We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

IMAGE QUALITY ENHANCEMENT BY APPLYING GENETIC ALGORITHM IN MEDIAN FILTERING

Sandra Sovilj-Nikic
University of Novi Sad
Serbia

1. Introduction

Images are often corrupted by impulse noise due to errors generated in noisy sensors or communication channels. Two types of impulse noise can be defined: 1) fixed-valued and 2) random-valued. Fixed-valued impulses occur when noisy pixel values are out of receiver's range and they can have one of two values (MIN or MAX). This type of noise is called "salt & pepper" noise and it appears as black and/or white impulses on the image. Another type of noise is generated after incorrect decoding of binary represented image. In this case noisy pixels can have arbitrary value.

In many applications it is very important to remove noise in the images before some subsequent processing such as edge detection, object recognition and image segmentation.

At the beginning of development of techniques in digital image processing, linear techniques are used extensively because of their mathematical simplicity and the existence of appropriate characteristics (e.g. principle of superposition) making them easy to design and implement. In the case where noise can be modulated as additive Gaussian noise linear techniques offer satisfactory performance for noise removal. However, in many cases the noise is impulsive and in this case linear techniques do not usually perform well. Another example where linear techniques fail is the case of nonlinear image degradations. Such degradations occur during image formation and during image transmission through nonlinear channels. The human visual perception mechanism has been shown to have nonlinear characteristics. Human vision is very sensitive to high-frequency information such as edges and fine details on the image (e.g. lines and corners). Therefore, high-frequency content of image is very important for visual perception. However, most of linear filters have low-pass characteristics. They tend to blur edges and to destroy lines and other fine details of image.

All above mentioned reasons led to leave linear techniques and to the use of nonlinear filtering techniques. One of nonlinear filters family includes filters based on order statistics (Pitas & Venetsanopoulos, 1992). The family of order statistics based filters is very rich, where the median filter is the best known. Median filter and its modification have shown good efficiency in suppressing impulse noise and capability of preserving image edges

(Pitas & Venetsanopoulos, 1991). Nevertheless, the median filters possess some disadvantages. The main disadvantage of median filter is location-invariant property, i.e. they are implemented uniformly across the entire image. They tend to alter both noise pixels and undisturbed good pixels. Therefore, they remove fine details in the image. Partition-based filtering is one of the concepts for eliminating disadvantages of median filtering.

In this chapter partition based median (PBM) filtering using genetic algorithm in the training process is proposed. With proposed PBM filter, at each location, observed vector is classified into one of M exclusive partitions, and an optimal particular filtering operation is then activated. The observation vector space is formed on differences between current pixel value and the outputs of the center weighted median (CWM) filters with variable center weights. The estimate at each location is formed as a linear combination of current pixel value and the outputs of CWM filters. Optimal weighting vector of each partition is derived using genetic algorithm in training the filter on the reference image. Optimal weights are derived to minimize the total square error of estimation at each location.

Performance of the proposed PBM filter has been evaluated by simulations on variety of test images. The proposed PBM filter in which genetic algorithm is used for optimization of the weighting vector for each partition had demonstrated better results in noise suppressing than competitive filters based on median filtering in terms of the SNR as well as the perceived image quality. The proposed filter outperforms other median based filters in removing different types of noise: impulse noise (fixed-valued and random-valued), Gaussian noise and mixed Gaussian and impulse noise.

In this chapter following introduction the definition of PBM filter and partition into regions are described. In section 3 application of genetic algorithm for solving the optimization problem is discussed. Finally in section 4 obtained results and comparison with competitive filters are given and discussed.

2. Definition of PBM filter

Schematic diagram of PBM filter is shown on Figure 1 (Chen & Wu, 2001).

Let $C = \{(c1, c2) \mid 1 \leq c1 \leq P, 1 \leq c2 \leq Q\}$ denote pixel coordinates of a digital image, where P and Q are its height and width, respectively. At each location $\mathbf{c}(c1, c2) \in C$ a filter window of size $W = 2n + 1$ symmetrically surrounding the current pixel, and

$$\mathbf{x}(\mathbf{c}) = \{x_i(c) : i = 1, 2, \dots, 2n + 1\} \quad (1)$$

denotes set of observed pixels via filter window, where $x(c) = x_{n+1}$ is original or center pixel.

