Robust filter for noise reduction in color images

M. Szczepanski [*] *B. Smolka* [*]
Department of Automatic Control
Silesian University of Technology
Akademicka 16 Str, 44-101 Gliwice, Poland
{mszczepa, 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 novel approach to the problem 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 standard methods of noise reduction in color images. In the paper a new smoothing operator, based on a random walk model and a fuzzy similarity measure between the image pixels connected by a digital geodesic path contained in the filter window is introduced. The efficiency of the proposed method was tested on the standard color images using the widely used objective image quality measures.

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 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 in this paper.

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].

[*] This work was partially supported by KBN grant 7 T11A 010 21

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 image) 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 $\rho^{\mathcal{H}}(x, y) = \min\limits_{P(x,y)} L[P(x, y)]$.

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). 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 a function which measures the similarity between two pixels connected along all geodesic digital paths leading from (i, j) and (k, l)

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

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).

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.

Results

The effectiveness of the new smoothing operator 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% impulsive 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 $n = 2$, $W = 3 \times 3$ and β increasing linearly from 10 to 30. 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. The efficiency of the new algorithm is depicted in Fig. 2 and 3.

Conclusions

In this paper, a new robust filter for noise reduction in color images has been presented. Experimental results indicate

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].

that the new filtering technique outperforms the 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 and Figures 2 and 3.

References

[1] A.N. Venetsanopoulos, K.N. Plataniotis, ‘Multichannel image processing’, Proceedings of the IEEE Workshop on Nonlinear Signal/Image Processing, vol. I, pp. 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, Berlin, ISBN 3-540-66953-1, August 2000.

[4] I. Pitas, P. Tsakalides, ‘Multivariate ordering in color image processing’, IEEE Trans. on Circuits and Systems for Video Technology, Vol.1, No. 3, pp. 247-256, 1991.

[5] I. Pitas, A.N. Venetsanopoulos, ‘Order statistics in digital image processing’, The Proceedings of IEEE, vol. 80, no. 12, pp. 1893-1923, 1992.

[6] J. Astola, P. Haavisto, Y. Neuovo, ‘Vector median filters’, The Proceedings of IEEE, vol. 78, pp. 678-689, 1990.

[7] B. Smolka, K. Wojciechowski, ‘Random walk approach to image enhancement’, Signal Processing, vol. 81, no. 3, pp. 465-482, 2001.

[8] G. Borgefors, ‘Distances transformations in digital images’. Computer Vision, Graphics and Image Processing, vol. 34, pp. 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.

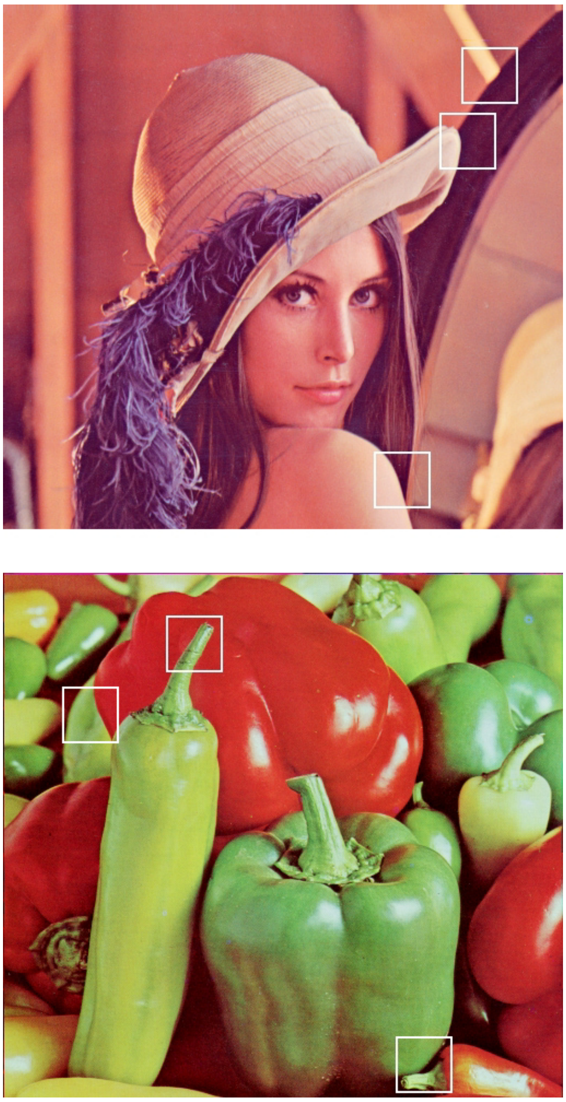

Figure 1. Test images (*Lena* and *Peppers*) with depicted region of interests.

METHOD$_N$	**NMSE** $[10^{-3}]$	**RMSE**	**SNR [dB]**	**PSNR [dB]**	**NCD** $[10^{-4}]$
NONE	420.550	29.075	13.762	18.860	250.090
AMF$_1$	66.452	11.558	21.775	26.873	95.347
AMF$_3$	69.307	11.803	21.592	26.691	76.286
AMF$_5$	91.911	13.592	20.366	25.465	75.566
VMF$_1$	136.560	16.568	18.647	23.745	153.330
VMF$_3$	93.440	13.705	20.295	25.393	123.500
VMF$_5$	87.314	13.248	20.589	25.688	117.170
BVDF$_1$	289.620	24.128	15.382	20.480	143.470
BVDF$_3$	279.540	23.705	15.536	20.634	117.400
BVDF$_5$	281.120	23.772	15.511	20.610	114.290
GVDF$_1$	112.450	15.035	19.490	24.589	119.890
GVDF$_3$	76.988	12.440	21.136	26.234	89.846
GVDF$_5$	76.713	12.418	21.151	26.250	84.876
DDF$_1$	150.830	17.412	18.215	23.314	143.530
DDF$_3$	106.900	14.659	19.710	24.809	114.770
DDF$_5$	100.500	14.213	19.979	25.077	108.960
HDF$_1$	119.100	15.473	19.241	24.339	131.190
HDF$_3$	72.515	12.073	21.396	26.494	99.236
HDF$_5$	66.584	11.569	21.766	26.865	92.769
AHDF$_1$	105.480	14.561	19.768	24.867	129.710
AHDF$_3$	64.519	11.388	21.903	27.002	97.873
AHDF$_5$	60.166	10.997	22.206	27.305	91.369
FVDF$_1$	78.927	12.596	21.028	26.126	101.950
FVDF$_3$	57.466	10.748	22.406	27.504	77.111
FVDF$_5$	62.269	11.188	22.057	27.156	74.235
ANNF$_1$	86.497	13.186	20.630	25.729	107.130
ANNF$_3$	63.341	11.284	21.983	27.082	82.587
ANNF$_5$	66.054	11.523	21.801	26.900	78.677
ANP-E$_1$	66.082	11.525	21.799	26.898	95.237
ANP-E$_3$	60.396	11.018	22.190	27.288	76.896
ANP-E$_5$	73.416	12.148	21.342	26.441	75.456
ANP-G$_1$	66.095	11.526	21.798	26.897	95.244
ANP-G$_3$	60.443	11.023	22.187	27.285	76.890
ANP-G$_5$	73.497	12.155	21.337	26.436	75.458
ANP-D$_1$	81.306	12.784	20.899	25.997	104.980
ANP-D$_3$	58.389	10.834	22.337	27.435	78.486
ANP-D$_5$	63.136	11.265	21.997	27.096	75.442
VBAMMF$_1$	66.452	11.558	21.775	26.873	95.347
VBAMMF$_3$	69.307	11.803	21.592	26.691	76.286
VBAMMF$_5$	91.911	13.592	20.366	25.465	75.566
NEW$_1$	84,314	13,018	20,741	25,840	108,340
NEW$_3$	46,221	9,639	23,352	28,450	73,777
NEW$_5$	47,039	9,724	23,275	28,374	70,189

Table 2. Comparison of the new algorithm with the standard techniques (Tab. 1) using the *LENA* 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$_1$	112,430	15,034	19,491	24,590	118,650
NEW$_3$	53,870	10,406	22,687	27,785	79,115
NEW$_5$	52,225	10,246	22,821	27,920	74,645

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.

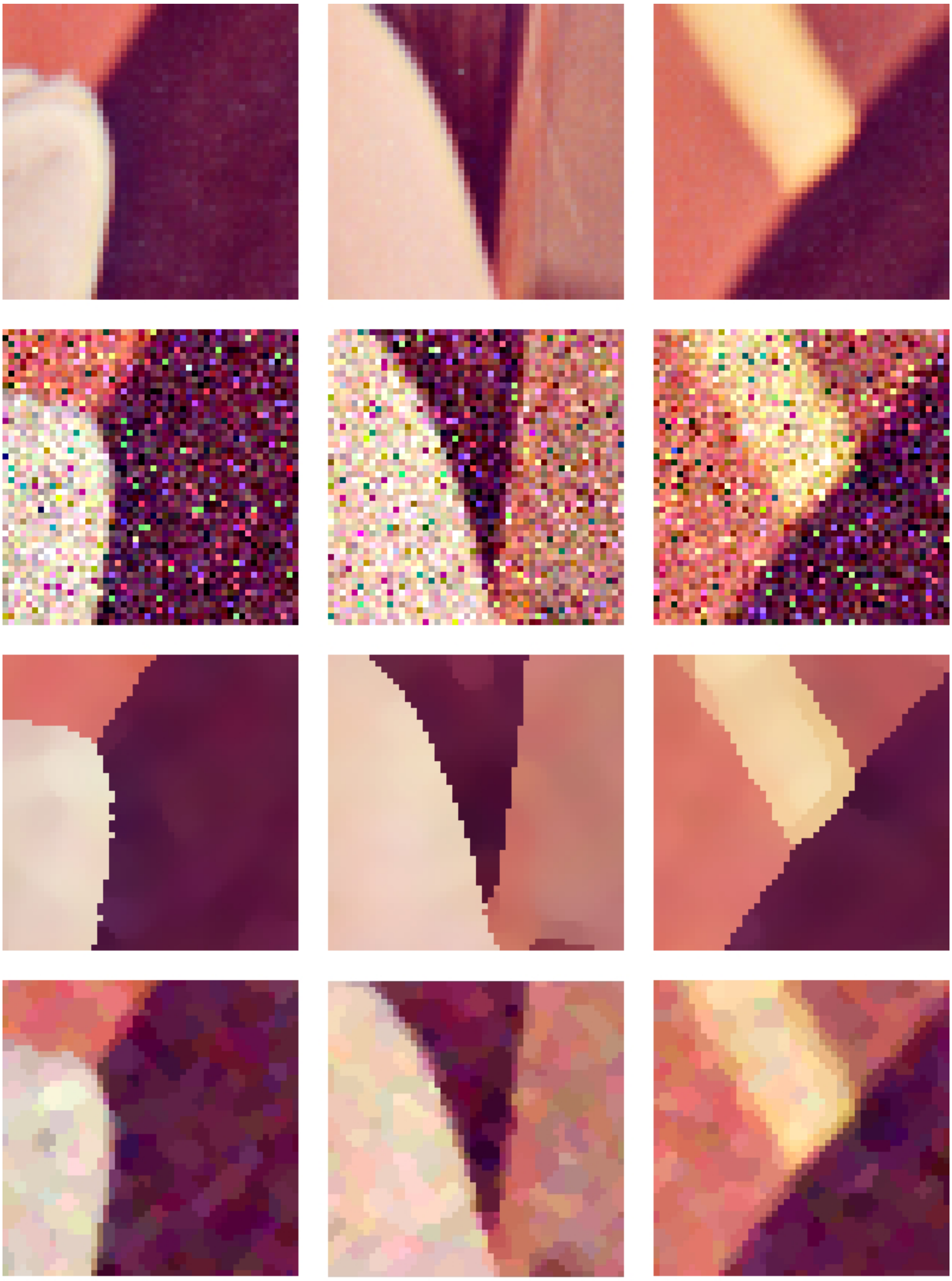

Figure 2. Comparison of the new algorithms with the standard vector median. From top to bottom : parts of the $Lena$ test image from Fig. 1, images distorted by 4% impulse and Gaussian noise $\sigma = 30$, images filtered with the new algorithm and the results obtained with the vector median.

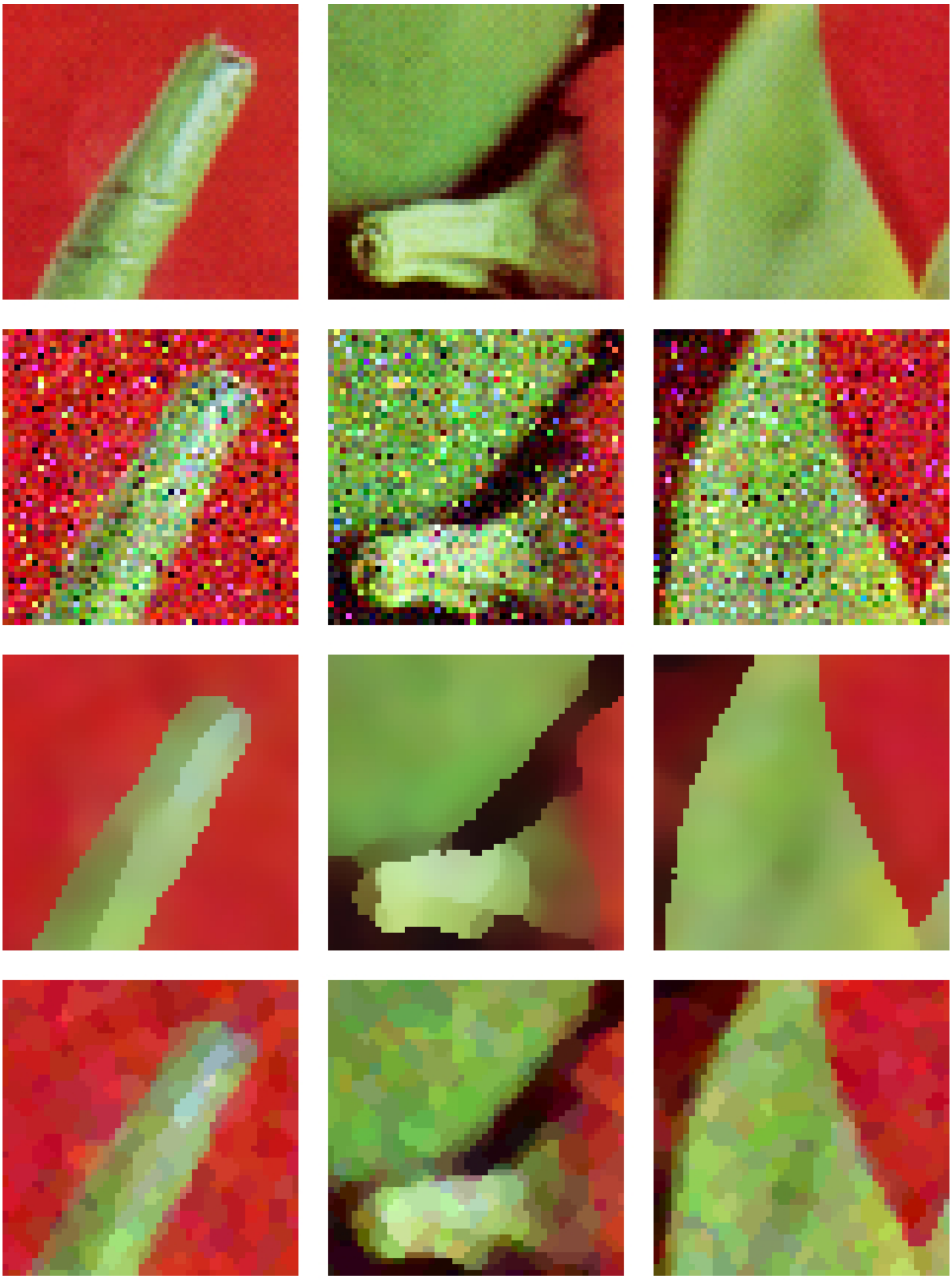

Figure 3. Comparison of the new algorithms with the standard vector median. From top to bottom: parts of the *Peppers* test image from Fig. 1, images distorted by 4% impulse and Gaussian noise $\sigma = 30$, images filtered with the new algorithm and the results obtained with the vector median.