Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 127 – 139
ISSN 1896-7094

Application of Genetic Algorithm in Median Filtering

Sandra Sovilj-Nikic, Ivan Sovilj-Nikic

University of Novi Sad, Faculty of Technical Sciences
Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia and Montenegro
sandrasn@eunet.yu

Abstract. Images are often corrupted by impulse noise due to errors generate d in noisy sensors or communication channels. Two types of impulse noise can be defined: 1) fixed-valued and 2) random-valued. 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. In this paper an adaptive filtering using genetic algorithm is proposed. In the simulations over various images, the proposed partition based median (PBM) filter using genetic algorithm in training have demonstrated better results in noise suppressing then competitive filters based on median filtering in terms of SNR(dB) as well as the perceived image quality.

1 Introduction

Images are often corrupted by impulse noise due to errors generate d 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 called salt&peper noise and it appears as black and/or white impulses on the image. Another type of noise generated after incorrect decoding of binary represented image and 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. The family of order statistics based filters is very rich, where the median filter is the best known. Median filter and its modification [11] have shown good efficiency in suppressing impulse noise and capability of preserving image edges. Nevertheless, because the median filters are location-invariant, 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 to exceed disadvantages of median filtering. In this paper an adaptive filtering using partition based median (PBM) filter is proposed. With proposed PBM filter, at each location, observed vector is classified into one of M exclusive partitions, and a 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. Optimal weights are derived to minimize the total square error of estimation at each location. In the simulation over variety of images, the proposed PBM filter using genetic algorithm in training on reference image has demonstrated better results in noise suppressing then competitive filters based on median filtering in terms of the SNR (signal/noise ratio) as well as the perceived image quality. The proposed filter outperforms other median based filters in removing both fixed-valued and random-valued impulses

2 Definition of PBM filter

Schematic diagram of PBM filter is shown on Figure 1 [4].

Let $C = \{(c1, c2) \mid 1 \le c1 \le P,\ 1 \le c2 \le Q\}$ 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 $W = 2n+1$ simetrically surrounding the current pixel, and

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

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

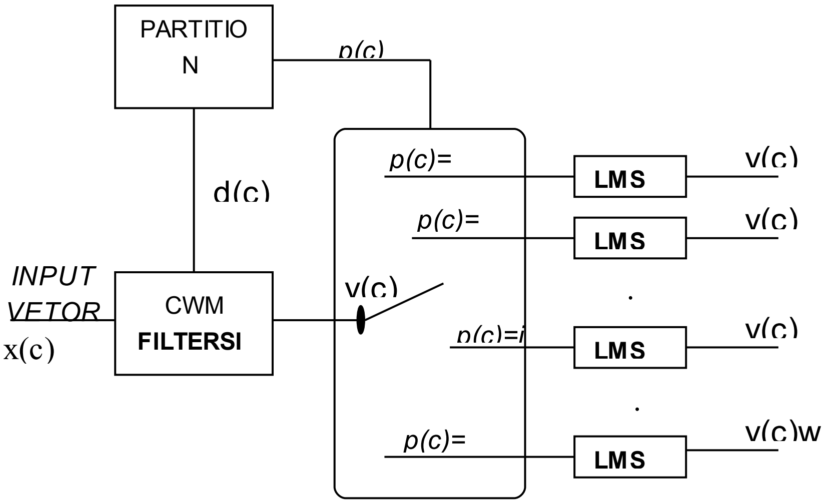

Fig. 1. - Schematic diagram of PBM filter

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

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

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

$$x_{\omega}(c) = [x_1(c),\, x_2(c),\, x_3(c),\ldots,\, x_n(c),\, \omega \quad x_{n+1}(c),\, x_{n+2}(c),\ldots,\, x_{2n+1}(c)] \tag{3}$$

Operator $\diamond$ denotes repetition operation. Two limit cases exist in the CWM filtering. Output of CWM filter y_0 (*k=0, w=1*) corresponds to the output of standard median filter, and for $k \geq n$ CWM filter is an identity filter $y_k = x(c)$. In the proposed filtering scheme k lies in $[0,\, n-1]$.

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

$$d_k(c) = |y_k(c) - x(c)| \tag{4}$$

where $k = 0,\, 1,\, \ldots,\, n-1$ and $\mathrm{d}_k(c) \leq \mathrm{d}_{k\text{-}1}(c)$.

