ON THE EFFICIENCY OF RANDOM WALK APPROACH TO NOISE REDUCTION IN COLOR IMAGES

B. Smolka * *M. Szczepanski* *

Silesian University of Technology
Department of Automatic Control
Akademicka 16 Str, 44-101 Gliwice, Poland
bsmolka@ia.polsl.gliwice.pl

K.N. Plataniotis *A. N. Venetsanopoulos*

Edward S. Rogers Sr. Department of
Electrical and Computer Engineering
University of Toronto
10 King's College Road, Toronto, Canada
kostas@dsp.toronto.edu

ABSTRACT

In this paper we propose a new algorithm of noise reduction in color images. The new technique of multichannel image enhancement is capable of reducing impulse and Gaussian noise and it outperforms the basic methods based on vector median used for the noise reduction in color images. In the paper a new smoothing operator, based on a random walk model and on a fuzzy similarity measure between pixels connected by a digital geodesic path is introduced. The efficiency of the proposed method was tested on the standard color images using the widely used objective image quality measures.

1. STANDARD NOISE REDUCTION FILTERS

Most popular nonlinear, multichannel filters are based on the ordering of vectors in a predefined moving window [1-6]. The output of these filters is defined as the lowest ranked vector according to a specific vector ordering technique.

Let $\mathbf{F}(x)$: represent a multichannel image and let W be a window of finite size n (filter length). The noisy image vectors inside the filtering window W are denoted as $\mathbf{F}_j$, $j = 0, 1, ..., n-1$. If the distance between two vectors $\mathbf{F}_i, \mathbf{F}_j$ is denoted as $\rho(\mathbf{F}_i, \mathbf{F}_j)$ then the scalar quantity $R_i = \sum_{j=0}^{n-1} \rho(\mathbf{F}_i, \mathbf{F}_j)$, is the distance associated with the noisy vector $\mathbf{F}_i$. The ordering of the R_i's: $R_{(1)} \leq \ldots \leq R_{(n-1)}$, implies the same ordering to the corresponding vectors $\mathbf{F}_i$: $\mathbf{F}_{(1)} \leq \ldots \leq \mathbf{F}_{(n-1)}$. Nonlinear ranked type multichannel estimators define the vector $\mathbf{F}_{(0)}$ as the filter output. 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.

To overcome this problem, distance functions are often utilized to order vectors. As an example, the *Vector Median Filter* (VMF) uses the L_1 or L_2 norm to order vectors according to their relative magnitude differences. The orientation difference between two vectors can also be used to remove vectors with atypical directions (*Vector Directional Filter* - VDF, *Basic Vector Directional Filters*- BVDF).

The reduction of image noise without major degradation of the image structure is one of the most important problems of the low-level image processing. A whole variety of algorithms has been developed, but none of them can be seen as a final solution of the noise problem and therefore a new filtering technique, which copes better with impulsive and Gaussian noise has been proposed.

2. NEW ALGORITHM OF NOISE REDUCTION

Let us assume, that R^2 is the Euclidean space, W is a planar subset of R^2 and x, y are points of of the set W.

A path from x to y is a continuous mapping $\mathcal{P}$: $[a, b] \rightarrow X$, such that $\mathcal{P}(a) = x$ and $\mathcal{P}(b) = y$. Point x is the starting point and y is the end point of the path $\mathcal{P}$ [8-10].

An increasing polygonal line P on the path $\mathcal{P}$ is any polygonal line $P = \{g(\lambda_i)\}_{i=0}^{n}$, $a = \lambda_0 < \ldots < \lambda_n = b$. The length of the polygonal line P is the total sum of its constitutive line segments $L(P) = \sum_{i=1}^{n} \rho(\mathcal{P}(\lambda_{i-1}, \lambda_i)$, where $\rho(x, y)$ is the distance between the points x and y, when a specific metric is adopted.

If $\mathcal{P}$ is a path from x to y then it is called rectifiable, if and only if $L(P)$, where P is an increasing polygonal line is bounded. Its upper bound is called the length of the path $\mathcal{P}$. The geodesic distance $\rho^W(x, y)$ between points x and y is the lower bound of the length of all paths leading from x to y totally included in W. If such paths do not exist, then the value of the geodesic distance is set to ∞. The geodesic distance verifies $\rho^W(x, y) \geq \rho(x, y)$ and in the case when W is a convex set then $\rho^W(x, y) = \rho(x, y)$.

The notion of the geodesic distance can be extended to a lattice, which is a set of discrete points, in our case image pixels.

