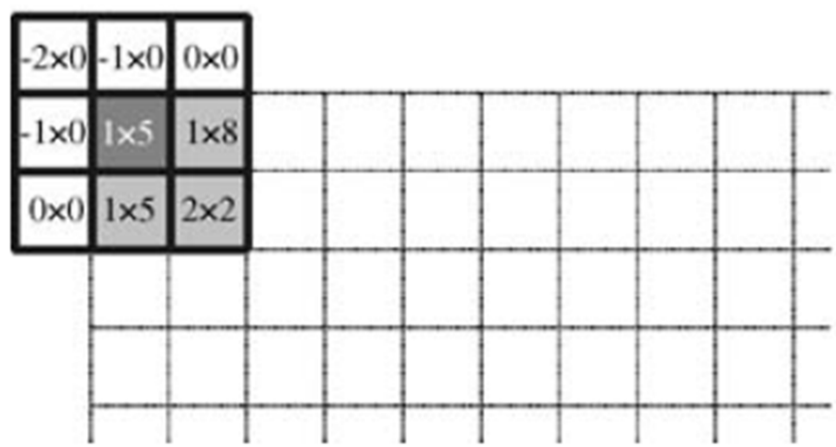
2D Convolution

$$g(x, y) = \sum_{k=-m/2}^{m/2} \sum_{l=-n/2}^{n/2} w(k, l) f(x-k, y-l)$$

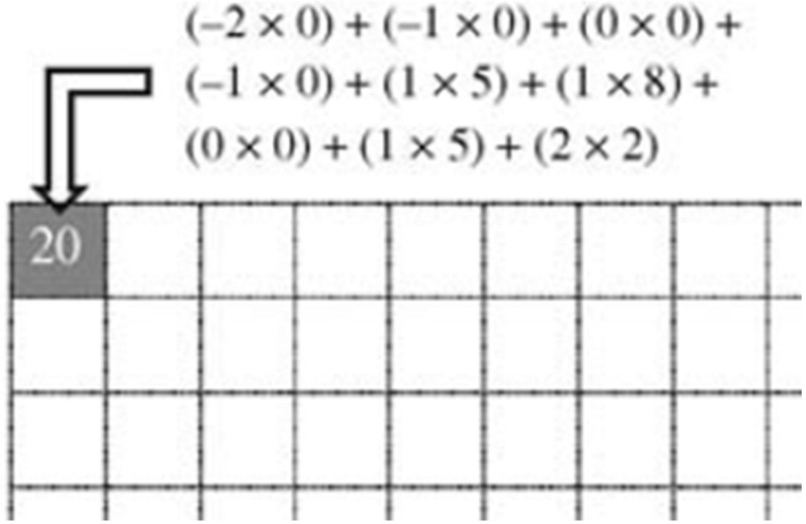
EXAMPLE

$$\boldsymbol{f} = \begin{bmatrix} 5 & 8 & 3 & 4 & 6 & 2 & 3 & 7 \\ 3 & 2 & 1 & 1 & 9 & 5 & 1 & 0 \\ 0 & 9 & 5 & 3 & 0 & 4 & 8 & 3 \\ 4 & 2 & 7 & 2 & 1 & 9 & 0 & 6 \\ 9 & 7 & 9 & 8 & 0 & 4 & 2 & 4 \\ 5 & 2 & 1 & 8 & 4 & 1 & 0 & 9 \\ 1 & 8 & 5 & 4 & 9 & 2 & 3 & 8 \\ 3 & 7 & 1 & 2 & 3 & 4 & 4 & 6 \end{bmatrix}$$

$$\boldsymbol{w} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$



2D Convolution



$$
w * f = \begin{bmatrix}
20 & 10 & 2 & 26 & 23 & 6 & 9 & 4 \\
18 & 1 & -8 & 2 & 7 & 3 & 3 & -11 \\
14 & 22 & 5 & -1 & 9 & -2 & 8 & -1 \\
29 & 21 & 9 & -9 & 10 & 12 & -9 & -9 \\
21 & 1 & 16 & -1 & -3 & -4 & 2 & 5 \\
15 & -9 & -3 & 7 & -6 & 1 & 17 & 9 \\
21 & 9 & 1 & 6 & -2 & -1 & 23 & 2 \\
9 & -5 & -25 & -10 & -12 & -15 & -1 & -12
\end{bmatrix}
$$

Image filtering

Filtering operations use masks as the *spatial filters*, which are named based on their behavior in the *spatial frequency:*

- *Low-pass* filters (LPFs) preserve low-frequency components (i.e., coarser details and homogeneous areas in the image).
- High-pass filters (HPFs) enhance high-frequency components (i.e., fine details in the image)

Image filtering

Original image

LPF image

HPF image

Spatial Averaging and Spatial Low-Pass Filtering

$$g(x,y) = \sum_{k=1}^{n}\sum_{l=1}^{n} w(k,l) f(x-k, y-l)$$

where *g* and *f* are output and input images.

w is weighted matrix with the size $n \times n$.

For average filter, all elements in w are 1, so that

$$g(x,y) = \frac{1}{N_w}\sum_{k=1}^{n}\sum_{l=1}^{n} f(x-k, y-l)$$

where $w = 1/N_w$ and N_w is numbers of element in *w*

Spatial Averaging and Spatial Low-Pass Filtering

$$g(x,y) = \frac{1}{2}\Big[f(x,y) + \frac{1}{4}\{f(x-1,y) + f(x+1,y) + f(x,y-1) + f(x,y+1)\}\Big]$$

$$w_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad w_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad w_3 = \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix}$$

$$g(x,y) = \frac{1}{N_w}\sum_{k=1}^{n}\sum_{l=1}^{n} w_1(k,l) f(x-k, y-l)$$
$$= \frac{1}{4}\sum_{k=1}^{n}\sum_{l=1}^{n} f(x-k, y-l)$$
$$= \frac{1}{4}[f(x,y) + f(x,y-1) + f(x-1,y) + f(x-1,y-1)]$$

```
Nw = 4;
for x=1:4
    for y=1:4
        w = f(x:x+1,y:y+1);
        g(x,y) = sum(w(:))/Nw;
    end
end
```

