ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 2003
Seria: AUTOMATYKA z. 138 Nr kol. 1581

Sebastian BUDZAN
Politechnika Śląska

ON THE NONPARAMETRIC NOISE REDUCTION IN COLOR IMAGES

Summary. In work question of efficiency of filtration the noise was talked over in color images at utilization of vectorial median. In this paper it was concetrated itself on utilization in filtration proces of Parzen estimator. In first part of this paper general questions were talked over – distance functions, ordering schemes. In second part Parzen estimator was talked over as well as leaning on him filtration algorithm. In last part, the experiments have been presented

NIEPARAMETRYCZNE METODY REDUKCJI SZUMU W OBRAZACH BARWNYCH

Streszczenie. W pracy omówiono zagadnienie efektywności filtracji szumów w obrazach barwnych przy wykorzystaniu mediany wektorowej. Skupiono się na omówieniu wykorzystania w procesie filtracji estymatora Parzena. W pierwszej części pracy omówiono ogólne zagadnienia dotyczące filtracji medianowej – schematy szeregowania, funkcje dystansu. W drugiej części omówiono estymator Parzena oraz oparty na nim algorytm filtracji. W ostatnim fragmencie zaprezentowano część badań przeprowadzonych z wykorzystaniem powstałego algorytmu.

1. Introduction

Noise removal is an important task in image processing. In this paper introduced is one from methods of image filtration – leaning on Parzen estimator. This algorithm is based on the maximization of the similarities between pixels in the filtering window. The new method removes the noise component, while adapting itself to the local image structures. The proposed algorithm eliminates impulsive noise from color images, while preserving edges and fine image details. The algorithm can be considered as a modification of the standard vector median filter driven by the smoothing kernels, used in the nonparametric density estimation. Experimental results show, that this approach can be equally well applied as different image filtering methods.

2. Standard Filters [13, 17]

The most nonlinear – multichannel filters are based on the ordering on vectors in the moving window. The output of these filters is defined as the lowest ranked vector according to a specific vector ordering technique.

The color images are represented by three components the RGB space. Let **X** denote a p-dimensional random variable, for example $X=[X_1, X_2,..., X_p]^T$. Now, let $x_1, x_2, ..., x_n$ be n random samples from multivariate X. Each one of the x_i is a p-dimensional vector of observations $x_i=[x_i^1, x_i^2,..., x_i^p]^T$, where p is the dimension (in our case p=3) in image filtering. The goal is to arrange the n values $(x_1, x_2, ..., x_n)$ in some sort of order. There are several ways to order the data, based on the so called *sub-ordering* principles.

The *sub-ordering* principles can be used to rank any kind of multivariate data, but for image processing the ordering scheme should be geared towards the ordering of color image vectors. Such an ordering scheme should satisfy the following criteria:

- The proposed ordering scheme should be useful from a robust estimation perspective.
- The proposed ordering scheme should take into consideration the type of multivariate data being used.

Based on these principles, we can propose:

$$R_a(x_i) = \sum_{j=1}^{n} R(x_i, x_j) \tag{1}$$

where $R(x_i, x_j)$ is the distance beetwen x_i and x_j. The scalar quantities $R_{ai}=R_a(x_i)$ are then ranked in order of magnitude and the associated vectors are ordered correspondingly:

$$R_{a1}; R_{a2}; \ldots; R_{an} \Rightarrow x_{(1)};\ x_{(2)}; \ldots;\ x_{(n)} \tag{2}$$

Using this ordering scheme proposed here, the ordered $x_{(i)}$ have a one-to-one relationship with the original samples x_i. However, the concept of input ordering, initially applied to scalar quantities is not easily extended to multichannel data, since there is no universal way to define ordering in vector spaces.

Distance functions are often utilized to order vectors. The most commonly used measure to quantify distance between two p-dimensional signals is the generalized Minkowski metric. It is defined for two vectors x_i and x_j as follows:

$$d_\gamma(i,j)=\left(\sum_{k=1}^{p}\left|\left(x_i^k-x_j^k\right)\right|^\gamma\right)^{\frac{1}{\gamma}} \tag{3}$$

Three special cases of the Minkowski metric. Metric are of particular interest. Namely:

