Lect. 11 Nonlinear filters for image restoration and enhancement

A priori knowledge on images and notion of pixel's neighbourhood.

Linear filters: $\hat{a}_{k,l} = \lambda_0 b_{k,l} + \bar{a}_{k,l}$; $\bar{a}_{k,l}$ is a weighted average over a spatial neighbourhood

<u>RANK FILTERS</u>: $\hat{a}_{k,l} = \Phi_{hst}\left(b_{k,l}\right)$ where $\Phi_{hst}(x)$ is a non-linear function determined by the local histogram over a certain neighbourhood of the pixel (***k,l***).

<u>Neighborhoods in terms of pixel values (*V*-neighborhoods):</u>

EV-neighborhood $EV\left\{a_{k,l}\right\} = \left\{a_{m,n} : a_{k,l} - \varepsilon_v^- \le a_{m,n} \le a_{k,l} + \varepsilon_v^+\right\}$

KNV-neighborhood

$$KNV\left(a_{k,l}\right) = \left\{a^p_{m,n} : \sum_{p=1}^{K}\left|a_{k,l} - a^p_{m,n}\right| = \min_{m,n}\right\}$$

<u>The neighbourhoods in terms of pixel position in the variational row (***R***-neighbourhoods),</u>:

$$ER\left\{a_{k,l}\right\} = \left\{a_{m,n} : R\left(a_{k,l}\right) - \varepsilon_r^- \le R\left(a_{m,n}\right) \le a_{k,l} + \varepsilon_r^+\right\};$$

$$KNR\left(a_{k,l}\right) = \left\{a^p_{m,n} : \sum_{p=1}^{K}\left|R\left(a_{k,l}\right) - R\left(a^p_{m,n}\right)\right| = \min_{m,n}\right\} ..$$

Cluster, (***CL***-neighbourhood), as a subset of pixels whose values fall into the same cluster, or mode, of the histogram as that of the central pixel.

Other possible neighbourhoods

<u>The standard neighbourhood operations:</u>

MEAN(***NBH***) - arithmetic mean over the neighbourhood ***NBH***;

ROS(***NBH***) - *R*-th Order Statistics over the neighbourhood; this includes as special cases

MED(***NBH***) - median;

MAX(***NBH***) - maximum;

MIN(***NBH***) - minimum;

MODE(***NBH***) - a value which corresponds to the highest maximum of the histogram over the neighbourhood;

RAND(***NBH***) - pseudo-random number taken from the same distribution as the histogram over the neighbourhood;

RANK(***NBH***) - position of the central pixel of the window in the variational row over the neighbourhood;

SIZE(***NBH***) -size (in pixels) of the neighbourhoods (applicable to ***EV***-, ***ER***- and ***CL***-neighbourhoods)

<u>TYPICAL ALGORITHMS:</u>

<u>Rank filters for smoothing additive and impulse noise</u> :

$$\hat{a}^{(t)}_{k,l} = SMTH\left(NBH\left(\hat{a}^{(t-1)}_{k,l}\right)\right)$$

and

$$\hat{a}^{(t)}_{k,l} = \begin{cases}\hat{a}^{(t-1)}_{k,l}, if \left|\hat{a}^{(t-1)}_{k,l} - SMTH\left(NBH\left(\hat{a}^{(t-1)}_{k,l}\right)\right)\right| \le thr \\ SMTH\left(NBH\left(\hat{a}^{(t-1)}_{k,l}\right)\right), otherwise\end{cases},$$

where ***SMTH*** is a smoothing operations (***MEAN***, ***MED***, ***ROS***, ***MODE*** or ***RAND)***, *thr* is a detection threshold *t* is number of the iteration.

<u>Rank order filters for local contrast enhancement:</u>

Unsharp masking:

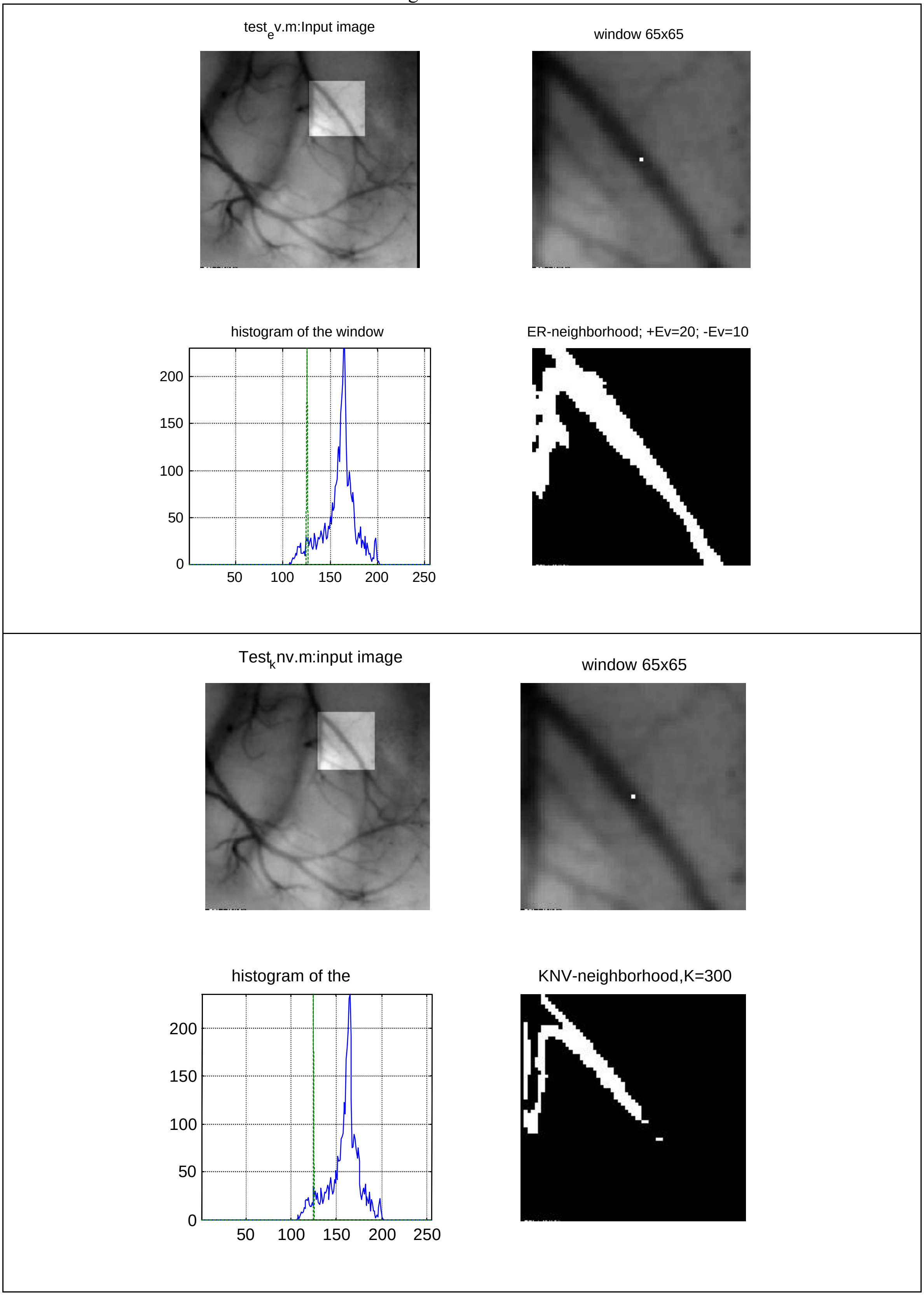
$\hat{a}_{k,l} = G\left(a_{k,l} - SMTH\left(NBH\left(a_{k,l}\right)\right)\right)$, where G is an enhancement coefficient,

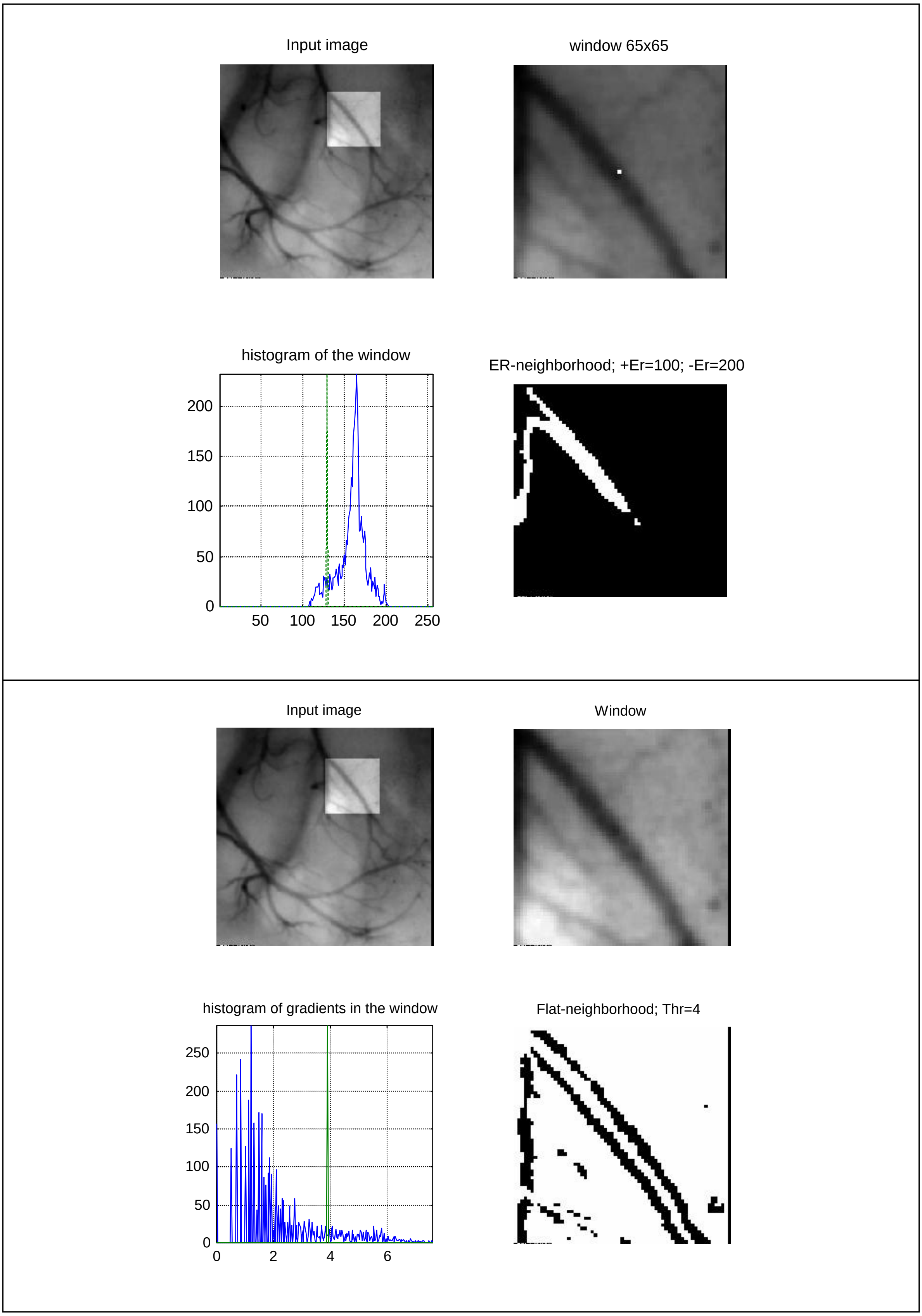
and local histogram equalization: $\hat{a}_{k,l} = RANK\left(NBH\left(a_{k,l}\right)\right)$