Let a digital lattice $\mathcal{H} = (\mathbf{F}, \mathcal{N})$ be defined by $\mathbf{F}$, which is the set of all points of the plane (pixels of a color im-

* This work was partially supported by KBN grant 8 T11E 013 19

0-7803-6725-1/01/$10.00 ©2001 IEEE

age) and the neighbourhood relation $\mathcal{N}$ between the lattice points.

A digital path $P = \{p_i\}_{i=0}^{n}$ on the lattice $\mathcal{H}$ is a sequence of neighbouring points $(p_{i-1}, p_i) \in \mathcal{N}$. The length $L(P)$ of digital path $P\,\{p_i\}_{i=0}^{n}$ is simply $\sum_{i=1}^{n} \rho^{\mathcal{H}}(p_{i-1}, p_i)$. If $P(x, y)$ denotes the digital path connecting the points x and y in F then the lattice distance between those points is defined as $\rho^{\mathcal{H}}(x, y) = \min_{P(x,y)} L[P(x, y)]$.

Constraining the paths to be totally included in a predefined set $W \in \mathbf{F}$ yields the digital geodesic distance ρ^W. In this paper we will assign to the distance of neighbouring points the value 1 and will be working with the 8-neighbourhood system.

Let the pixels (i, j) and (k, l) be called connected , denoted as $(i, j) \leftrightarrow (k, l)$, if there exists a geodesic path $P^W\{(i, j), (k, l)\}$ contained in the set W starting from (i, j) and ending at (k, l) (Fig. 1).

If two pixels (x_0, y_0) and (x_n, y_n) are connected by a geodesic path $P^W\{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$ of length n then let χ

$$\chi^{W,n}\{(x_0, y_0),(x_n, y_n)\} = \sum_{k=0}^{n-1} ||\mathbf{F}(x_{k+1}, y_{k+1}) - \mathbf{F}(x_k, y_k)|| \tag{1}$$

be a measure of dissimilarity between pixels (x_0, y_0) and (x_n, y_n), along a specific geodesic path P^W joining (x_0, y_0) and (x_n, y_n). If a path joining two distinct points x, y, such that $\mathbf{F}(x) = \mathbf{F}(y)$ consists of lattice points of the same values, then $\chi^{W,n}(x, y) = 0$ otherwise $\chi^{W,n}(x, y) > 0$.

Let us now define the similarity function between two pixels connected along all geodesic digital paths leading from (i, j) and (k, l) (Fig. 1) [7]

$$\mu^{W,n}\{(i,j),(k,l)\} = \frac{1}{\omega} \sum_{l=1}^{\omega} \exp\left[-\beta \cdot \chi_l^{W,n}\{(i,j),(k,l)\}\right] \tag{2}$$

where ω is the number of all geodesic paths connecting (i, j) and (k, l), β is a parameter and $\chi_l^{W,n}\{(i, j), (k, l)\}$ is a dissimilarity value along a specific path from a set of all ω possible paths leading from (i, j) to (k, l). In this way $\mu^{W,n}\{(i, j), (k, l)\}$ is an average value, taken over all routes joining the starting point (i, j) and the end point (k, l).

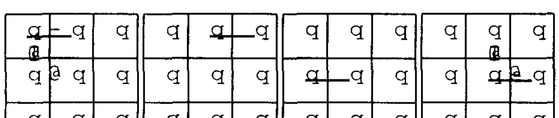

Fig. 1. There are four geodesic paths of length 2 connecting two neighbouring points contained in the 3×3 window W when the 8-neighbourhood system is applied.

For $n = 1$ and W a square mask of the size 3×3, we have $\mu^{W,1}\{(i, j), (k, l)\} = \exp\{-\beta||\mathbf{F}(i, j) - \mathbf{F}(k, l)||\}$ and when $\mathbf{F}(i, j) = \mathbf{F}(k, l)$ then $\chi^{W,n}\{(i, j), (k, l)\} = 0$, $\mu\{(i, j), (k, l)\} = 1$, and for $||\mathbf{F}(i, j) - \mathbf{F}(k, l)|| \to \infty$ then $\mu \to 0$.