1. The city-block distance corresponding to $\gamma=1$. In this case, the distance between the two p-dimensional vectors is considered to be the sum of the absolute diferences of the components values between their components:

$$d_1(i,j)=\sum_{k=1}^{p}\left|x_i^k-x_j^k\right| \tag{4}$$

2. The Euclidean distance corresponding to $\gamma=2$. In this case, the distance between the two p-dimensional vectors is set to be a square root of the sum of the square of diferences of their components:

$$d_2(i,j)=\sqrt{\sum_{k=1}^{p}\left(x_i^k-x_j^k\right)^2} \tag{5}$$

3. The chess-board distance corresponding to $\gamma=\propto$. In this case, the distance between two p-dimensional vectors is equal to the maximum difference bettwen their components:

$$d_\infty(i,j)=\max_k\left|x_i^k-x_j^k\right| \tag{6}$$

The orientation difference between two vectors can also be used as their distance measure. This so-called *vector angle criterion* is used by the *VectorDirectionalFilters* - VDF [10] to remove vectors with atypical directions. The *Basic Vector Directional Filter* – BVDF [2] is a ranked-order, nonlinear filter which parallelizes the VMF operation. However, a distance criterion, different from the *L*1, *L*2 norms used in VMF is utilized to rank the input vectors. The output of the BVDF is that vector from the input set, which minimizes the sum of the angles with the other vectors. In other words, the BVDF chooses the vector most centrally located without considering the magnitudes of the input vectors. To improve the efficiency of the directional filters, a new method called *Directional-Distance Filter* – DDF was proposed [18]. This filter retains the structure of the BVDF but utilizes a new distance criterion to order the vectors inside the processing window.

All standard filters detect and replace well noisy pixels, but their property of preserving pixels which were not corrupted by the noise process is far from the ideal. In this paper we show the construction of a efficient and fast filter which removes noisy pixels, but has the ability of preserving original image pixel values.

3. New filter using Parzen estimator

3.1. Parzen Theory

The Parzen-window approach to estimating densities can be intoduced by temporarily assuming that the region R_n is a p-dimensional hypercube. If h_n is the lenght of an edge of that hypercube, then its volume is given by

$$V_n = h_n^p \tag{7}$$

Window function K is defined as:

Thus, K(u) defines a unit hypercube centered at the origin. It follows that

$$K(u) = \begin{cases} 1 & |u_j| \le \frac{1}{2} \quad j = 1,\dots,p \\ 0 & otherwise \end{cases} \tag{8}$$

is equal to unity if x_i falls within the hypercube of volume V_n centered at x, and is zero

$$K\left(\frac{(x - x_i)}{h_n}\right) \tag{9}$$

otherwise. Hence the number of samples in this hypercube is given by

where is n-th estimate to p(x).

$$k_n = \sum_{i=1}^{n} K\left(\frac{(x - x_i)}{h_n}\right) \tag{10}$$

$$p_n(x) = \frac{k_n / n}{V_n} \tag{11}$$

And when we substitute equation (10) with equation (11) we obtain the estimate

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} K\left(\frac{(x - x_i)}{h_n}\right) \tag{12}$$

Here we give some examples using the window function

Frequently used is Gaussian function.

$$K(\xi) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\xi^2\right) \tag{13}$$

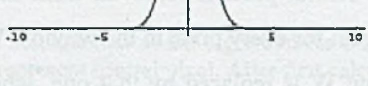


Fig. 1. Gaussian function
Rys. 1. Funkcja Gaussa

We have some windows function, like triangle, cosinus, uniform functions.

3.2. *Smoothing coefficients*

Parzen estimator with normal kernel function [6]:

$$p_n(x,h) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\left(h\sqrt{2\pi}\right)^p} \exp\left(\frac{\|x - x_i\|^2}{2h^2}\right) \tag{14}$$

where h is the smoothing parameter.

A) Silverman coefficient [8].

$$h_i = \Theta * \sigma * n^{-\frac{1}{5}} \tag{15}$$

where θ - scaling coefficient,

σ - standard deviation,

n – number of the pixels.

B) From maximum likelihood principle – MLP [6], we can obtain:

$$h_i = \Theta * \sqrt{\frac{1}{pn}\sum_{j=1}^{n}\left\| x_j^* - x_j \right\|^2} \tag{16}$$

where x^*_j – represents the nearest neighbour of the sample x_j,

n – number of the pixels,

θ – scaling coefficient.

3.3. Practical side of algorithm

Let **X** represents a multichannel image and **W** be a window of finite size **n**.

In the construction of our filter, the central pixel in the window **W** is replaced by that one, which maximizes the sum of similarities between all its neighbors.