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:

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

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

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

Vector $\mathbf{w}_{p(c)} = \mathbf{w}_i = \lfloor w_{i,o}, w_{i,1}, \dots, w_{i,n} \rfloor$ 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 satisfies a location-invariance constraint:

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

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

If $x(c) = median\,(\mathbf{x(c)})$, 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:

$$\hat{x}(c) = y(c) \cdot w_i^T = e_{n+1} \cdot x(c) \cdot w_i^T = x(c) \quad (8)$$

2.1 Region partitioning

At each location $\mathbf{c}$, difference vector is obtained by applying (2.4) as follows:

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

The difference vector space is divided into M exclusive regions, $\{\Omega_i : i = 1, 2, \dots, M\}$. The pertition index for each input vector $\mathbf{x(c)}$ is given such that $p(c) = i$ for $\mathbf{d(c)} \in \Omega_i$. 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 is applyed in this paper. Quantization levels are determined experimentally and remain fixed through simulations.

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

$$q_{k,v} < q_{k,v+1} \quad , \qquad 0 \leq v \leq L-1 \quad (10)$$

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

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

In the other words scalar quantization is described as:

$$\mathbf{Q}(\mathbf{d(c)}) = \mathbf{z}_i \tag{11}$$

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

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 [4]

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

	v						
k	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

2.2 LMS weights optimization

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

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

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 imto consideration we can write:

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

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

Following (5) we have:

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

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

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

2.3 Recursive implementation

In this paper PBM filter is implemented recursively because of obtained results through 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 in recursive filtering depends on past filter outputs, with input

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

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 [4].

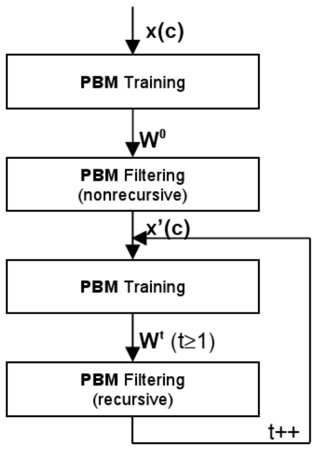

Fig. 2. Schematic diagram of recursive PBM filter

3 Optimization using genetic algorithm

Evolution programs are methods for solving optimization problems which based on the principles of Darwin's evolution theory, i.e. natural selection and survival of the fittest. 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.

Taking the type of optimization problem in this paper into consideration, genetic algorithm has been choose n to solve the problem.

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 and estimate error of current pixel should be minimal for each partition.

3.1.1 O ptimization 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. 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 is compromise solution for the problem in this paper. 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

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

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:

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

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 individuals until convergence criteria would not be satisfied. 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 paper tournament selection is applied to select potential parents. Using Matlab's random numbers generator two parents are choosed randomly and they create their two offsprings in the crossover operation. Crossover enables to exchange information between different potential solution. In this paper decade encoding and real arithmetic crossover is applyed.

Crossovered offspring genes are formed as:

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

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 $i = 1, 2, \ldots, 5$

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

$$c1(i) = k \cdot r1(i) \tag{19}$$

where is:

$c1(i)$ mutated gene of individual at *i* th position

$r1(i)$ parent gene at *i* th position

$i \in [1, 5]$

Coefficient *k* is selecting 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 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 operations is repeated until convergence criteria would not be satisfied. 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 **w** which in a linear combination with vector **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. In simulations 3×3 filter window is used, and then $W = 2n + 1$, 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 2.1. Each dimension has $L = 6$ intervals in the range of $[0, 255]$ and the total number og 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:

- IMPULSE NOISE
- GAUSSIAN NOISE
- 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.

In this paper 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 3 for various images corrupted by fixed-valued impulses with *p* =20%.

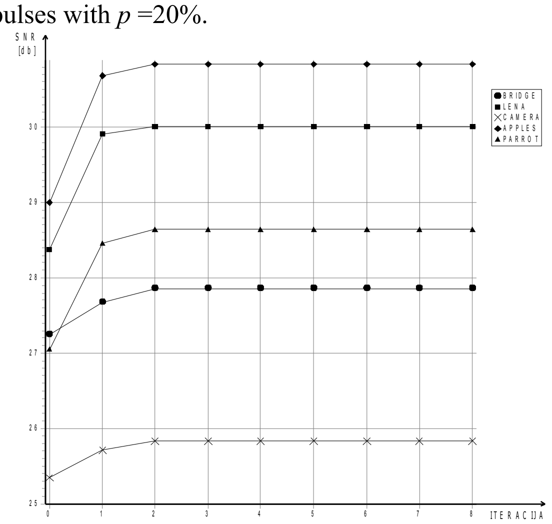

Figure 3 – Results of recursive PBM filtering for different images corrupted by fixed-valued impulses with p=20%

Noticable gain has been achieved by recursive filtering and its value is image depen dent. It is interesting to note that recursive filtering converges after two iterations of training. This can see on Figure 3, too.

