LECTURE 3.2

Image restoration and enhancement: Nonlinear filters.

3.2.1 Local criteria of image quality

The design of filters for image processing should be based on a processing quality criterion. Every criterion assumes certain method for numerical evaluation of pixel wise deviation of processing results from desired ones and certain procedure of averaging pixel wise deviations over a set of pixels and image ensembles. In image processing it is also preferable to design filters that fit local rather then global peculiarities of images (see Lect. 3.1). It follows from this requirement that processing quality evaluation should also be local rather then global. Mathematical formulation of local processing criteria is given by equation:

$$\boldsymbol{AVLOSS}(k,l) = \mathbf{AV}\left\{ \sum_{m,n} \boldsymbol{LOC}\left(m,n;a^{tr}_{k,l}\right) \boldsymbol{LOSS}\left(a^{tr}_{m,n}, a^{out}_{m,n}\right) \right\} \qquad (3.2.1)$$

where $\{m,n\}$ and $\{k,l\}$ are pixel co-ordinates, $\{a^{tr}_{m,n}\}$ and $\{a^{out}_{m,n}\}$ are “true” image samples that are processing goal and output image samples obtained as a result of processing. Loss function ***LOSS* (.)** evaluates pixel wise deviation of processing results from the desired ones. Locality function ***LOC*(.)** that takes values between 0 and 1 selects a subset of pixels and weights the deviations in them for averaging by weighted summation. **AV** is a statistical averaging operator over random factors (such as, for instance, imaging system noise) that may be involved or should be taken into account when evaluating processing quality. Locality function ***LOC*(.)** and loss function ***LOSS* (.)** are supposed to be selected on the base of a priori knowledge on images and imaging systems purposes and also on purposes of the processing.

As one can see from this definition, local criteria evaluate image processing quality for every given pixel on the base of averaging losses over a subset of pixels that are selected from all available pixels by the locality function. We will call this subset of pixels neighborhood of the given pixel:

$$\boldsymbol{NBH}_{k,l} = \forall (m,n):\ \boldsymbol{LOC}\left(m,n;a_{k,l}\right) \geq 0 \qquad (3.2.2)$$

3.2.2 Filter parameterization: Linear and nonlinear filters

Filter design should be aimed at minimization of criterion function (Eq.(3.2.1)) by means of selecting mappings input signal $\left\{a^{inp}_{k,l}\right\} \Rightarrow \left\{a^{out}_{k,l}\right\}$ to output signal. In some cases optimal mapping can be derived from signal/noise model (see Lect. 4 on optimal algorithms for target detection). Most frequently, however, one need to parameterize mapping algorithm and look for optimal parameters. Linear filters represent the most known example of the filter parameterization. Linear filtering can be described by equation:

$$a^{out}_{m,n} = \lambda_0 a^{inp}_{k,l} + \bar{a}_{k,l}\ , \tag{3.2.3}$$

in which $\bar{a}_{k,l}$ is a weighted average over a spatial neighborhood of pixel (k,l). (Note that filters are called linear for equation (3.2.3) is equation of a straight line in coordinates $\left(a^{inp}_{k,l}, a^{out}_{k,l}\right)$). One can naturally extend this example to nonlinear filters:

$$a^{out}_{k,l} = \Phi_{NBH\left(a^{out}_{k,l}\right)}\left(a^{inp}_{k,l}\right), \tag{3.2.4}$$

where $\Phi_{NBH\left(a^{out}_{k,l}\right)}(\bullet)$ is a nonlinear function that depends on pixel's neighborhood $NBH\left(a^{inp}_{k,l}\right)$. We will call this function that maps input pixel value $a^{inp}_{k,l}$ to output pixel value $a^{out}_{k,l}$ estimation operation $ESTM(\bullet)$ and define nonlinear filters as mapping:

$$\left\{a^{inp}_{k,l}\right\} \rightarrow a^{out}_{k,l}\ : a^{out}_{k,l} = ESTM\left(NBH\left(a^{inp}_{k,l}\right)\right). \tag{3.2.5}$$

All known nonlinear filters work in sliding window and can be classified in terms of neighborhood building and estimation operations ([1-3]). Generally, neighborhood building operations are multi stage. Fig. 3.2.1 shows with two stages neighborhood building flow diagram of signal nonlinear filtering.

A popular and powerful subclass on nonlinear filters form ***rank filters*** ([3,4]). In rank filters estimation operations are determined by the local histogram over a certain neighborhood of pixel (***k,l***):

$$a^{out}_{k,l} = ESTM_{\mathrm{hst}\left(NBH\left(a^{inp}_{k,l}\right)\right)}\left(a^{inp}_{k,l}\right). \tag{3.2.6}$$

3.2.3 Neighborhoods used in rank filters

Neighborhood is a set of elements and attributes that are inputs to the estimation operation. Neighborhood attributes may be associated both with elements of the neighborhood and with the neighborhood as a whole. The following attributes may be used as attributes of neighborhood elements:

- **Co-ordinate** in the window
- **Value**.
- **Rank**, or position of the element in the variational row formed from all elements of the neighborhood by sorting them from minimum to maximum. Rank of an element shows number of elements of the neighborhood that have lower value than that of the given element.
- **Cardinality**, or number of neighborhood elements with the same (quantized) value.
- **Geometrical features**, such as gradient, curvature, etc.

Attributes **Rank** and **Cardinality** are interrelated and can actually be regarded as two implementations of the same quality. While **Rank** is associated with variational row, **Cardinality** is associated with histogram over the neighborhood. Choice between them is governed, in particular, by their computational complexity. The computational complexity of operating with histogram is $O(Q)$, where Q is the number of quantization levels used for representing signal values. Computational complexity of operating with variational row is $O(N)$, where N is the number of neighborhood elements. Histogram of the window elements in each window position can be easily computed recursively from the histogram in the window previous position with only $O(1)$ operations. Recursive formation of the variational row is also possible with the complexity of $O(N)$ operations.

According to these attributes one can distinguish the following type of neighborhoods used in the design of rank filters.

- Co-ordinate based neighborhoods (***C***-neighborhoods)

SHnbh-, or Shape-neighborhoods that are formed from window samples by selecting them according to their co-ordinates. In 2-D and multi-dimensional case this corresponds to forming spatial neighborhoods of a certain shape.

- Pixel values based neighborhoods(***V***-neighborhoods):

EV-neighborhood:

$$EV\{a_{k,l}\} = \{a_{m,n} : a_{k,l} - e_v^- \le a_{m,n} \le a_{k,l} + e_v^+\}. \qquad (3.2.7)$$

e_v^- and e_v^+ are parameters of the neighborhood.

K-nearest values (***KNV***-) neighborhood

$$KNV(a_{k,l}) = \left\{ a_{m,n}^p : \sum_{p=1}^{K} \left| a_{k,l} - a_{m,n}^p \right| = \min_{m,n} \right\}. \qquad (3.2.8)$$

K is neighborhood parameter.