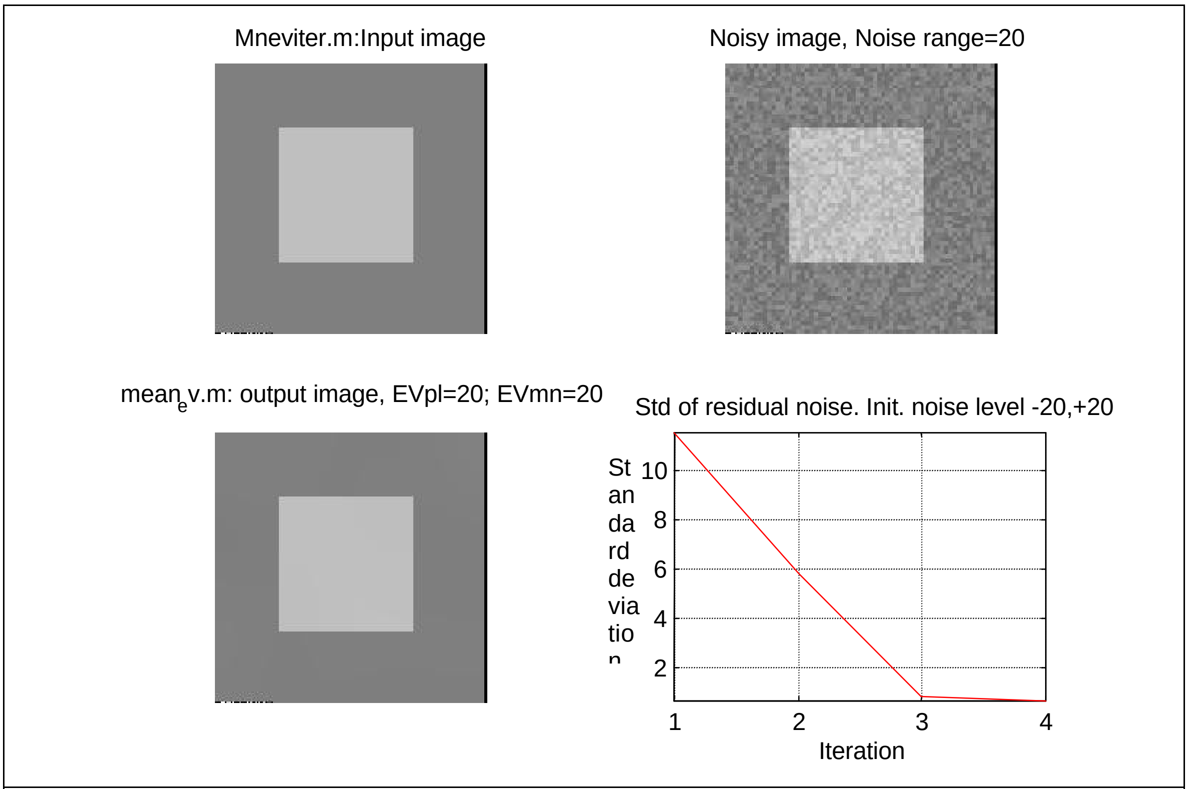
<u>Problems</u>

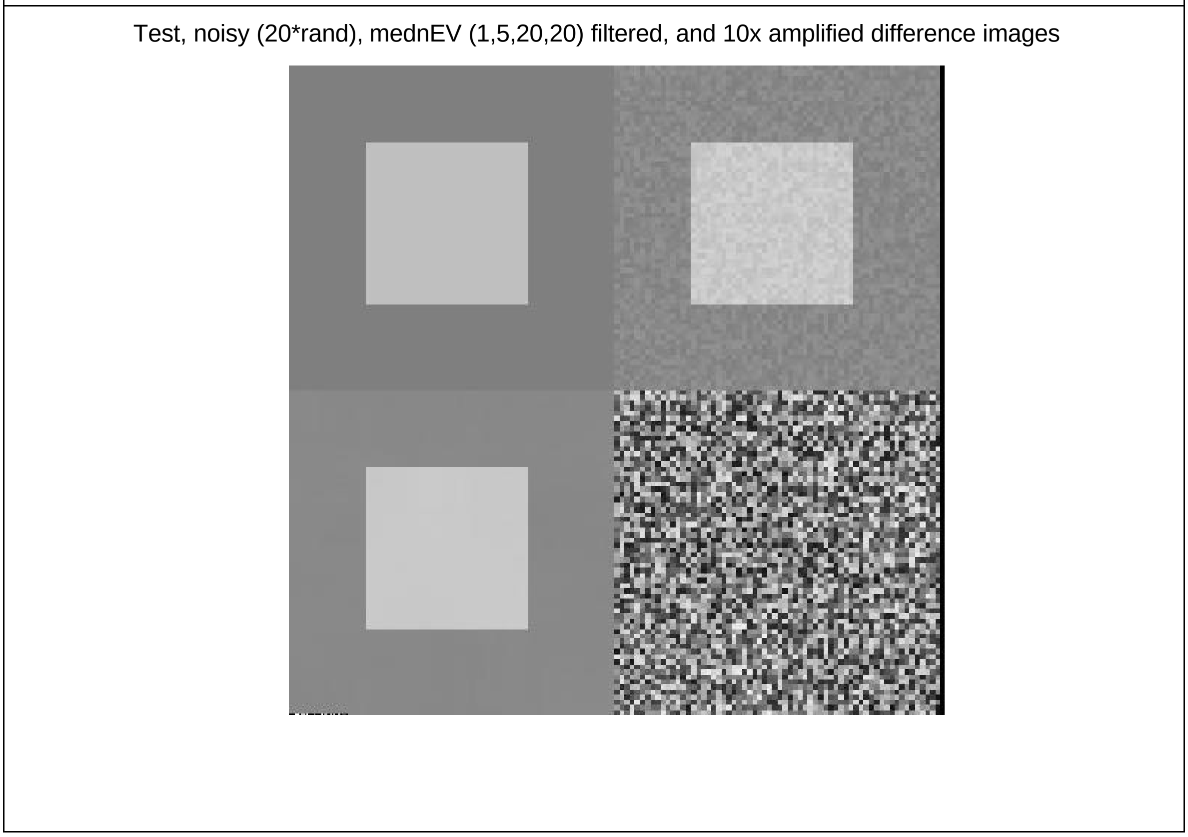
1. Explain basic idea of rank filters and principle of classification of rank filters
2. Give types of neighbourhoods and estimation operations used for the design of rank filters
3. Explain principles of efficient computer implementation of rank filters
4. Give examples of rank filters for image denoising, and enhancement

Neighborhoods:

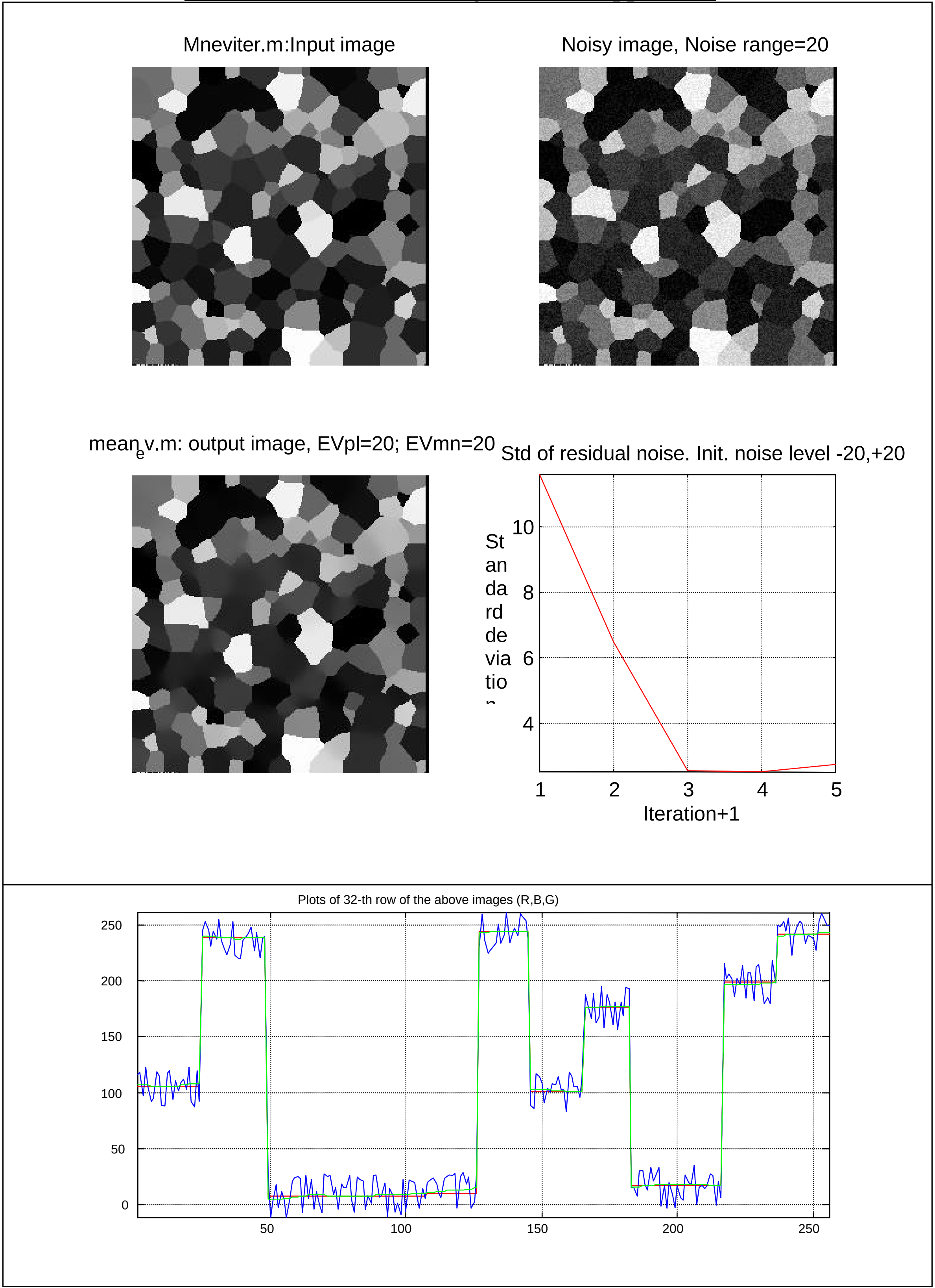



Input image
window 65x65
histogram of the window
200
150
100
50
0
50
100
150
200
250
ER-neighborhood; +Er=100; -Er=200
Input image
Window
histogram of gradients in the window
250
200
150
100
50
0
0
2
4
6