The normalized similarity function takes the form

$$\psi^{W,n}\{(i,j),(k,l)\} = \frac{\mu^{W,n}\{(i,j),(k,l)\}}{\sum\limits_{(l,m)\Leftrightarrow(i,j)} \mu^{W,n}\{(i,j),(l,m)\}} \tag{3}$$

The normalized similarity function has the property

$$\sum_{(k,l)\Leftrightarrow(i,j)} \psi^{W,n}\{(i,j),(k,l)\} = 1 \tag{4}$$

Now we are in a position to define a smoothing transformation $\mathcal{J}$

$$\mathbf{J}(i,j) = \sum_{(k,l)\Leftrightarrow(i,j)} \psi^{W,n}\{(i,j),(k,l)\} \cdot \mathbf{F}(k,l \tag{5}$$

where (k, l) are points which are connected with (i, j) by geodesic digital paths of length n included in W.

3. RESULTS

The effectiveness of the new smoothing operator defined by (5) was tested using the *LENA* and *PEPPERS* standard images contaminated by Gaussian noise of $\sigma = 30$. We also used the LENA image contaminated by 4% impulsiv noise (salt & pepper added on each channel) mixed with Gaussian noise ($\sigma = 30$).

The performance of the presented method was evaluated by means of the objective image quality measures RMSE, PSNR, NMSE and NCD [3]. Tables 2 and 3 show the obtained results for $n = 3$ and β increasing linearly from 10 to 30. After 3 iterations the filtered image was being sharpened and was visually more pleasing, however the quality measures were decreasing. Therefore only the results of 3 iterations are shown in the Tab. 2 and 3. Additionally Fig. 2 shows the comparison of the new filtering technique with the standard vector median.

For the calculation of the similarity function we used the L_1 metric and an exponential function, however we have obtained good results using other convex functions and different vector metrics.

4. CONCLUSIONS

In this paper, a new filter for noise reduction in color images has been presented. Experimental results indicate that the new filtering technique outperforms standard procedures used to reduce mixed impulsive and Gaussian noise in color images. The efficiency of the new filtering technique is shown in Tables 2 and 3.

Notation	Filter
AMF	Arithmetic Mean Filter
VMF	Vector Median Filter
BVDF	Basic Vector Directional Filter
GVDF	Generalized Vector Directional Filter
DDF	Directional-Distance Filter
HDF	Hybrid Directional Filter
AHDF	Adaptive Hybrid Directional Filter
FVDF	Fuzzy Vector Directional Filter
ANNF	Adaptive Nearest Neighbor Filter
ANP-EF	Adaptive Non Parametric (Exponential) Filter
ANP-GF	Adaptive Non Parametric (Gaussian) Filter
ANP-DF	Adaptive Non Parametric (Directional) Filter
VBAMMF	Vector Bayesian Adaptive Median/Mean Filter

Table 1. Filters taken for comparison with the proposed filter [1-5].

5. REFERENCES

[1] A.N. Venetsanopoulos, K.N. Plataniotis, Multichannel image processing, Proceedings of the IEEE Workshop on Nonlinear Signal/Image Processing, 2-6, (1995)

[2] I. Pitas, A. N. Venetsanopoulos, 'Nonlinear Digital Filters : Principles and Applications', Kluwer Academic Publishers, Boston, MA, (1990)

[3] K.N. Plataniotis, A.N. Venetsanopoulos, 'Color Image Processing and Applications', Springer Verlag, (June 2000)

[4] I. Pitas, P. Tsakalides, Multivariate ordering in color image processing, IEEE Trans. on Circuits and Systems for Video Technology, 1, 3, 247-256, (1991)

[5] I. Pitas, A.N. Venetsanopoulos, Order statistics in digital image processing, Proceedings of IEEE, 80, 12, 1893-1923, (1992)

[6] J. Astola, P. Haavisto, Y. Neuovo, Vector median filters, IEEE Proceedings, 78, 678-689, (1990)

[7] 12. B. Smolka, K. Wojciechowski, Random walk approach to image enhancement, Signal Processing, Vol. 81, No. 4

[8] G. Borgefors, Distances transformations in digital images. Computer Vision, Graphics and Image Processing, 34:334-371,1986

[9] G. Matheron, Random Sets and Integral Geometry. John Willey, New York, 1975

[10] Henk J.A.M. Heijmans., Mathematical Morphology: Basic Principles, Proceedings of the Summer School on Morphological Image and Signal Processing, Zakopane, Poland, 1995

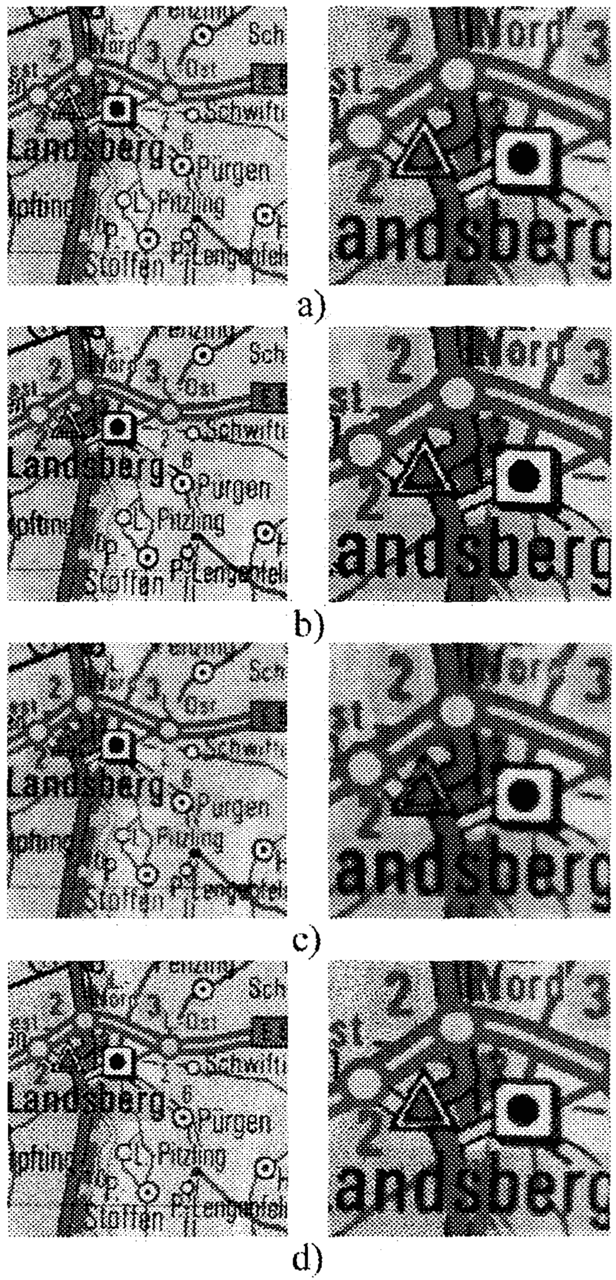

Fig. 2. Comparison of the efficiency of the vector median and the proposed filter: **a**) test image (part of a scanned map), **b**) result of the new filtering technique ($\beta = 20, \alpha = 1.25$, 3 iterations), **c**) result of the standard vector median filtration (3×3 mask), **c**) result of the DDF (3×3 mask).

METHOD_N	NMSE $[10^{-3}]$	RMSE	SNR [dB]	PSNR [dB]	NCD $[10^{-4}]$
NONE	502.410	28.683	12.989	18.978	244.190
AMF_1	90.184	12.152	20.449	26.438	115.210
AMF_3	88.815	12.060	20.515	26.504	99.043
AMF_5	113.840	13.653	19.437	25.426	98.853
VMF_1	168.830	16.627	17.725	23.714	158.920
VMF_3	113.420	13.628	19.453	25.442	129.700
VMF_5	105.180	13.124	19.781	25.770	123.390
BVDF_1	372.320	24.691	14.291	20.280	153.420
BVDF_3	363.390	24.394	14.396	20.385	129.040
BVDF_5	367.740	24.539	14.345	20.334	124.350
GVDF_1	144.640	15.390	18.397	24.386	127.370
GVDF_3	99.400	12.758	20.026	26.015	97.348
GVDF_5	100.490	12.828	19.979	25.968	92.583
DDF_1	184.620	17.387	17.337	23.326	149.540
DDF_3	127.260	14.436	18.953	24.942	120.400
DDF_5	118.820	13.949	19.251	25.240	114.400
HDF_1	147.060	15.518	18.325	24.314	139.380
HDF_3	87.730	11.986	20.569	26.558	107.600
HDF_5	79.698	11.424	20.986	26.975	101.140
AHDF_1	131.390	14.668	18.814	24.803	137.650
AHDF_3	78.739	11.355	21.038	27.027	106.180
AHDF_5	72.331	10.883	21.407	27.396	99.673
FVDF_1	103.950	13.047	19.832	25.821	112.450
FVDF_3	72.888	10.925	21.373	27.362	89.743
FVDF_5	77.012	11.230	21.134	27.123	88.023
ANNF_1	112.660	13.583	19.482	25.471	120.270
ANNF_3	80.934	11.512	20.919	26.908	96.789
ANNF_5	84.101	11.735	20.752	26.741	93.171
ANP-E_1	88.827	12.060	20.515	26.504	115.100
ANP-E_3	79.688	11.423	20.986	26.975	100.860
ANP-E_5	94.793	12.459	20.232	26.221	101.070
ANP-G_1	88.787	12.058	20.517	26.506	115.080
ANP-G_3	79.674	11.422	20.987	26.976	100.850
ANP-G_5	94.741	12.455	20.235	26.224	101.050
ANP-D_1	105.280	13.130	19.776	25.765	113.610
ANP-D_3	73.211	10.949	21.354	27.343	89.078
ANP-D_5	78.419	11.332	21.056	27.045	87.650
VBAMMF_1	90.184	12.152	20.449	26.438	115.210
VBAMMF_3	88.815	12.060	20.515	26.504	99.043
VBAMMF_5	113.840	13.653	19.437	25.426	98.853
NEW_1	**65.412**	**10.349**	**21.843**	**27.832**	**95.248**
NEW_2	**57.921**	**9.739**	**22.372**	**28.361**	**88.917**
NEW_3	**61.473**	**10.033**	**22.113**	**28.102**	**88.561**

Table 2. Comparison of the new algorithm with the standard techniques (Tab. 1) using the *PEPPERS* standard image corrupted by Gaussian noise $\sigma = 30$. The subscripts denote the iteration number.

METHOD_N	NMSE $[10^{-3}]$	RMSE	SNR [dB]	PSNR [dB]	NCD $[10^{-4}]$
NONE	905.930	42.674	10.429	15.528	305.550
AMF_1	128.940	16.099	18.896	23.995	122.880
AMF_3	97.444	13.996	20.112	25.211	95.800
AMF_5	113.760	15.122	19.440	24.539	92.312
VMF_1	161.420	18.013	17.920	23.019	161.700
VMF_3	104.280	14.478	19.818	24.916	128.620
VMF_5	96.464	13.925	20.156	25.255	121.790
BVDF_1	354.450	26.692	14.504	19.603	152.490
BVDF_3	336.460	26.006	14.731	19.829	123.930
BVDF_5	338.940	26.102	14.699	19.797	118.500
GVDF_1	140.970	16.833	18.509	23.607	126.820
GVDF_3	93.444	13.705	20.294	25.393	94.627
GVDF_5	91.118	13.534	20.404	25.503	89.277
DDF_1	176.670	18.845	17.528	22.627	152.050
DDF_3	119.330	15.488	19.232	24.331	119.940
DDF_5	110.620	14.912	19.561	24.660	113.390
HDF_1	143.190	16.966	18.441	23.539	139.360
HDF_3	82.413	12.871	20.840	25.939	104.620
HDF_5	74.487	12.236	21.279	26.378	97.596
AHDF_1	132.710	16.333	18.771	23.869	138.180
AHDF_3	75.236	12.298	21.236	26.334	103.410
AHDF_5	68.563	11.740	21.639	26.738	96.327
FVDF_1	108.760	14.786	19.635	24.734	111.220
FVDF_3	73.796	12.179	21.320	26.418	83.629
FVDF_5	76.274	12.382	21.176	26.275	80.081
ANNF_1	110.720	14.919	19.558	24.656	113.560
ANNF_3	75.652	12.332	21.212	26.310	86.836
ANNF_5	76.757	12.421	21.149	26.247	82.825
ANP-E_1	128.590	16.077	18.908	24.007	122.890
ANP-E_3	90.509	13.488	20.433	25.532	97.621
ANP-E_5	96.930	13.959	20.135	25.234	94.131
ANP-G_1	128.600	16.078	18.908	24.006	122.900
ANP-G_3	90.523	13.489	20.432	25.531	97.603
ANP-G_5	96.990	13.963	20.133	25.231	94.134
ANP-D_1	113.900	15.131	19.435	24.533	115.230
ANP-D_3	74.203	12.213	21.296	26.394	85.026
ANP-D_5	76.265	12.381	21.177	26.275	81.202
VBAMMF_1	128.940	16.099	18.896	23.995	122.880
VBAMMF_3	97.444	13.996	20.112	25.211	95.800
VBAMMF_5	113.760	15.122	19.440	24.539	92.312
NEW_1	**74.762**	**12.259**	**21.263**	**26.362**	**83.585**
NEW_2	**55.239**	**10.537**	**22.578**	**27.676**	**72.115**
NEW_3	**56.078**	**10.617**	**22.512**	**27.611**	**70.008**

Table 3. Comparison of the new algorithm with the standard techniques (Tab. 1) using the *LENA* standard image corrupted by 4% Impulse and Gaussian noise $\sigma = 30$. The subscripts denote the iteration number.