Systematic approach to nonlinear filtering associated with aggregation operators. Part 1. SISO-filters

V. Labunets[a], E. Osthaimer[b]

[a] Ural State Forest Engineering University, 620100, Ekaterinburg, Russia
[b] Capricat LLC, Pompano Beach, Florida, USA

Abstract

There are various methods to help restore an image from noisy distortions. Each technique has its advantages and disadvantages. Selecting the appropriate method plays a major role in getting the desired image. Noise removal or noise reduction can be done on an image by linear or nonlinear filtering. The more popular linear technique is based on average (on mean) linear operators. Denoising via linear filters normally does not perform satisfactorily since both noise and edges contain high frequencies. Therefore, any practical denoising model has to be nonlinear. In this work, we introduce and analyze a new class of nonlinear SISO-filters that have their roots in aggregation operator theory. We show that a large body of non-linear filters proposed to date constitute a proper subset of aggregation filters.

Keywords: nonlinear filtering; multicolor images; aggregation operators; nonlinear SISO-filters

1. Introduction

The basic idea behind this paper is the estimation of the uncorrupted image from the distorted or noisy image, and is also referred to as image "denoising". To denoise images is to filter out the noise. The challenge is to preserve and enhance important features during the denoising process. For images, for example, an edge is one of the most universal and crucial features. There are various methods to help restore an image from noisy distortions [1,2]. Each technique has its advantages and disadvantages. Selecting the appropriate method plays a major role in getting the desired image. Noise removal or noise reduction can be done on an image by linear or nonlinear filtering. The more popular linear technique is based on average (on mean) linear operators. Denoising via linear filters normally does not perform satisfactorily since both noise and edges contain high frequencies. Therefore, any practical denoising model has to be nonlinear. In this paper, we propose a new type of nonlinear data-dependent denoising filter called the *aggregation digital filter (ADF)*.

2. The object of the study

Let us introduce the observation model and notion used throughout the paper. We consider noise signals or images of the form $\mathbf{f}(\mathbf{x})=\mathbf{s}(\mathbf{x})+\boldsymbol{\eta}(\mathbf{x})$, where $\mathbf{s}(\mathbf{x})=\left(s_1(\mathbf{x}),s_2(\mathbf{x}),...,s_K(\mathbf{x})\right)$ is the original multichannel signal, $\boldsymbol{\eta}(\mathbf{x})=\left(\eta_1(\mathbf{x}),\eta_2(\mathbf{x}),...,\eta_K(\mathbf{x})\right)$ denotes the multichannel noise introduced into the signal $\mathbf{s}(\mathbf{x})$ to produce the corrupted signal $\mathbf{f}(\mathbf{x})=\left(f_1(\mathbf{x}),f_2(\mathbf{x}),\dots,f_K(\mathbf{x})\right)$. Here $\mathbf{x}=i\in\mathbf{Z}$, $\mathbf{x}=(i,j)\in\mathbf{Z}^2$, or $\mathbf{x}=(i,j,k)\in\mathbf{Z}^3$ are a 1D, 2D, or 3D coordinates, respectively, that belong to the signal (image) domain and represent the pixel location. If $\mathbf{x}\in\mathbf{Z},\mathbf{Z}^2,\mathbf{Z}^3$ then $\mathbf{f}(\mathbf{x}),\mathbf{s}(\mathbf{x})$, $\boldsymbol{\eta}(\mathbf{x})$ are 1D, 2D and 3D multichannel signals, respectively. The aim of image enhancement is to reduce the noise as much as possible or to find a method which, given $\mathbf{s}(\mathbf{x})$, derives an image $\mathbf{y}(\mathbf{x})=\hat{\mathbf{s}}(\mathbf{x})$ as close as possible to the original $\mathbf{s}(\mathbf{x})$, subject to a suitable optimality criterion [1].

In 2D standard linear and median filters with a square window $\left[\mathrm{M}_{(i,j)}(m,n)\right]_{m=-s,n=-s}^{m=+s,n=+s}$ of size $(2s+1)\times(2s+1)$ is located at (i,j) the arithmetic mean and median replace the central pixel

$$\hat{\mathbf{s}}(i,j)=\underset{(m,n)\in M_{(i,j)}}{\mathbf{Arithm}}\left\{\mathbf{f}(m,n)\right\},\quad \hat{\mathbf{s}}(i,j)=\underset{(m,n)\in M_{(i,j)}}{\mathbf{Median}}\left\{\mathbf{f}(m,n)\right\},$$

where $\hat{\mathbf{s}}(i,j)$ is the filtered image, $\left\{\mathbf{f}(m,n)\right\}_{(m,n)\in M_{(i,j)}}$ is image block of the fixed size $(2s+1)\times(2s+1)$ extracted from $\vec{\mathbf{f}}$ by moving window $\mathrm{M}_{(i,j)}$ at the position (i,j). Symbols **Arithm** and **Median** are the arithmetic mean (average) and median operators, respectively. When those filters are modified as follows

$$\hat{\mathbf{s}}(i,j)=\underset{(k,l)\in M(i,j)}{\mathbf{Aggreg}}\left\{\mathbf{f}(k,l)\right\},\tag{1}$$

it becomes an aggregation digital filter, where **Aggreg** is a *generalized average* or an *aggregation operator* [3].

In the first part, we are going consider a general theory of nonlinear SISO-filters (single-input single-output) associated with aggregation operators of averaging types. In the next parts, we will consider a general theory of nonlinear filters associated with aggregation operators of conjunctive and disjunctive types. We show that a large body of non-linear filters proposed to date constitute a proper subset of aggregation digital filters.

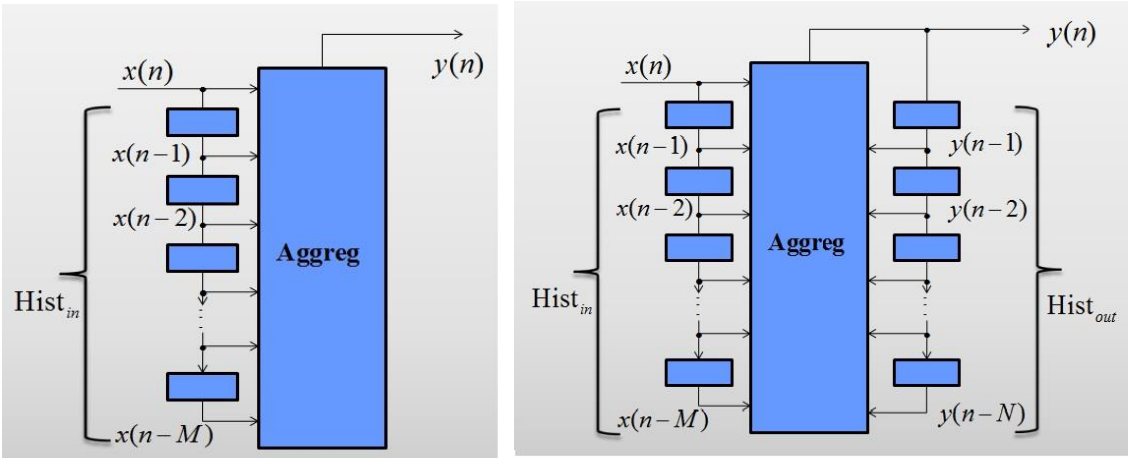


Fig. 1. Discrete non-recursive and recursive r filters.