Flat-neighborhood; Thr=4

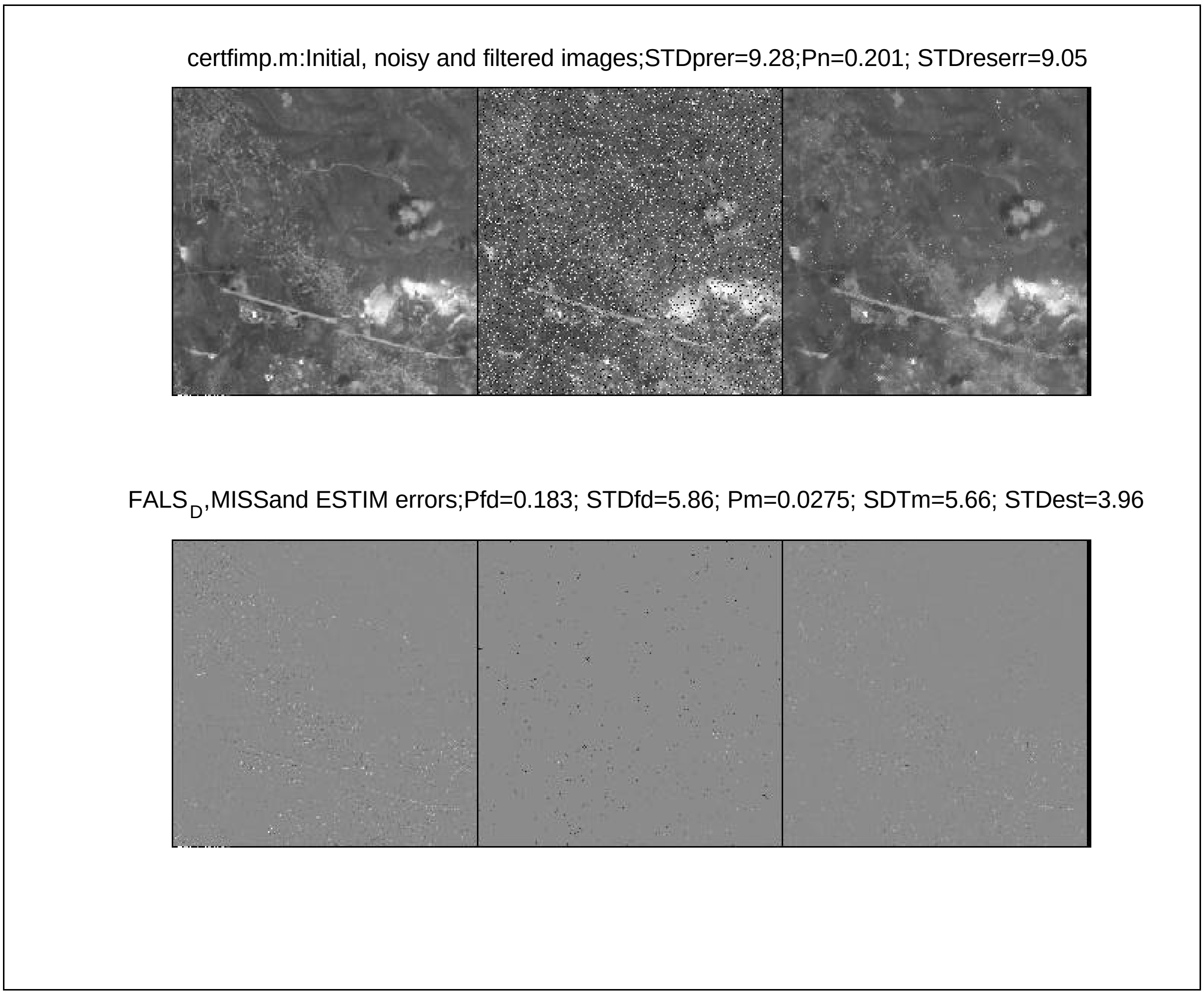

Mneviter.m:Input image
Noisy image, Noise range=20
mean_ev.m: output image, EVpl=20; EVmn=20
Std of residual noise. Init. noise level -20,+20
Standard deviation
10
8
6
4
2
1
2
3
4
Iteration