In first step we must count parzen estimator – according to equation [14], for every pixel in window **W**, in next step the central pixel is removed from the window, and than calculation parzen estimator follows again for every pixel in the window **W**.

The central pixel in the window **W** is replaced by that one, which parzen estimator value is smallest among all values in window **W** and simultaneously parzen estimator value is larger from parzen estimator value for central pixel in the Window **W**.

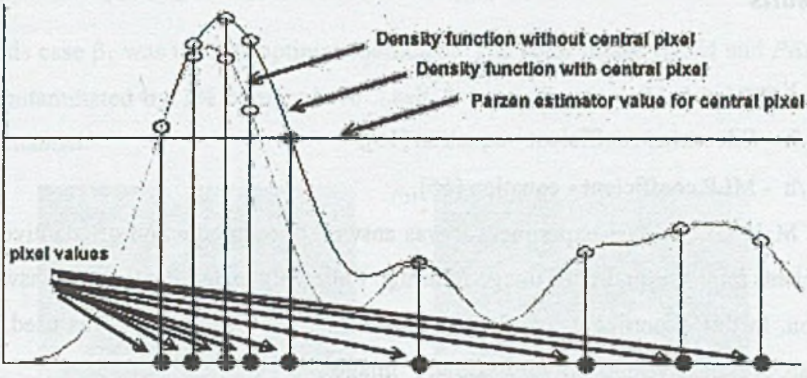


Fig. 2. Ilustration of the construction of the algorithm for 8 neighbourhood case. On bottom of the figure are placed pixel's values: n = 9 values. Density function was counted for window with central pixel and without central pixel. For this case central pixel became removed and replaced by first pixel from left

Rys. 2. Ilustracja algorytmu w przypadku 8 sąsiadów. Na dole rysunku widoczne 9 pikseli. Funkcja gęstości obliczona dla okna z pikselem centralnym oraz bez piksela centralnego. W tym przypadku piksel centralny zamieniany jest przez piksel nr.1 – pierwszy od lewej

X_1	X_2	X_3
X_8	X_0	X_4
X_7	X_6	X_5

	X_2	X_3
X_8	X_1	X_4
X_7	X_6	X_5

Fig. 3. Ilustration of the construction of the algorithm for 8 neigbourrhood case. Vector X_i represent pixel no. i, X_0 represent central pixel. After first calculation of the Density function for every pixel in the window $p(X_i)$, i=0,..., n - with central pixel, and second calculation of the Density function for every pixel in the window $p(X_i)$, i=0,..., n-1 – without central pixel, checked condition, for which pixel parzen estimator value is smallest among all values and simultaneously parzen estimator value is larger from parzen estimator value for central pixel, and than central pixel became removed and replaced by pixel, which it is represented through vector X_1 – in this case

Rys. 3. Ilustracja algorytmu w przypadku 8 sąsiadów. Wektor X_I reprezentuje piksel nr. I, X_0 reprezentuje piksel centralny. Po pierwszym obliczeniu funkcji gęstości dla każdego piksela w oknie – uwzględniając piksel centralny, oraz po drugim obliczeniu funkcji gęstości dla każdego piksela w oknie – bez piksela centralnego, sprawdzany jest warunek, dla którego piksela wartość estymatora Parzena jest najmniejsza i jednocześnie estymator Parzena jest większy od estymatora Parzena dla piksela centralnego, i wówczas piksel centralny jest usuwany i zamieniany na piksel, który jest reprezentowany przez wektor X_1 – w tym przypadku

4. Results

Keys:

β_1=1/h - Silverman coefficient - equation [15],

β_2=1/h - MLP coefficient - equation [16],

Main task in ouer experiment it was answer, does intoduction of adaptive smoothing coefficients improve quality of image filtering. Following experiments gives answer onto this question. In this experiment the „LENA" and the „PEPPERS" image was used, but in this paper are presented results only for "LENA" image.

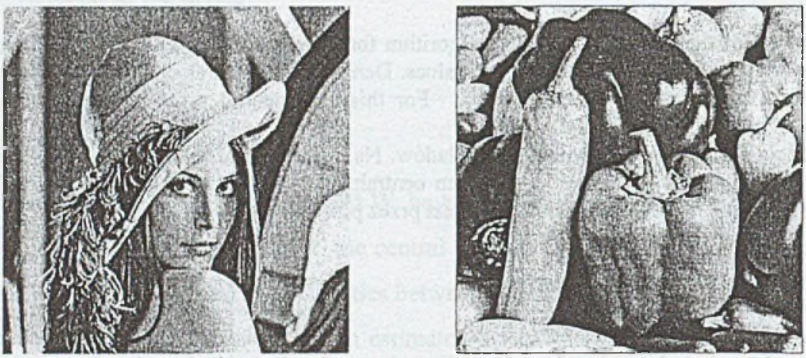


Fig. 4. Testing color images; 512x512 pixels
Rys. 4. Barwne obrazy testowe; 512x512 pikseli

The performance of the new algorithm was compared with the standard procedures of noise reduction used in color image processing. The root of the mean squared error (RMSE), peak signal to noise ratio (PSNR) have been used as quantitative measures of quality for evaluation purposes. Here

$$MSE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\left\| X(i,j) - \hat{X}(i,j) \right\|^2}{N \bullet M} \quad (17)$$

$$RMSE = \sqrt{MSE} \quad (18)$$

$$PSNR = 20\log\left\{\frac{255}{RMSE}\right\} \quad (19)$$

where N and M are the image dimensions, X(i; j) and ^X(i; j) denote the original.

4.1. Determination of the optimal norm

In this case β_1 was used as optimal coefficient. The color image *LENA* and *PEPPERS* has been contaminated by 1% of impulsive " salt & pepper" noise added independently to each RGB channel.

(a)  (b) 

Fig. 5. (a) – original image, (b) – after corruption
Rys. 5. (a) – obraz oryginalny, (b) – po dodaniu szumu

Table 1

PSNR values

Beta	L1	L2	Lmax
0,011	37,85	40,89	41,92
Beta adapt. 1	36,97	39,45	40,75
Beta adapt. 2	37,31	39,38	40,92

Table 2

RMSE values

Beta	L1	L2	Lmax
0,011	3,281	2,312	2,051
Beta adapt. 1	3,612	2,713	2,336
Beta adapt. 2	3,471	2,732	2,291

4.2. Dependence of the noise reduction efficiency on the percentage of impulsive noise

Table 3

PSNR & RMSE values

Noise	Beta adapt. 1		Beta adapt. 2	
	PSNR [dB]	RMSE	PSNR [dB]	RMSE
1%	40,751	2,33	40,921	2,29
2%	39,371	2,74	39,227	2,78
3%	38,062	3,18	37,667	3,33
4%	36,712	3,72	36,244	3,92
5%	35,571	4,24	34,924	4,57
6%	34,206	4,96	33,666	5,28
7%	32,909	5,74	32,334	6,16
8%	31,587	6,71	31,047	7,14
9%	30,202	7,87	29,763	8,28
10%	28,934	9,11	28,581	9,48

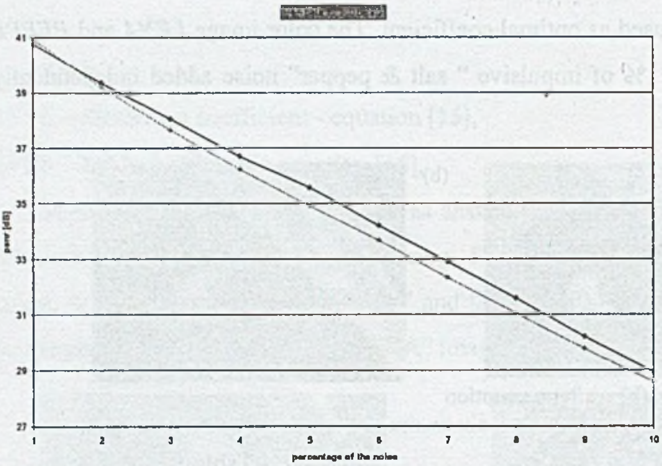


Fig. 6. PSNR values
Rys. 6. Wartości współczynnika PSNR

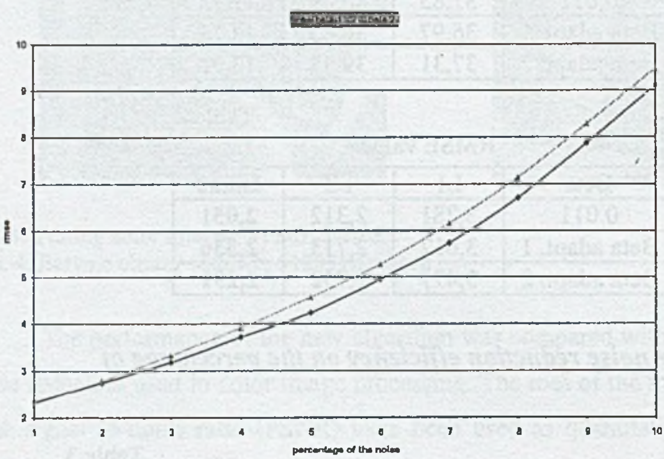


Fig. 7. RMSE values
Rys. 7. Wartości współczynnika RMSE

In this experiment was studied what has degree of noise onto quality of filtration process. Test images has been contaminated by 1% to 10% of impulsive and color "salt & pepper" noise added independently to each RGB channel.

4.3. Dependence of the noise reduction efficiency on the number of iterations for color test image distorted by 5% impulsive noise

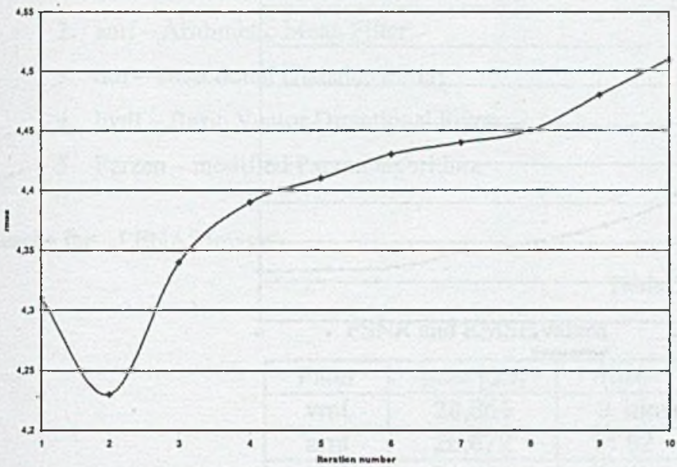


Fig. 8. RMSE values
Rys. 8. Wartości współczynnika RMSE

Table 4
RMSE & PSNR values
for iteration 1,...,10

Iteration	RMSE	PSNR [db]
1	4,31	35,454
2	4,23	35,613
3	4,34	35,371
4	4,39	35,262
5	4,41	35,223
6	4,43	35,201
7	4,44	35,189
8	4,45	35,169
9	4,48	35,165
10	4,51	35,163

On basis of *table 4* we can say, that we got best result for second iterations – test image distorted by 5 % impulsive color noise added independently to each RGB channel. For iteration 3 to 10 We get fall of PSNR value – *figure 9* and height of RMSE value – *figure 8.* On *figure 10* We can notice differences beetwen images after filtering in iterations – 1, 4, 7, 10.

