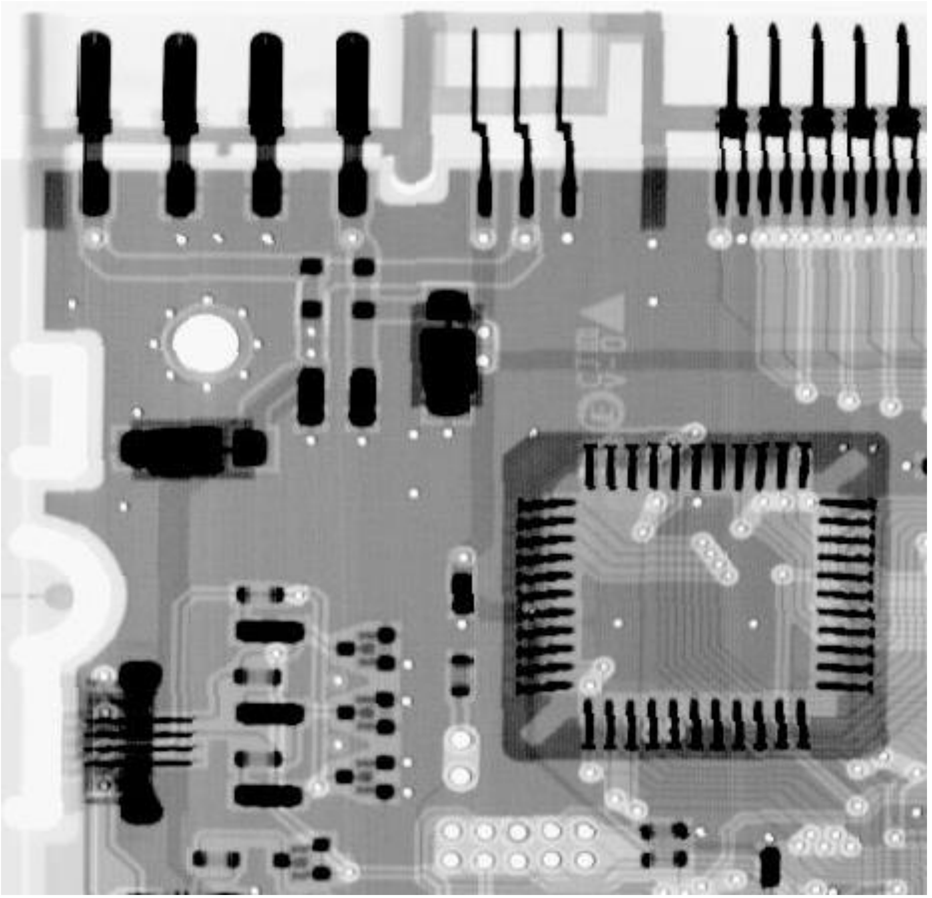
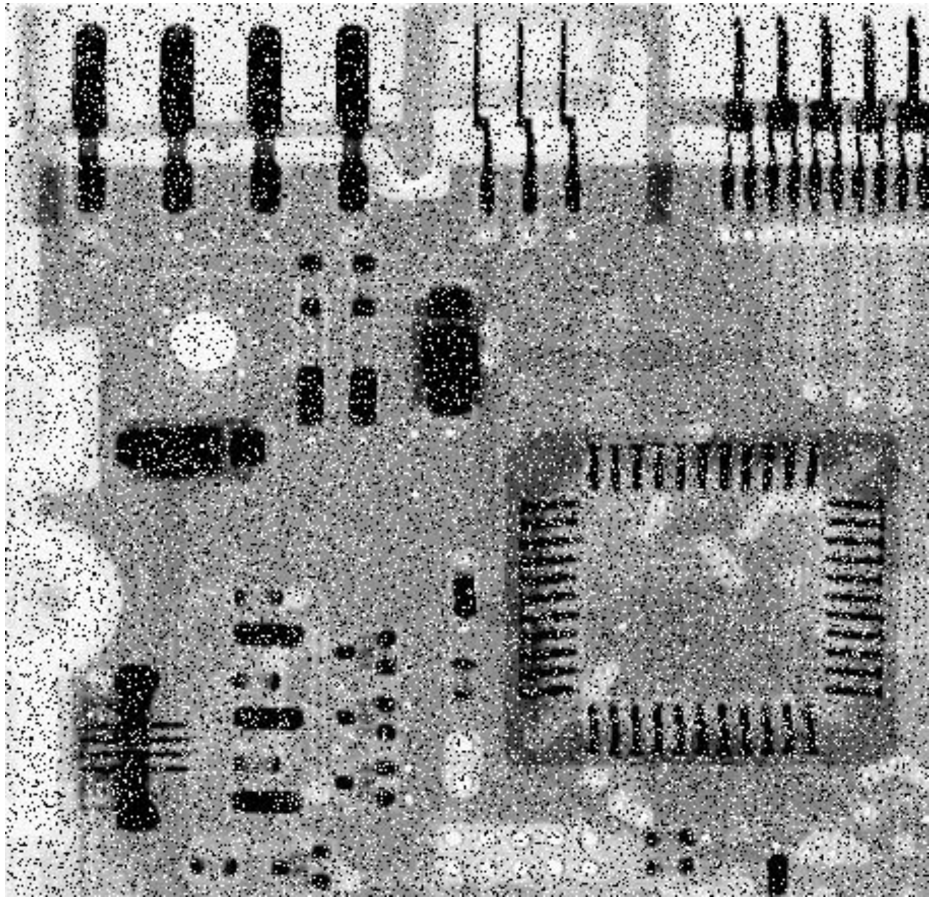
Nonlinear Image Smoothing –Outline

- Introduction
- Problems of Linear Filter
- Order-Statistic Filters
- Median Filter vs. Mean Filter
- Median Filter Properties
- Other order-statistic Filters