3. Methods

3.1. Filters as discrete dynamic systems

A discrete-time system (DTS) is a device or algorithm that, according to some well-defined input/output rule, operates on a discrete-time signal called the input signal $x(n)$ or excitation to produce another discrete-time signal called the output signal or response $y(n)$. For a DTS the output $y(n)$ theoretically can depends on all earlier input values $\{x(m)\}_{m\leq n}$. DTS must memorizes these values. It requests infinite volume of memory. In real, discrete-time systems have finite memory and for this reason can memorize only a finite set of earlier input values $\{x(n-1),x(n-2),...,x(n-N+1)\}$. This set of earlier input values is called the prehistory of the input sample $x(n)$ and denotes as $\mathrm{Hist}_{in}(n,N-1):=\{x(n-1),x(n-2),...,x(n-N+1)\}$. Hence, for a DTS the output $y(n)$ can depends on only a finite set of earlier input values. Mathematically speaking, a system is also a function of N variables. The input signal $x(n)$ is transformed by the system into a signal $y(n)$, which we express mathematically as

$$y(n)=\mathbf{Aggreg}\{x(n),x(n-1),...,x(n-N+1)\}=\mathbf{Aggreg}\{x(n),\mathrm{Hist}_{in}(n;N-1)\}$$

where $\mathbf{Aggreg}\{.\}$ is some well-defined transformation input/output rule (a function of N variables) of input samples into an output sample $y(n)$ at the discrete moment n. Block diagram representation of similar discrete-time system is illustrated in Fig.1a. It is called a non-recursive filter (NRF).

More "clever" system have to analyzes to self-behavior and memorizes a finite set of earlier output values $\{y(n-1),y(n-2),...,y(n-M)\}$. This set is called the prehistory of the output sample $y(n)$ and denotes as $\mathrm{Hist}_{out}(n,M):=\{y(n-1),y(n-2),...,y(n-M)\}$. In this case DTS analyzes both a input and output prehistories and after that the input signal $f(n)$ is transformed by the system into a signal $y(n)$, which we express mathematically as

$$y(n)=\mathbf{Aggreg}\{x(n),x(n-1),...,x(n-N+1);y(n-1),...,y(n-M)\}=\mathbf{Aggreg}\{x(n),\mathrm{Hist}_{in}(n;N-1),\mathrm{Hist}_{out}(n,M)\},$$

where $\mathbf{Aggreg}\{.\}$ is some well-defined transformation input/output rule (a function of $N+M$ variables) of input samples into an output sample $y(n)$ at the discrete moment n. Block diagram representation of similar discrete-time system is illustrated in Fig.1b. It is called the recursive filter (RF).

If если $\mathbf{Aggreg}\{.\}$ is a linear function, then NRF is an infinite-impulse response filter (IIF). In this case, we have

$$\begin{aligned} y(n) &= \alpha\cdot\sum_{k=0}^{N-1} w_k x(n-k)+\beta\cdot\sum_{k=1}^{M} v_k y(n-k)= \\ &= \alpha\cdot\mathbf{WArithm}\{x(n),...,x(n-N+1)\}+\beta\cdot\mathbf{WArithm}\{y(n-1),...,y(n-M)\}, \end{aligned}$$

where $w_0,w_1,...,w_N$ и $v_1,...,v_M$ are weights, $\alpha=\sum_{k=0}^{N-1} w_k\left(\sum_{k=0}^{N-1} w_k+\sum_{k=1}^{M} v_k\right)^{-1}$, $\beta=\sum_{k=1}^{M} v_k\left(\sum_{k=0}^{N-1} w_k+\sum_{k=1}^{M} v_k\right)^{-1}$ and

$$\mathbf{WArithm}\{x(n),...,x(n-N+1)\}=\sum_{k=0}^{N-1}w_k x(n-k),\quad \mathbf{WArithm}\{y(n-1),...,y(n-M)\}=\sum_{k=1}^{M}v_k y(n-k).$$

3.2. Aggregation operators

The aggregation problem consist in aggregating N-tuples of objects $(x_1,x_2,...,x_N)$ all belonging to a given set D, into a single object of the same set D, *i.e.*, $\mathbf{Aggreg}: D^N\to D$. In fuzzy logic theory, the set D is an interval of the real $D=[0,1]\subset\mathbf{R}$. In image processing theory $D=[0,255]\subset\mathbf{Z}$. In this setting, an aggregation operator is simply a function, which assigns a number y to any N-tuple $(x_1,x_2,...,x_N)$ of numbers that satisfies [3]:

1) $\mathbf{Aggreg}_N(x_1,x_2,...,x_N)$ is continuous and monotone in each variable; to be definite, we assume that **Aggreg** is increasing in each variable.
2) The aggregation of identical numbers is equal to their common value: $\mathbf{Aggreg}_N(x,x,...,x)=x$.
3) $\mathbf{Min}(x_1,...,x_N)\le\mathbf{Aggreg}(x_1,...,x_N)\le\mathbf{Max}(x_1,...,x_N)$. Here $\mathbf{Min}(x_1,x_2,...,x_N)$ and $\mathbf{Max}(x_1,x_2,...,x_N)$ are the *minimum* and the *maximum* values among the elements of $(x_1,x_2,...,x_N)$.
4) $\mathbf{Aggreg}(x_1,x_2,...,x_N)$ is a symmetric function: $\mathbf{Aggreg}(x_{\sigma(1)},x_{\sigma(2)},...,x_{\sigma(N)})=\mathbf{Aggreg}(x_1,x_2,...,x_N)$, $\forall\sigma\in\mathbf{S}_\mathrm{N}$ of $\{1,2,...,N\}$, where $\mathbf{S}_N$ is the set of all permutations of $1,2,...,N$. In this case $\mathbf{Aggreg}(x_1,...,x_N)$ is invariant (symmetric) with respect to the permutations of the elements of $(x_1,x_2,...,x_N)$. In other words, as far as means are concerned, the *order* of the elements of $(x_1,x_2,...,x_N)$ is - and must be - completely irrelevant.