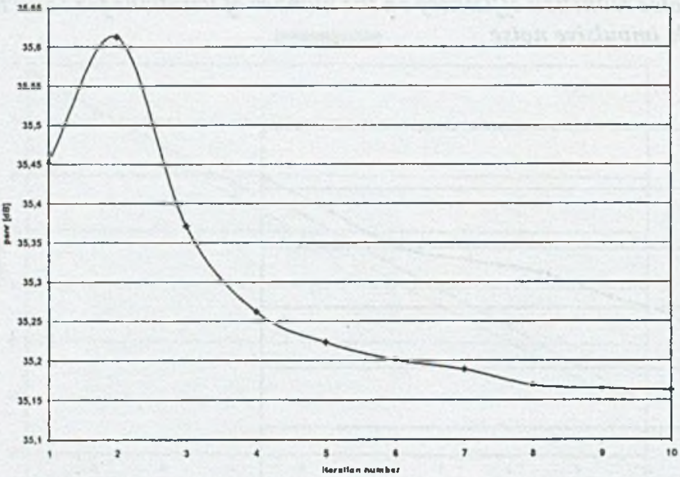


Fig. 9. PSNR values
Rys. 9. Wartości współczynnika PSNR

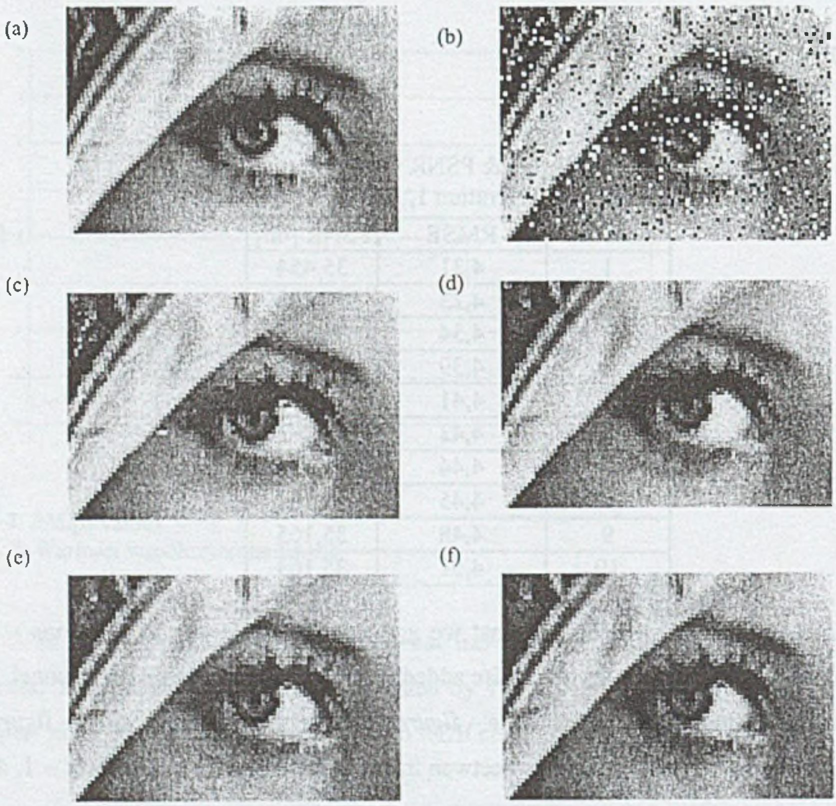


Fig. 10. (a) – original image, (b) – image after corruption – 5% salt & pepper, images after filtering: (c) – after 1 iteration, (d) – after 4 iterations, (e) – after 7 iterations, (f) – after 10 iterations
Rys. 10. (a) – obraz oryginalny, (b) – obraz po dodaniu szumu – 5% sól i pieprz, Obrazy po filtracji: (c) – po 1 iteracji, (d) – po 4 iteracjach, (e) – po 7 iteracjach, (f) – po 10 iteracjach

4.4. Comparison algorithm with another algorithms

Keys:

1. vmf – Vector Median Filter,
2. amf – Arithmetic Mean Filter,
3. ddf – Directional Distance Filter,
4. bvdf – Basic Vector Directional Filter,
5. Parzen – modified Parzen algorithm.

Results for „LENA" image

Table 5

PSNR and RMSE values

Filter	psnr [dB]	rmse
vmf	28,851	9,19
amf	26,672	11,82
ddf	29,204	8,83
bvdf	28,322	9,78
Parzen	35,454	4,31

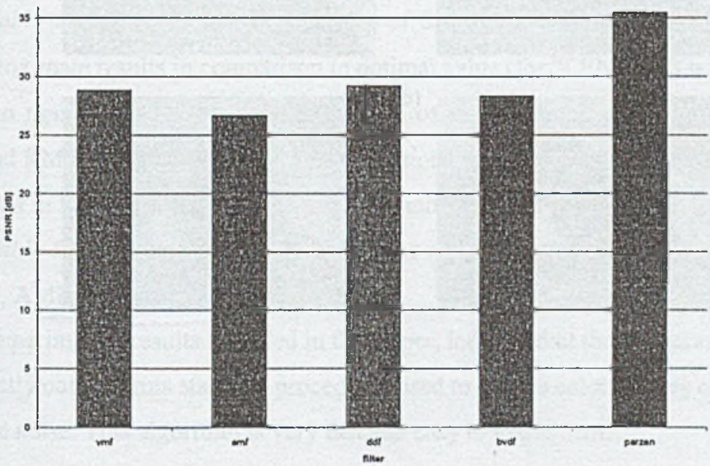


Fig. 11. PSNR values: VMF, AMF, DDF, BVDF, PARZEN (from left)
Rys. 11. Wartości współczynnika PSNR: VMF, AMF, DDF, BVDF, PARZEN (od lewej)