156	159	158	155	158
160	154	157	158	157
156	159	158	155	158
160	154	157	158	157
156	153	155	159	159

157.25	157.00	157.00	157.00
157.25	157.00	157.00	157.00
157.25	157.00	157.00	157.00
155.75	154.75	157.25	158.25

$$= \frac{1}{4}\sum_{k=1}^{n}\sum_{l=1}^{n} f(x-k, y-l)$$

$$= \frac{1}{4}[f(x,y) + f(x,y-1) + f(x-1,y) + f(x-1,y-1)]$$

=(156+159+160+154)/4

=157.25

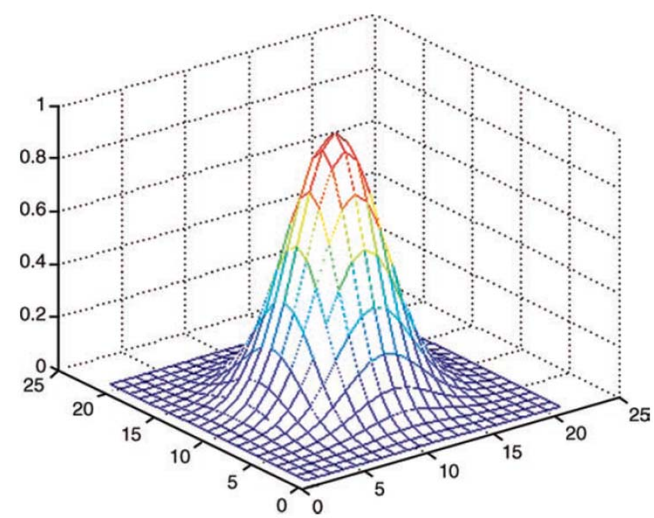
157.25	159	158	155	158
160	154	157	158	157
156	159	158	155	158
160	154	157	158	157
156	153	155	159	159

Variations of averaging filter

$$h(x, y) = \begin{bmatrix} 0.075 & 0.125 & 0.075 \\ 0.125 & 0.2 & 0.125 \\ 0.075 & 0.125 & 0.075 \end{bmatrix}$$

Modified Mask Coefficients

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



Variations of averaging filter

a, b,
c, d

a) Original image
b) 3x3 average
c) Modified Mask Coefficients
d) Gaussian blur filter, $\sigma = 0.5$

$$h = \begin{bmatrix} 0.0113 & 0.0838 & 0.0113 \\ 0.0838 & 0.6193 & 0.0838 \\ 0.0113 & 0.0838 & 0.0113 \end{bmatrix}$$

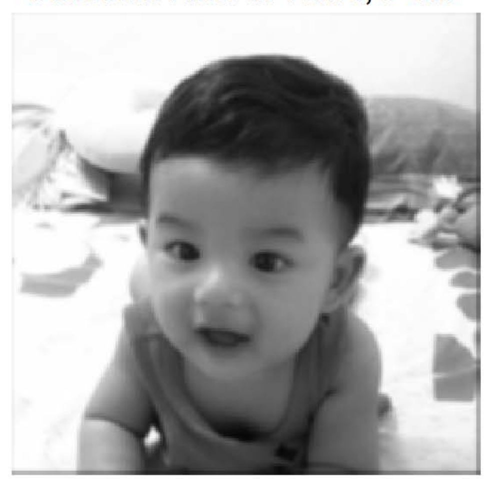
Ex. Gaussian blur filters

Original Image

Gaussian Filter in 5x5,s=0.9

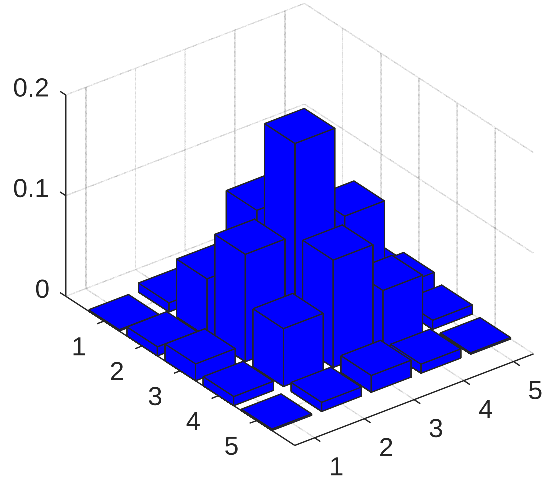
average Filter in 13x13

Gaussian Filter in 13x13, s=0.9

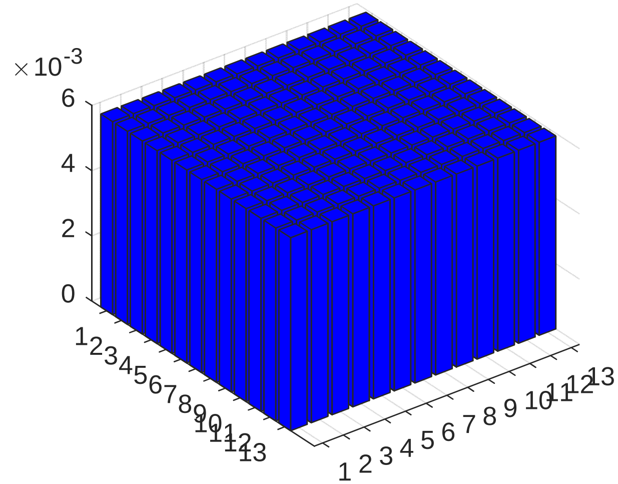


Ex. Gaussian blur filters

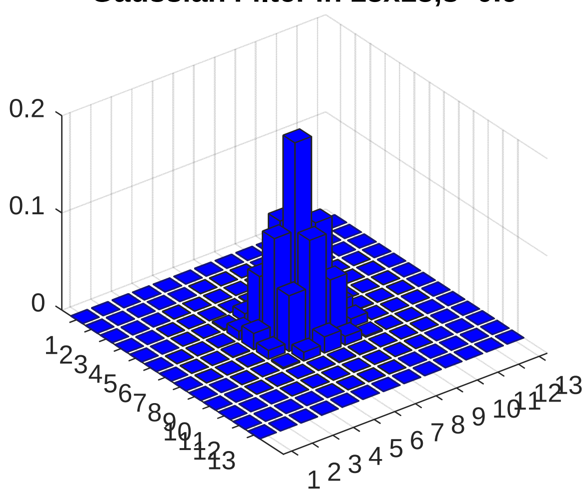
Gaussian Filter in 5x5,s=0.9



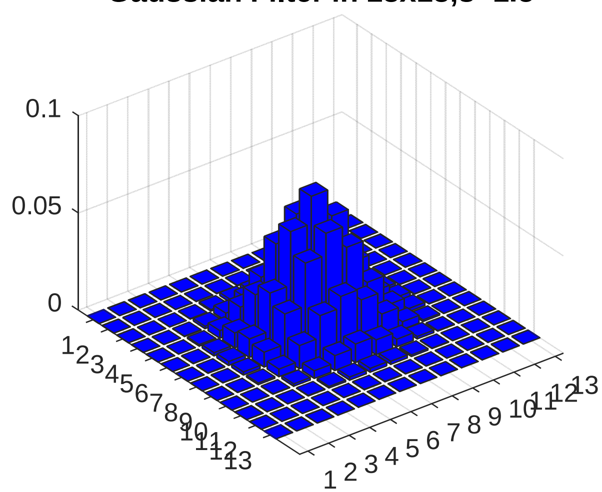
average Filter in 13x13



Gaussian Filter in 13x13,s=0.9



Gaussian Filter in 13x13,s=1.5



Properties of the Gaussian blur filter

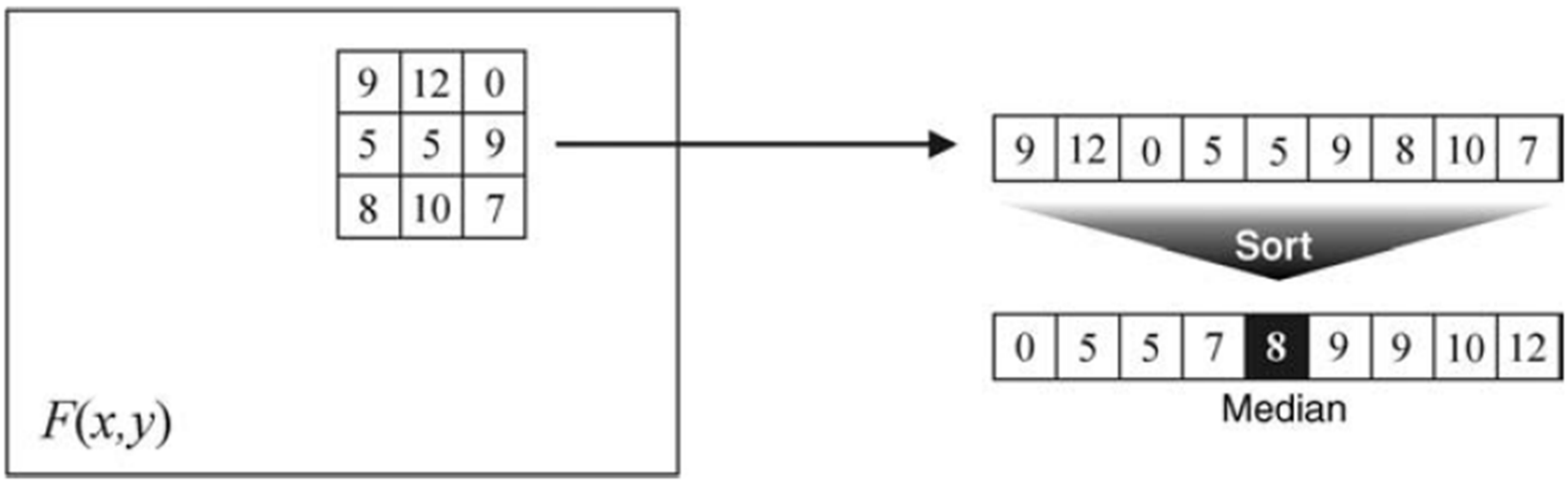
- The window is symmetric with respect to rotation.
- The window's coefficients fall off to (almost) zero at the kernel's edges.
- The output image obtained after applying the Gaussian blur filter is more pleasing to the eye than the one obtained using other low-pass filters.
- The kernel is separable, which can lead to fast computational implementations.

Median filter

The median filter is a nonlinear filter used in image processing. It works by:

- sorting the pixel values within a neighborhood,
- finding the median value, and
- replacing the original pixel value with the median.

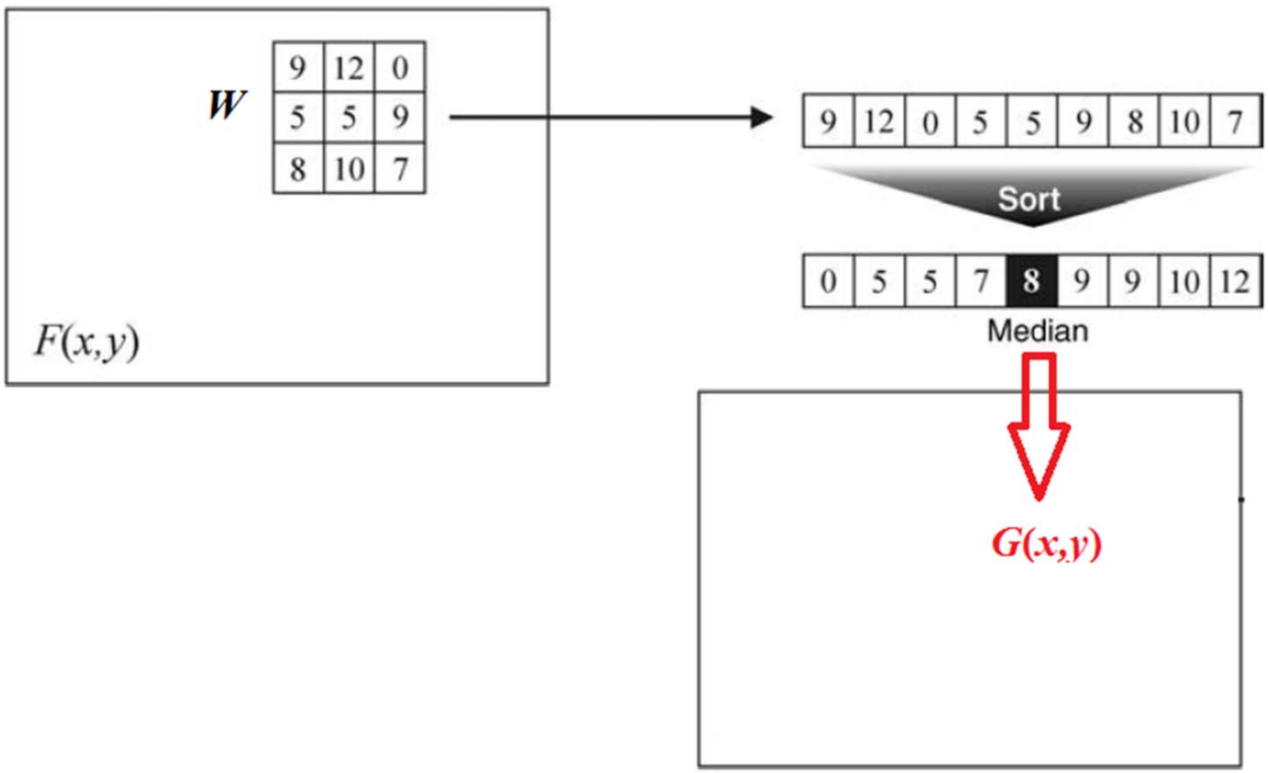
Median filtering process



Median filtering

Median filter operates in the window *W*, given by:

$$g(x,y) = median\{f(x-k, y-l), (k,l) \in W\}$$



Window *W* has the size, for ex., 3×3, 5× 5, 7 × 7 etc.

Median filtering

Ex. Let $f(x)$ = [2 3 8 4 2], W=[-1,0,1], the filtering operation can be performed by:

[-1,0,1]

[2 3 8 4 2]

$g(1) = 2;$

[-1,0,1]

[2 3 8 4 2]

$g(2) = \text{median}[2\ 3\ 8] = 3$

[-1,0,1]

[2 3 8 4 2]

$g(3) = \text{median}[3\ 8\ 4] = 4$

[-1,0,1]

[2 3 8 4 2]

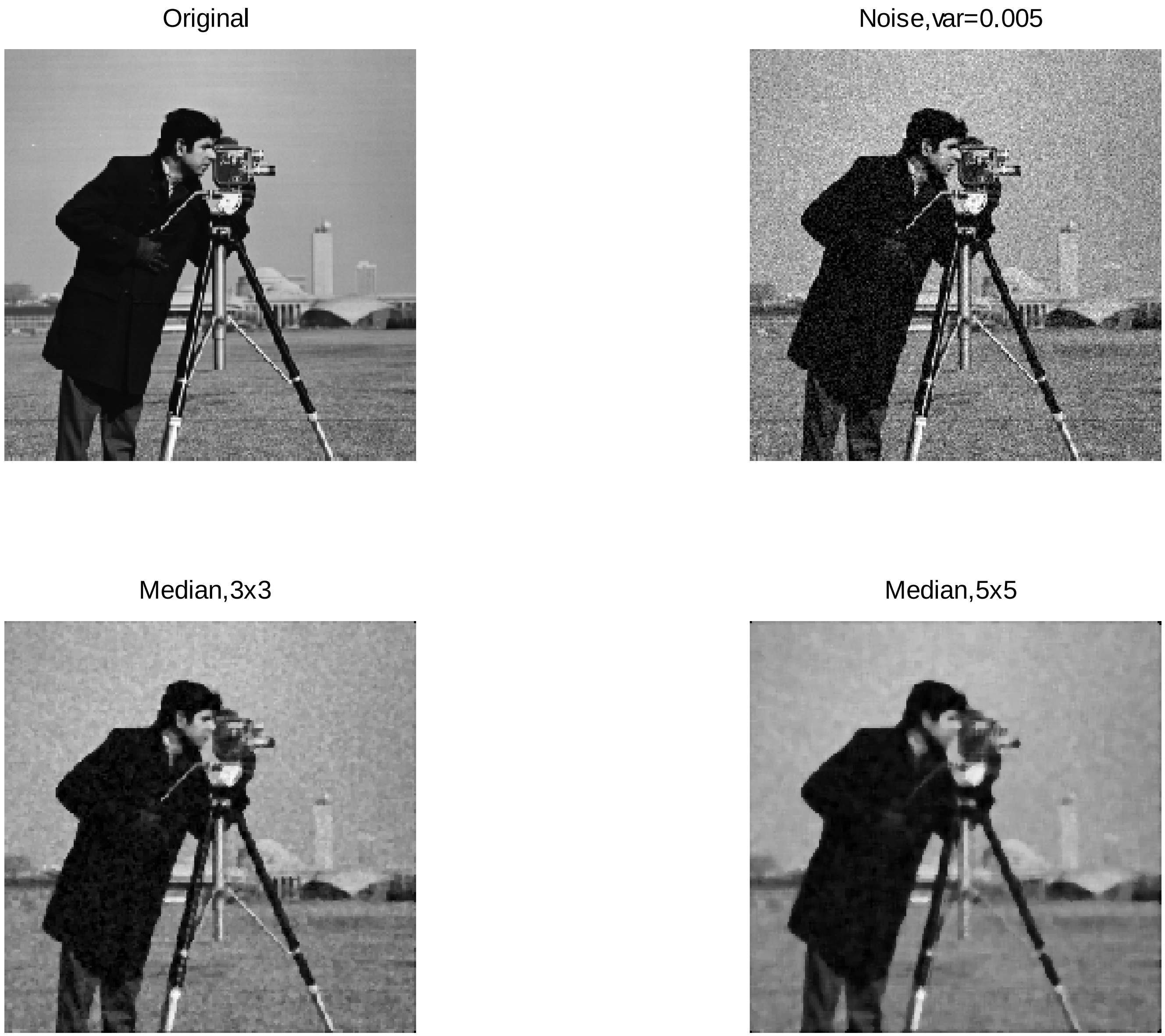
$g(4) = \text{median}[8\ 4\ 2] = 4$

[-1,0,1]

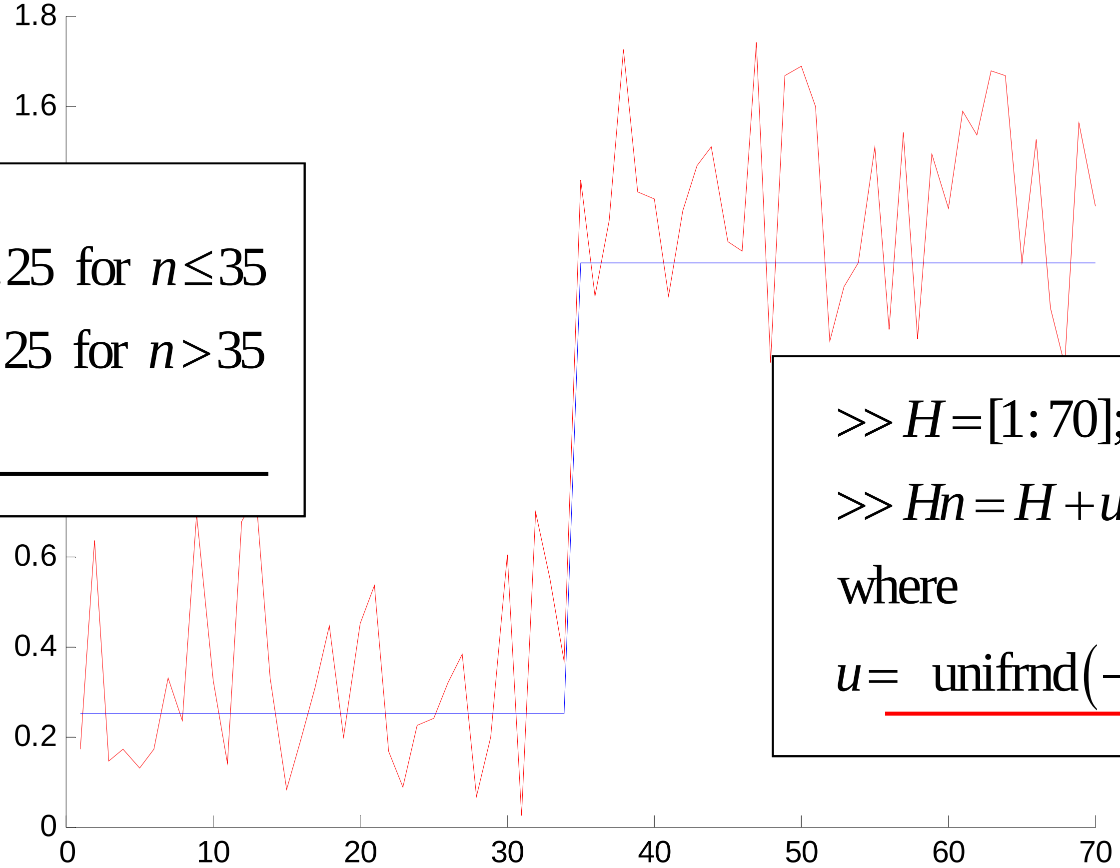
[2 3 8 4 2]

$g(5) = 2;$

The output of median filter is $g(x)$ = [2 3 4 4 2].


Original
Noise,var=0.005
Median,3x3
Median,5x5

A Noisy Step Edge



$$H(n)=\begin{cases}0.25 & \text{for } n\le 35\\ 1.25 & \text{for } n>35\end{cases}$$

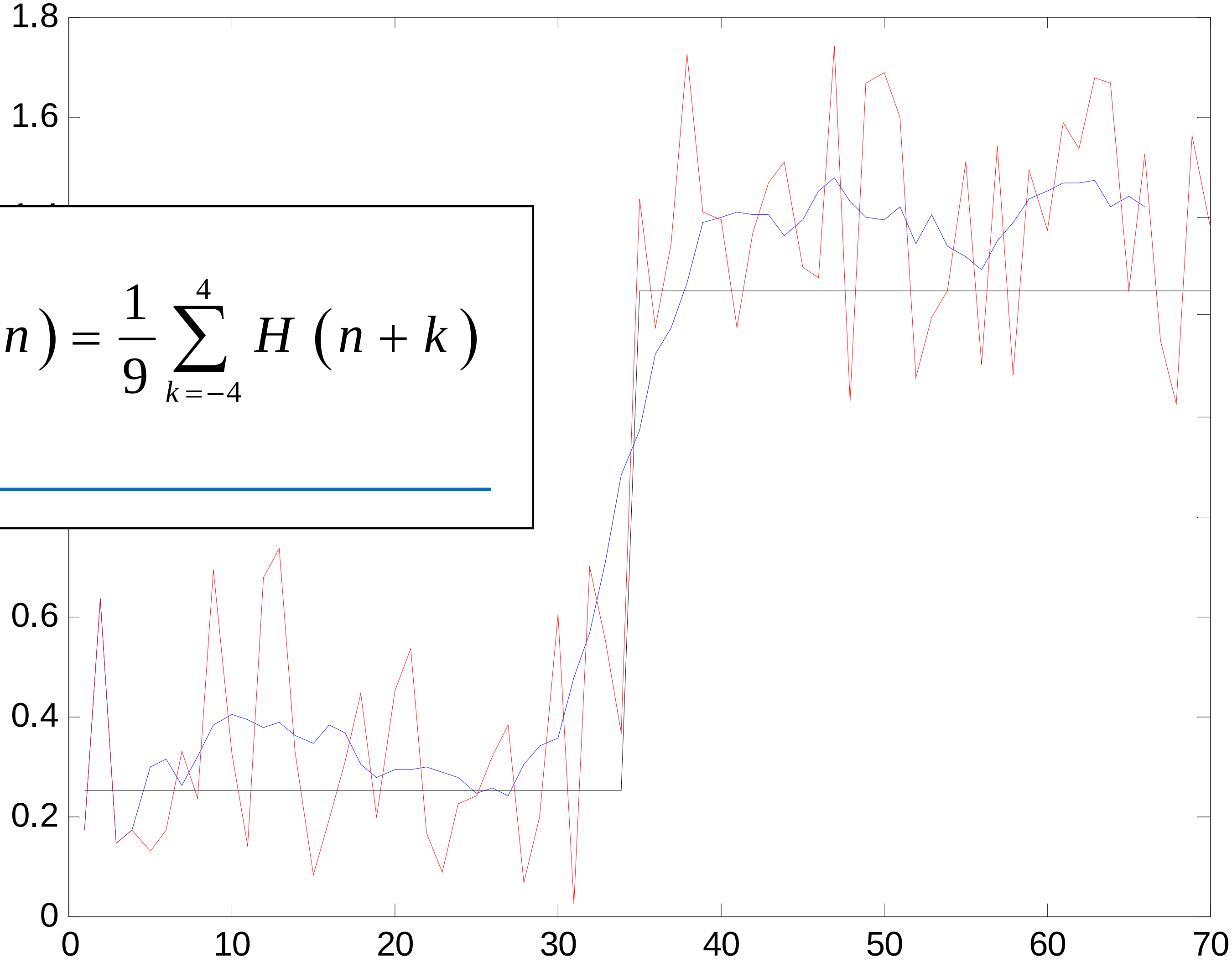
$>> H=[1:70];$

$>> Hn=H+u;$

where

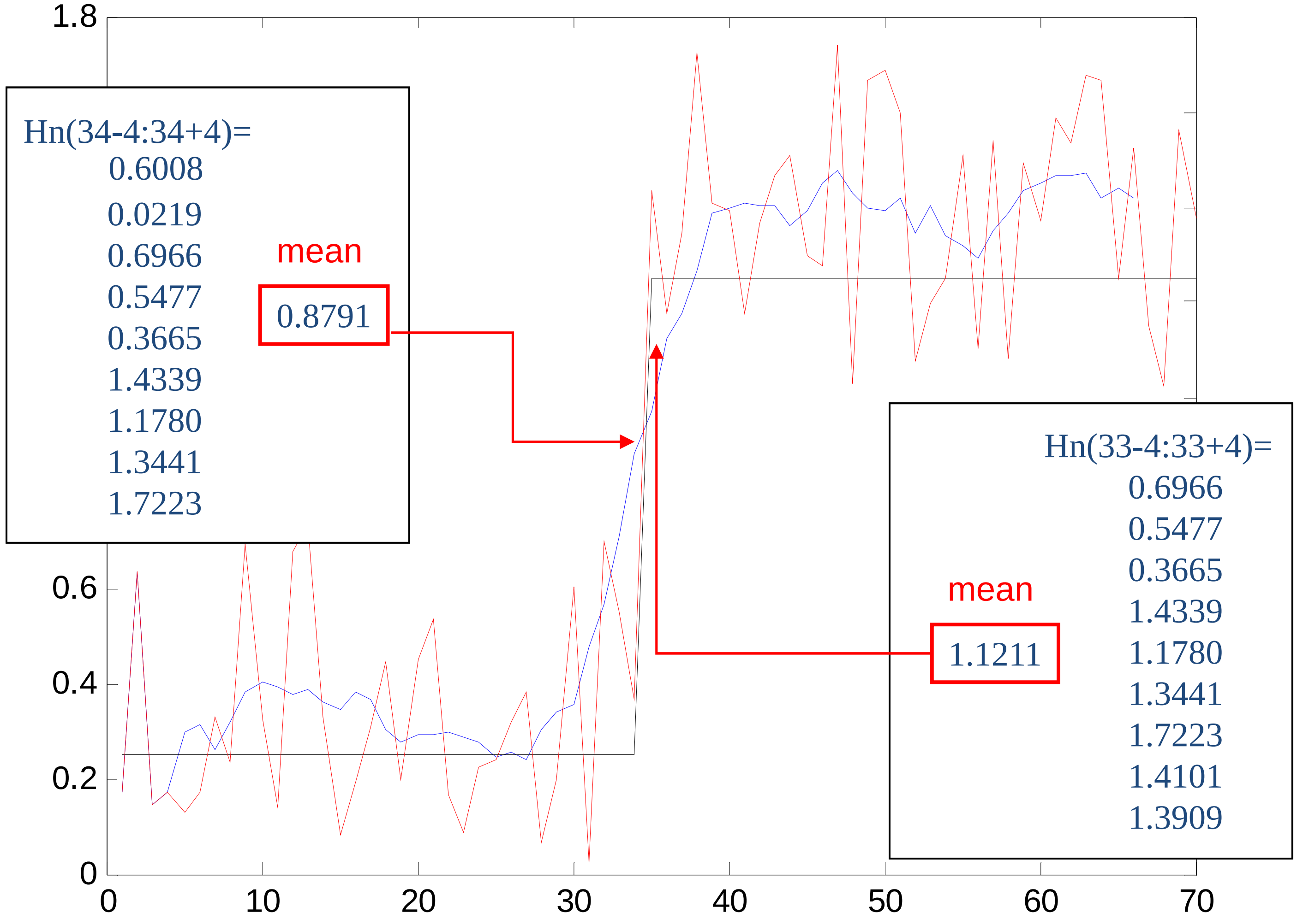
$u=\ \text{unifrnd}\left(-0.25\,,\,0.25,1,70\right)$