Another step in filtering is to apply CWM filters with variable center weights on input vector $\mathbf{x}(\mathbf{c})$ (Ko & Lee, 1991; Pitas & Venetsanopoulos, 1991). The output of CWM filter is described as:

$$y_k(c) = \text{median}(\mathbf{x}_{2k+1}(c)) = \text{median}(\mathbf{x}_\omega(c)) \quad (2)$$

where ω denotes center weight and $\omega = 2k + 1$, k is nonnegative integer.

$$\mathbf{x}_\omega(c) = \{x_1(c), x_2(c), x_3(c), \dots, x_n(c), \omega \diamond x_{n+1}(c), x_{n+2}(c), \dots, x_{2n+1}(c)\} \quad (3)$$

Fig. 1. Schematic diagram of PBM filter

Operator $\diamond$ denotes repetition operation. Two limit cases exist in the CWM filtering. Output of CWM filter y_0 ($k=0, \omega=1$) corresponds to the output of standard median filter, and for $k \geq n$ CWM filter is an identity filter $y_k = x(c)$. Taking two limit cases into consideration, CWM filter with higher center weight is superior in preserving fine details in the image than filter with smaller center weight. But, CWM filter with smaller center weight is better in noise suppressing. Therefore, we should find a compromise solution between preserving of fine details and noise suppressing. That was the reason for applying the CWM filter with variable center weights. In the proposed filtering scheme k lies in $[0, n-1]$.

At each location, for the current pixel, the following differences are defined:

$$d_k(c) = |y_k(c) - x(c)| \quad (4)$$

where $k = 0, 1, \dots, n-1$ and $d_k(c) \leq d_{k-1}(c)$.

On the basis of these differences, it can be found out whether the current pixel is corrupted one. For instance, we can observe the difference $d_{n-1}(c)$. If it is too large, that implies the current pixel value is the biggest or the smallest in the set of observed pixels $\mathbf{x}(c)$ (1) and it is probably the corrupted pixel. On the other side, if the difference $d_0(c)$ is small, it can be concluded that the current pixel is uncorrupted one and it should be preserved unaltered.

The information about noise presence on the location $\mathbf{c}$ can be obtained on the basis of differences from $d_0(c)$ to $d_{n-1}(c)$.

At each location $\mathbf{c}$, the estimation of pixel value is derived as a linear combination of outputs of CWM filter with variable center weights and a current pixel value, which is equivalent to the output of the CWM filter with $k = n$ as follows:

$$\hat{x}(c) = \mathbf{y}(\mathbf{c}) \cdot \mathbf{w}_{p(c)}^T \quad (5)$$

where $p(c)$ corresponds to the partition index for $\mathbf{x}(\mathbf{c})$ which will be defined later and $\mathbf{y}(\mathbf{c})$ is the input vector for filtering at observed location:

$$\mathbf{y}(\mathbf{c}) = [y_0(c), y_1(c), \dots, y_{n-1}(c), y_n(c)] = [y_0(c), y_1(c), \dots, y_{n-1}(c), x(c)] \quad (6)$$

Vector $\mathbf{w}_{p(c)} = \mathbf{w}_i = [w_{i,0}, w_{i,1}, \dots, w_{i,n}]$ represents weighting vector of the particular filter for the i th partition ($i = 1, 2, \dots, M$). Optimal weighting vector for each partition is derived using genetic algorithm.

To control the dynamic range of outputs, weighting vector should satisfy a location-invariance constraint:

$$\mathbf{e}_{n+1} \cdot \mathbf{w}_i^T = \mathbf{w}_i \cdot \mathbf{e}_{n+1}^T = 1 \quad (i = 1, 2, \dots, M) \quad (7)$$

where $\mathbf{e}_N = [1, 1, \dots, 1]$ is $1 \times N$ vector.