Proposition 1. (Kolmogorov [4]). If conditions 1)–4) are satisfied, the aggregation $\mathbf{Aggreg}(x_1,x_2,...,x_N)$ of the average type are as of the forms:

$$\begin{aligned}\mathbf{Kolm}(K\mid x_1,x_2,...,x_N)&=K^{-1}\left[\frac{1}{N}\sum_{i=1}^{N}K(x_i)\right]=K^{-1}\left[\mathbf{Arithm}\{K(x_1),K(x_2),...,K(x_N)\}\right],\\ \mathbf{WKolm}(K\mid x_1,x_2,...,x_N)&=K^{-1}\left[\sum_{i=1}^{N}\bar{w}_i K(x_i)\right]=K^{-1}\left[\mathbf{WArithm}\{K(x_1),K(x_2),...,K(x_N)\}\right],\end{aligned}\tag{2}$$

or

$$\begin{aligned}\mathbf{Kolm}(K^{-1}\mid x_1,x_2,...,x_N)&=K\left[\frac{1}{N}\sum_{i=1}^{N}K^{-1}(x_i)\right]=K^{-1}\left[\mathbf{Arithm}\{K^{-1}(x_1),K^{-1}(x_2),...,K^{-1}(x_N)\}\right],\\ \mathbf{WKolm}(K^{-1}\mid x_1,x_2,...,x_N)&=K\left[\sum_{i=1}^{N}\bar{w}_i K^{-1}(x_i)\right]=K\left[\mathbf{WArithm}\{K^{-1}(x_1),K^{-1}(x_2),...,K^{-1}(x_N)\}\right],\end{aligned}\tag{3}$$

where K is a strictly monotone continuous function in the extended real line.

In (2) and (3) we can use an arbitrary aggregation operator instead of **Arithm**, that gives new the Kolmogorov aggregation operators

$$\begin{aligned}\mathbf{KolmAgg}(K\mid x_1,x_2,...,x_N)&=K^{-1}\left[\mathbf{Aggreg}\{K(x_1),K(x_2),...,K(x_N)\}\right],\\ \mathbf{KolmAgg}(K^{-1}\mid x_1,x_2,...,x_N)&=K\left[\mathbf{Aggreg}\{K^{-1}(x_1),K^{-1}(x_2),...,K^{-1}(x_N)\}\right].\end{aligned}\tag{4}$$

We list below a few particular cases of aggregation means:

1) Arithmetic and weighted means ($K(x)=x$):

$$\mathbf{Arithm}(x_1,x_2,...,x_N)=N^{-1}\sum_{i=1}^{N}x_i,\quad \mathbf{WArithm}(x_1,x_2,...,x_N)=\sum_{i=1}^{N}\bar{w}_i x_i,\tag{5}$$

where $\sum_{i=1}^{N}\bar{w}_i=1$. Classical operator $\mathbf{Arithm}(x_1,x_2,...,x_N)$ is interesting because it gives an aggregated value that is smaller than the greatest argument and bigger than the smallest one. Therefore, the resulting aggregation is "a middle value". This property is known as the compensation property that is described mathematically by: $\mathbf{min}(x_1,x_2,...,x_N)\le\mathbf{Arith}(x_1,x_2,...,x_N)\le$ $\le\mathbf{max}(x_1,x_2,...,x_N)$, where $\mathbf{min}(x_1,x_2,...,x_N)$) and $\mathbf{max}(x_1,x_2,...,x_N)$ are the algebraic minimum and maximum functions, respectively. The mappings **min** and **max** both satisfy the defining conditions and therefore are aggregations (means), even though they are rarely mentioned - or even perceived - as such. It is often used since it is simple and satisfies the properties of monotonicity, continuity, symmetry, associativity, idempotence and stability for linear transformations. However, it has neither absorbent nor neutral element and has no behavioral properties.

2) Another operator that follows the idea obtaining "a middle value" is the median. It consists in ordering the arguments from the smallest one to the biggest one

$$\left(x_1, x_2, ..., x_{2m-1}, x_{2m}, x_{2m+1}, ..., x_N\right) \to \left(x_{(1)}, x_{(2)}, ..., x_{(2m-1)}, x_{(2m)}, x_{(2m+1)}, ..., x_{(N)}\right), \quad N = 2m+1$$

$$\left(x_1, x_2, ..., x_{2m}, x_{2m+1}, ..., x_N\right) \to \left(x_{(1)}, x_{(2)}, ..., x_{(2m)}, x_{(2m+1)}, ..., x_{(N)}\right), \qquad N = 2m$$

where taking the element in the middle:

$$\mathbf{Med}\left(x_1, x_2, ..., x_N\right) = \begin{cases} x_{(2m)}, & N = 2m+1, \\ 0,5\left(x_{(2m)} + x_{(2m+1)}\right), & N = 2m. \end{cases} \tag{6}$$

This aggregation operator satisfies the boundary conditions, the monotonicity, the symmetry, the idempotence and evidently the compensation behavior.

3) There exists a generalization of this operator: the k-order statistic, with which we can choose the element on the k th position on the ordered list (from the smallest to the biggest element):

$$\mathbf{OS}_k\left(x_1, x_2, ..., x_k, ..., x_N\right) = \mathbf{OS}_k\left(x_{(1)}, x_{(2)}, ..., x_{(k)}, ..., x_{(N)}\right) = x_{(k)}. \tag{7}$$

4) Two remarkable particular cases of the k-order statistic are the minimum and the maximum:

$$\mathbf{min}\left(x_1, x_2, ..., x_N\right) = x_{(1)}, \quad \mathbf{max}\left(x_1, x_2, ..., x_N\right) = x_{(N)}. \tag{8}$$

The minimum gives the smallest value of a set, while the maximum gives the greatest one. They are aggregation operators since they satisfy the axioms of the definition. The main properties of these operators are monotonicity, symmetry, associativity, idempotence. Mathematically speaking they have a compensation behavior, but these are the limit cases. Using these operators, we will never obtain an aggregated value "in the middle". For this reason, we do not consider that we can talk about compensation behavior in this case.

5) Geometric and weighted geometric means ($K(x) = \ln(x)$):

$$\mathbf{Geo}\left(x_1, x_2, ..., x_N\right) = \sqrt[N]{\prod_{i=1}^{N} x_i} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \ln x_i\right), \quad \mathbf{WGeo}(x_1, x_2, ..., x_N) = \left(\prod_{j=1}^{N} x_j^{w_j}\right)^{1/\sum_{i=1}^{N} w_i}. \tag{9}$$

6) Harmonic and weighted harmonic means ($K(x) = x^{-1}$):

$$\mathbf{Harm}(x_1, x_2, ..., x_N) = \left(N^{-1}\sum_{i=1}^{N} x_i^{-1}\right)^{-1} = \frac{N}{\left(\sum_{i=1}^{N} x_i^{-1}\right)}, \quad \mathbf{WHarm}\left(x_1, x_2, ..., x_N\right) = \frac{\sum_{i=1}^{N} w_i}{\sum_{i=1}^{N} w_i \dfrac{1}{x_i}}. \tag{10}$$

7) Very notable particular case corresponds to the function $K(x) = x^p$. We obtain then power mean:

$$\mathbf{Power}_p\left(x_1, x_2, ..., x_N\right) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i^p\right)^{\frac{1}{p}}. \tag{11}$$

In mathematics, the power mean, also known as Hölder mean (named after Otto Holder), is an abstraction of the Pythagorean means including arithmetic, geometric, and harmonic means.

3.3. Ordinary aggregation 2D SISO-filters

The simplest and most common way to aggregate input data in 2D SISO-filter is to use a simple arithmetic and weighted mean:

$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Arithm}}\left\{f(m,n)\right\}, \quad \hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{WArithm}}\left\{\overline{w}(m,n)\, f(m,n)\right\}. \tag{12}$$

Some extensions of the simple arithmetic filters (12) have been introduced as geometric and harmonic filters

$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Geo}} \left\{ f(m,n) \right\} = \left[\prod_{(m,n)\in M_{(i,j)}} f(m,n) \right]^{\frac{1}{N}},$$
$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{WGeo}} \left\{ f(m,n) \right\} = \left[\prod_{(m,n)\in M_{(i,j)}} f^{w(m,n)}(m,n) \right]^{1/\sum\limits_{(m,n)\in M_{(i,j)}} w(m,n)} \quad (13)$$

$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Harm}} \left\{ f(m,n) \right\} = \left[\sum_{(m,n)\in M_{(i,j)}} f^{-1}(m,n) \right]^{-1},$$
$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{WHarm}} \left\{ f(m,n) \right\} = \frac{\sum\limits_{(m,n)\in M_{(i,j)}} w(m,n)}{\sum\limits_{(m,n)\in M_{(i,j)}} w(m,n)\cdot f^{-1}(m,n)}. \quad (14)$$

3.4. Kolmogorov aggregation 2D SISO-filters

Many extensions of the simple ordinary linear filters are defined as Kolmogorov filters

$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Kolm}} \left\{ K \mid f(m,n) \right\} = K^{-1}\left[\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} K\left(f(m,n) \right) \right] = K^{-1}\left[\underset{(m,n)\in M_{(i,j)}}{\mathbf{Arithm}} \left\{ K\left(f(m,n) \right) \right\} \right]. \quad (15)$$

and as dual Kolmogorov filters

$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Kolm}} \left\{ K \mid f(m,n) \right\} = K\left[\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} K^{-1}\left(f(m,n) \right) \right] = K\left[\underset{(m,n)\in M_{(i,j)}}{\mathbf{Arithm}} \left\{ K^{-1}\left(f(m,n) \right) \right\} \right]. \quad (16)$$

If $K(x) = x^p$ and $K^{-1}(x) = \sqrt[p]{x}$ then we have the Hölder (or power) and the dual Hölder filters of the following forms:

$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{p}} \left\{ f(m,n) \right\} = \sqrt[p]{\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} f^{p}(m,n)},$$
$$\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{1/p}} \left\{ f(m,n) \right\} = \left(\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} \sqrt[p]{f(m,n)} \right)^{p}. \quad (17)$$

This family is particularly interesting, because it generalizes a group of common filters, only by changing the value of p:

$$1)\ \hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{1}} \left\{ f(m,n) \right\} = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Arithm}} \left\{ f(m,n) \right\} = \frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} f(m,n),$$
$$2)\ \hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{2}} \left\{ f(m,n) \right\} = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Square}} \left\{ f(m,n) \right\} = \sqrt[2]{\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} f^{2}(m,n)}, \quad (18)$$

$$3)\ \hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{1/2}} \left\{ f(m,n) \right\} = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Root}} \left\{ f(m,n) \right\} = \left(\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} \sqrt{f(m,n)} \right)^{2},$$
$$4)\ \hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{3}} \left\{ f(m,n) \right\} = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Triple}} \left\{ f(m,n) \right\} = \sqrt[3]{\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} f^{3}(m,n)}, \quad (19)$$
$$5)\ \hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{1/3}} \left\{ f(m,n) \right\} = \underset{(m,n)\in M_{(i,j)}}{\mathbf{Triple}^{-1}} \left\{ f(m,n) \right\} = \left(\frac{1}{N} \sum_{(m,n)\in M_{(i,j)}}^{N} \sqrt[3]{f(m,n)} \right)^{3}.$$

In particular, using Hölder filters we can construct new Kolmogorov-Lehmer filters as:

$$\underset{(m,n)\in M_{(i,j)}}{\mathbf{Lehm}^p}\left\{f(m,n)\right\}=\frac{\underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^p}\left\{f(m,n)\right\}}{\underset{(m,n)\in M_{(i,j)}}{\mathbf{Hold}^{p-1}}\left\{f(m,n)\right\}}. \tag{20}$$

The next extensions of the Kolmogorov filters are based on (4):

$$\begin{aligned}\hat{s}(i,j)&=\underset{(m,n)\in M_{(i,j)}}{\mathbf{Kolm}}\left\{K\mid f(m,n)\right\}=K^{-1}\left[\underset{(m,n)\in M_{(i,j)}}{\mathbf{Aggreg}}\left\{K\left(f(m,n)\right)\right\}\right],\\ s(i,j)&=\underset{(m,n)\in M_{(i,j)}}{\mathbf{Kolm}}\left\{K\mid f(m,n)\right\}=K\left[\underset{(m,n)\in M_{(i,j)}}{\mathbf{Aggreg}}\left\{K^{-1}\left(f(m,n)\right)\right\}\right].\end{aligned} \tag{21}$$

3.5. The Heronian aggregation filters

The classical Heronian mean definition of two positive real numbers a and b are

$$\mathbf{ArithHeron}(a,b)=\left(a+\sqrt{ab}+b\right)/3=\left(\sqrt{aa}+\sqrt{ab}+\sqrt{bb}\right)/3. \tag{22}$$

Hero of Alexandria is the Greek mathematician [5]. Along with the Heronian mean, we introduce the Heronian median as follows

$$\mathbf{MedHeron}(a,b)=\mathbf{Med}\left\{\sqrt{aa},\sqrt{ab},\sqrt{bb}\right\}. \tag{23}$$

Let $(x_1,x_2,...,x_N)$ be an N-tuple of positive real numbers. An obvious way to generalize Eqs. (22)-(23) is by including inside the parentheses the square roots of all possible products of two elements.

Definition 1. The 2-generalized Heronian mean and median of an N-tuple of positive real numbers $(x_1,x_2,...,x_N)$ are defined as

$$\begin{aligned}\mathbf{MeanHeron}_2(x_1,x_2,...,x_N)&=\frac{2}{N(N+1)}\sum_{i}\underset{\le}{}\sum_{j}\sqrt{x_ix_j},\\ \mathbf{MedHeron}_2(x_1,x_2,...,x_N)&=\mathbf{Med}\left[\left\{\sqrt{x_ix_j}\right\}_{i\le j}\right].\end{aligned} \tag{24}$$

Now we can generalize this definition using k*-th* roots of all possible distinct products of k elements of $(x_1,x_2,...,x_N)$, again with repetition. The number of all such products corresponds to extracting k elements from a bag of N, with replacement, where C^k_{N+k-1} is the binomial coefficient. This determines the normalization factor.

Definition 2. The Heronian k-mean and k-median of an N-tuple of positive real numbers $(x_1,x_2,...,x_N)$ are defined as

$$\mathbf{ArithHeron}_k(x_1,x_2,...,x_N)=\frac{2}{C^k_{N+k-1}}\sum_{r_1}\le\sum_{r_2}\le...\le\sum_{r_k}\sqrt[k]{x_{r_1}x_{r_2}\cdots x_{r_k}}. \tag{25}$$

$$\mathbf{MedHeron}_k(x_1,x_2,...,x_N)=\mathbf{Med}\left[\left\{\sqrt[k]{x_{r_1}x_{r_2}\cdots x_{r_k}}\right\}_{r_1\le r_2\le...\le r_k}\right]. \tag{26}$$

Obviously, $C^k_{N+k-1}=\mathbf{ArithHeron}_k(1,1,...,1)$.

As we see, two types of aggregation operators ($\mathbf{Arith}$ and $\mathbf{Med}$) are used in (25) and (26). We can use an arbitrary aggregation operator

$$\mathbf{AggHeron}_k(x_1,x_2,...,x_N)=\underset{r_1\le r_2\le...\le r_k}{\mathbf{Aggreg}}\left[\left\{\sqrt[k]{x_{r_1}x_{r_2}\cdots x_{r_k}}\right\}\right] \tag{27}$$

that gives a wide family of Heronian filters. Indeed, in the standard linear and nonlinear 2D-filters the square window $\left[\mathrm{M}_{(i,j)}(m,n)\right]_{m=-s,n=-s}^{m=+s,n=+s}$ of size $N=M\times M==\left(2s+1\right)\times\left(2s+1\right)$ is used, where $M=2s+1$. Obviously, $\left\{f_{(i,j)}(n,m)\right\}_{(n.m)\in M(i,j)}$ is an image block of the fixed size $N=M\times M$ extracted from f by moving window $\mathrm{M}_{(i,j)}$ at the position (i,j). Our idea consists in ordering the pixels according to Radix-s number system:

$$\left\{f_{(i,j)}(n,m)\right\}_{(n.m)\in M(i,j)}=\left\{f_{(i,j)}(-s,-s),...,f_{(i,j)}(0,0),...,f_{(i,j)}(s,s)\right\}=\left\{f_{(i,j)}(r)\right\}_{r=1}^{N}=\left\{f_{(i,j)}(1),f_{(i,j)}(2),...,f_{(i,j)}(N)\right\}.$$

where the map $(n,m)\to r$ has the following form $r = M(n+s)+(m+s)$. For example, for the window of size 3×3 we have $(-1,-1)\to 0,\ (-1,0)\to 1,\ (-1,1)\to 2,\quad (0,-1)\to 3,\ (0,0)\to 4,\ (0,1)\to 5,\ (1,-1)\to 6,\ (1,0)\to 7,\ (1,1)\to 8.$ Here $M=3,\ s=1$ and $r=3(n+1)+(m+1)$. Now we define a product of k pixels $f_{(i,j)}(r_1)f_{(i,j)}(r_2)...f_{(i,j)}(r_k)$ from the image block $\left\{f_{(i,j)}(r)\right\}_{r=1}^{N}=\left\{f_{(i,j)}(n,m)\right\}_{(n.m)\in M(i,j)}$. Using this product we define the generalized aggregation Heronian filter as

$$\hat{s}(i,j)=\underset{(m,n)\in M_{(i,j)}}{\mathbf{AggHeron}^k}\left\{f(m,n)\right\}=\underset{r_1\le r_2\le ...\le r_k}{\mathbf{Aggreg}}\left[\left\{\sqrt[k]{f_{(i,j)}(r_1)f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}\right\}\right]. \tag{28}$$

In particular cases, we have the following Heronian filters.

1) The arithmetic k-Heronian filter

$$\hat{s}(i,j)=\underset{(m,n)\in M_{(i,j)}}{\mathbf{ArithHeron}^k}=\underset{r_1\le r_2\le ...\le r_k}{\mathbf{Arith}}\left[\left\{\sqrt[k]{f_{(i,j)}(r_1)f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}\right\}\right]=\frac{1}{C_{N+k-1}^{k}}\sum_{r_1}\ \le\ \sum_{r_2}\ \le\ ...\le\sum_{r_k}\sqrt[k]{f_{(i,j)}(r_1)f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}. \tag{29}$$

2) The median k-Heronian filter

$$\hat{s}(i,j)=\underset{(m,n)\in M_{(i,j)}}{\mathbf{MedHeron}^k}\left\{f(m,n)\right\}=\underset{r_1\le r_2\le ...\le r_k}{\mathbf{Med}}\left[\left\{\sqrt[k]{f_{(i,j)}(r_1)f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}\right\}\right]. \tag{30}$$

3) The Kolmogorov-Heronian filter

$$\begin{aligned}\hat{s}(i,j)&=\underset{(m,n)\in M_{(i,j)}}{\mathbf{KolmHeron}^r}\left\{K\mid f(m,n)\right\}=K^{-1}\left[\underset{r_1\le r_2\le ...\le r_k}{\mathbf{Arith}}\left(\left\{K\left(\sqrt[k]{f_{(i,j)}(r_1)f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}\right)\right\}\right)\right]=\\&=K^{-1}\left[\frac{1}{C_{N+k-1}^{k}}\sum_{r_1}\ \le\ \sum_{r_2}\ \le\ ...\le\sum_{r_k}K\left(\sqrt[k]{f_{(i,j)}(r_1)f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}\right)\right].\end{aligned} \tag{31}$$

It is easy to see that $\sum_{i_1}\ \le\ \sum_{i_2}\ \le\ ...\le\sum_{i_k}\sqrt[k]{x_{i_1}x_{i_2}\cdots x_{i_k}}$ is a symmetric polynomial in the variables $x_1,...,x_N$. There are a few types of symmetric polynomials in variables $x_1,...,x_N$, which that are associated new symmetric means.

3.6. Symmetric aggregation filters

Any monomial in $x_1,x_2,...,x_N$ can be written as $x_1^{p_1}x_2^{p_2}...x_N^{p_N}$, where the exponents p_i are natural numbers (possibly zero); writing $\mathbf{p}=\left(p_1,p_2,...,p_N\right)$ this can be abbreviated to $\mathbf{X}^{\mathbf{p}}=x_1^{p_1}x_2^{p_2}...x_N^{p_N}$. If $p=p_1+p_2+...+p_N$ then we write $\sqrt[p]{\mathbf{X}^{\mathbf{p}}}=\sqrt[p]{x_1^{p_1}x_2^{p_2}...x_N^{p_N}}$.

Definition 3. The monomial symmetric polynomial is defined as the sums of all monomials $\sqrt[q]{\mathbf{X}^{\mathbf{q}}}$, where $\mathbf{q}$ ranges over all distinct permutations of $\mathbf{p}$:

$$\mathbf{Mon}_{\mathbf{p}}(x_1,x_2,...,x_N)=\sum_{\sigma\in S_N}\sqrt[p]{x_1^{\sigma(p_1)}x_2^{\sigma(p_2)}...x_N^{\sigma(p_N)}}. \tag{32}$$

where S_N is the set of all permutations of $p_1,p_2,...,p_N$. These monomial symmetric polynomials form a vector space basis: every symmetric polynomial can be written as a linear combination of the monomial symmetric polynomials.

Definition 4. Let $x_1,x_2,...,x_N$ be positive real numbers and $\mathbf{p}=\left(p_1,p_2,...,p_N\right)\in R^N$. The $\mathbf{p}$-Muirhead symmetric *polynomial* [6] is defined by

$$\mathbf{Mui}_{\mathbf{p}}(x_1,x_2,...,x_N)=\sum_{\sigma\in S_N}\sqrt[p]{x_{\sigma(1)}^{p_1}x_{\sigma(2)}^{p_2}...x_{\sigma(N)}^{p_N}}, \tag{33}$$

For example,

$$\begin{aligned}&\mathbf{Mui}_{(1,0,...,0)}(x_1,x_2,...,x_N)=\sum_{i=1}^{N}x_i=\mathbf{Mean}(x_1,x_2,...,x_N),\\&\mathbf{Mui}_{(1,1,...,1)}\left(x_1,x_2,...,x_N\right)=\sqrt[N]{x_1x_2\cdots x_N}=\mathbf{Geo}(x_1,x_2,...,x_N),\\&\mathbf{Mui}_{(\underbrace{1,1,...,1}_{k},0...,0)}(x_1,x_2,...,x_N)=\sum_{r_1}\ \le\ \sum_{r_2}\ \le\ ...\le\sum_{r_2}\sqrt[k]{x_{r_1}x_{r_2}\cdots x_{r_k}}=\mathbf{Heron}_k(x_1,x_2,...,x_N).\end{aligned} \tag{34}$$

For each nonnegative integer $0 \le k \le N$ the elementary $\mathbf{El}_k^I(x_1, x_2, ..., x_N)$ and homogeneous $\mathbf{Hom}_k^I(x_1, x_2, ..., x_N)$ symmetric polynomials are the sums of all distinct products of k distinct variables:

$$\mathbf{El}_k\left(x_1, x_2, ..., x_N\right) = \sum_{r_1 < r_2 < .... < r_k} \sqrt[k]{x_{r_1} x_{r_2} \cdots x_{r_k}}, \ \mathbf{Hom}_k\left(x_1, x_2, ..., x_N\right) = \sum_{r_1 \le r_2 \le .... \le r_k} \sqrt[k]{x_{r_1} x_{r_2} \cdots x_{r_k}} \tag{35}$$

We then define

$$\mathbf{El}_{k_1 k_2 ... k_q}\left(x_1, x_2, ..., x_N\right) = \sqrt[q]{\prod_{t=1}^{q} \mathbf{El}_{k_t}\left(x_1, x_2, ..., x_N\right)}, \ \mathbf{Hom}_{k_1 k_2 ... k_q}\left(x_1, x_2, ..., x_N\right) = \sqrt[q]{\prod_{t=1}^{q} \mathbf{Hom}_{k_q}\left(x_1, x_2, ..., x_N\right)}. \tag{36}$$

To each of polynomial $\mathbf{Mon}_{p_1, p_2, ..., p_N}, \mathbf{Mui}_{p_1, p_2, ..., p_N}$, $\mathbf{El}_{k_1 k_2 ... k_r}$, $\mathbf{Hom}_{k_1 k_2 ... k_r}$ we will associate normalized symmetric function:

$$\begin{aligned}
&\overline{\mathbf{Mon}}_{p_1, ..., p_N}\left(x_1, ..., x_N\right) = \mathbf{Mon}_{p_1, ..., p_N}\left(x_1, ..., x_N\right) / \mathbf{Mon}_{p_1, ..., p_N}\left(1, ..., 1\right), \\
&\overline{\mathbf{Mui}}_{p_1, ..., p_N}\left(x_1, ..., x_N\right) = \mathbf{Mui}_{p_1, ..., p_N}\left(x_1, ..., x_N\right) / \mathbf{Mui}_{p_1, ..., p_N}\left(1, ..., 1\right), \\
&\overline{\mathbf{El}}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right) = \mathbf{El}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right) / \mathbf{El}_{k_1 k_2 ... k_r}\left(1, ..., 1\right), \\
&\overline{\mathbf{Hom}}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right) = \mathbf{Hom}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right) / \mathbf{Hom}_{k_1 k_2 ... k_r}\left(1, ..., 1\right),
\end{aligned} \tag{37}$$

We obtain four families of a generalized symmetric means:

$$\begin{aligned}
&\mathbf{MonMean}_{p_1, p_2, ..., p_N}\left(x_1, ..., x_N\right) = \sqrt[p]{\overline{\mathbf{Mon}}_{p_1, p_2, ..., p_N}\left(x_1, ..., x_N\right)}, \\
&\mathbf{MuiMean}_{p_1, p_2, ..., p_N}\left(x_1, ..., x_N\right) = \sqrt[p]{\overline{\mathbf{Mui}}_{p_1, p_2, ..., p_N}\left(x_1, ..., x_N\right)}, \\
&\mathbf{ElMean}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right) = \sqrt[k]{\overline{\mathbf{El}}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right)}, \\
&\mathbf{HomMean}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right) = \sqrt[k]{\overline{\mathbf{Hom}}_{k_1 k_2 ... k_r}\left(x_1, ..., x_N\right)}.
\end{aligned} \tag{38}$$

where $k = k_1 + k_2 + ... + k_r$, and $p = p_1 + p_2 + ... + p_N$. Using generalized symmetric means, we can construct the following families of symmetric **MonArith**-, **MuiArith**-, **ElMean**- and **HomMean**-filters:

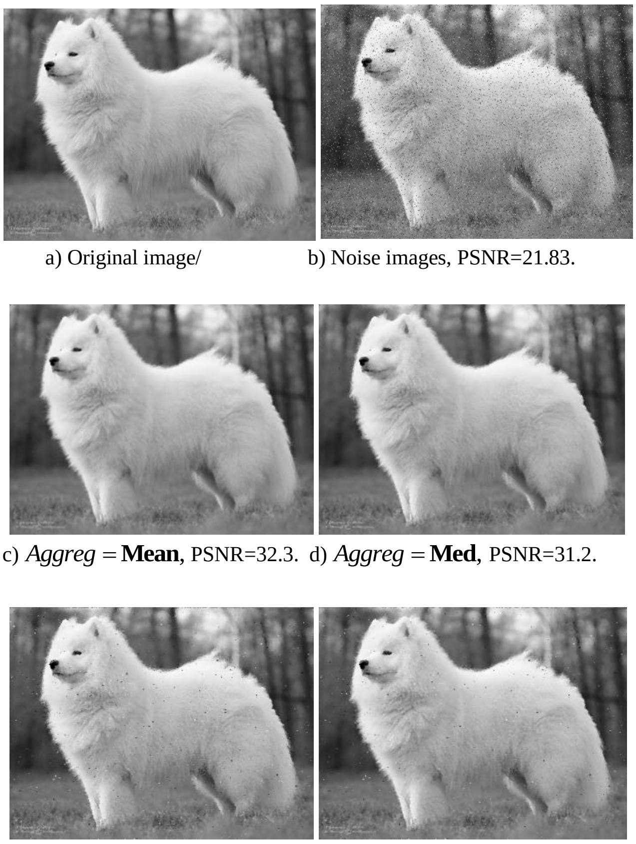
$$\begin{aligned}
&\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{MonArith}}\left\{p_1, p_2, ..., p_N \mid f(m,n)\right\} = \underset{M_{(i,j)}}{\mathbf{Arith}}\left[\left\{\sqrt[p]{\prod_{l=1}^{N} f_{(i,j)}^{\sigma(p_l)}(l)}\right\}_{\sigma \in S_N}\right] = \\
&= \underset{M_{(i,j)}}{\mathbf{Arith}}\left[\left\{\sqrt[p]{f_{(i,j)}^{\sigma(p_1)}(1) f_{(i,j)}^{\sigma(p_2)}(2) \cdots f_{(i,j)}^{\sigma(p_N)}(N)}\right\}_{\sigma \in S_N}\right] = \frac{1}{\mathbf{Mon}_{p_1, ..., p_N}\left(1, ..., 1\right)} \sum_{\sigma \in S_N} \sqrt[p]{\prod_{l=1}^{N} f_{(i,j)}^{\sigma(p_l)}(l)}
\end{aligned} \tag{39}$$

$$\begin{aligned}
&\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{MuiArith}}\left\{p_1, p_2, ..., p_N \mid f(m,n)\right\} == \underset{M_{(i,j)}}{\mathbf{Arith}}\left[\left\{\sqrt[p]{\prod_{l=1}^{N} f_{(i,j)}^{p_l}\left(\sigma(l)\right)}\right\}_{\sigma \in S_N}\right] = \\
&= \underset{M_{(i,j)}}{\mathbf{Arith}}\left[\left\{\sqrt[p]{f_{(i,j)}^{p_1}\left(\sigma(r_1)\right) f_{(i,j)}^{p_2}\left(\sigma(r_2)\right) \cdots f_{(i,j)}^{p_N}\left(\sigma(r_N)\right)}\right\}_{\sigma \in S_N}\right] = \frac{1}{\mathbf{Mui}_{p_1, ..., p_N}\left(1, ..., 1\right)} \sum_{\sigma \in S_N} \sqrt[p]{\prod_{l=1}^{N} f_{(i,j)}^{p_l}\left(\sigma(l)\right)}
\end{aligned} \tag{40}$$

$$\begin{aligned}
&\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{ElArith}}\left\{k \mid f(m,n)\right\} = \underset{r_1 < r_2 < ... < r_{k_t}}{\mathbf{Arith}}\left[\left\{\sqrt[k]{f_{(i,j)}(r_1) f_{(i,j)}(r_2) \cdots f_{(i,j)}(r_k)}\right\}\right]. \\
&= \sum_{r_1 < r_2 < .... < r_k} \left\{\sqrt[k]{f_{(i,j)}(r_1) f_{(i,j)}(r_2) \cdots f_{(i,j)}(r_k)}\right\} / \mathbf{El}_k\left(1, ..., 1\right), \\
&\hat{s}(i,j) = \underset{(m,n)\in M_{(i,j)}}{\mathbf{ElArith}}\left\{q; k_1 k_2 ... k_q \mid f(m,n)\right\} = \prod_{t=1}^{q}\left(\underset{r_1 < r_2 < ... < r_{k_t}}{\mathbf{Arith}}\left[\left\{\sqrt[k_t]{f_{(i,j)}(r_1) f_{(i,j)}(r_2) \cdots f_{(i,j)}(r_{k_t})}\right\}\right]\right) \\
&= \prod_{t=1}^{q}\left(\sum_{r_1 < r_2 < .... < r_k} \left\{\sqrt[k]{f_{(i,j)}(r_1) f_{(i,j)}(r_2) \cdots f_{(i,j)}(r_{k_t})}\right\} / \mathbf{El}_{k_t}\left(1, ..., 1\right)\right),
\end{aligned} \tag{41}$$

$$
\begin{aligned}
\hat{s}(i,j) &= \underset{(m,n)\in M_{(i,j)}}{\mathbf{HomArith}}\left\{k \mid f(m,n)\right\} = \underset{r_1\le r_2\le\ldots\le r_k}{\mathbf{Arith}}\left[\left\{\sqrt[k]{f_{(i,j)}(r_1)\,f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}\right\}\right].\\
&= \sum_{r_1<r_2<\ldots<r_k}\left\{\sqrt[k]{f_{(i,j)}(r_1)\,f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_k)}\right\} / \mathbf{Hom}_k\,(1,\ldots,1),\\
\hat{s}(i,j) &= \underset{(m,n)\in M_{(i,j)}}{\mathbf{HomArith}}\left\{q;k_1k_2\ldots k_q \mid f(m,n)\right\} = \prod_{t=1}^{q}\left(\underset{r_1\le r_2\le\ldots\le r_{k_t}}{\mathbf{Arith}}\left[\left\{\sqrt[k_t]{f_{(i,j)}(r_1)\,f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_{k_t})}\right\}\right]\right)\\
&= \prod_{t=1}^{q}\left(\sum_{r_1\le r_2\le\ldots\le r_{k_t}}\left\{\sqrt[k]{f_{(i,j)}(r_1)\,f_{(i,j)}(r_2)\cdots f_{(i,j)}(r_{k_t})}\right\} / \mathbf{Hom}_{k_t}\,(1,\ldots,1)\right).
\end{aligned}
\tag{42}
$$

As we see, aggregation operator **Arith** is used in (39)-(42). We can use here an arbitrary aggregation operator **Aggreg** . Some of similar filters can find in [7-10].