- Pixel rank (position in the variational row) based neighborhoods (***R***-neighborhoods):

ER-neighborhood:

$$ER\{a_{k,l}\} = \{a_{m,n} : R(a_{k,l}) - e_r^- \le R(a_{m,n}) \le a_{k,l} + e_r^+\}; \qquad (3.2.9)$$

ER-neighborhood:

$$KNR(a_{k,l}) = \left\{ a_{m,n}^p : \sum_{p=1}^{K} \left| R(a_{k,l}) - R(a_{m,n}^p) \right| = \min_{m,n} \right\}. \qquad (3.2.10)$$

Qnbh-, or quantile neighborhood:

$$Qnbh = \{a_{m,n} : R_{left} \le R(a_{m,n}) \le R_{right}\}. \qquad (3.2.11)$$

$e_r^-, e_r^-, K, R_{left}$ and R_{right} are parameters of the corresponding neighborhoods.

- Histogram based neighborhood (***H***-neighborhood):

Cluster, (***CL***-neighborhood): subset of pixels that belong to the same cluster, or mode, of the histogram as that of the central pixel.

- Image geometrical feature based neighborhoods:

Flat-neighborhood:

$$FLAT\{a_{k,l}\} = \{a_{m,n} : |\nabla_{m,n}| < Thr\}, \qquad (3.2.12)$$

where $\nabla_{m,n}$ is image gradient in pixel (m,n). ***Thr*** is neighborhood parameter.

Selection of the neighborhood type and its parameters is based on a priori information concerning processed image and processing task.

The notions of ***EV***, ***KNV***, ***ER*** and ***Flat*** neighborhoods are illustrated in Figs. 3.2.2 a - d. As it was mentioned, rank algorithms are built on the base of local histograms over pixel neighborhoods (Eq. 3.2.6). Since histograms completely ignore spatial relationships between pixels it may seem that rank algorithms may fail to use spatial information which is one of the most important attributes of images. But no matter how strange it may appear, this property of rank algorithms is more often an advantage then disadvantage; it is one more aspect of their adaptivity. As a matter of fact, spatial relations between pixels (defined, for example, by their membership in one image detail or object) manifest themselves indirectly in local histograms. It is illustrated in Fig.3.2.3 which shows how closely one can imitate and image by building its copy from random numbers using only local histograms over an appropriately selected neighborhood.

3.2.4 Estimation operations

Estimation operations used to generate filter current sample output value are a many-to-one mappings of neighborhood attributes to one output value. Selection of a particular estimation operation is governed by a priori knowledge regarding signal properties and the criterion of the processing quality. Here is a list of typical estimation operations composed on the base of analysis of nonlinear filters known from literature.

- **MEAN(*NBH*)**: arithmetic mean of elements of the neighborhood. As it is well known, **MEAN** is an optimal MAP- (Maximal A Posteriori Probability) estimation of a location parameter of data in the assumption that data are observations of a single value distorted by an additive noncorrelated Gaussian random values (noise).
- **PROD(*NBH*)**: product of elements of the neighborhood. Product is an operation homomorphic to the addition involved in **MEAN** operation: sum of logarithms of a set of values is logarithm of their product.
- **K_ROS(*NBH*)**: ***K***-th rank order statistics over the neighborhood. The operation picks up the element that occupies ***K***-th place (has *rank* ***K***) in the variational row. Special cases of **K_ROS(*NBH*)** operation are:
- **MIN(*NBH*)**: minimum over the neighborhood;
- **MAX(*NBH*)**: maximum over the neighborhood;
- **MEDN(*NBH*)**: median over the neighborhood, that is central element of the variational row.

  These operations are optimal MAP estimations for other then additive Gaussian noise models. For instance, if neighborhood elements are observations of a constant distorted by addition to it independent random values with exponential distribution density, **MEDN(*NBH*)** is known to be optimal MAP estimation of the constant. If additive noise samples have one-sided distribution and affect not all data, **MIN(*NBH*)** or **MAX(*NBH*)** might be optimal estimations. One can imagine models for which intermediate Rank Order Statistics may be optimal estimations.

- **MODE(*NBH*).** This estimation operation provides value of the neighborhood element with highest cardinality. It is sort of an analogue to **MAX(*NBH*).**
- **RAND(*NBH*)**. This estimation produces a pseudo-random number taken from an ensemble with the same distribution density as that of elements of the neighborhood. This operation may look exotic and is not a traditional one in the theory of estimation. However it has a remarkable property that it generates an estimation that is statistically (in terms of the distribution density) akin to the neighborhood elements: it has the same mean value, standard deviation, all other moments, etc. In certain cases it may serve as a sort of a "soft" alternative to the above "hard" estimations .
- **SIZE(*NBH*)**. This operation provides number of elements of the neighborhood.
- **SPRD(*NBH*) –** spread od data within the neighborhood. As a measure of data spread, standard deviation **STDEV(*NBH*)** over the neighborhood can be used. However, conventional standard deviation operation for evaluation of data variance is nonrobust against outliers. In case of outliers present in the data, Interquantil Distance **IQDIST(*NBH*)** may be preferable:

$$\mathbf{IQDIST}\left(NBH\right) = \mathbf{R\_ROS}\left(NBH\right) - \mathbf{L\_ROS}\left(NBH\right), \qquad (3.1.13)$$

where $1 \le L < R \le \mathrm{SIZE}\left(NBH\right)$.

There is also a number of auxiliary relation operations between individual elements of the neighborhood and entire neighborhood such as:

- **MEMB(*NBH*,*a*)**. This is a binary operation that evaluates by 0 and 1 membership of a certain window element **a** in the neighborhood.
- **ATTR(*NBH*,*a*)**. This operation provides certain attribute of element **a** in the neighborhood defined by its another attribute. Examples of this operation are:
- **RANK(*NBH*,a_k)**: rank or value of the element defined by its co-ordinate k in the window. As it was mentioned, ranks of neighborhood elements can be found from histogram over the neighborhood. A natural generalization to ordering through the histogram is histogram modification by some function prior to the ordering:

$$f\_histogram = f\left(histogram\right). \qquad (3.2.14)$$

One of the useful special cases is that of P-th low nonlinearity:

$$P\_histogram = (histogram)^P . \qquad (3.2.15)$$

Modified in this sense **RANK(*NBH,a$_k$*)** we'll denote ***P*_RANK(*NBH,a$_k$*).**

- **COORD(NBH,a$_r$):** co-ordinate of the element with a certain rank **r**.

In addition note that different combinations of the basic operations can also be used for generating filter output.