If $x(c) = \text{median}(\mathbf{x}(\mathbf{c}))$, then $y_k(c) = x(c)$ for any $k \geq 0$ and current pixel $x(c)$ remains unaltered. In this case, let $p(c) = i$, we have:

$$\hat{x}(c) = \mathbf{y}(\mathbf{c}) \cdot \mathbf{w}_i^T = \mathbf{e}_{n+1} \cdot x(c) \cdot \mathbf{w}_i^T = x(c) \quad (8)$$

2.1 Region partitioning

At each location $\mathbf{c}$, difference vector is obtained by applying (4) as follows:

$$\mathbf{d}(\mathbf{c}) = [d_0(c), d_1(c), \dots, d_{n-1}(c)] \in \mathbf{R}^n. \quad (9)$$

The difference vector space is divided into M exclusive regions, $\{\Omega_i : i = 1, 2, \dots, M\}$. The partition index for each input vector $\mathbf{x}(\mathbf{c})$ is given such that $p(c) = i$ for $\mathbf{d}(\mathbf{c}) \in \Omega_i$. Partition in regions can be fulfilled by different methods, e.g. the scalar quantization, the vector quantization etc. Due to its simplicity and computational efficiency, scalar quantization has been chosen to apply for partitioning n -dimensional difference vector

space into regions. Quantization levels are determined experimentally and remain fixed through the simulations.

Let $\{q_{k,v} : v = 0, 1, \dots, L\}$ be a set of monotonically ascending points, i.e.:

$$q_{k,v} < q_{k,v+1} \quad , \quad 0 \leq v \leq L-1$$

(10)

for the k th dimension of difference vector space, where $k = 0, 1, \dots, n-1$.

At each location, for each input vector $\mathbf{x}(\mathbf{c})$ and appropriate difference vector $\mathbf{d}(\mathbf{c})$ is defined vector of the quantization level indices as $\mathbf{z}_i = [z_{i,0}, z_{i,1}, \dots, z_{i,n-1}]$ for $i = 1, 2, \dots, M$.

In the other words scalar quantization is described as:

$$\mathbf{Q}(\mathbf{d}(\mathbf{c})) = \mathbf{z}_i$$

(11)

where $z_{i,k} = v$ if $q_{k,v} \leq d_k(c) \leq q_{k,v+1}$ for $k = 0, 1, \dots, n-1$.

k	v						
	0	1	2	3	4	5	6
0	0	5	20	35	50	70	256
1	0	5	15	25	35	55	256
2	0	5	10	15	25	35	256
3	0	2	5	10	15	20	256

Table 1. Quantization levels for $\{q_{k,v} : 0 \leq k \leq n-1, 0 \leq v \leq L\}$ obtained via a 3×3 ($n = 4, L = 6$) filter window

In that way partitioning of n -dimensional space in $M = L^n$ regions is fulfilled, and unique vector $\mathbf{z}_i$ defines a distinct region for $i = 1, 2, \dots, M$. Experimentally determined quantization levels for 3×3 filter window are given in Table 1 (Chen & Wu, 2001). In this study 4-dimensional space has been partitioned in $M = L^n = 6^4 = 1296$ different regions. Using the scalar quantization an estimation of partition index is fulfilled without multiplication which leads to the decrease of computational complexity in the training and the filtering process.

2.2 LMS weights optimization

Optimal weights are determined to minimize the mean square error, or equivalently the total square error:

$$\varepsilon = \sum_{c \in C} \left[s(c) - \hat{x}(c) \right]^2 \quad (12)$$

where $s(c)$ and $\hat{x}(c)$ denote the original pixel value and its estimate, respectively. Taking partitioning of n -dimensional space in M exclusive regions into consideration we can write:

$$\varepsilon = \sum_{i=1}^M \left\{ \sum_{c: p(c)=i} \left[s(c) - \hat{x}(c) \right]^2 \right\} \quad (13)$$

where the inner sum, i.e. ε_i is error attributed to the pixels classified into the i th partition. Thus, entire error ε can be minimized by achieving the independent minimization of ε_i for $i = 1, 2, \dots, M$

Following (5) we have:

$$\varepsilon_i = \sum_{c: p(c)=i} \left[s(c) - \hat{x}(c) \right]^2 = \sum_{c: p(c)=i} \left[s(c) - \mathbf{y}(c) \cdot \mathbf{w}_i^T \right]^2 \quad (14)$$

Optimization problem is to find optimal weighting vector $\mathbf{w}_i$ of i th partition for $i = 1, 2, \dots, M$ and ε_i of each partition should be minimal.

In this study genetic algorithm is used to derive optimal weighting vector $\mathbf{w}_i$ for each partition.

2.3 Recursive implementation

In this study PBM filter is implemented recursively because of obtained results through the simulations on variety of images. Superior results have been produced by the recursive PBM filter in removing noise than nonrecursive design.