Test, noisy (20*rand), mednEV (1,5,20,20) filtered, and 10x amplified difference images

Iterative Mean(EV)filtering for noise suppression



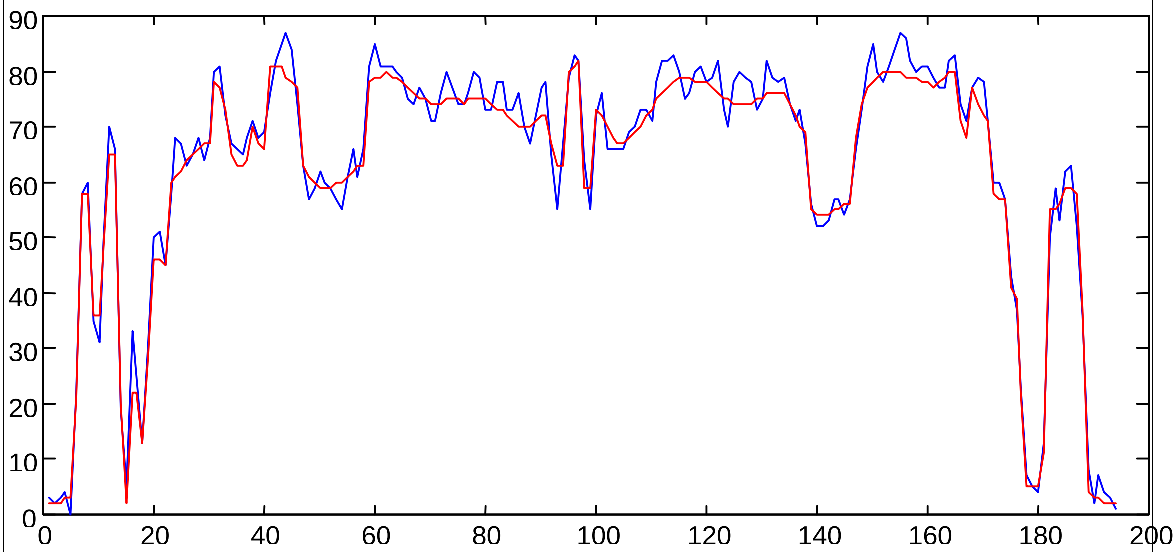
Rank filtering impulse noise


certfimp.m:Initial, noisy and filtered images;STDprer=9.28;Pn=0.201; STDreserr=9.05
FALS$_D$,MISSand ESTIM errors;Pfd=0.183; STDfd=5.86; Pm=0.0275; SDTm=5.66; STDest=3.96

Rank smoothing for image segmentation

Initial image

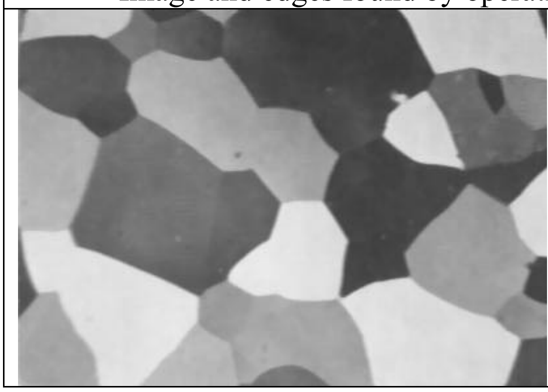
itermnev(InImg,3,3,7,7,3)

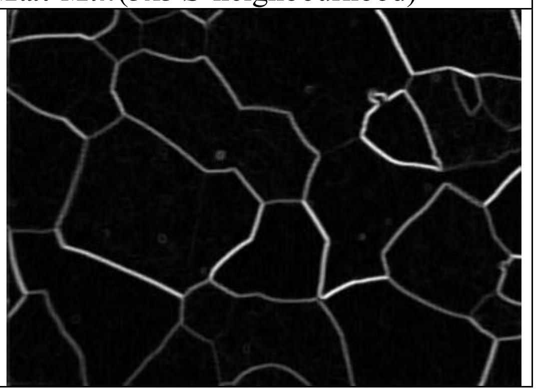


Middle rows of the initial image (blue) and itermnev-smoothed one (red)

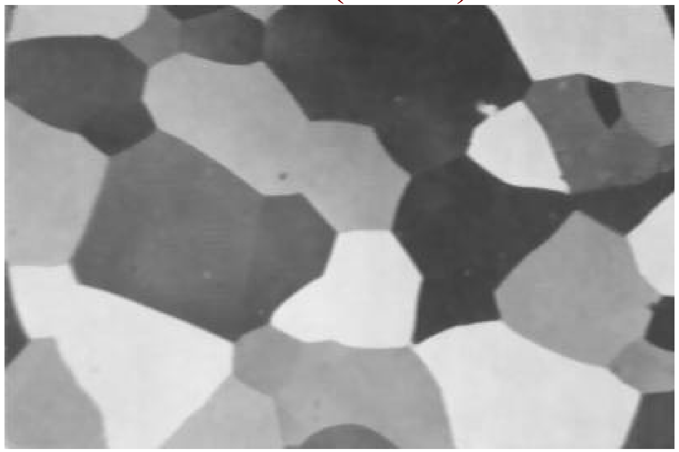
Image segmentation and boundary detection