3.2.6 Typical algorithms

According to local criteria (Eq. 3.2.1) locality function $LOC\left(m,n;a_{k,l}^{tr}\right)$ for pixel (k,l) is defined by its "true" value $a_{k,l}^{tr}$. This value is unknown and its approximation is the goal of the processing. A possible solution of this vicious circle is iterative processing that begins process of estimating of $a_{k,l}^{tr}$ from $a_{k,l}^{inp}$ and then proceeds iteratively taking, for building neighborhood at t-th iteration an estimate $a_{k,l}^{out(t-1)}$ obtained on the previous step:

$$\hat{a}_k^{out(t)} = \mathrm{ESTM}\left(NBH\left(a_{k,l}^{out(t-1)}\right)\right). \qquad (3.2.16)$$

In what following we provide some examples of typical rank filters

Rank filters for smoothing additive noise and image segmentation:

Rank filtering additive noise can be described by equation:

$$\hat{a}_{k,l}^{(t)} = SMTH\left(NBH\left(\hat{a}_{k,l}^{(t-1)}\right)\right), \qquad (3.2.17)$$

where ***SMTH*** is one of smoothing operations such as ***MEAN***, ***MED***, ***ROS***, ***MODE*** or ***RAND***. (an example of using ***RAND*** operation one can see in Fig. 3.2.3). Figs. 3.2.4 and 3.2.5 illustrate efficiency and edge preserving capability of rank filters in suppressing additive noise in images.

Image segmentation can usually be interpreted as generating from input image its piece wise constant model. In image segmentation, one can regard image tiny and low contrast details that have to be eliminated in the process of segmentation as "noise". Noise smoothing rank filters described by Eq. (3.2.16) are very well suited for such a processing. One illustrative example of segmentation of MRI image is shown in Fig. 3.2.6.

Rank filters for smoothing impulse noise

Filtering impulse noise assumes two stages: detection of pixels replaced by noise and estimation of values of the detected corrupted pixels. For the detection

$$\hat{a}_{k,l}^{(t)} = MEMB\left(NBH_{det}, \hat{a}_{k,l}^{(t-1)}\right) \times \hat{a}_{k,l}^{(t-1)} + \left(1 - MEMB\left(NBH_{det}, \hat{a}_{k,l}^{(t-1)}\right)\right) \times SMTH\left(NBH_{est}\right), \tag{3.2.18}$$

where NBH_{det} and NBH_{det} are neighborhoods used for detection and estimation of corrupted pixels, performed by $MEMB(.)$ and $SMTH(.)$ respectively. Fig. 3.2.7 illustrates filtering impulse noise using one of the simplest version of filters (3.2.18):

$$\hat{a}_{k,l}^{(t)} = MEMB\left(Qnbh_{det}, \hat{a}_{k,l}^{(t-1)}\right) \times \hat{a}_{k,l}^{(t-1)} + \left(1 - MEMB\left(Qnbh_{det}, \hat{a}_{k,l}^{(t-1)}\right)\right) \times MEDN\left(NBH_{est}\right) \tag{3.2.19}$$

where $Qnbh_{det}$ is quantile-neighborhood of the window of 3×3 pixels with $R_{left} = 1; R_{right} = 8$.

Rank order filters for image enhancement

Enhancement of monochrome images is most frequently associated with different methods of amplification of local contrasts and edge extraction. Three families of rank filters can be used for such local contrast enhancement: ***unsharp masking filters***, ***histogram modification filters*** and ***range-filters***.

Unsharp masking filters work according to equation:

$$\hat{a}_{k,l}^{out} = G\left(a_{k,l} - SMTH\left(NBH\left(\hat{a}_{k,l}^{inp}\right)\right)\right), \tag{3.2.20}$$

where G is a contrast enhancement coefficient.

Histogram modification algorithms are described by equation:

$$\hat{a}_{k,l}^{out} = P\_RANK\left(NBH\left(\hat{a}_{k,l}^{inp}\right)\right). \tag{3.2.21}$$

A version of this filter for histogram nonlinearity index in Eq. (3.2.15) $P = 1$ and neighborhood formed by all pixels in the filter window is known as ***local histogram equalization***. Processing with $P \neq 1$ we cal p-histogram equalization. An example of local contrast enhancement with local p-histogram equalization is shown in Fig. 3.2.8.

Range-filters can be regarded as a generalization of unsharp masking filters. They amplify difference between pixel values and local mean inversely proportionally to image local range that is evaluated by rank algorithms:

$$\hat{a}_{k,l}^{out} = \frac{\left(a_{k,l}^{inp} - MEAN\left(NBH\left(a_{k,l}^{inp}\right)\right)\right)}{SPRD\left(NBH\left(a_{k,l}^{inp}\right)\right)} \qquad (3.2.22)$$

Edge extraction is a processing very akin to local contrast enhancement. Unsharp masking carried out in sliding window of small size (3×3 to 5×5) can be, for instance, used for this purpose. Rank algorithms for evaluating image local range represent another and very efficient option. Fig. 3.2.8 c) illustrates edge extraction by one of the most simple algorithms of this sort:

$$\hat{a}_{k,l}^{out} = MAX\left(3 \times 3SHnbh\right) - MIN\left(3 \times 3SHnbh\right), \qquad (3.2.23)$$

where $3 \times 3SHnbh$ is sliding window of 3×3 pixels.