The estimate of current pixel value in recursive filtering depends on past filter outputs, with input

$$\mathbf{x}'(\mathbf{c}) = \left\{ \hat{x}_1(c), \hat{x}_2(c), \dots, \hat{x}_n(c), x_{n+1}(c), x_{n+2}(c), \dots, x_{2n+1}(c) \right\} \quad (15)$$

In the recursive PBM filtering the partition index for current pixel depends on previous filter outputs.

Recursive design of PBM filtering is shown on Figure 2 (Chen & Wu, 2001).

Fig. 2. Schematic diagram of recursive PBM filter

3. Optimization using genetic algorithm

Evolution algorithms are methods for solving optimization problems which are based on the principles of Darwin's evolution theory, i.e. natural selection and survival of the fittest (Koza, 1992; Michalewicz, 1996). The best known algorithms in this class include genetic algorithm, genetic programming, evolutionary programming, evolution strategies, classifier systems, and neural networks. All above mentioned algorithms are based on the same concept-simulating the evolution of organisms of some population through selection, recombination and mutation process (Michalewicz, 1996; Brezocnik, 2000). Having in mind the nature of optimization problem and available data at the beginning of optimization process, it was decided that genetic algorithm will be used for solving the optimization problem. This choice has been shown as good one because the optimization process is not time-consuming, i.e. genetic algorithm converges very quickly.

3.1 Training process

Matrix of weighting vectors for each partition is obtained by training the filter over reference image. Optimal weights for each partition are obtained using genetic algorithm in the training process (Sovilj-Nikic S. & Sovilj-Nikic I., 2007). Structure of genetic algorithm is shown on Figure 3 (Sovilj-Nikic et al., 2008). Total square error of the current pixel value estimate between all combinations (previously memorized) of vector **y(c)** and the original pixel value should be minimal for each partition. In this study the number of memorized combinations was chosen to be 1000.

Fig. 3. Structure of genetic algorithm

3.1.1 Optimization function by applying genetic algorithm

Genetic algorithm has been applied for each partition and final result of that procedure is a matrix which consists of weighting vectors for each partition. First step in application of genetic algorithm is to choose population size, i.e. number of individuals in the population (Figure 4). Typical values of population size are from 20 to 500 individuals. Simulations on different values of population size have been shown that population of 120 individuals represents compromise solution for the optimization problem in this study regarding a quality of obtained solution and computational complexity of the algorithm. Individuals of population are generated randomly using Matlab's random numbers generator. Each individual represents potential problem solution, i.e. optimal weighting vector $\mathbf{w}$ which is one 1×5 vector. Columns of vector $\mathbf{w}$ are chromosomes (genes) of individuals. Range of

values for individual chromosomes is $[-1, 1]$ and weighting vectors satisfy location-invariance constraint:

$$\mathbf{e}_{n+1} \cdot \mathbf{w}^T = \mathbf{w} \cdot \mathbf{e}_{n+1}^T = 1 \quad (16)$$

where $\mathbf{e}_N = [1, 1, \dots, 1]$ is an $1 \times N$ vector.

After creating an initial population should evaluate fitness function for each individual in the population, In this case fitness function is a total square error ε , i.e. deviation between estimate of current pixel value and original (uncorrupted) pixel value:

$$\varepsilon = (s - \mathbf{y} \cdot \mathbf{w}^T)^2 \quad (17)$$

where is:

- s uncorrupted pixel value
- $\mathbf{y}$ vector of CWM filter outputs
- $\mathbf{w}$ weighting vector (individual of population)

After generating initial population comes iterative procedure of genetic operators (selection, crossover and mutation) application to the individuals. This procedure is repeated until the satisfaction of convergence criteria. If convergence criteria is not satisfied the fittest individual in the current population is found. That individual has been kept in the memory. Then, a new population is formed by selecting the more fit individuals. In this study tournament selection is applied to select potential parents. Using Matlab's random numbers generator two parents are chosen randomly and they create their two offsprings in the crossover operation. Crossover enables to exchange information between different potential solutions. In this study decade encoding and real arithmetic crossover is applied (Michalewicz, 1996).