Nonlinear Image Smoothing –Introduction

Observe an image and its noise contaminated version

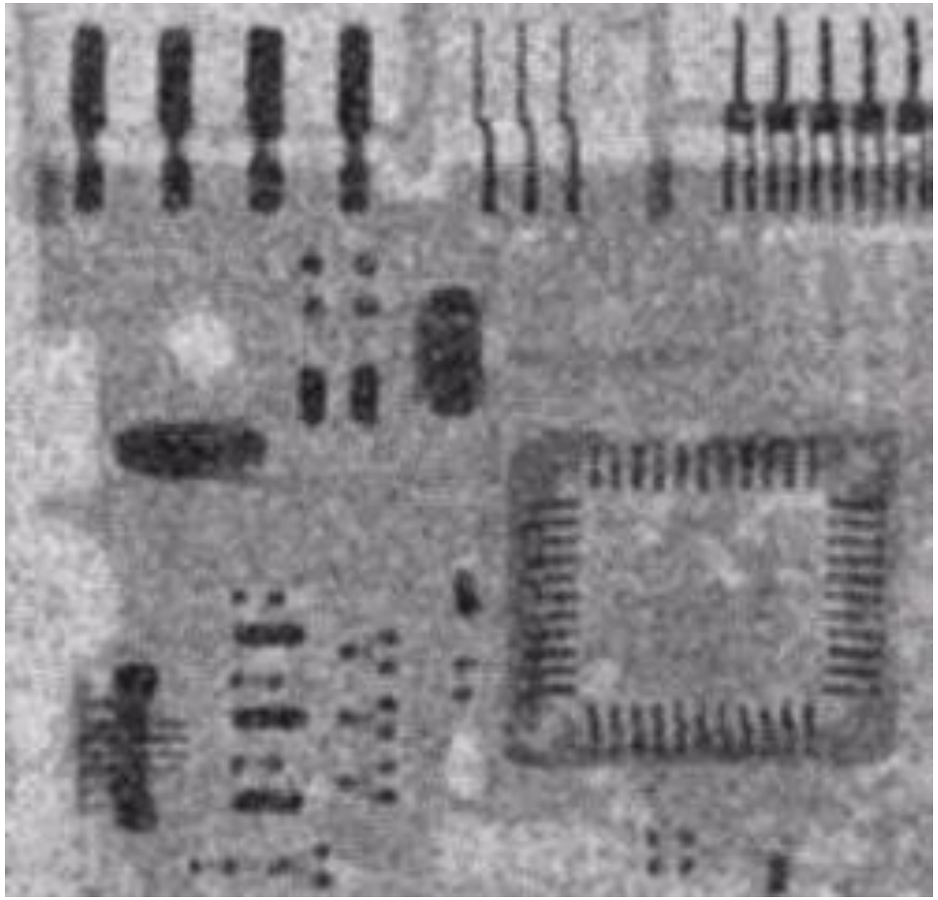


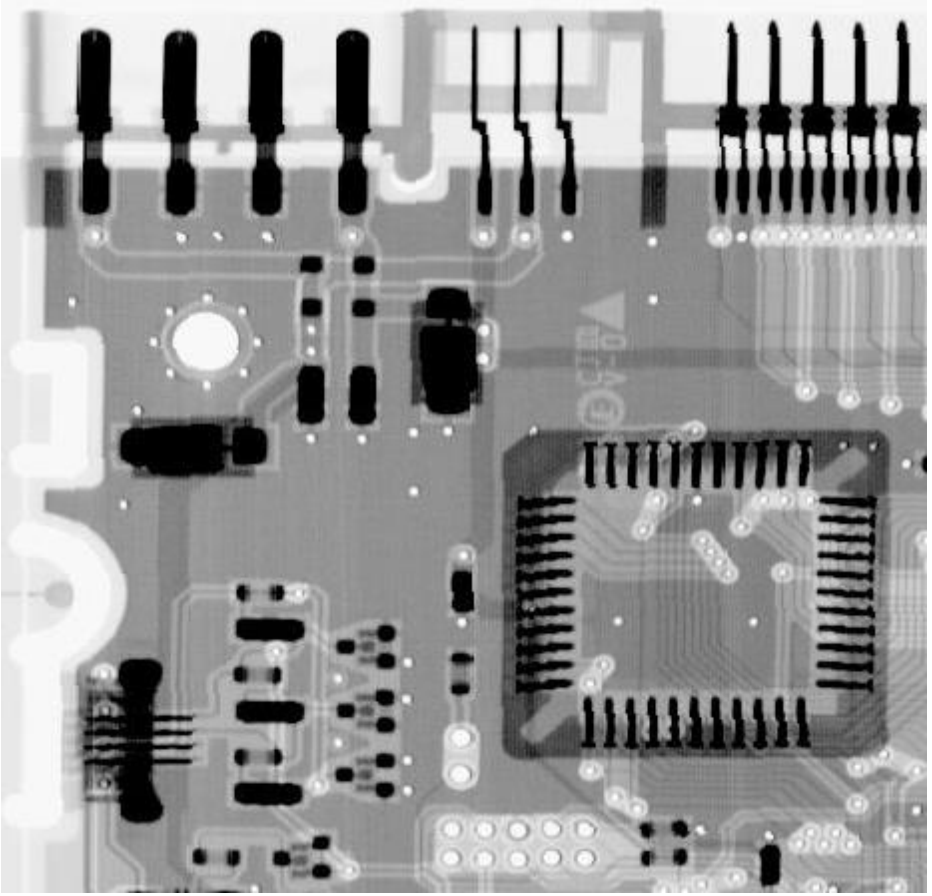


What are the noise characteristics? How to remove such noise?