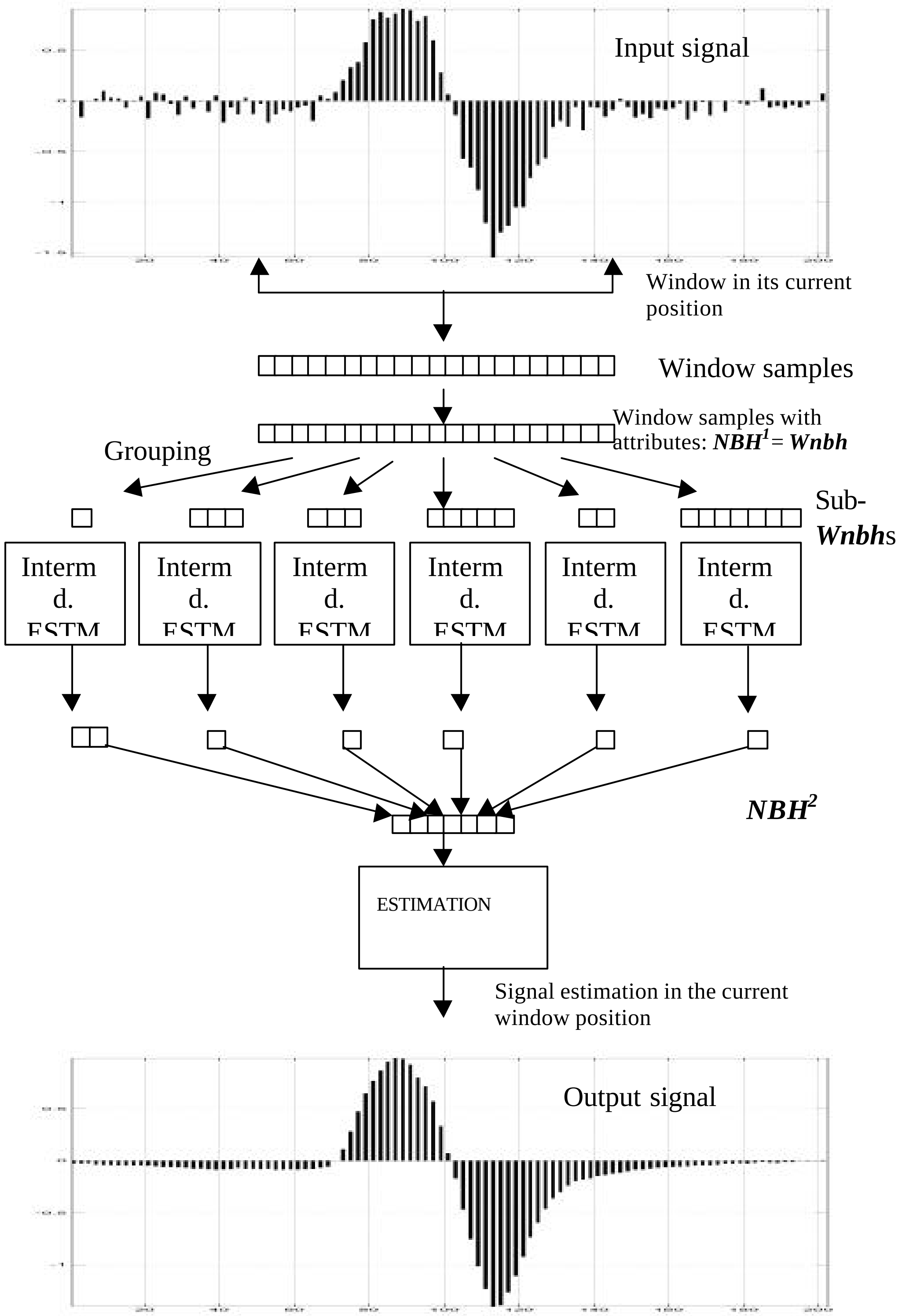


Fig. 3.2.1 Schematic flow diagram of nonlinear filters with two stages of neighborhood building.

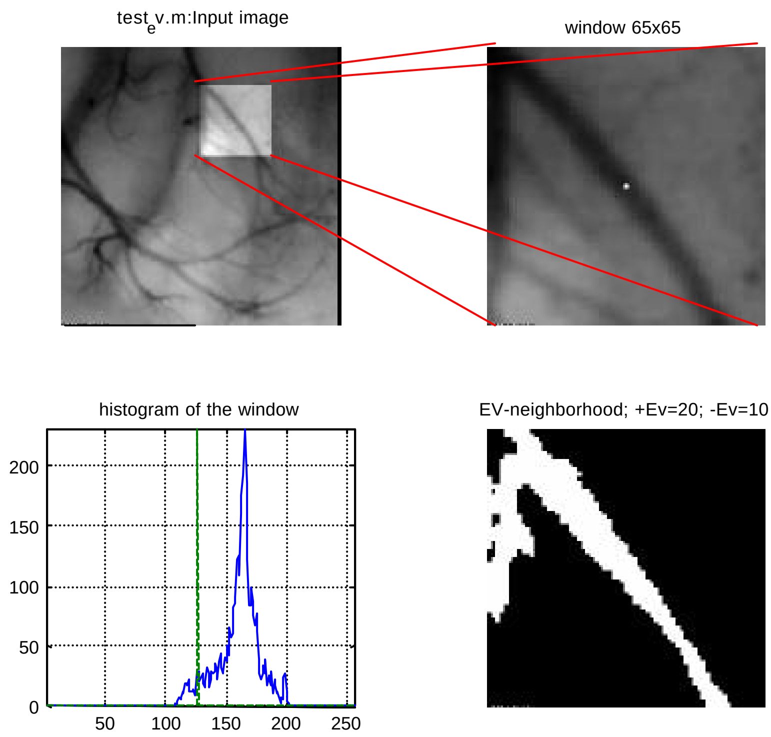


Fig. 3.2.2 a)

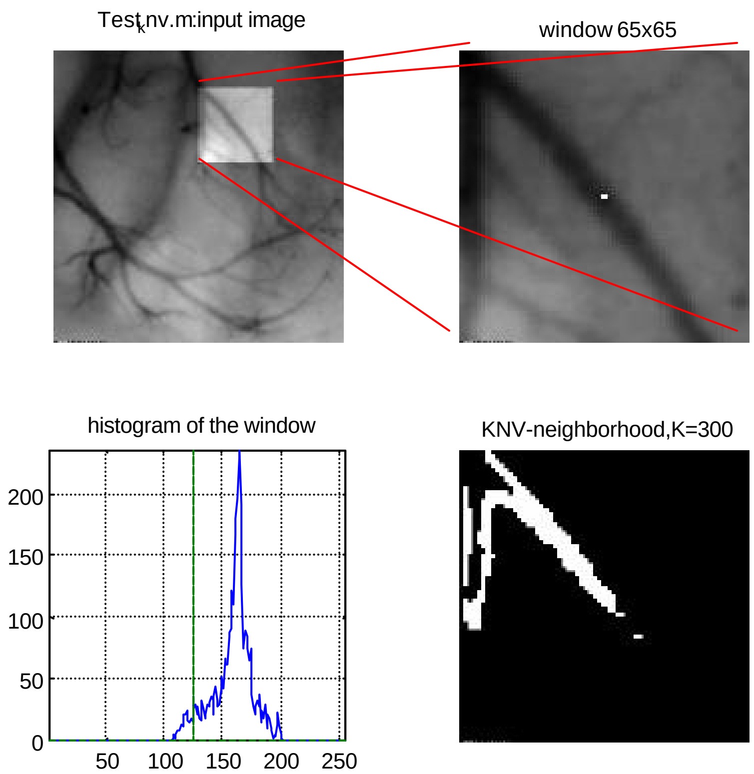


Fig. 3.2.2 b)

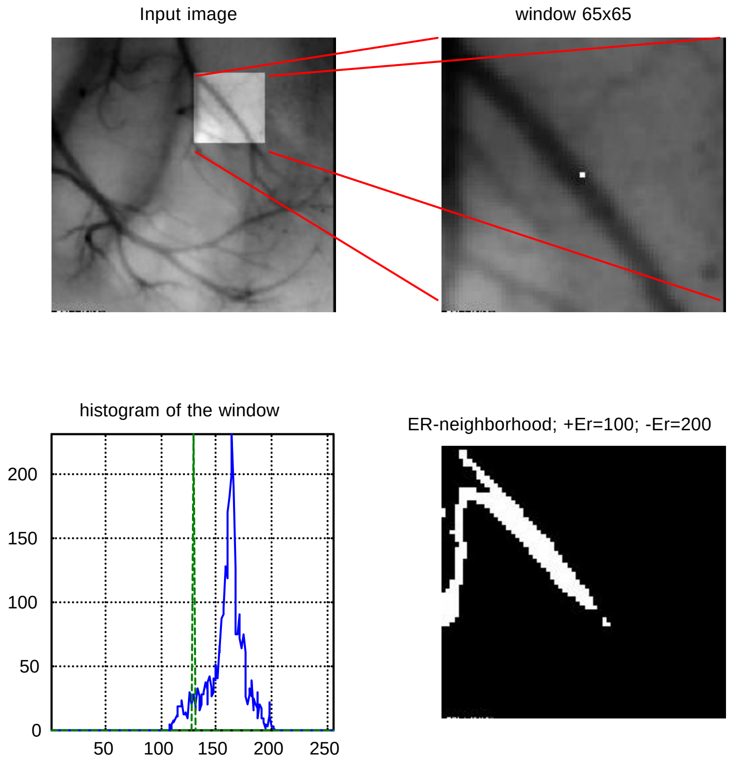


Fig. 3.2.2 c)