Crossovered offspring genes are formed as:

$$\begin{aligned} c1(i) &= \frac{k1 \cdot r1(i) + k2 \cdot r2(i)}{2} \\ c2(i) &= \frac{k2 \cdot r1(i) + k1 \cdot r2(i)}{2} \end{aligned} \quad (18)$$

where are:

- $c1(i)$ and $c2(i)$ offspring genes at i th position
- $r1(i)$ and $r2(i)$ parent genes at i th position
- for $i = 1, 2, \dots, 5$

Coefficients $k1$ and $k2$ are selected randomly from range $[0.9, 1.0]$, taking the limitation for range of values of individual genes into consideration. Individual genes, i.e. elements of vector $\mathbf{w}$ have to be in range $[-1, 1]$. Offsprings become members of population instead two individuals which have the largest value of fitness function. Then using Matlab's random generator one individual is selected and also one its gene which will be alter.

Fig. 4. Schematic diagram of optimization function by applying genetic algorithm

Mutation introduces new genetic material into population. Mutated gene of individual is formed as:

$$cl(i) = k \cdot r1(i)$$

(19)

where is:

- $cl(i)$ mutated gene of individual at i th position
- $r1(i)$ gene of individual at i th position before mutation
- $i \in [1, 5]$

Coefficient k is selected randomly from range $[0.9, 1.0]$ with the same reason as crossover operation.

After mutation convergence criteria is examined. New population, after application of genetic operators, becomes initial population for the next generation. The worst individual in the current generation is replaced with a best one in the previous generation. Then, convergence criteria is tested and iterative procedure of applying genetic operators is repeated until the satisfaction of convergence criteria. Simulation have been shown that genetic algorithm converges after 50 generations. Satisfying of convergence criteria represents the end of genetic algorithm. And a result is the the fittest individual (with an optimal fitness function), i.e. optimal weighting vector $\mathbf{w}$ which in a linear combination with vector $\mathbf{y}$ gives the minimal error of current pixel value estimate.

4. Simulation results and comparison with competitive filters

Performance of PBM filter has been evaluated by simulations on variety of 256×256 test images. Within this study 5 different images (*Bridge, Lena, Camera, Apples, Parrot*) was analyzed. In simulations 3×3 filter window is used, and then $W = 2n + 1 = 9$, i.e. $n=4$. Therefore, $k = 0, 1, 2, 3$ and the difference vector space is 4-dimensional. Quantization levels are determined experimentally and they are shown in Table 1. Each dimension has $L = 6$ intervals in the range of $[0, 255]$ and the total number of regions is $M = L^n = 6^4 = 1296$. Simulations have been fulfilled on variety of images which have been corrupted with different type of noise:

- 1. IMPULSE NOISE
- 2. GAUSSIAN NOISE
- 3. MIXED GAUSSIAN AND IMPULSE NOISE

For a corruption by impulses with a noise ratio p , only p of total pixels are replaced with impulses and the others keep noise-free. Here, both fixed-valued and random-valued impulses are used. For gray-scale images, noise intensity in the first case corresponds to 0 or 255 with equal probability (i.e. $p/2$), while in the second case, it is uniformly distributed within $[0, 255]$. Another type of noise assumed is the zero-mean additive Gaussian noise with standard deviation σ . The mixed noise is also used by adding Gaussian noise and the fixed-valued impulse noise together.

Fig. 5. Results of recursive PBM filtering for different images corrupted by fixed-valued impulses with $p=20\%$

Fig. 6. Results of recursive PBM filtering for different images corrupted by random-valued impulses with $p=20\%$

In this study PBM filter is implemented recursively, because of better results in SNR (signal-noise ratio) obtained by the recursive filtering over nonrecursive design. Results of recursive filtering are shown on Figure 5 and Figure 6 for various images corrupted by fixed-valued and random-valued impulses with $p=20\%$.

Noticable gain has been achieved by recursive filtering and its value is image dependent. It is interesting to note that recursive filtering converges after two iterations of training. This can also see on Figure 5 and Figure 6.

4.1 Comparison with competitive filters

The performance of PBM filter is examined through comparison with competitive filters based on median filtering, including median filter (MED), CWM filter ($\omega = 3$) in recursive

Fig. 7. Performance of different filters in filtering the Bridge image corrupted by fixed-valued impulses

Fig. 8. Performance of different filters in filtering the Bridge image corrupted by random-valued impulses

and nonrecursive design, fuzzy median filter, florencio&schafer, SDROM (state dependent rank order median) filter in recursive and nonrecursive design and nonrecursive PBM filter. The comparative SNR results of filtering the *Bridge* image are given on Figure 7 and Figure 8 where p ranges from 10% to 30%. The PBM filter is trained assuming the corruption by 20% impulses, while the same type of noise is used in training and filtering.