Image and edges found by operation ***Max-Min***(3x3 ***S***-neighbourhood)

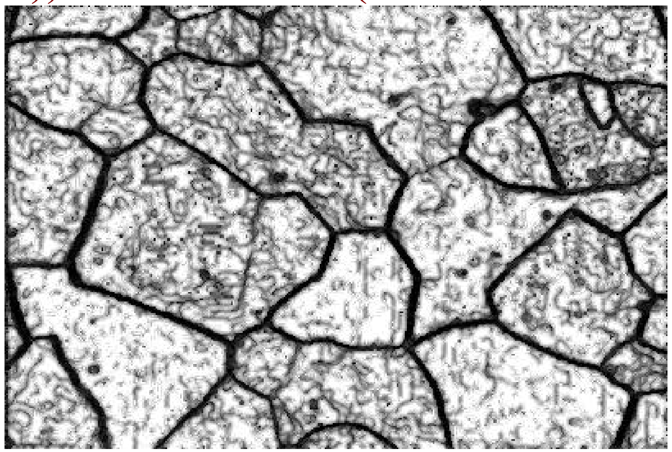




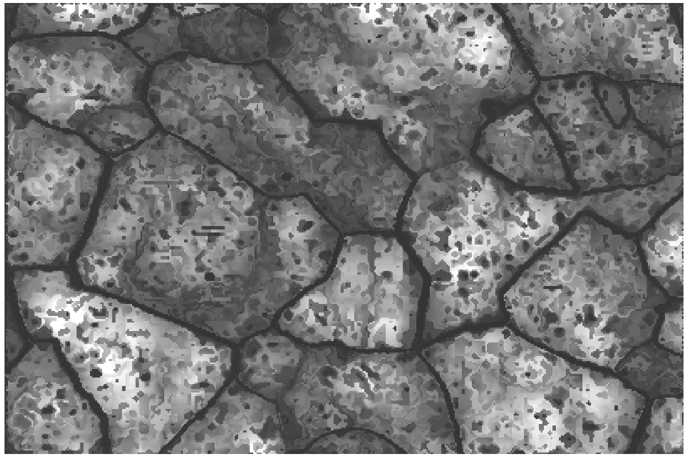
HIST(*Wnbh*) and **SIZE**(*Evnbh*(*Wnbh*)) versus **RANK**(*Wnbh*)



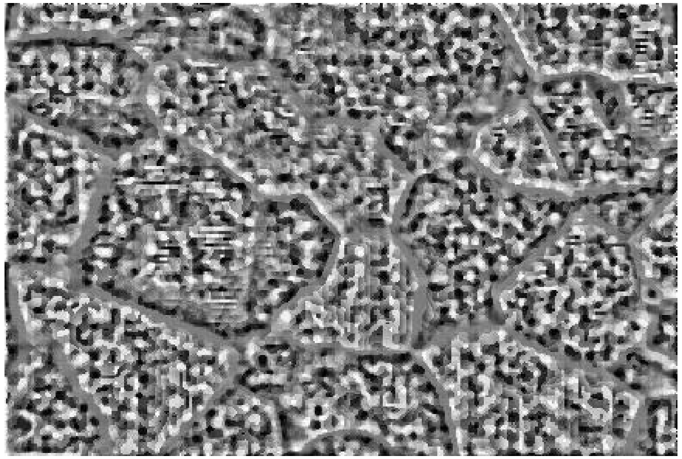
Initial image



SIZE(Evnbh(Wnbh5x5,2,2))



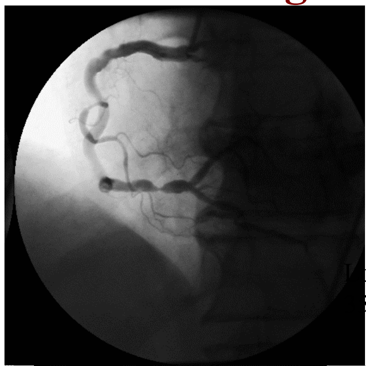
HIST(***NBH***)-filter



RANK(***NBH***)-filter (histogram equalization)