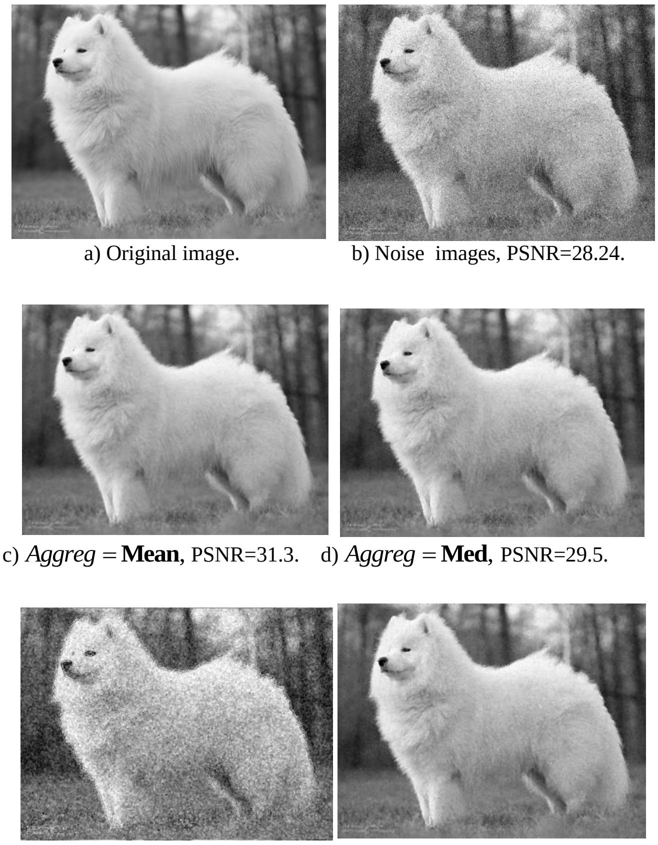


a) Original image/ b) Noise images, PSNR=21.83.

c) $Aggreg = \mathbf{Mean}$, PSNR=32.3. d) $Aggreg = \mathbf{Med}$, PSNR=31.2.

e) $Aggreg = \mathbf{Min}$, PSNR=27.5. f) $Aggreg = \mathbf{Geo}$, PSNR=28.6.

Fig. 2. Original (a) and noise (b) images.
Noise: "Salt-Pepper PD". Denoised images (c)-(f).



a) Original image. b) Noise images, PSNR=28.24.

c) $Aggreg = \mathbf{Mean}$, PSNR=31.3. d) $Aggreg = \mathbf{Med}$, PSNR=29.5.

e) $Aggreg = \mathbf{Min}$, PSNR=19.9. f) $Aggreg = \mathbf{Geo}$, PSNR=28.7.

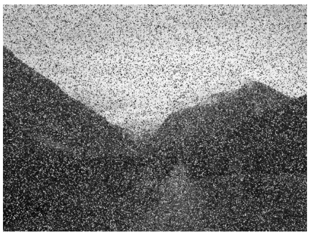
Fig. 3. Original (a) and noise (b) images.
Noise: "Laplasian PDF". Denoised images (c)-(f).

4. Results and Discussion

The following generalized aggregation Heronian filtering $\mathbf{MeanHeron}^2$, $\mathbf{MedHeron}^2$, $\mathbf{MinHeron}^2$, $\mathbf{GeoHeron}^2$ for $N = |\mathbf{M}| = M \times M = 5\times 5$ has been applied to noised 256x256 gray level "*Dog*" images (Figures 2b,3b). The denoised images are shown in Figures 2–3. In Fig.4-5, we present examples of **MonArith**-, **MuiArith**-, **ElMean- and HomMean-**filtering. All filters have very good denoising properties. This fact confirms that further investigation of these new filters is perspective. Particularly, very interesting is a question about the types of noises, for which such filters are optimal.

5. Conclusion

We developed a new theoretical framework for image filtering using aggregation operators. The main goal of the work is to show that aggregation operators can be used to solve problems of image filtering in a natural and effective manner. Some properties of a nonlinear aggregation filters are exploited in this paper. Unlike the linear masking filter, they avoid amplification thanks to the nonlinearity of the response to luminance variations; unlike the classical linear and median filters, they are able to sharpen even small details as its impulse response demonstrates.

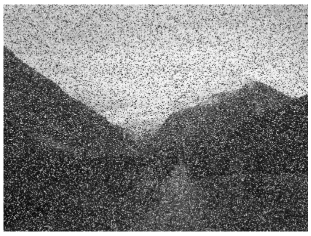


a) Original image. b) Noise images, PSNR=23.

c) **MonMean,** PSNR =29. d) **MuiMean,** PSNR =28.

e) **ElMean,** PSNR =28. f) **HomMean,** PSNR =27.

Fig. 1. Original (a) and noise (b) images.
Noise: “Salt-Peper PD”. Denoised images (c)-(f).



a) Original image. b) Noise images, PSNR=23.

c) **MonMean,** PSNR =30. d) **MuiMefn,** PSNR =31.

e) **ElMean,** PSNR =31. f) **HomMean,** PSNR =32.

Fig. 2. Original (a) and noise (b) images.
Noise: “Laplasian PDF”. Denoised images (c)-(f).

Acknowledgements

This work was supported by grants the RFBR No. 17-07-00886 and by Ural State Forest Engineering’s Center of Excellence in ”Quantum and Classical Information Technologies for Remote Sensing Systems”.

References

[1] Gonzalez, R.C., Woods, R.E. Digital Image Processing. ─ New York: Addison-Wesley. ─ 1992. – 582 p.

[2] Mitra S.K., Nonlinear Image Processing // Academic Press Series in Communications, Networking, and Multimedia. – San Diego, New York.– 2001–248 p.

[2] Mayor, G., On the representation of some Aggregation functions/ Trillas, E. // Proceeding of ISMVL. – 1986. – P. 111-114.

[3] Ovchinnikov, S. On Robust Aggregation Procedures // Aggregation Operators for Fusion under Fuzziness / Bouchon-Meunier B. (eds.) . – 1998. – P. 3-10.

[4] Kolmogorov, A. Sur la notion de la moyenne // Atti Accad. Naz. Lincei. – 1930. – Vol. 12. – P. 388–391.

[5] Sykora, S. Mathematical Means and Averages // Generalized Heronian means I, II, III / Stan's Library, Ed. S. Sykora . – Vol. III. – 2009. – 57 p.

[6] Muirhead, R. F. Some methods applicable to identities of symmetric algebraic functions of *n* letters // Proc. Edinburgh Math. Soc. – 1902. – Vol. 21. – P. 144 – 157.

[7] Labunets, V. G. Filters based on aggregation operators. Part 1. Aggregation Operators // Proc. of the 24th Int. Crimean Conference “Microwave & Telecommunication Technology”. – Sevastopol, Crimea, Russia. – 2014. – Vol. 24. – P. 1239 – 1240

[8] Labunets, V.G. Filters based on aggregation operators. Part 2. The Kolmogorov filters / Gainanov, D.N., Ostheimer, E. // Proc. of the 24th Int. Crimean Conference “Microwave & Telecommunication Technology” . – Sevastopol, Crimea, Russia. – Vol. 24. – P. 1241 – 1242.

[9] Labunets, V.G. Filters based on aggregation operators. Part 3. The Heron filters/ Gainanov, D.N., Tarasov A.D., Ostheimer E. // Proc. of the 24th Int. Crimean Conference Microwave & Telecommunication Technology” – Sevastopol, Crimea, Russia. – Vol. 24. – P. 1243 – 1244.

[10] Labunets, V.G. Filters based on aggregation operators. Part 4. Generalized vector median flters / Gainanov, D.N., Arslanova, R.A., Ostheimer, E. // Proc. of the 24th Int. Crimean Conference “Microwave& Telecommunication Technology” . – Sevastopol, Crimea, Russia. – Vol. 24. – P. 1245—1246.