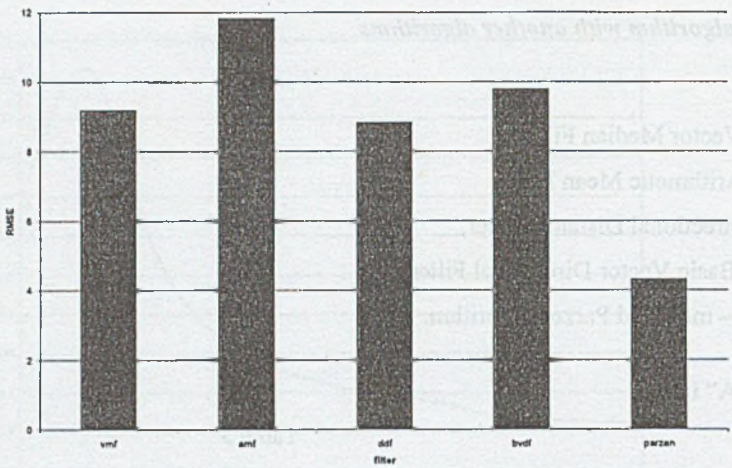
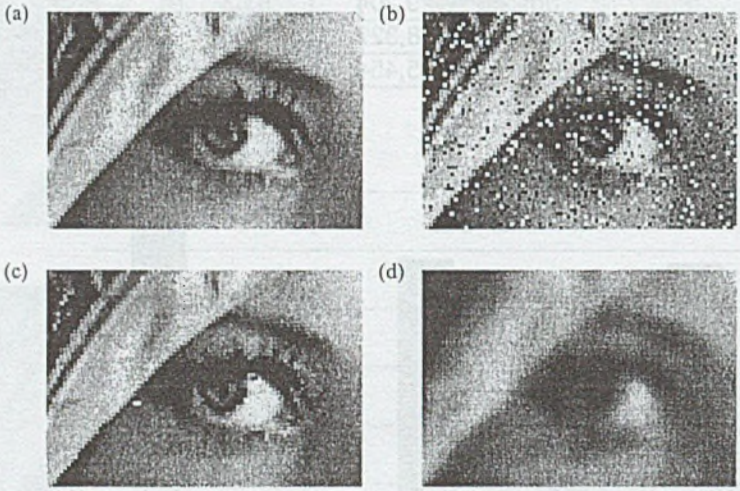


Fig. 12. RMSE values: VMF, AMF, DDF, BVDF, PARZEN (from left)
Rys. 12. Wartości współczynnika RMSE: VMF, AMF, DDF, BVDF, PARZEN (od lewej)



(e) (f)

Fig. 13. (a) – original image, (b) – image after 5% corruption, (c) – image after filtering - Parzen, (d) – image after filtering - VMF, (e) – image after filtering - BVDF, (f) – image after filtering - AMF

Rys. 13. (a) – obraz oryginalny, (b) – obraz po dodaniu 5% szumu, c) – obraz po filtracji – Parzen, (d) – obraz po filtracji – VMF, (e) – obraz po filtracji – BVDF, (f) – obraz po filtracji – AMF

5. Conclusions

All standard filters detect and replace well noisy pixels, but their property of preserving pixels which were not corrupted by the noise process is far from the ideal. In this paper we show the construction of a efficient and fast filter which removes noisy pixels, but has the ability of preserving original image pixel values.

Table 1 and *table 2* show, that use of the adaptive smoothing coefficients, it gives very approximate results in comparison to optimal value (for "LENA" - $\beta = 0{,}011$).

On *figure 8* and *9* interesting property of the presented method is shown. Both for PSNR and RMSE coefficients after a few iterations the filter reaches "constant" value, which means, that in further iterations no changes are introduced in to the image.

Table 5 and *figures 11 and 12* shows the efficiency of our method in comparison with the VMF, AMF, DDF, BVDF method.

Experimental results included in this paper, indicate that the nonparametric filter significantly outperforms standard procedures used to restore color images contaminated with impulsive noise. This algorithm is very fast and easy to implement.

REFERENCES

1. Silverman B.W.: Density estimation for statistics and data analysis. Published in Monographs on Statistics and Applied Probability, London: Chapman and Hall, 1986.
2. Astolla J., Haavisto P., Neuvo Y.: Vector median filters – Proceedings of the IEEE, Vol. 78, no. 4 April 1990.
3. Tapia R.A, Thompson J.R.: Nonparametric Probability Density Estimation 1978.
4. Pitas I., Venetsanopoulos A. N.: Nonlinear Digital Filters: Principles and Applications. Kluwer Academic Publishers, Boston, MA 1990.
5. Duda R.O., Hart P.E.,. Stork D.G.: Pattern classification. J.Willey & Sons Inc, New York 2001.
6. Kraaijveld Martin A.: A Parzen classifier with an improved robustness against deviations between training and test data. Pattern Recognition Letters 17 1996.
7. Babich Gregory A., Camps Octavia I.: Weighted Parzen Windows for Pattern Classifica-tion. Proceedings of the IEEE, Vol. 18, no. 5 May 1996.
8. Belanche L., Nebot A.: Density Estimation (UPC 2001/2002).
9. Plataniotis K.N., Androutsos D., Venetsanopoulos A.N.: Color image processing using adaptive vector directional filters. IEEE Trans. on Circuit and Systems-II, 1998.
10. Venetsanopoulos A.N.,. Trahanias P.E.: Vector Directional Filters – A new class of multichannel image processing filters. IEEE Trans. Image Processing 2(4), October 1993.
11. Kurzyński M: Rozpoznawanie obiektów – metody statystyczne. Oficyna Wydawnicza P. Wr., Wroclaw 1997.
12. Wu Ying, Li Bin, Ping Fan Yan: Nonparametric Density Estimation using Wavelet Transformation and Scale-space zero-crossing reconstruction. Proceedings of ICSP' 96.
13. Vardavoulia M.I,. Tsalides Ph.: A new vector median filter for colour image processing Pattern Recognition Letters 22 (2001).
14. Simon Tong, Daphne Koller: Restricted Bayes Optimal Classifiers. Proceedings of the 17-th National Conference on Artificial Intelligence.
15. Darryl Morrell: Statistical Pattern Recognition Lecture Note 6a: More. Nonparametric Estimation, 1996.

16. Herbin M,. Bonnet N, Vautrot P.: Estimation of the number of clusters and influence zones. Pattern Recognition Letters, Vol. 17, 1996.

17. Alparone L., Barni M, Bartolini F., Cappellini V.: Adaptively weighted vector-median filters for motion- fields smoothing University of Florence.

18. Karakos D.G.,. Trahanias P.E: Generalized multichannel image filtering structures, IEEE Trans. on Image Processing, 6, 7, 879-881, 1996.

Recenzent: Dr hab.inż. Marian WYSOCKI
Prof. Politechniki Rzeszowskiej

Wpłynęło do Redakcji dnia 04 lipca 2002 r.

Streszczenie

Do dzisiaj specjaliści z dziedziny przetwarzania obrazów zaprojektowali wiele filtrów, które mają za zadanie uzyskanie jak najlepszej jakości obrazu po filtracji. Metody te oparto początkowo na regułach szeregowania zwanych *Marginal Ordering*, a następnie *Vector Ordering*. Zwłaszcza te drugie zmieniły kierunek rozwoju technik redukcji szumu z obrazów opartych na szeregowaniu wektorów w oknie, które przesuwamy po macierzy obrazu. Na tej podstawie powstały filtry tj. *Vector Median Filter* – który stanowi podstawę dla pozostałych filtrów opartych na medianie wektorowej. W tych filtrach wykorzystuje się jako miarę dystansu pomiędzy pikselami, miary odległości między wektorami, które reprezentują piksele. Następnym krokiem w rozwoju filtracji obrazów było wykorzystanie jako miary dystansu pomiędzy pikselami, miary kąta pomiędzy wektorami, stąd powstał filtr *Vector Directional Filter*. Jakkolwiek filtry te dokonywały redukcji szumu z zakłóconego obrazu, to jednak powodowały duże zniszczenia obrazu, zauważalne szczególnie w zaniku krawędzi obiektów, bądź w niektórych przypadkach szczegóły zajmujące niewielkie obszary obrazu ulegały całkowitym zniszczeniu.

Wskutek braku materiału w postaci nowych propozycji redukcji szumów zaczęto szukać innych dróg. I to skierowało tok poszukiwań w kierunku metod statystycznych przetwarzania obrazów. Po badaniach filtrów opartych na estymowaniu wartości pikseli

obrazu, tj. estymatora Parzena, estymatora Bayesa, estymatora Kn najbliższych sąsiadów, stwierdzono, że poprawiają one jakość obrazów, jak pokazały badania, przy odpowiednich warunkach można uzyskać znacznie lepszą jakość obrazów po filtracji w stosunku do filtrów parametrycznych, co zauważalne jest głównie w szczegółach obrazów.

Celem tej pracy zaprezentowano możliwości, jakie daje zastosowanie w filtracji obrazów, metod statystycznych w postaci estymatora Parzena. Zaprezentowano również fragment badań z wykorzystaniem powstałego algorytmu filtracji.