Nonlinear Image Smoothing –Introduction

Observe the Image smoothed by a linear low pass filter

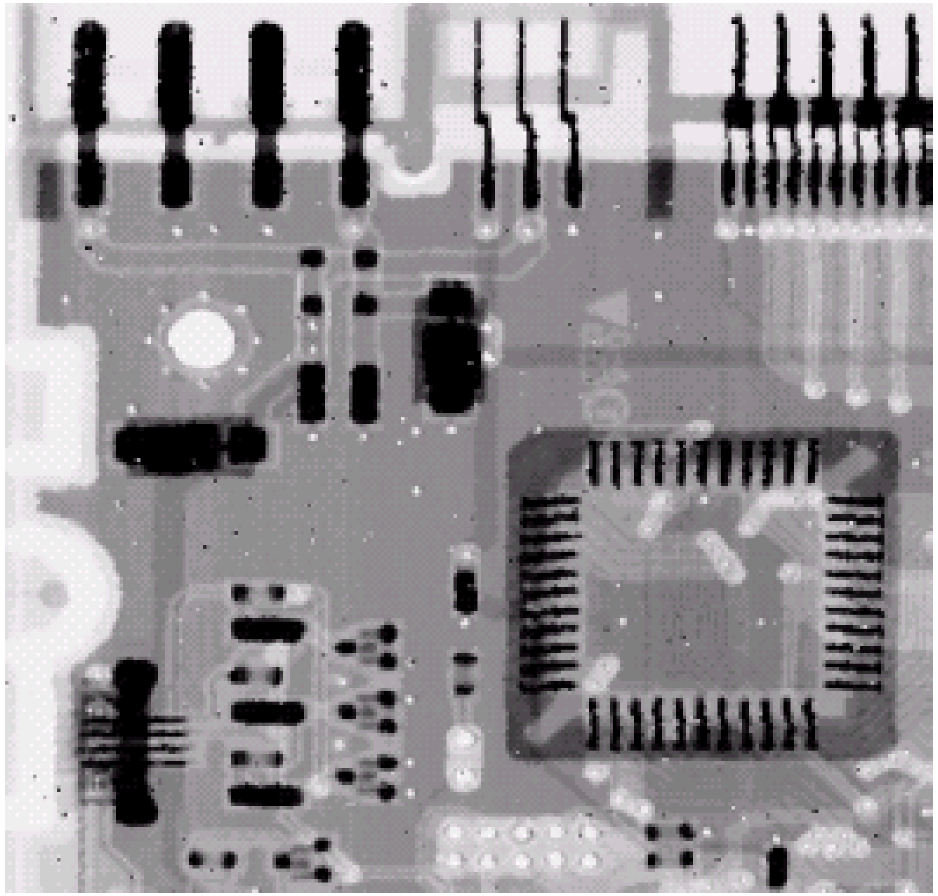


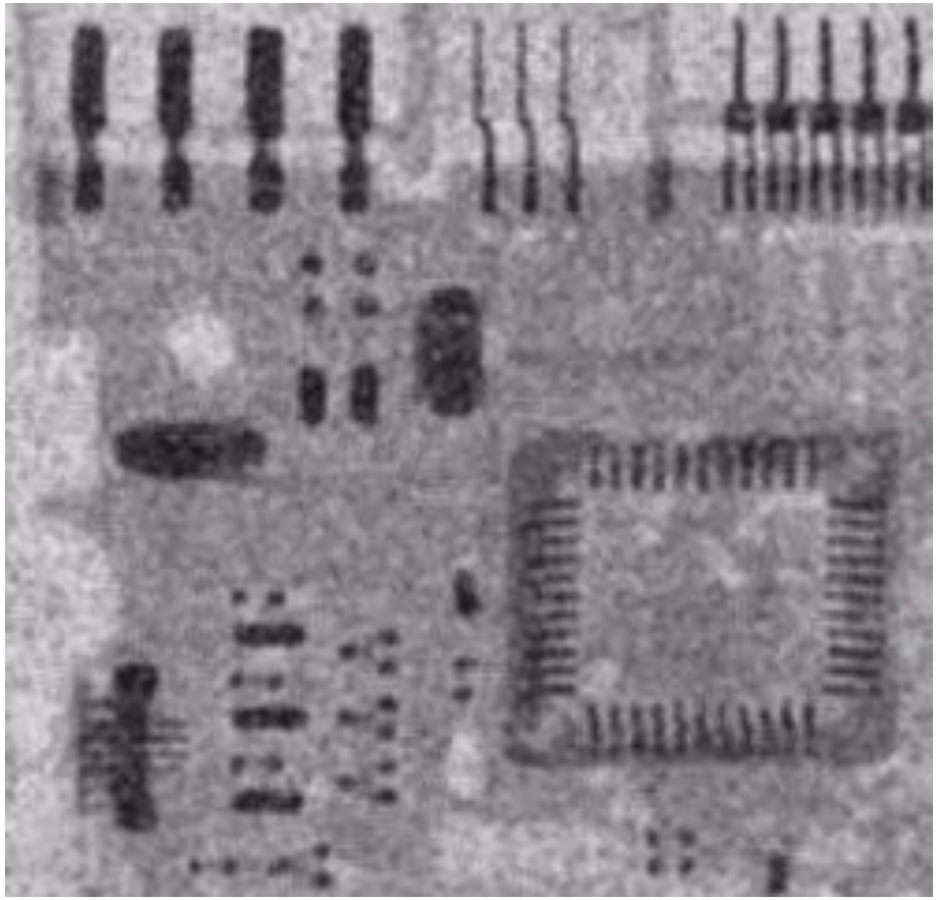


What are its problems comparing to the original image? Why?

Nonlinear Image Smoothing –Introduction

See another smoothed image comparing to the previous one





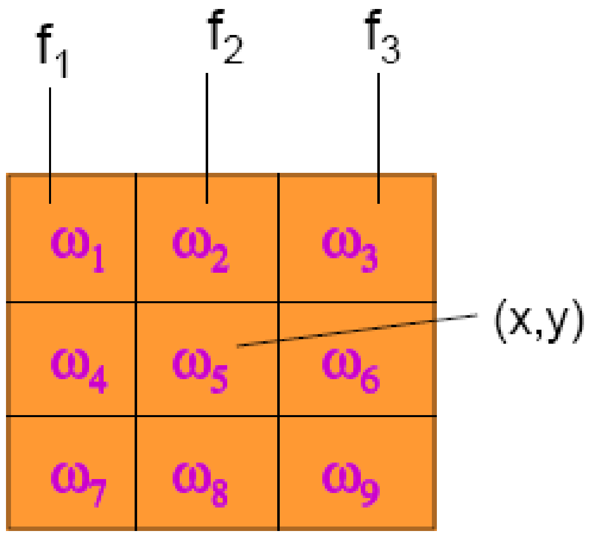
How is this smoothed image much better than the previous one?