Blurred Noisy 1D Step Edge

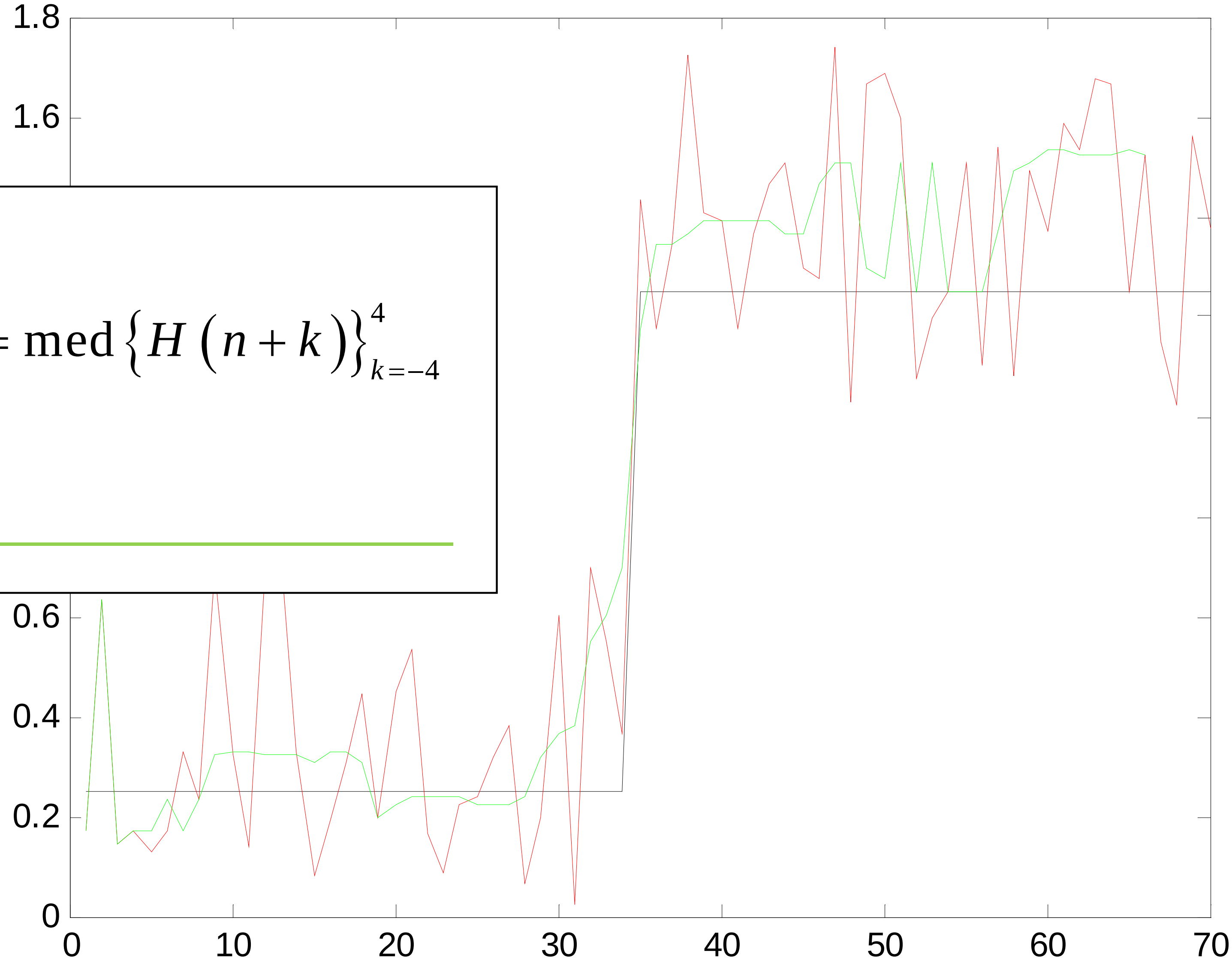


$$h(n) = \frac{1}{9}\sum_{k=-4}^{4} H(n+k)$$

Blurred Noisy 1D Step Edge

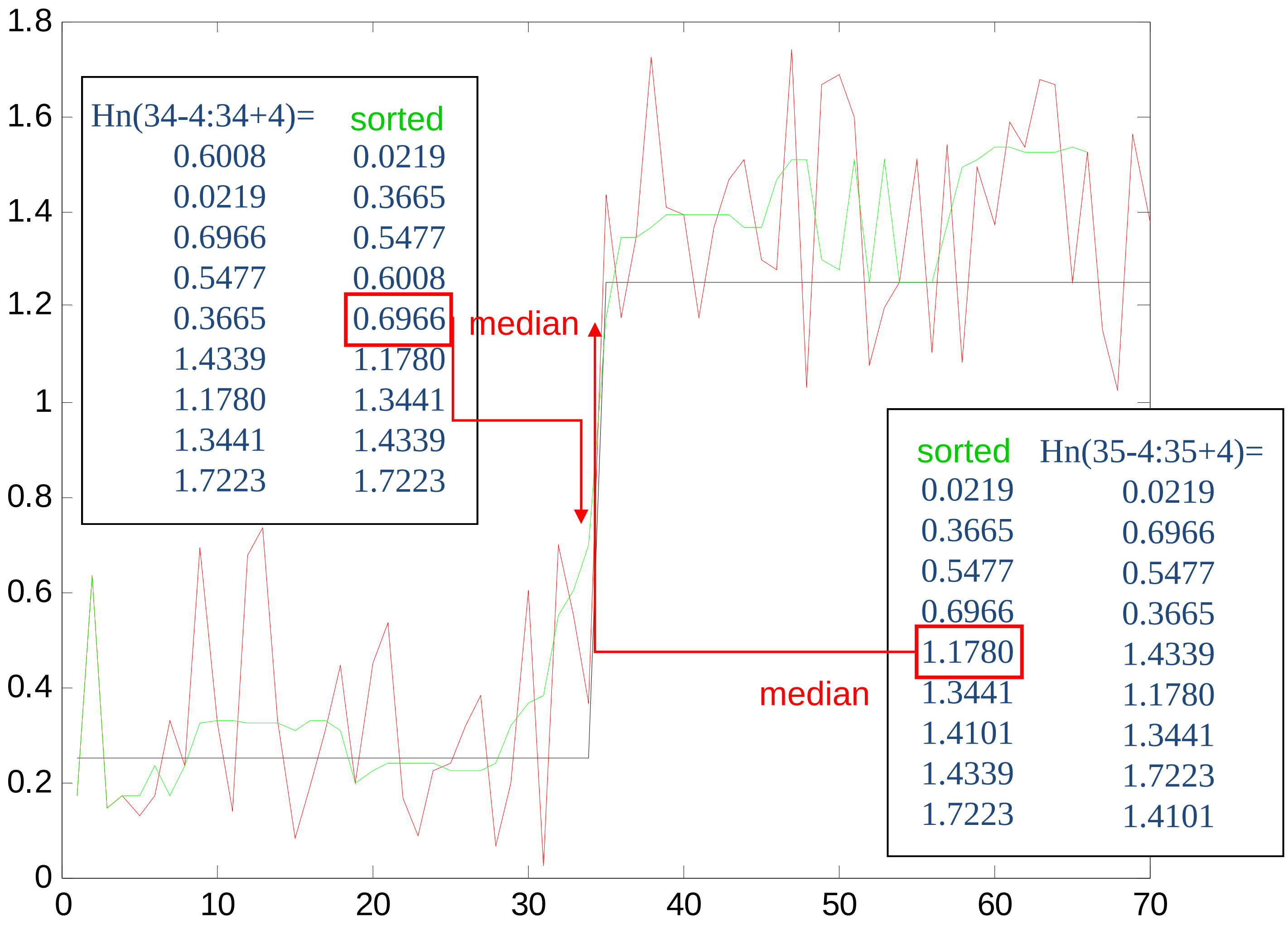


Median Filtered Noisy 1D Step Edge

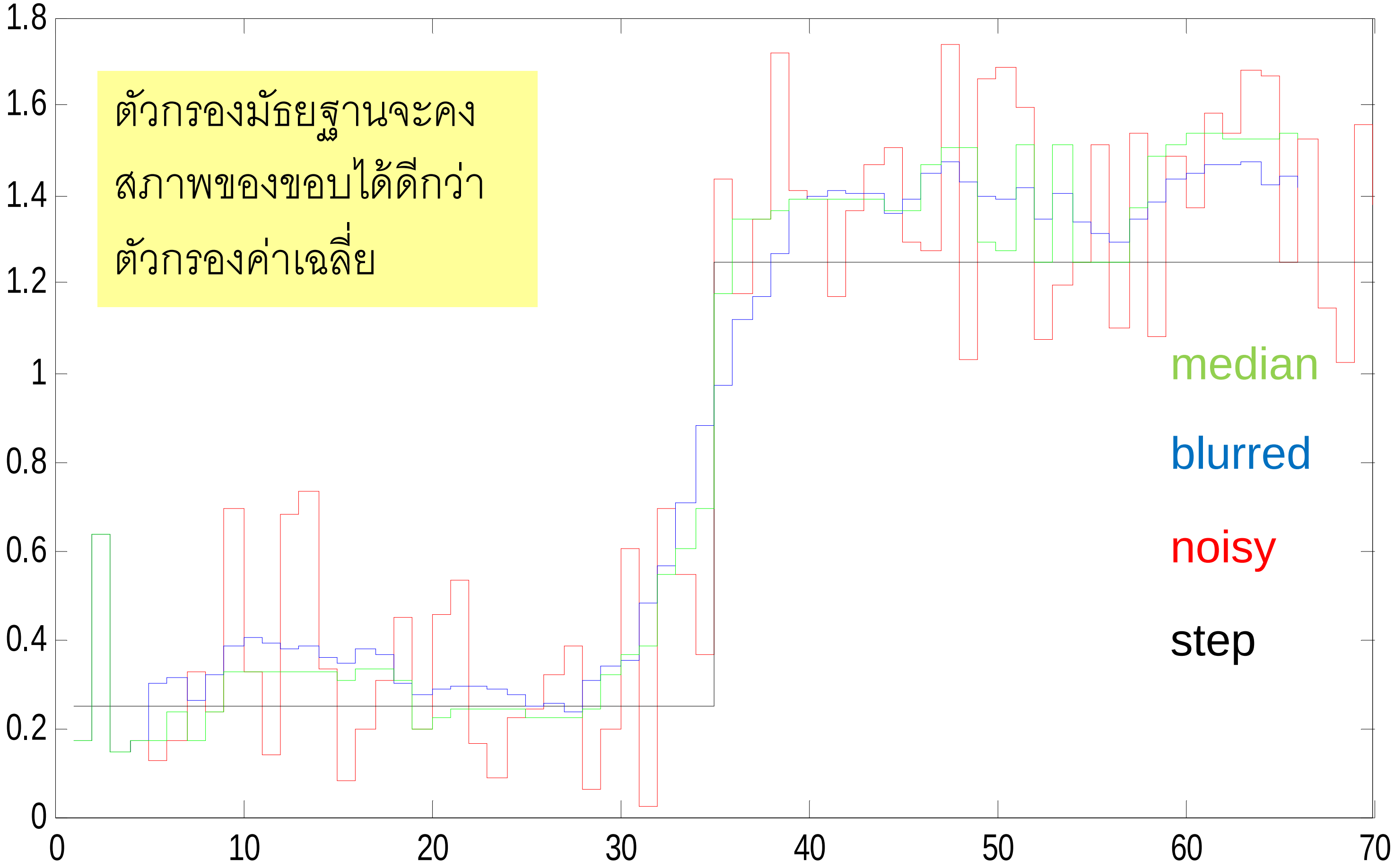


$$h(n) = \operatorname{med}\left\{H(n+k)\right\}_{k=-4}^{4}$$

Median Filtered Noisy 1D Step Edge



Median vs. Blurred



Median vs. Average

Properties of the median filter

- It is a nonlinear filter, thus, any x(m) and y(m)

$$\mathrm{median}(x(m)+y(m)) \neq \mathrm{median}(x(m))+\mathrm{median}(y(m))$$

- Noise remove in spatial resolution, the median filter has performance better than the average filter.
- The performance of noise remove will be decreased, when the noisy quantity more than haft of the window.

Smoothing image containing Gaussian noise

Original

noise,var=0.1

average with 3x3

average with 5x5

average with 7x7

average with 9x9