Local histogram and P-histogram equalization

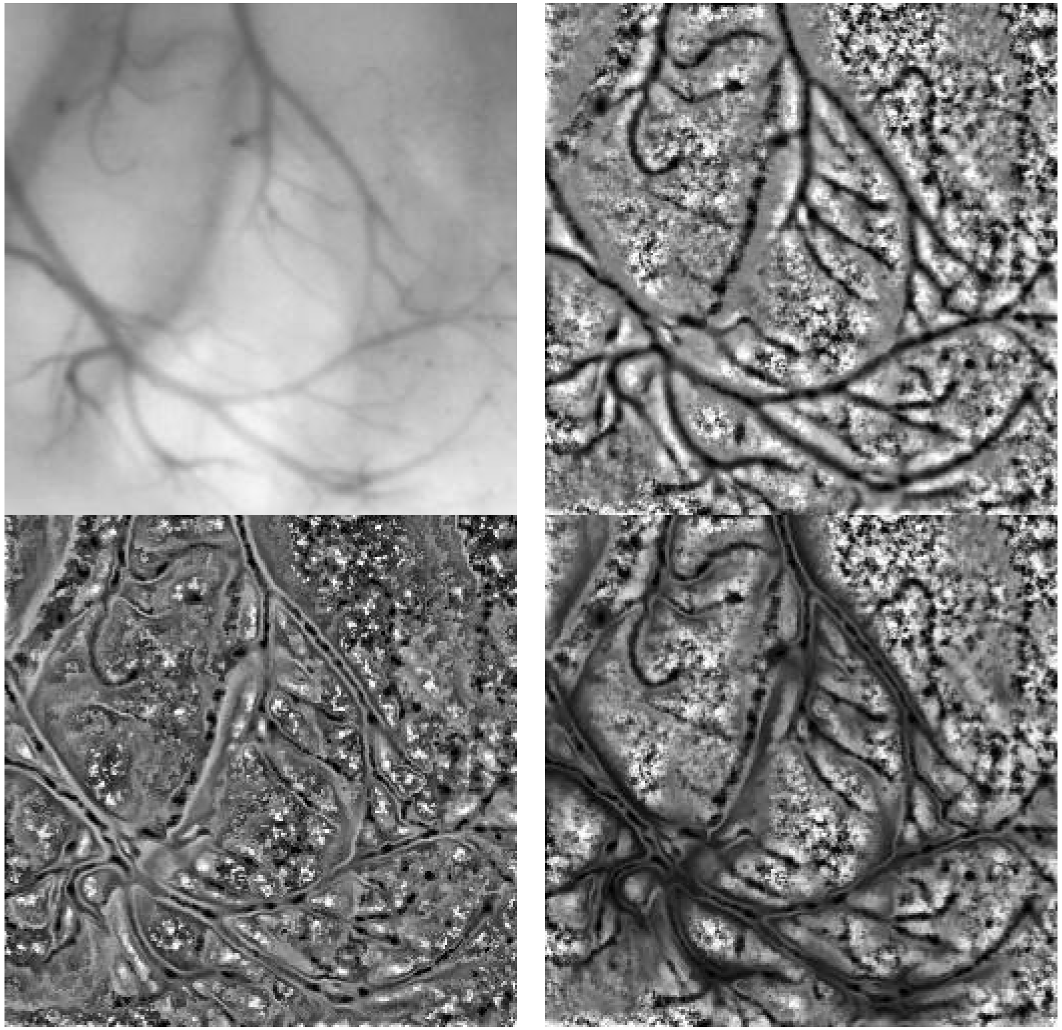
Initial image



Local histogram equalization in the window of 35x35 pixels

Local histogram P-equalization in the window of 35x35 pixels, *P*=0.3

Local histogram equalization: *Wnbh*, *EV-nbh* and *KNV-nbh*



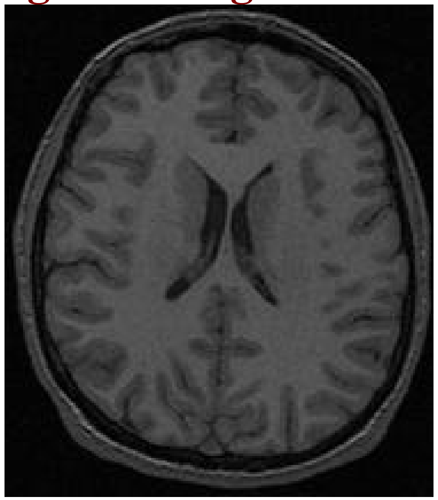
Initial image

RANK(*Wnbh*15x15)

RANK(*KNV*
(*Wnbh*15x15;113))

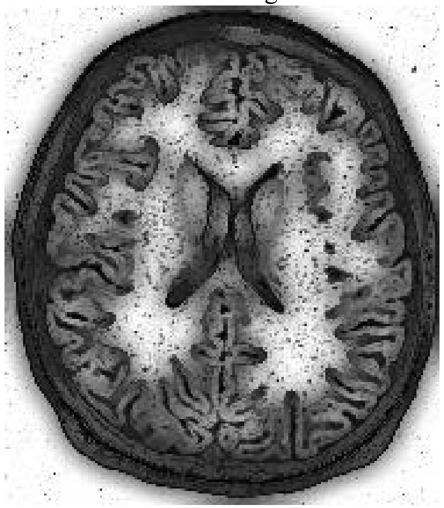
RANK(*EV*
(*Wnbh*15x15;10,10))

Smoothing and image enhancement algorithms on the base of V-neighborhoods

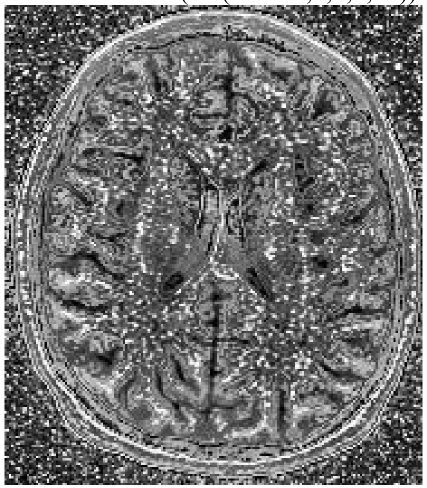


Initial image

Iterated **MEAN**(*EV*(*Wnbh*5,5,2,2,10))



SIZE(*EV*(*Wnbh*15x15,7,7))



RANK(*KNV*(*Wnbh*,25,25,205))