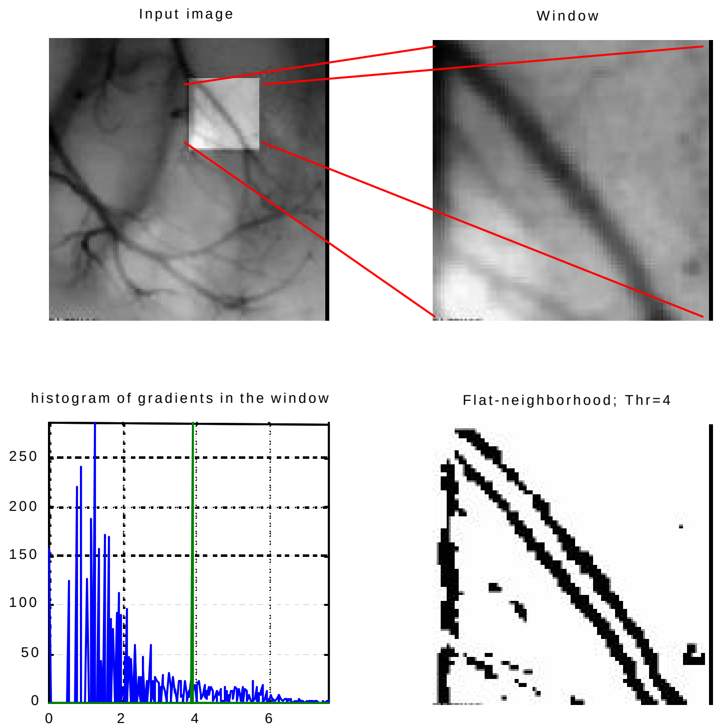


Fig 3.2.2 d)

Fig.3.2.2 ***EV*** (a), ***KNV***(b), ***ER*** (c) and ***Flat*** (d) neighborhoods (from left to right from top to bottom): input image with window highlighted, window magnified with central pixel highlighted, gray level histogram over the window with central pixel gray level shown by a vertical bar and neighborhood segmented (white)

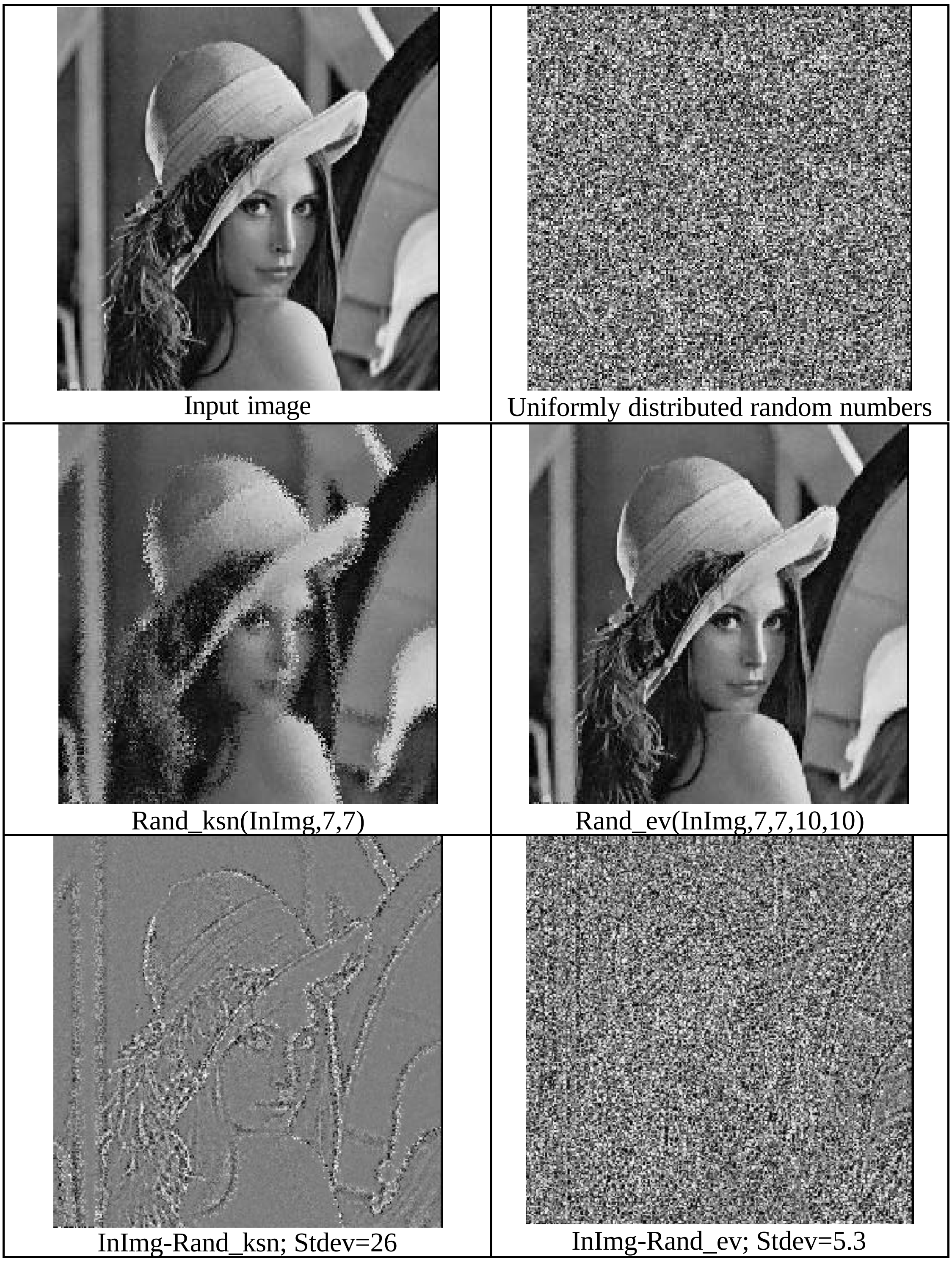


Fig. 3.2.3 Representation of spatial relations in images through local histograms: input image (top left), pattern of uniformly distributed random numbers (top right), image generated from the random pattern keeping local histograms in the window of 7x7 pixels same as those of input image (middle left), image generated from the random pattern keeping local histograms over ***EV***-neighborhood ($\boldsymbol{e_v^+ = e_v^- = 10}$) in the window of 7x7 pixels same as those of input image (middle left) and corresponding differences with the input image (bottom left and right)