Nonlinear Image Smoothing –Problems of Linear Filter

- Any linear filter output is a weighted average of the input pixels

$$\hat{f}(x,y) = h(x,y) * f(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} h(i,j) f(x-i, y-j)$$

$$= \sum_{(s,t)\in S_{xy}} \omega(s,t) f(s,t)$$



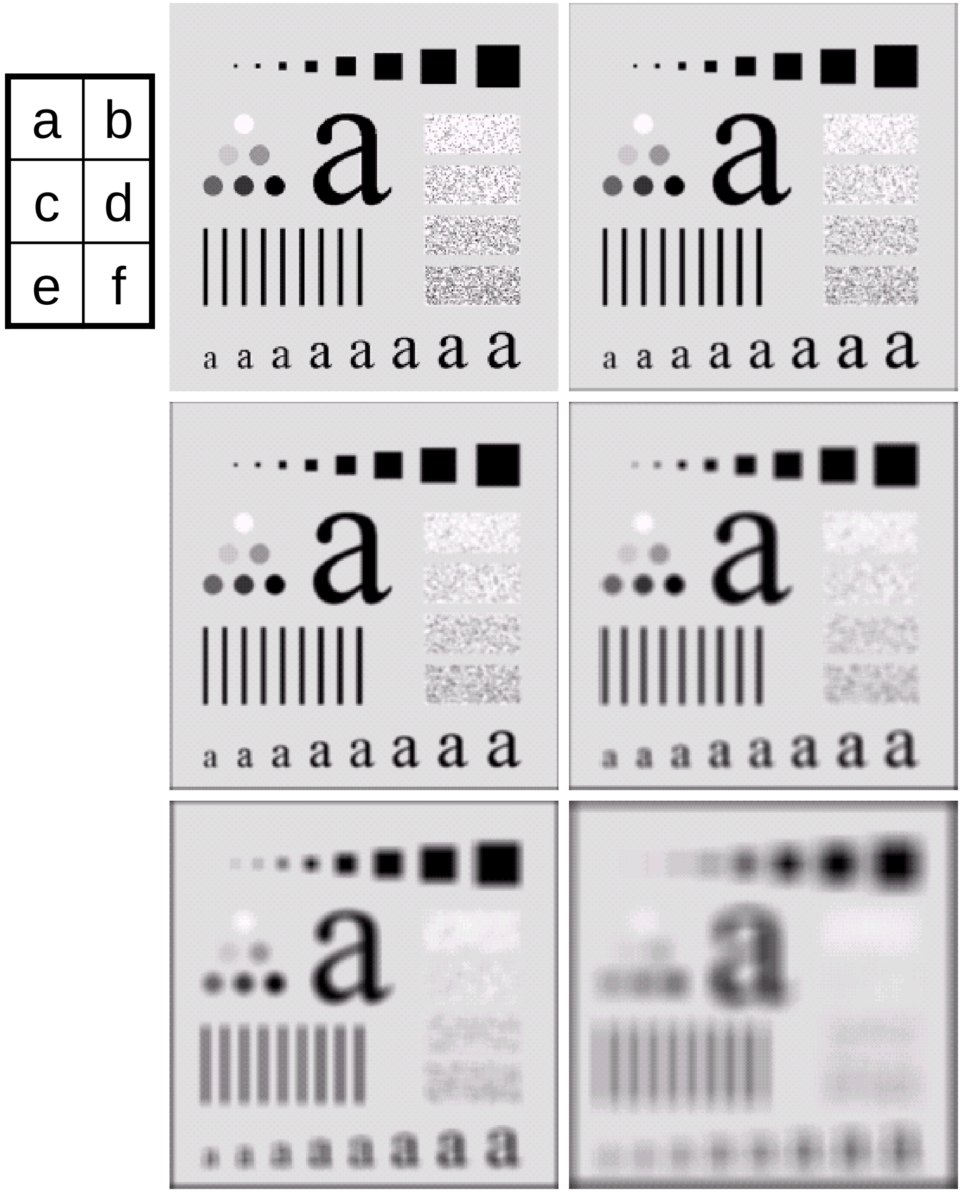
➢ What are problems of the average of pixel grey values?

image blurring, sharpness details are lost,
difficult to smooth strong noise
Why?

Nonlinear Image Smoothing –Problems of Linear Filter

a) original image 500x500 pixel

b). - f). results of smoothing with square averaging filter masks of size *nxn*, *n* = 3, 5, 9, 15 and 35, respectively.



Note:

- The size of the mask establishes the relative size of the objects that will be blended with the background.

Nonlinear Image Smoothing –Order-Statistic Filters

- The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter.

➢ The best-known example is median filter, which replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel.

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

10	20	20
20	15	20
25	20	100

→

(10,15,20,20,20,20,20,25,100)

Median=20

So replace (15) with (20)

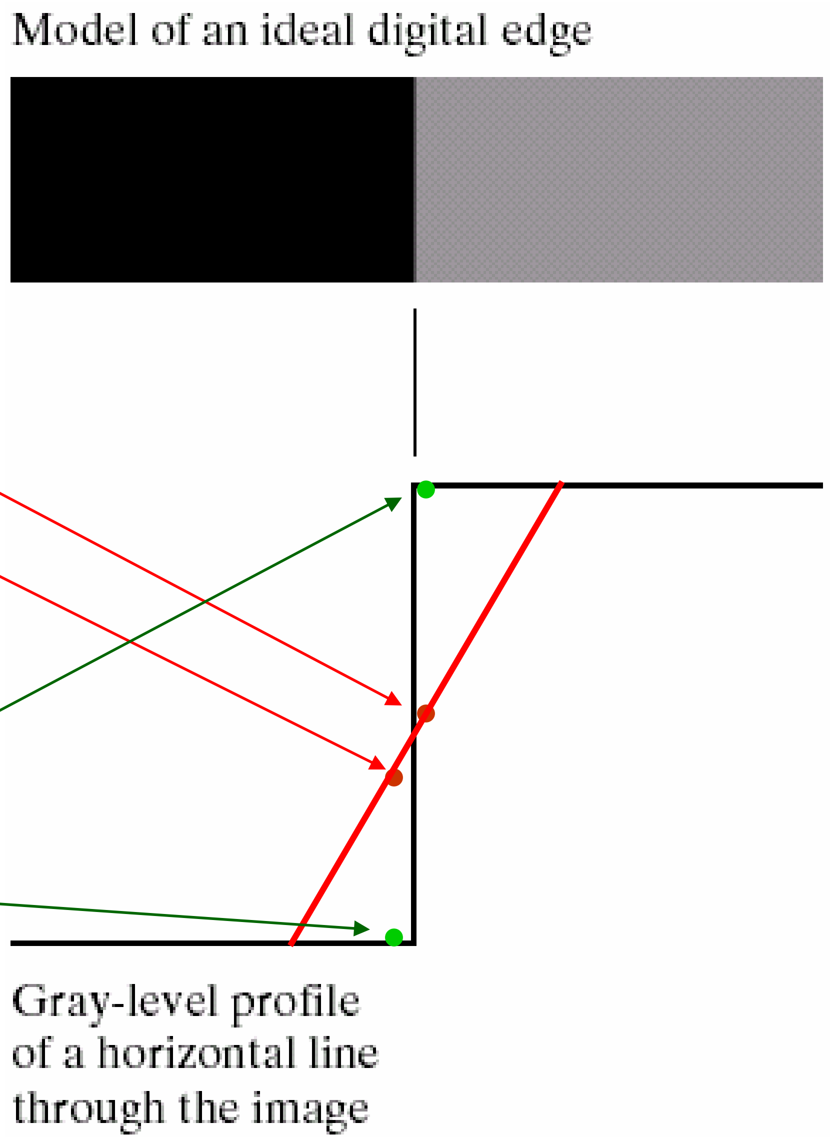
Nonlinear Image Smoothing –Median Filter

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

- Median filter forces the points with distinct gray levels to be more like their neighbors.
- Isolated clusters of pixels that are lighter or darker with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n x n median filter.
- eliminated = forced to have the value equal the median intensity of the neighbors.
- Larger clusters are affected considerably less.

Nonlinear Image Smoothing –Median Filter

- Edge is a basic and significant structure of an image.



What is the outputs of a mean filter?

$$mean\{0,0,0,\underset{\leftrightarrow}{1},1,1,1\} = 0.57$$

$$mean\{0,0,0,\underset{\leftrightarrow}{0},1,1,1,\} = 0.43$$

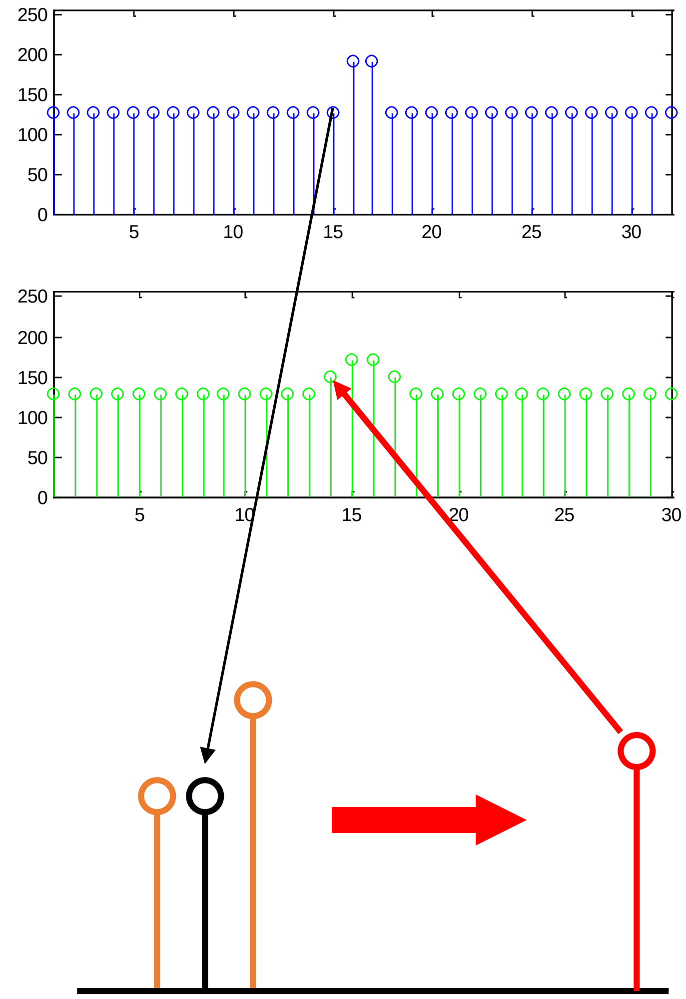
What is the outputs of a median filter?

$$median\{0,0,0,\underset{\leftrightarrow}{1},1,1,1\} = 1$$

$$median\{0,0,0,\underset{\leftrightarrow}{0},1,1,1,\} = 0$$

Nonlinear Image Smoothing –Mean vs. Median Filter

- Consider a uniform 1-D image with a pulse function.
- Pulse function corresponds to fine image detail such as lines and curves.



➢ Mean filter 'blurs' the image details.

➢ If the pulse is noise, mean filter suppress it only for some extent but spread the noise.