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 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 4 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.

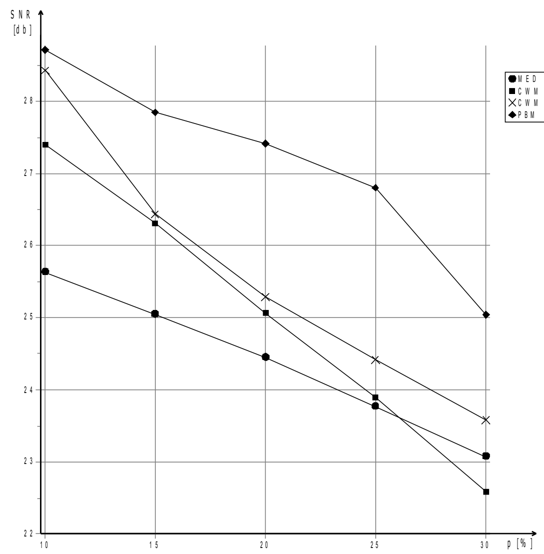

Figure 4 – Performance of different filters in filtering the Bridge image corrupted by random-valued impulses with p=20%

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

Tables 2 and 3 present the comparative SNR results of removing different type of noise on various images. The PBM filter yields better results in suppressing impulse noise as well as Gaussian noise.

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	24.00	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 NON-RECURSIVE	27.25	28.38.	25.35	29.00	27.05
PBM RECURSIVE	27.86	30.01	25.84	30.84	28.64

Table 3. - Comparative results in SNR (dB) of filtering different images corrupted by Gaussian noise with $\sigma = 20$

FILTE R	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 RE CUR-SIVE	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 RE CUR-SIVE	22.95	23.90	22.76	24.14	23.51
FUZZY MEDIAN	22.75	22.8 0	22.0 6	22.81	22.67
PBM NONRE-CURSIVE	26.13	27.53	26.63	27.87	27.02
PBM RECURSIVE	25.96	27.71	25.57	28.11	27.11

The performance of PBM filter in reducing fixed-valued impulse noise can notice on Figure 5. PBM filter yields better subjective quality with respect to noise suppression and detail preservation than the other methods, by producing a visually more pleasing image.

Figure 5. - Comparison of the restoration performance of different filtering type for the Bridge image corrupted by 20% fixed-valued impulses

5 Conclusion

In this paper an adaptive filtering using partition based median filter is proposed. With proposed PBM filter, at each location, observed vector is classified into one of *M* exclusive partitions, and a 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 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.

References

1. Balič J., *Manufacturing Systems for the Third Millenium,* University of Maribor, Faculty of Mechanical Engineering, Laboratory for Intelligent Manufacturing Systems, Maribor, 2001.

2. Brezočnik M., *Uporaba genetskega programiranja v inteligentnih proizvodnih sistemih,* Univerza v Mariboru, Fakultet za strojništvo, Maribor, 2000.
3. Brezočnik M., Balič J., *Comparison of Genetic Programming with Genetic Algorithm,* University of Maribor, Slovenia
4. Chen T., Wu R. H., *Application of Parition-Based Median Type Filter for Suppressing Noise in Images,* IEEE Trans. Image Processing, vol. 10, pp. 829-836, June 2001.
5. Florencio D. A. F., *Decision-based median filter using local statistics*, Proc. SPIE Symp.Visual Communication Image Processing, vol. 2038, pp. 268-275, September 1994.
6. Herzeg D., *Modelovanje frekvencijski zavisnih impedansi uz pomoć genetskog algoritma,* Novi Sad, 2001.
7. Ko S. J., Lee Y. H., *Center weighted median filters and their application to image enhancement,* IEEE Trans. Circuits System, vol. 38, pp. 984-993, September 1991.
8. Kotropoulos C., Pitas I., *Nonlinear Model-Based Image/Video Processing and Analysis,* John Wiley&Sons, 2001.
9. Kuzmanović S., Dautović S., *Genetički algoritam za određivanje minimuma udruživanja na grafu,* Fakultet tehničkih nauka, Univerzitet u Novom Sadu, 1997.
10. Novak L., Dautović S., Kuzmanović S., *Genetski program za približno rešavanje problema nalaženja minimum udruživanja na grafu,* Fakultet tehničkih nauka, Univerzitet u Novom Sadu, 1997.
11. Pitas I., Venetsanopoulos A., *Nonlinear Digital Filters-Principles and Applications,* Kluwer International Series, 1991.
12. Pitas I., Venetsanopoulos A., *Order Statistics in Digital Image Processing,* Proceedings of the IEEE, vol. 80, no. 12, December 1992.