FILTER	IMAGE				
	BRIDGE	LENA	CAMERA	APPLES	PARROT
MEDIAN	24.00	26.97	24.20	27.37	25.87
CWM	24.08	25.96	24.39	24.68	23.78
CWM RECURSIVE	24.53	27.36	25.30	27.81	26.03
FLORENCIO&SCHAFER	24.29	24.67	23.85	24.24	23.97
SDROM	25.43	27.55	24.54	27.77	26.20
SDROM RECURSIVE	26.71	29.51	25.48	30.41	28.35
FUZZY MEDIAN	27.42	28.70	25.67	28.25	28.41
PBM NONRECURSIVE	27.25	28.38.	25.35	29.00	27.05
PBM RECURSIVE	27.86	30.01	25.84	30.84	28.64

Table 2. Comparative results in SNR [dB] of filtering different images corrupted by fixed-valued impulse noise with $p=20\%$

FILTER	IMAGE				
	BRIDGE	LENA	CAMERA	APPLES	PARROT
MEDIAN	24.44	27.68	25.25	28.33	25.86
CWM	25.05	28.43	25.76	26.70	25.19
CWM RECURSIVE	25.26	28.37	26.06	28.62	26.59
FLORENCIO&SCHAFER	23.31	24.03	23.02	22.81	22.42
SDROM	25.81	28.60	25.56	28.64	26.77
SDROM RECURSIVE	26.13	29.56	25.27	30.45	27.95
FUZZY MEDIAN	26.03	27.53	24.67	27.79	27.05
PBM NONRECURSIVE	26.79	28.33	25.74	28.17	26.62
PBM RECURSIVE	27.41	29.91	26.46	30.70	28.47

Table 3. Comparative results in SNR [dB] of filtering different images corrupted by random-valued impulse noise with $p=20\%$

The PBM filter yields significant improvement over the other filters. Proposed filter has demonstrated excellent robustness in respect to impulse ratios, regardless of that used in the training.

Tables 2,3,4 and 5 present the comparative SNR results of removing different types of noise on various images. The proposed PBM filter yields better results in suppressing impulse

noise (fixed-valued and random-valued), Gaussian noise as well as mixed Gaussian and impulse noise.

FILTER	IMAGE				
	BRIDGE	LENA	CAMERA	APPLES	PARROT
MEDIAN	24.37	26.95	25.15	27.75	26.07
CWM	25.14	26.73	25.75	27.10	26.21
CWM RECURSIVE	25.11	27.20	26.00	27.78	26.53
FLORENCIO&SCHAFER	23.34	24.17	23.95	24.39	23.84
SDROM	22.90	23.76	22.84	24.07	23.44
SDROM RECURSIVE	22.95	23.90	22.76	24.14	23.51
FUZZY MEDIAN	22.75	22.80	22.06	22.81	22.67
PBM NONRECURSIVE	26.13	27.53	26.63	27.87	27.02
PBM RECURSIVE	25.96	27.71	25.57	28.11	27.11

Table 4. Comparative results in SNR [dB] of filtering different images corrupted by Gaussian noise with $\sigma = 20$

FILTER	IMAGE				
	BRIDGE	LENA	CAMERA	APPLES	PARROT
MEDIAN	22.29	23.91	22.57	24.10	23.66
CWM	21.21	22.05	21.32	21.81	21.91
CWM RECURSIVE	22.61	24.08	23.09	24.07	23.84
FLORENCIO&SCHAFER	20.12	20.64	20.13	20.62	20.84
SDROM	21.06	22.07	21.09	22.15	21.95
SDROM RECURSIVE	22.06	23.15	21.95	23.36	22.89
FUZZY MEDIAN	21.92	22.27	21.48	22.21	22.22
PBM NONRECURSIVE	23.73	24.30	23.11	24.50	24.25
PBM RECURSIVE	24.09	25.87	23.88	26.48	25.71

Table 5. Comparative results in SNR [dB] of filtering different images corrupted by mixed Gaussian ($\sigma = 20$) and impulse noise (p=20%)

The performance of PBM filter proposed in this study in reducing different types of noise can also notice on Figure 9, Figure 10, Figure 11 and Figure 12. PBM filter yields better image quality with respect to noise suppression and detail preservation than the other methods, by producing a visually more pleasing image.