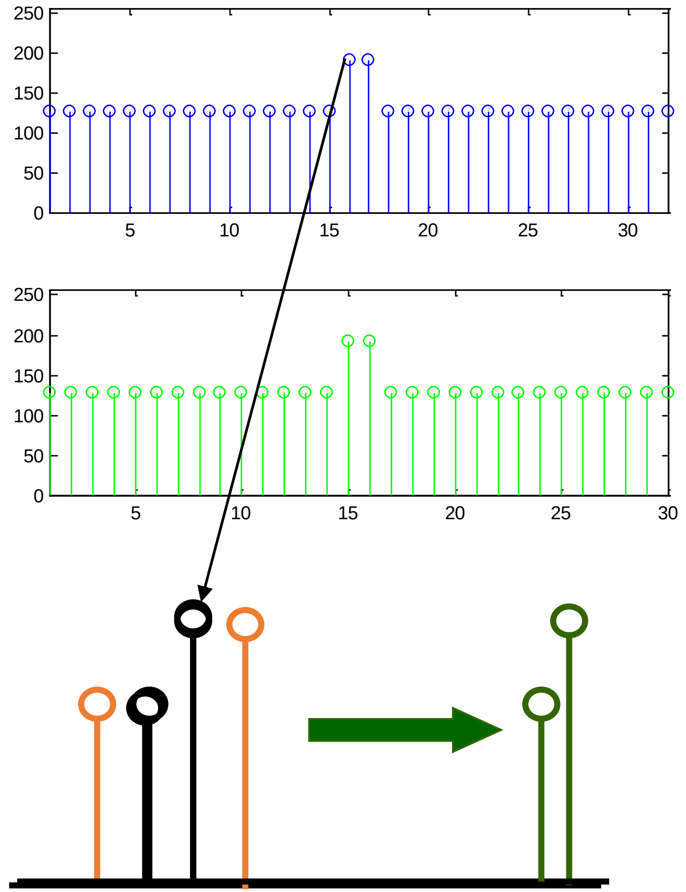
Nonlinear Image Smoothing –Mean vs. Median Filter

- Consider a uniform 1-D image with a pulse function.

- Pulse function corresponds to fine image detail such as lines and curves.

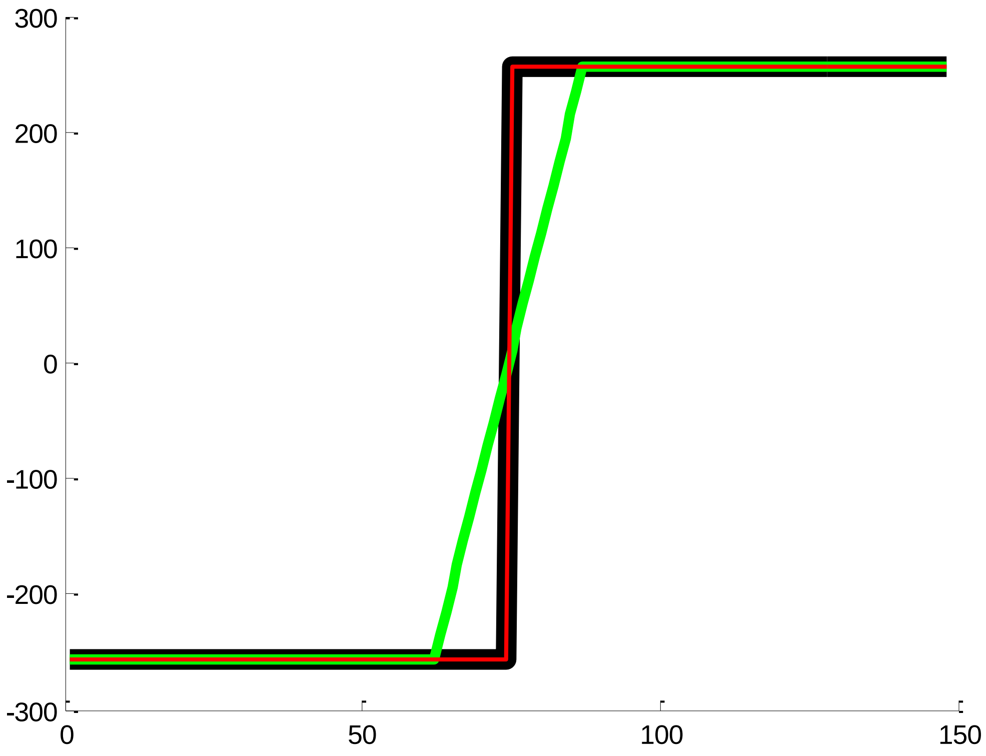
- Median filter does not ‘blur’ the edge.

➢ If the pulse is noise, 5X5 median filter totally remove such noise.



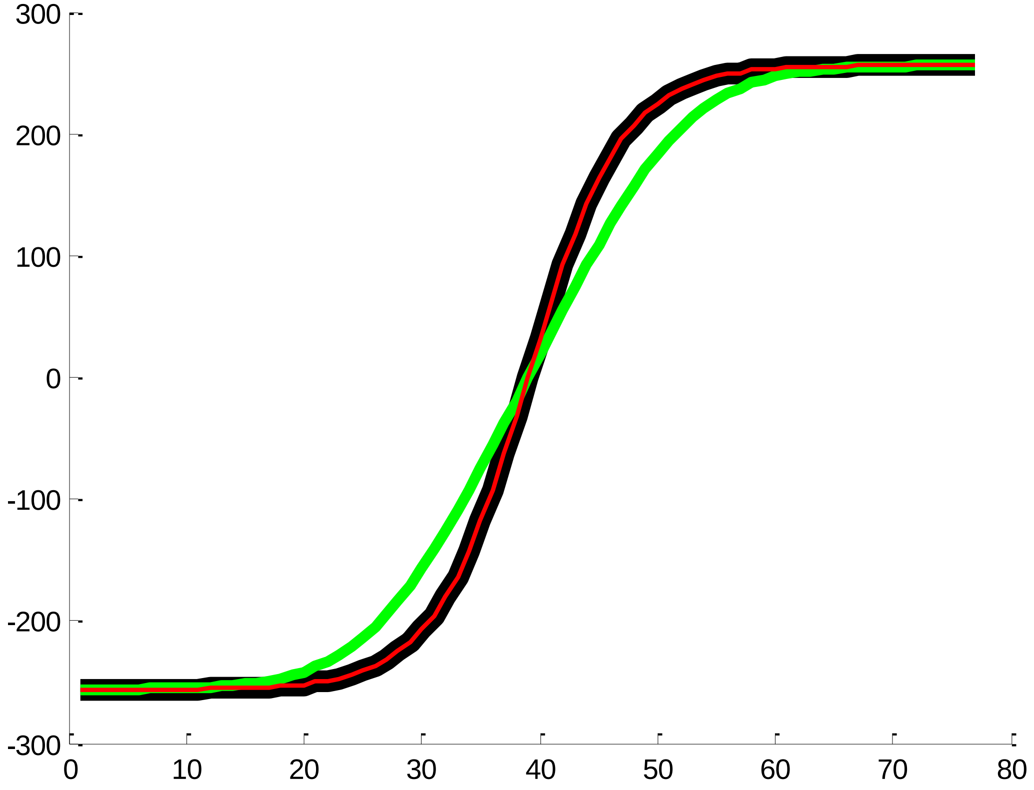
Nonlinear Image Smoothing –Mean vs. Median Filter

A simple MATLAB program can show: Mean filter blurs the step edge. Median filter preserves the step edge.



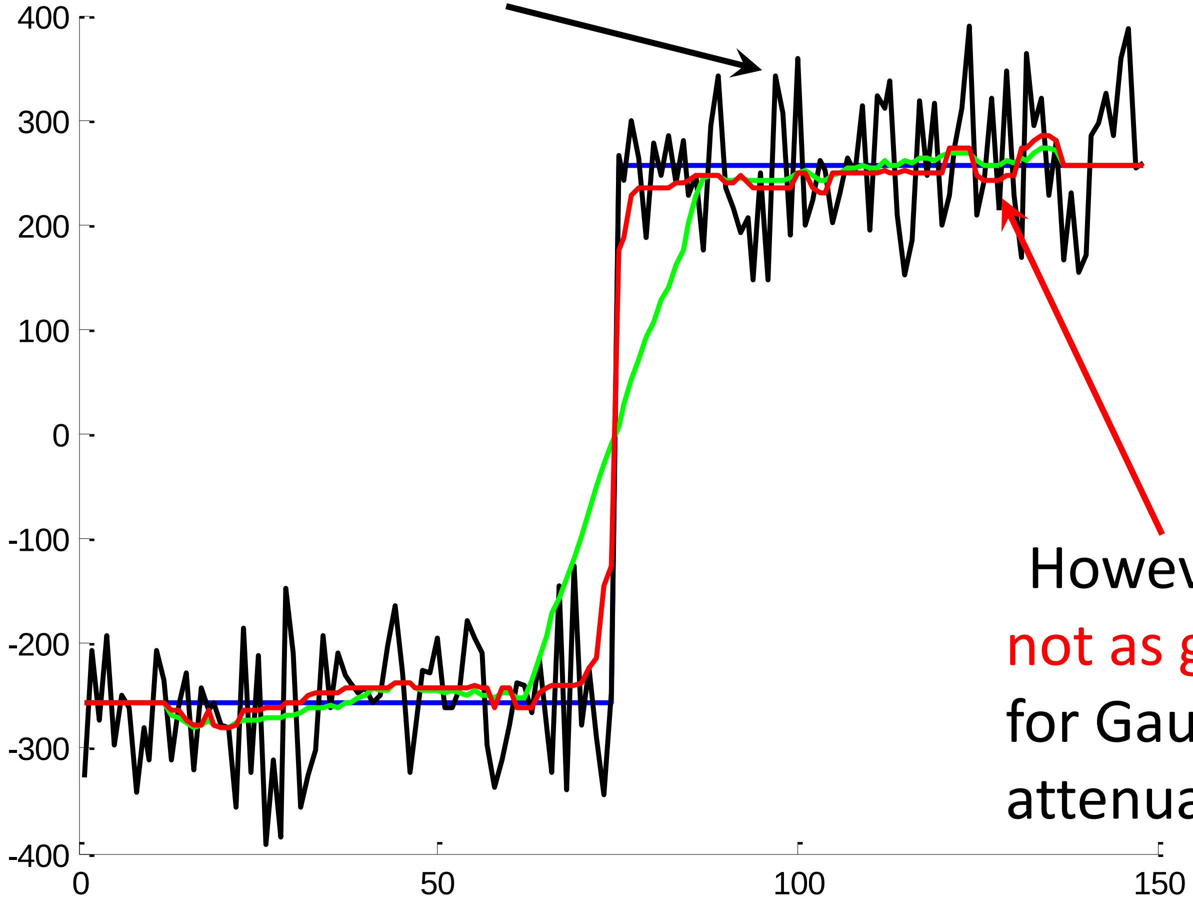
Nonlinear Image Smoothing –Mean vs. Median Filter

A simple MATLAB program can show: Mean filter further blurs the smooth edge. Median filter preserves the smooth edge.



Nonlinear Image Smoothing –Mean vs. Median Filter

A simple MATLAB program can show: Mean filter attenuates additive Gaussian noise but blurs the edge. Median filter attenuates Gaussian noise and preserves the edge.



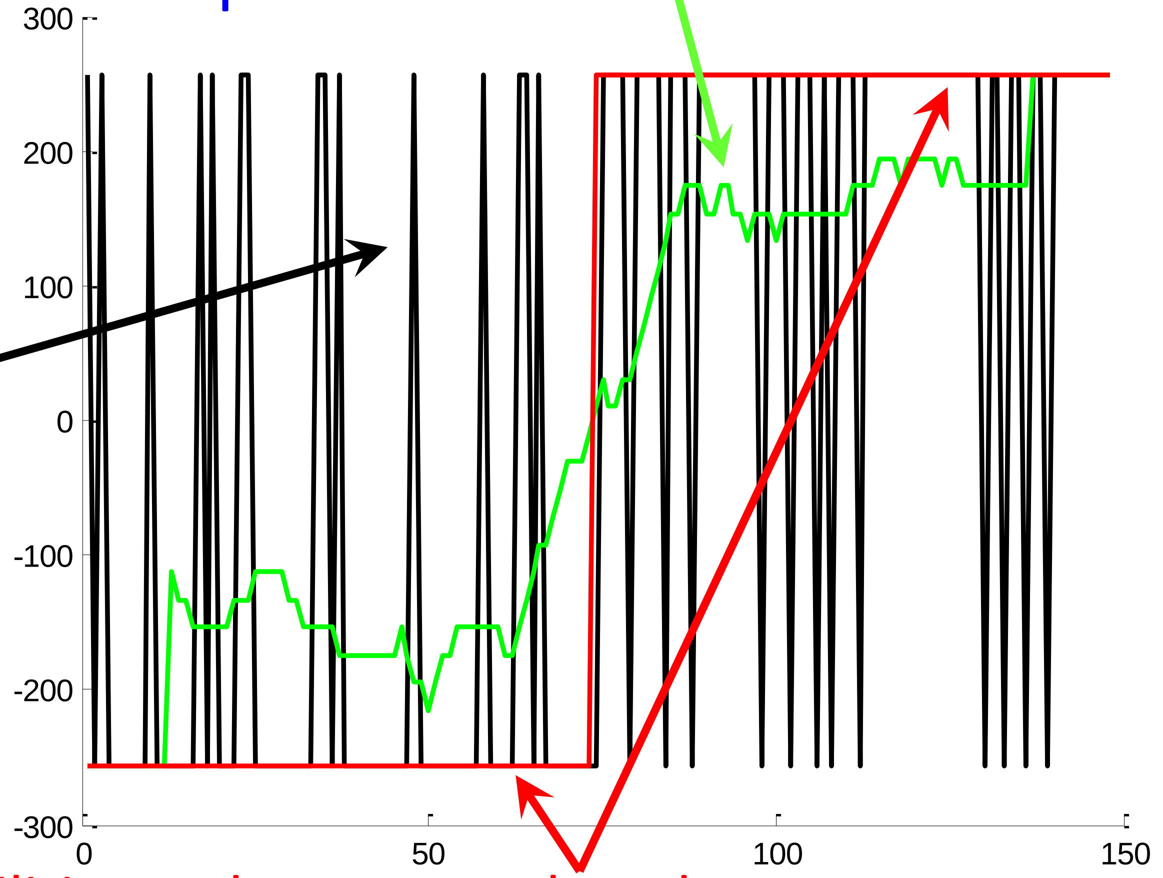
However, median filter is not as good as mean filter for Gaussian noise attenuation.

Nonlinear Image Smoothing –Mean vs. Median Filter

A simple MATLAB program can show: Mean filter is ineffective to attenuate impulsive noise

and blurs the edge.

Impulsive noise is strong in amplitude and spatial sparse



Median filter provides excellent noise-reduction capabilities and preserves the edge.

Nonlinear Image Smoothing –Mean vs. Median Filter

Original and noise corrupted images

impulse noise ⇨ salt and pepper noise.

Nonlinear Image Smoothing –Mean vs. Median Filter

Example outputs of

mean filter and median filter.

Nonlinear Image Smoothing –Median Filter Properties

- Linear filter has established theory to analyze its properties, especially in the frequency domain.
- However, It is difficult to analyze Median filter and other order-statistic filters due to their nonlinearity.

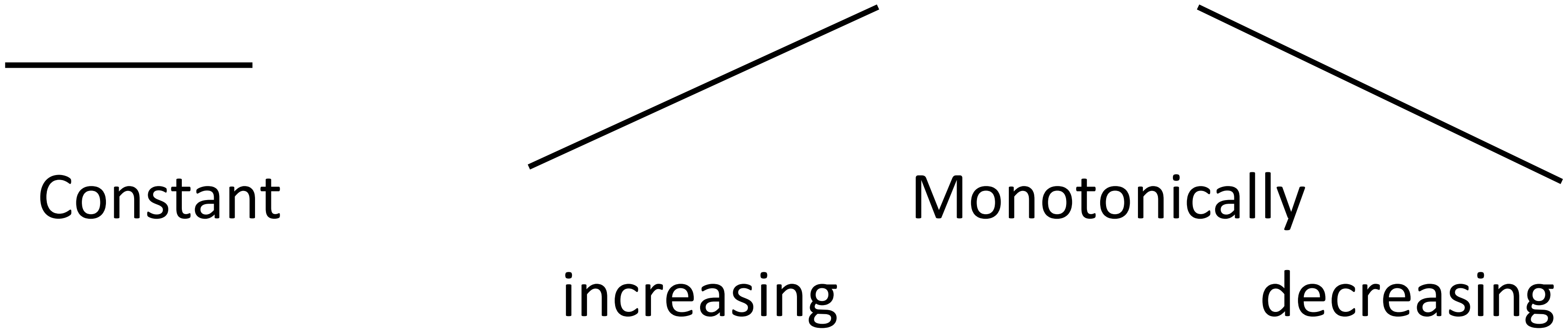
- Repeated applications of median filter to a signal result in an invariant signal called the "root signal". A root signal is invariant to further application of the median filter.
- Example: 1-D signal: Median filter length = 3

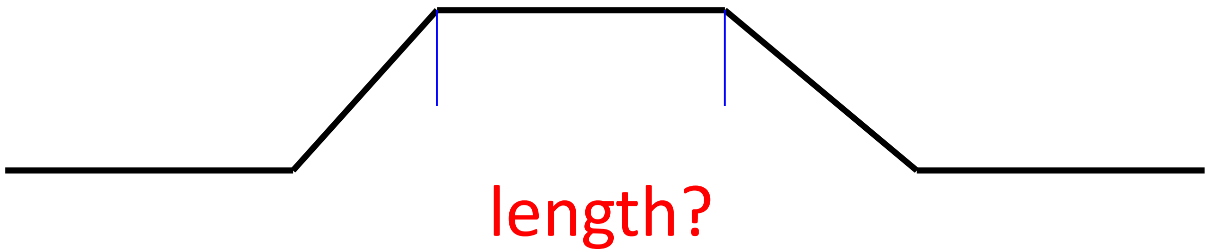
0 0 0 1 2 1 2 1 2 1 0 0 0

0 0 0 1 1 2 1 2 1 1 0 0 0
0 0 0 1 1 1 2 1 1 1 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 ~ root signal

Nonlinear Image Smoothing –Median Filter Properties

- Invariant signals to a median filter:





Nonlinear Image Smoothing –Other Order-Statistics Filter

- Max filter

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

- Min filter

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

- Midpoint filter

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

Nonlinear Image Smoothing –Other Order-Statistics Filter

- Alpha-trimmed mean filter

  -- a combination of the median and mean filter concepts

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

where $f_r(s,t)$ are the remaining $mn-d$ pixels after deleting $d/2$ lowest and the $d/2$ highest gray level pixels in the neighborhood S_{xy}