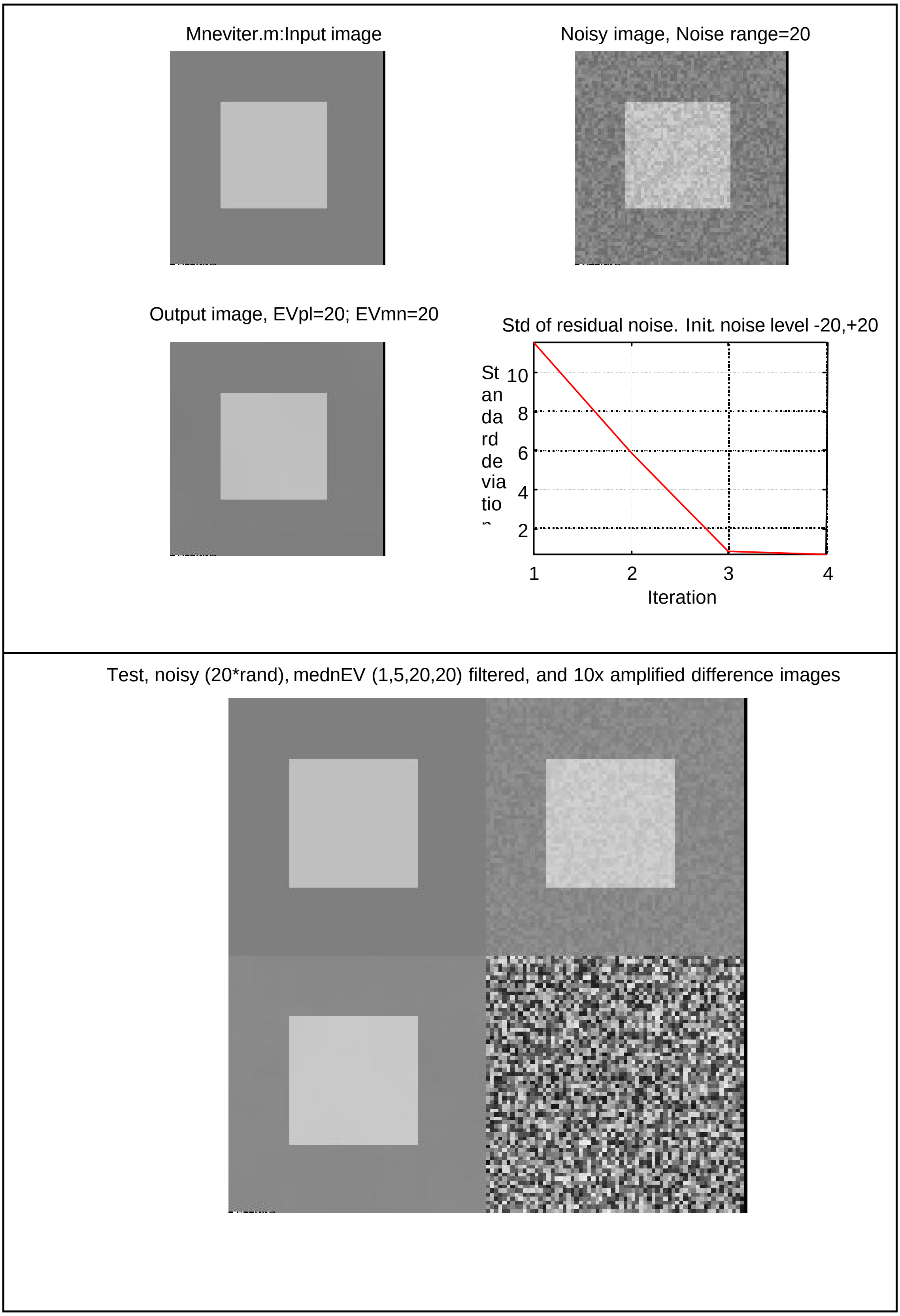
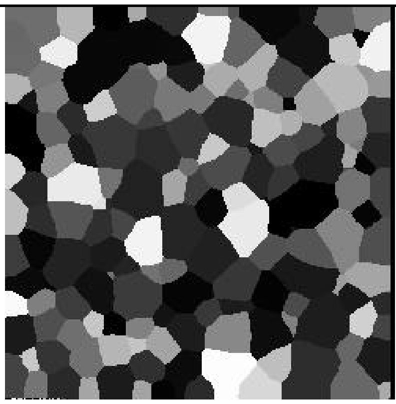


Fig. 3.2.4 Iterative denoising of a piece wise constant image with rank filter **MEDN(*EVnbh*)**. Upper half of the figure shows that noise is almost completely removed in 3 iterations; bottom half demonstrates that difference (bottom right) between noisy (upper right) and filtered (bottom left) images contains only noise that has been removed and does not contain any image details.



Input image

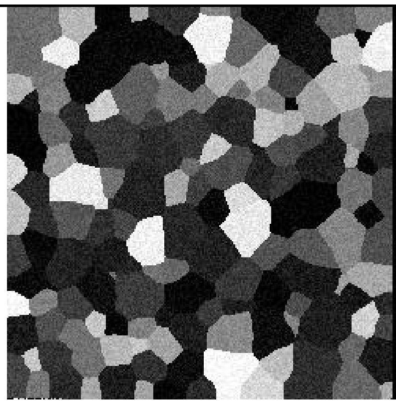
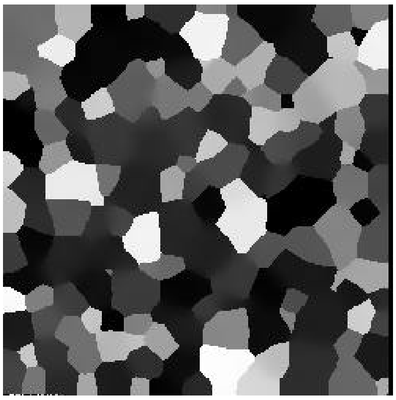
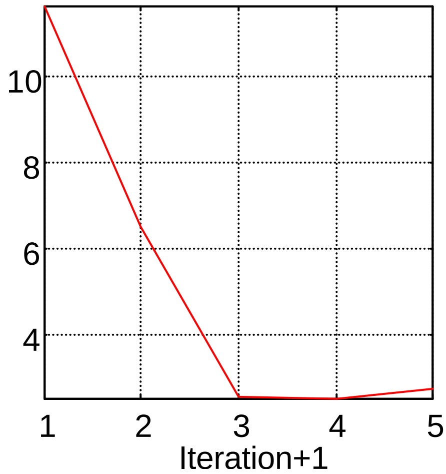


Image corupted by additive noise with the range ± 20 gray levels

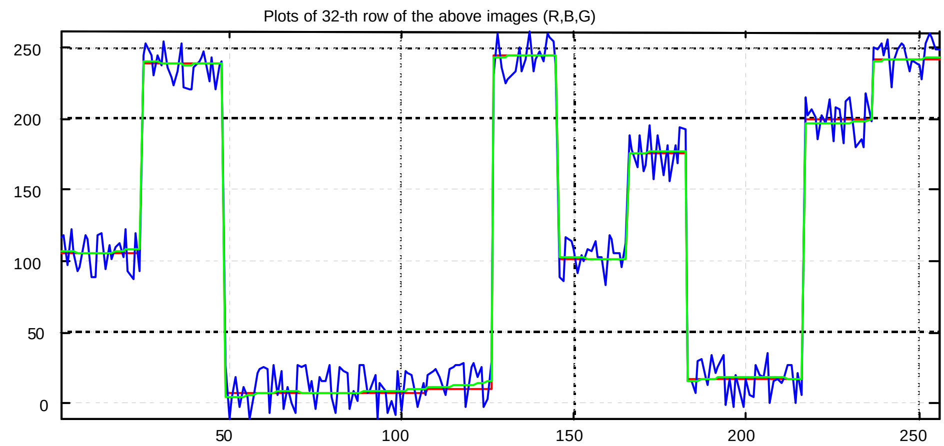


Filtered image ($\boldsymbol{e}_v^{+} = \boldsymbol{e}_v^{-} = 20$)



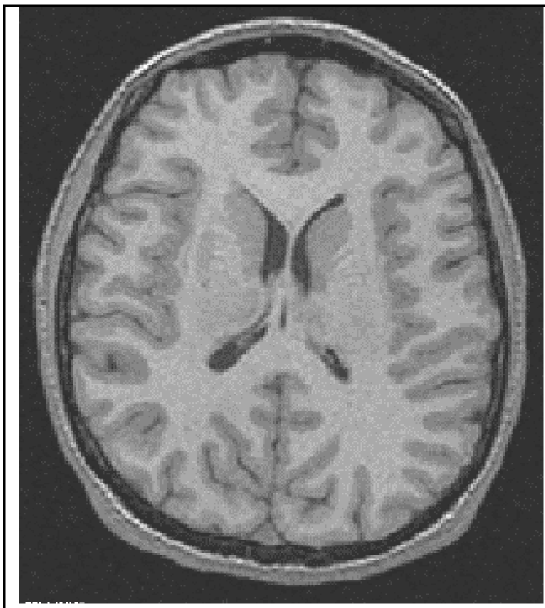
Standard deviation of residual noise in course of iterations

a)

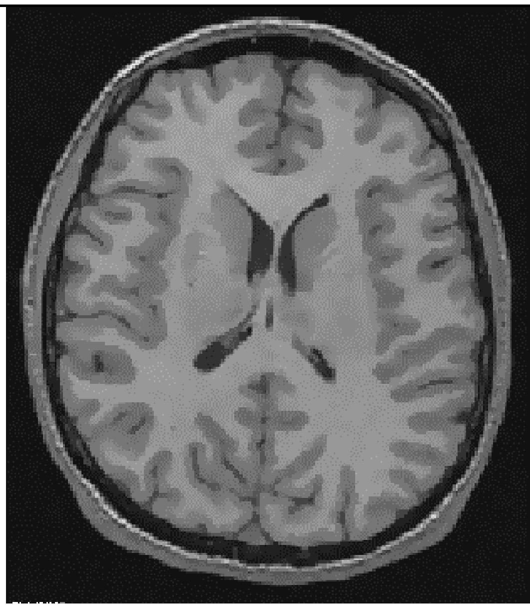


b)

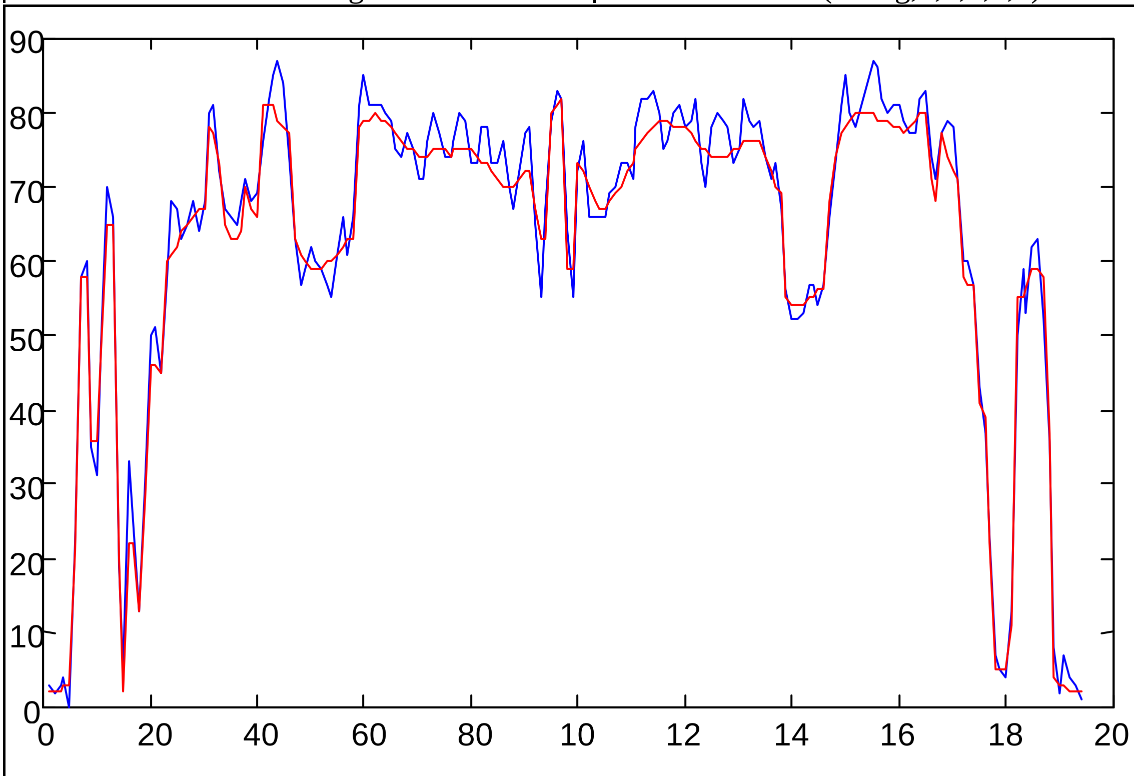
Fig. 3.2.5 Iterative denoising of a piece wise constant test image with rank filter ***MEAN(EVnbh)*** (“sigma-filter”) : a) initial, noisy and filtered images and the plot of standard deviation of the residual noise as a function of iteration number; b) – a plot of an image row (initial image – green, noisy image – blue, filtered image - red). One can see from this plot how successfully the filter removes noise while preserving image edges.



Initial image

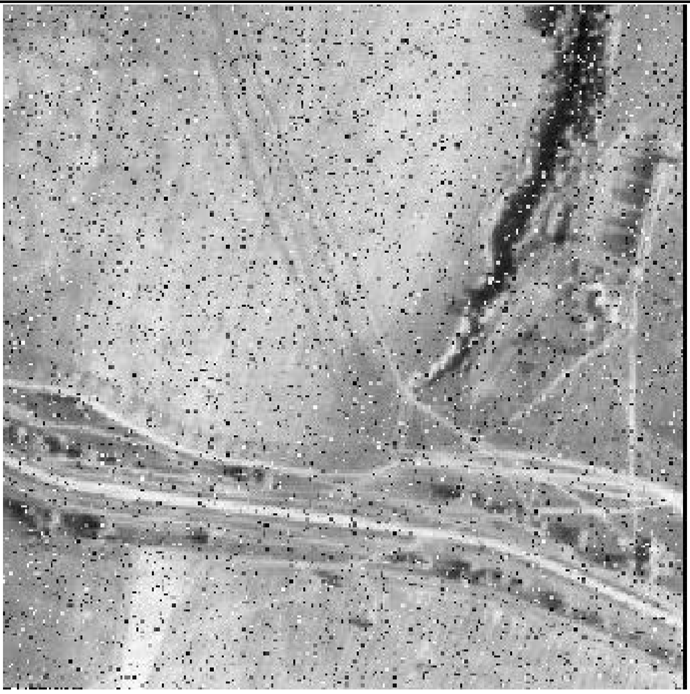


itermnev(InImg,3,3,7,7,3)



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

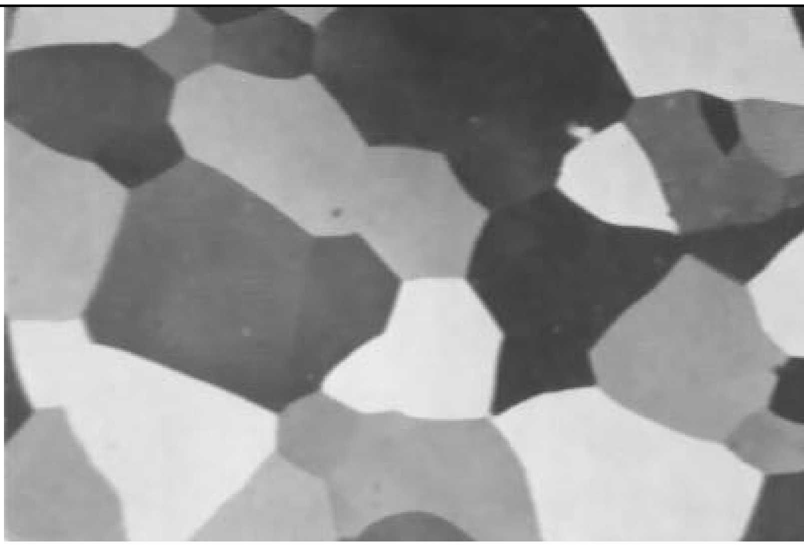
Fig. 3.2.6 Iterative rank filtering for image segmentation with (filter ***MEAN***(***EVnbh***)): initial image (top left), segmented image (top right) and image row (blue: initial image, red – filtered image)



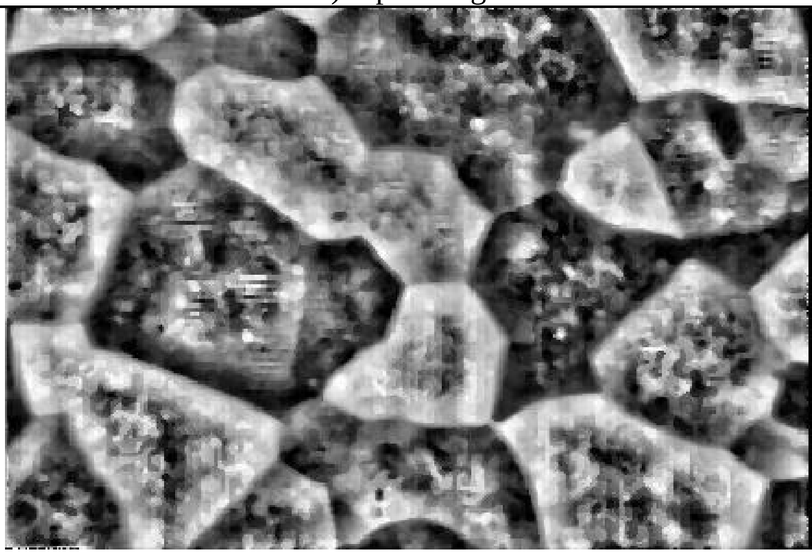
Input image, Perr=0.2

Filtered image

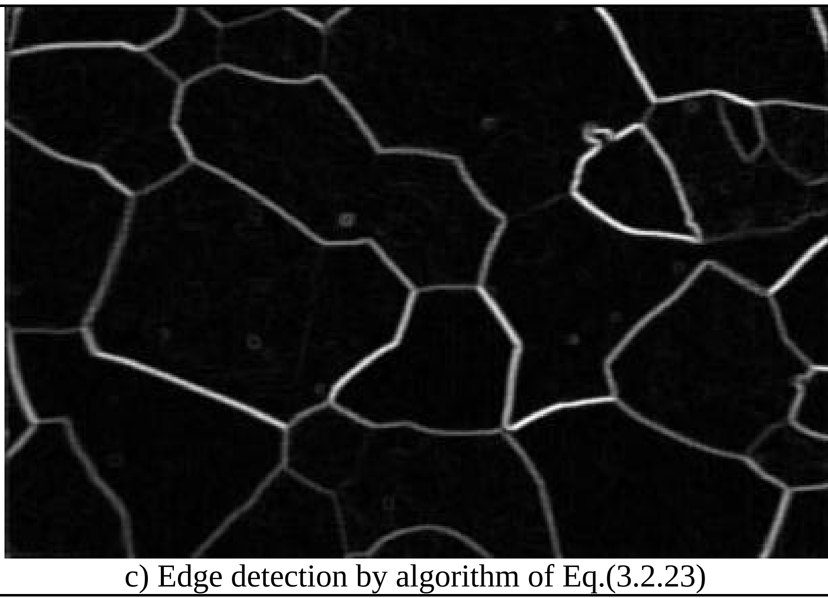
Fig. 3.2.7 Filtering impulse noise with iterative algorithm 3.2.19.



a) Input image



b) Enhancement of image local contrasts by filter of Eq. 3.2.21 in the window $\mathbf{31 \times 31}$ and $\boldsymbol{P = 0.5}$



c) Edge detection by algorithm of Eq.(3.2.23)

Fig. 3.2.8 Image local contrast enhancement and edge extraction

References

1. L. Yaroslavsky, Nonlinear signal processing filters: a unification approach, EUSIPCO2000, Tampere, Finland, Sept. 5-8, 2000
2. L. Yaroslavsky, Nonlinear Signal Processing Filters: A Unification Approach. Manuscript: http://www.eng.tau.ac.il/~yaro/recent
3. L. Yaroslavsky, M. Eden, Fundamentals of Digital Optics, Birkhauser, Boston, 1996
4. V. Kim, L.P. Yaroslavsky, Rank Algorithms for Picture Processing, Computer Vision, Graphics and Image Processing, v. 35, 1986, p. 234-258
5. J. Serra, Image analysis and mathematical morphology, Academic Press, London, 1982
6. S.R. Sternberg, "Grayscale morphology", Computer Vision, Graphics and Image Processing, v. 35, pp.333-355, 1986
7. Astola, P. Kuosmanen, Fundamentals of Nonlinear Digital Filtrering, CRC Press, Roca Baton, New York, 1997
8. I. Pitas, A. N. Venetsanopoulos, Nonlinear Digital Filters. Principles and Applications. Kluwer, 1990