Fig. 9. Comparison of the restoration performance of different filtering type for the Bridge image corrupted by 20% fixed-valued impulses

Fig. 10. Comparison of the restoration performance of different filtering type for the Parrot image corrupted by 20% random-valued impulses

Fig. 11. Comparison of the restoration performance of different filtering type for the Lena image corrupted by Gaussian noise with $\sigma = 30$

Fig. 12. Comparison of the restoration performance of different filtering type for the Apple image corrupted by mixed Gaussian noise ($\sigma = 20$) and impulse noise ($p=20\%$)

5. Conclusion

In this chapter an adaptive filtering using genetic algorithm in the training process is proposed. In the proposed filtering scheme, at each location, observed vector is classified into one of M exclusive partitions, and an optimal particular filtering operation is then activated. Optimal weighting vector of each partition is derived using genetic algorithm in training the filter over a reference image. Recursive implementation of the proposed filter is applied and shown to produce better results than its nonrecursive design. In the simulations over variety of images, the proposed PBM filter using genetic algorithm in training process has demonstrated better results in suppressing different types of noise than competitive filters in terms of the SNR [dB] as well as the perceived image quality.

In the future, research will be directed to the forming of reference image which will provide more information for the genetic algorithm in the training process and improve the filtering process.

6. References

- Brezocnik, M. (2000). Uporaba genetskega programiranja v inteligentnih proizvodnih sistemih, Fakulteta za strojninstvo, ISBN 86-435-0306-1, Maribor
- Chen, T. & Wu, R. H. (2001). Application of Partition-Based Median Type Filter for Suppressing Noise in Images, *IEEE Trans. on Image Processing*, Vol. 10, (June 2001), pp. 829-836
- Ko, S. J. & Lee, Y. H. (1991). Center Weighted Median Filters and Their Application to Image Enhancement, *IEEE Trans. on Circuits System*, Vol. 38, (September 1991). pp. 984-993
- Koza, J. R. (1992). *Genetic Programming*, The MIT Press, Massachusetts
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, ISBN 3-540-60676-9, Berlin
- Pitas, I. & Venetsanopoulos, A. (1991). *Nonlinear Digital Filters-Principles and Applications*, Kluwer International Series, ISBN 0-7923-9049-0, Massachusetts
- Pitas, I. & Venetsanopoulos, A. (1992), Order Statistics in Digital Image Processing, *Proceedings of the IEEE*, Vol. 80, No. 12, (December 1992), pp. 1893-1921
- Sovilj-Nikic, I.; Sovilj, B. & Brezocnik, M. (2008). Application of Genetic Algorithm in Analysis of Influence of Gear Hob Geometric Parameters on the Tool Life, *Proceedings of ICMEN*, pp. 145-154, ISBN 978-960-243-649-3, Kallithea, October 2008, Aristoteles University of Thessaloniki, Thessaloniki
- Sovilj-Nikic, S. & Sovilj-Nikic, I. (2007). Application of Genetic Algorithm in Median Filtering, *Proceedings of IMCSIT*, pp. 127-139, ISSN 1896-7094, Wisla, October 2007, Polish Information Processing Society in cooperation with IEEE Computer Society (Poland Chapter) and Systems Research Institute of the Polish Academy of Science, Katowice

Advances in Computer Science and IT

Edited by D M Akbar Hussain

ISBN 978-953-7619-51-0

Hard cover, 420 pages

Publisher InTech

Published online 01, December, 2009

Published in print edition December, 2009

The book presents some very interesting and excellent articles for this divergent title. The 22 chapters presented here cover core topics of computer science such as visualization of large databases, security, ontology, user interface, graphs, object oriented software developments, and on the engineering side filtering, motion dynamics, adaptive fuzzy logic, and hyper static mechanical systems. It also covers topics which are combination of computer science and engineering such as meta computing, future mobiles, colour image analysis, relative representation and recognition, and neural networks. The book will serve a unique purpose through these multi-disciplined topics to share different but interesting views on each of these topics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Sandra Sovilj-Nikic (2009). Image Quality Enhancement By Applying Genetic Algorithm In Median Filtering, *Advances in Computer Science and IT*, D M Akbar Hussain (Ed.), ISBN: 978-953-7619-51-0, InTech, Available from: <http://www.intechopen.com/books/advances-in-computer-science-and-it/image-quality-enhancement-by-applying-genetic-algorithm-in-median-filtering>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen