Electronic Notes in Theoretical Computer Science 46 (2001)

URL: <http://www.elsevier.nl/locate/entcs/volume46.html> 15 pages

Geodesic Paths Approach to Color Image Enhancement

M. Szczepanski 1, B. Smolka, and D. Slusarczyk

*Department of Automatic Control Silesian University of Technology*

*Akademicka 16 Str, 44-101 Gliwice, Poland*

K. N. Plataniotis 2 and A. N. Venetsanopoulos

*Edward S. Rogers Sr. Department of Electrical and Computer Engineering University of Toronto*

*10 King’s College Road, Toronto, Canada*

**Abstract**

New filter class for multichannel image processing is introduced and analyzed. The new technique of image enhancement is capable of reducing impulsive and Gaussian noise and it significantly outperforms the standard methods of noise reduction. In the paper a smoothing operator, based on a random walk model and 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 objective image quality measures. Obtained results show that the new method not only outperforms standard noise reduction algorithms, but has some interesting features useful for segmentation of noisy color images.

# Introduction

Numerous noise filtering techniques have been proposed for multichannel im- age processing [6,7]. The nonlinear filters are required to preserve edges, cor- ners and other image details, and to remove Gaussian and impulsive noise. One of the most important families of nonlinear filters is based on order statis- tics. A number of different vector processing filters using order statistics have been developed in the last decade. The output of these filters is defined as the lowest ranked vector according to a specific vector ordering technique.

1 Email: [mszczepa@ia.polsl.gliwice.pl](mailto:mszczepa@ia.polsl.gliwice.pl)

2 Email: [kostas@dsp.toronto.edu](mailto:kostas@dsp.toronto.edu)

*◯*c *2001 Published by Elsevier Science B. V.* Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

Let **F**(*x*) represents 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 **F***j*, *j* = 0*,* 1*, ..., n −* 1 . If the distance between two vectors **F***i,* **F***j* is denoted as *ρ*(**F***i,* **F***j*) then the scalar quantity *Ri* = *n−*1 *ρ*(**F***i,* **F***j*)*,* is the distance associated with the noisy vector **F***i* . The ordering of the *Ri* ’s: *R*(0) *≤ R*(1) *≤ ... ≤ R*(*n−*1)*,* implies the same ordering to the corresponding vectors **F***i* : **F**(0) *≤* **F**(1) *≤ ... ≤* **F**(*n−*1)*.* Nonlinear ranked type multichannel estimators define the vector **F**(0) as the filter output.

*j*=0

Σ

The best known order statistics filter is the so called Vector Median Filter (VMF). The definition of the multichannel median is a direct extension of the ordinary single-channel median definition [1]. VMF uses the *L*1 or *L*2 norm to order vectors according to their relative magnitude differences.

The orientation difference between color vectors can also be used to remove vectors with atypical directions (*Vector Directional Filter* - VDF, *Basic Vector Directional Filter* - BVDF) [14]

Another efficient rank-ordered technique called Hybrid Directional Filter was presented in [4]. This filter operates on the directional and the magni- tude of the color vectors independently and then combines them to produce a unique final output. Another more complex hybrid filter, which involves the utilization of an Arithmetic Mean Filter (AMF), has also been proposed [4]. The reduction of image noise without major degradation of the image structure is one of the most important problems of the low-level image pro- cessing. A whole variety of algorithms has been developed, however none of them can be seen as a final solution of the noise problem and therefore a new

filtering technique is proposed in this paper.

# Digital Paths Approach

Let us assume, that R2 is the Euclidean space, *W* is a planar subset of R2 and *x*, *y* are points of the set *W* . A path from *x* to *y* is a continuous mapping *P*: [*a, b*] *→ X*, such that *P*(*a*) = *x* and *P*(*b*) = *y*. Point *x* is the starting point and *y* is the ending point of the path *P* [2].

An increasing polygonal line *P* on the path *P* is any polygonal line *P* =

*{g*(*λi*)*}n*

*i*=0

*,a* = *λ*0 *< . . . < λn* = *b*. The length of the polygonal line *P* is

*n*

Σ

the total sum of its constitutive line segments *L*(*P* )=

*i*=1 *ρ*(*P*(*λi−*1)*, P*(*λi*))

where *ρ*(*x, y*) is the distance between the points *x* and *y*, when a specific metric is adopted. If *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 *P*.

The geodesic distance *ρW* (*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 *ρW* (*x, y*) *≥ ρ*(*x, y*) and in the case when *W* is a convex set then *ρW* (*x, y*)= *ρ*(*x, y*).

The notion of the path can be extended to a lattice, which is a set of discrete points, in our case image pixels. Let a digital lattice *H* = (**F***, N* ) be defined by **F**, which is the set of all points of the plane (pixels of a color image) and the neighborhood relation *N* between the lattice points [11].

A digital path *P* = *{pi}n*

on the lattice *H* is a sequence of neighboring

points (*p*

*,p* ) *∈ N*

*i*=0

*L*(*P* ) of digital path *P {p }n*

is simply

*i−*1 *i*

Σ

. The length

*i i*=0

*n i*=1

*ρH*(*pi−*1*, pi*), where *ρH* denotes the distance between two neighboring

points on the lattice *H* (Fig. 1).

Constraining the paths to be totally included in a predefined set *W ∈* **F** yields the digital geodesic distance *ρW* . In this paper we will assign to the distance of neighboring points the value 1 and will be working with the 8- neighborhood system.

Let the pixels (*i, j*) and (*k, l*) be called connected (denoted as (*i, j*) *↔* (*k, l*) ), if there exists a geodesic path *PW {*(*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 (*xn, yn*) are connected by a geodesic path

*PW {*(*x*0*, y*0)*,* (*x*1*, y*1)*,...,* (*xn, yn*)*}* of length *n* then let *χW,n*

*m m*

*n−*1

Σ

*χW,n{*(*x*0*, y*0)*,* (*xn, yn*)*}* = *||***F**(*xk*+1*, yk*+1) *−* **F**(*xk, yk*)*|| ,* (1)

*m*

*k*=0

where *m* is the path index be a measure of dissimilarity between pixels (*x*0*, y*0)

and (*xn, yn*), along a specific geodesic path *PW*

*m*

joining (*x*0*, y*0) and (*xn, yn*)

[13,3]. If a path joining two distinct points *x, y*, such that **F**(*x*) = **F**(*y*) consists of lattice points of the same values, then *χW,n*(*x, y*) = 0 otherwise *χW,n*(*x, y*) *>* 0.

.**<❅** .

.

.

.

.

**❅**.

.

**)**.

.**(** .

**/’**

.

.

.

.

.**/**

.

.

.**(/** .

.

.

.

.

.

**/’**.

.

.

.**❅<**

.

. . **)❅**.

. . .

Fig. 1. There are four geodesic paths of length 2 connecting two neighboring points contained in the specific window *W* when the 8-neighborhood system is applied.

Let us now define a fuzzy similarity function between two pixels connected along all geodesic digital paths leading from (*i, j*) to (*k, l*)

*µW,n {*(*i, j*)*,* (*k, l*)*}* = 1 Σ exp *−β · χW,n {*(*i, j*)*,* (*k, l*)*}* (2)

*ω*

*ω m*

*m*=1

where *ω* is the number of all paths connecting (*i, j*) and (*k, l*), *β* is a design parameter and *χW,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 *µW,n{*(*i, j*)*,* (*k, l*)*}* is a value, taken over all routes linking the starting point

*m*

. . . . .

. . . . .

. . . . .

. . . .. *y* .

. . **)**. **)**.. *y* .

**✻**

. . . .. *y* .

**✻**

. . . . .

. .**✻**

. . .

. . . .**✻** .

. *x* .. . . .

. *x* ..

. . .

. *x* .. **)**. **)**. .

. . . . .

. . . . .

. . . . .

. . . . .

. . . . .

. . . . .

. . . **)**.. *y* .

. . .

.. *y* .

. . . **)**.. *y* .

**✻**

. . .**✻** . .

. . .

**✻**

**)**. .

**✻**

. . **)**. . .

. *x* ..

**)**. . .

**✻**

. *x* .. **)**. . .

**✻**

. *x* ..

. . .

. . . . .

. . . . .

. . . . .

Fig. 2. There are five paths of length 4 connecting point *x* and *y* when the 4-neigh- borhood system is used.

(*i, j*) and the endpoint (*k, l*). For *n* = 1 and *W* a square mask of the size 3 *×* 3 (Fig. 1), we have

*µW,*1*{*(*i, j*)*,* (*k, l*)*}* = exp *{−β||***F**(*i, j*) *−* **F**(*k, l*)*||} ,* (3)

and when **F**(*i, j*) = **F**(*k, l*) then *χW,n{*(*i, j*)*,* (*k, l*)*}* = 0, *µ{*(*i, j*)*,* (*k, l*)*}* = 1, and for *||***F**(*i, j*) *−* **F**(*k, l*)*|| → ∞* then *µ →* 0 [8]. The normalized similarity function takes the form

*ψW,n{*(*i, j*)*,* (*k, l*)*}* =

*µW,n{*(*i, j*)*,* (*k, l*)*}*

*µW,n{*(*i, j*)*,* (*l, m*)*} .* (4)

Σ

(*l,m*)*⇔*(*i,j*)

and has the property that

### Σ

(*k,l*)*⇔*(*i,j*)

*ψW,n{*(*i, j*)*,* (*k, l*)*}* = 1 *.* (5)

Now we are in a position to define a smoothing transformation **Fˆ**

**Fˆ**(*i, j*)= Σ

(*k,l*)*⇔*(*i,j*)

*ψW,n{*(*i, j*)*,* (*k, l*)*}·* **F**(*k, l*) *,* (6)

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

# New Filter Design

* 1. *Models of Digital Paths*

The features of the new filter strongly depend on the type of digital paths chosen. Numerous models of paths produce specific filters with the ability to suppress certain kinds of noise. In this paper three types of random paths are introduced: Self avoiding path (*SAP*) Non-reversing path model (*NRP*) and Escaping Path Model (*EPM*)(Fig. 3).

. . . . . .

. . . . . .

. . . . . .

. **)**. . . **)**.. *y* . . . . . .. *y* . . . . . **)**.. *y* .

**^ ❅ ^ /’ ^**

.**(** . **❅❘**. .**(** . . . . . .**/** . . . . . **)**. . .

**^ ❅ /’ ❅< /’**

. .**(** . **❅❘**.**/** . . . . .**(** . **❅** **/**

. . . .

**^ ^ ^ ^**

. *x* .. **)**. . . . . *x* .. **)**.**7)**. **/’**. . . *x* .. . . . .

. . . . . . . . .**7)**.**/** . . . . . . . .

## b) c)

Fig. 3. Different types of geodesic paths **a)** Self avoiding path (*SAP*),

* + 1. Non-reversing path (*NRP*), **c)** Escaping Path (*EPM*) with *L*2 metric .

Self avoiding path (*SAP*) is a special path along the image lattice such that adjacent pairs of edges in the sequence share a common vertex of the lattice, but no vertex is visited more than once and in this way the trajectory never intersects itself. In other words *SAP* is a path on a lattice that does not pass through the same point twice (Fig. 3a).

On the two-dimensional lattice *SAP* is a finite sequence of distinct lattice points (*x*0*, y*0)*,* (*x*1*, y*1)*,...,* (*xn, yn*), which are in neighborhood relation and (*xi, yi*) */*=*xj*(*, yj*) for all *i /*=*j*.

Non-reversing path is (*NRP*) is a special trajectory along the image lattice such that adjacent pairs of edges in the sequence share a common vertex of the lattice, but no vertex can be revisited in one step (Fig. 3b).

Escaping path model(*EPM*) is a model of random walk in which the distance from the starting point cannot be decreased in subsequent steps (Fig. 3c).

For two steps all described paths are equivalent.

* 1. *Iterative Nature of the New Class of Filters*

The smoothing operator **Fˆ** in (6) has to be applied in an iterative way. Starting with low value of *β* enables the smoothing of the image noise components. At each iteration steps the parameter *β* has to be increased, like in simulated

annealing, so we have used:

*β*(*k*)= *β*(*k −* 1) *· α, k* = 1*,..., n,* (7)

However, in this case two parameters *α* and *β* are needed to define the filter. In order to make the new filter less dependent on the initial parameter values, adaptive version of our filter was introduced. Parameter *β* in (2) is obtained from the data in the filter window and is defined as a standard deviation of samples in *W* , multiplied by a normalizing constant factor *γ*,

*γ* , Σ Σ  2

*l*

,

*β* = *N · l*

*i,j∈W k*=1

*Fk* (*i, j*) *− Fk*

*,* (8)

where *N* is the number of pixels in the processing window *W* , *l* is the number of channels of the image (in the RGB color space *l* = 3), *Fk* denotes the average value of the *k*th component in window *W* and *γ* is a normalizing parameter. Using adaptive version of our filter, there is no need to use parameter *α* from

(7) and in this way there remains only one design parameter, while performing the filtering. As shown in Tabs. 2 and 3 the adaptive version of our filter yields better results especially for heavily distorted images.

# Results

The effectiveness of the new filters was tested on the color test image *LENA* contaminated by a Gaussian noise of *σ* = 30 and on the same original image contaminated by 4% impulsive noise (salt & pepper in each channel) mixed with Gaussian noise (*σ* = 30). The performance of the presented method was evaluated by means of the objective image quality measures RMSE, PSNR, NMSE and NCD [7].

Tables 2 and 3 show the results obtained for *n* = 2 and *n* = 3 in comparison with the standard noise reduction algorithms shown in Tab. 1. Additionally Fig. 7 shows the comparison of the new filtering technique with the standard vector median.

In our experiments wide range of filter parameters was examined. Figures 4 and 5 show the peak signal to noise ratio (PSNR) and normalized color distance (NCD) dependence on the *α* and *β* values for the *LENA* standard image corrupted by 4% impulse mixed with Gaussian noise (*σ* = 30). As can be easily observed the extrema of PSNR and NCD are rather flat and in this way the new filter is robust to improper values of chosen parameters. Results obtained for adaptive version of our filter are presented in Fig. 6

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.

|  |  |  |
| --- | --- | --- |
| **Notation** | **METHOD** | **REF**. |
| AMF | Arithmetic Mean Filter | [7] |
| VMF | Vector Median Filter | [1] |
| BVDF | Basic Vector Directional Filter | [14] |
| GVDF | Generalized Vector Directional Filter | [15] |
| DDF | Directional-Distance Filter | [5] |
| HDF | Hybrid Directional Filter | [4] |
| AHDF | Adaptive Hybrid Directional Filter | [4] |
| FVDF | Fuzzy Vector Directional Filter | [8] |
| ANNF | Adaptive Nearest Neighbor Filter | [9] |
| ANP-EF | Adaptive Non Parametric (Exponential) Filter | [10] |
| ANP-GF | Adaptive Non Parametric (Gaussian) Filter | [10] |
| ANP-DF | Adaptive Non Parametric (Directional) Filter | [10] |
| VBAMMF | Vector Bayesian Adaptive Median/Mean Filter | [10] |

Table 1

Filters taken for comparison with the proposed noise reduction technique.

The efficiency of the new algorithm as compared with the vector median filter is shown in Figs. 7 and 8. After the application of the new filter, the impulse pixels introduced by noise process are removed, the contrast is improved, the image is smoothed and what is important the edges are well preserved.

Figures 9 and 10 show the features of the new filter class for *LENA* and *PEPPERS* images. The results show that they are quite similar to those ob- tained using anisotropic diffusion. However, our filter is robust to the impulse noise, which is a main obstacle, when using the anisotropic diffusion approach to smooth noisy images.

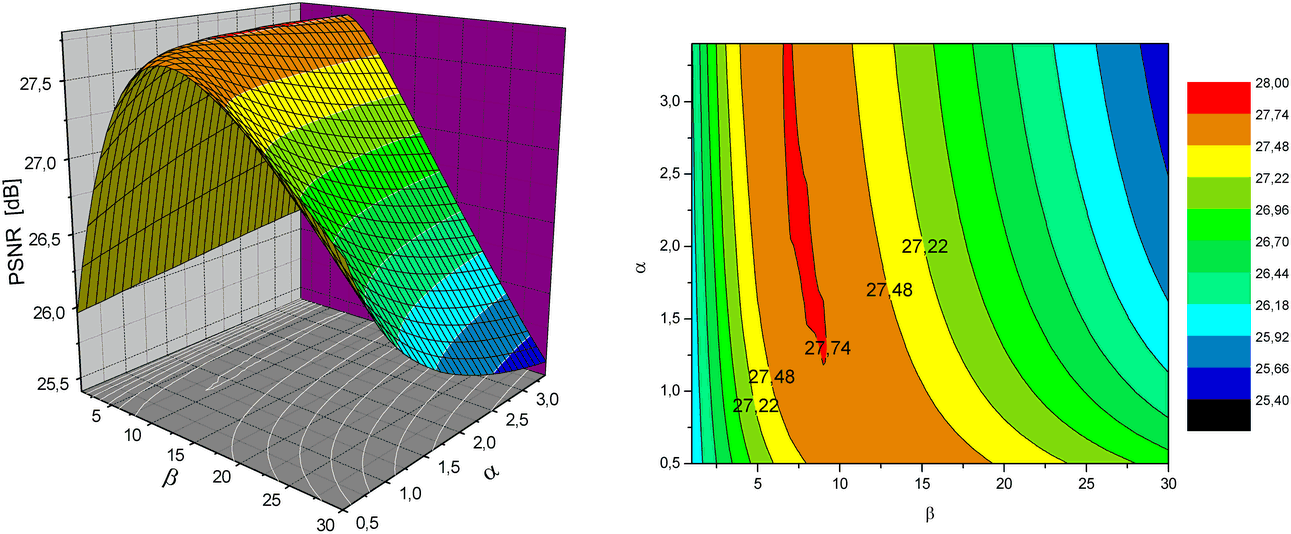


Fig. 4. Efficiency of the new filter in terms of PSNR and its dependence on the *α* and *β* values for the *LENA* standard image corrupted by 4% impulse and Gaussian noise (*σ* = 30) (SAP *n* = 3, 2 iterations).

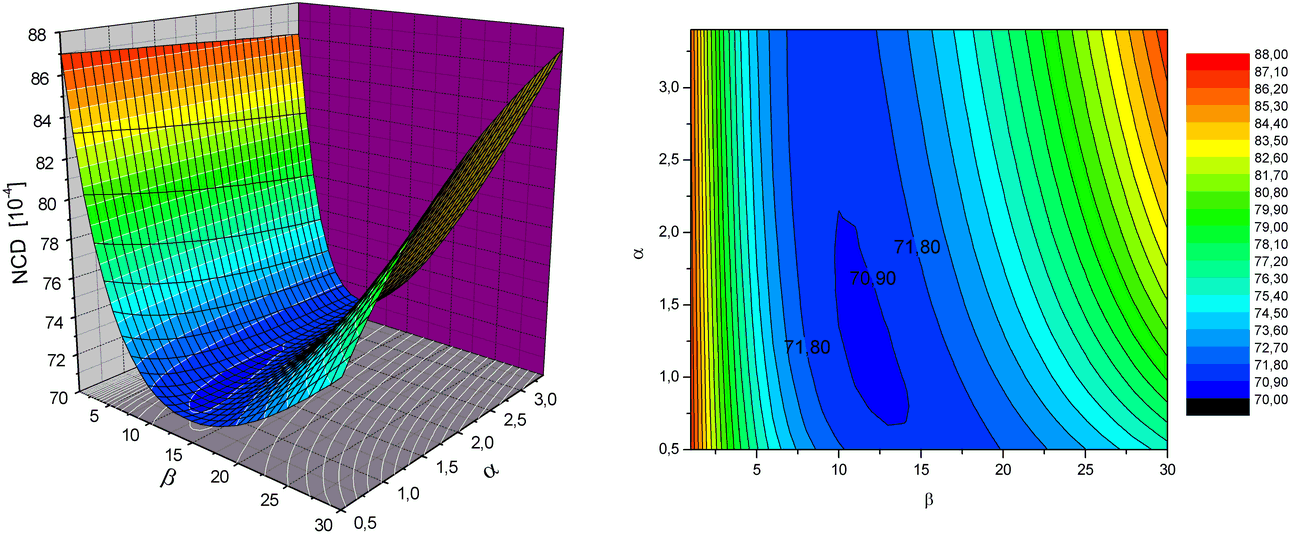


Fig. 5. Efficiency of the new filter in terms of NCD and its dependence on the *α* and *β* values for *LENA* standard image corrupted by 4% impulse and Gaussian noise (*σ* = 30) (SAP *n* = 3, 2 iterations).

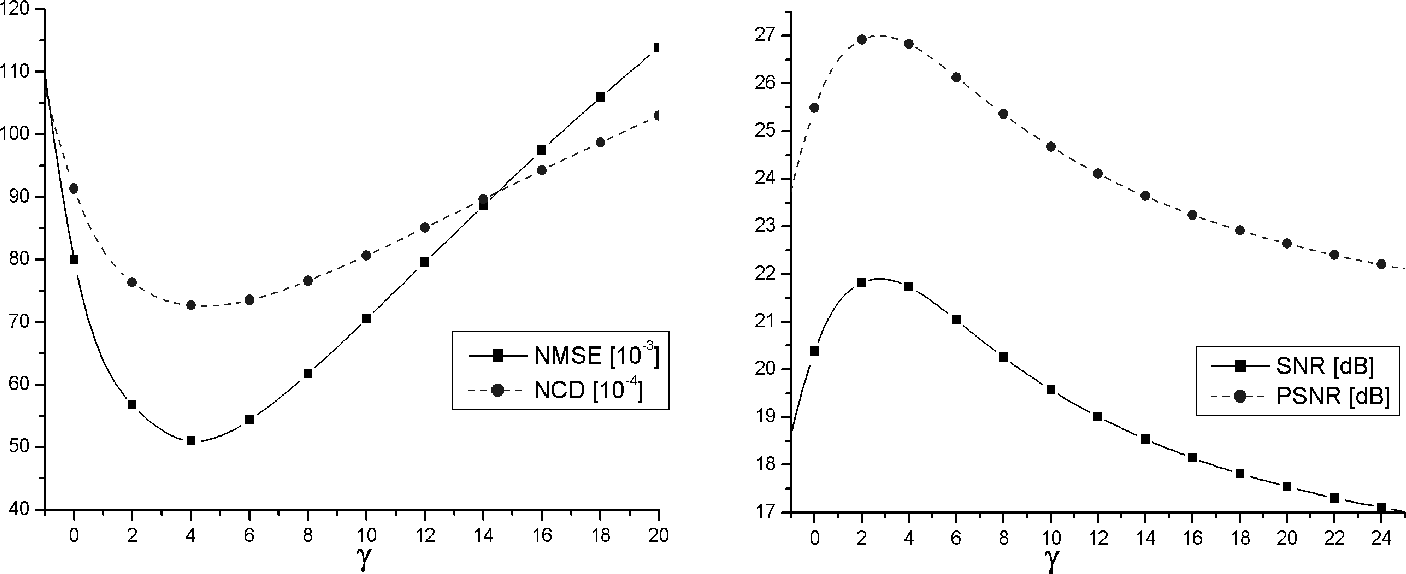


Fig. 6. Efficiency of the new adaptive filter in terms of PSNR, SNR, NCD and NMSE for *LENA* standard image corrupted by 4% impulse and Gaussian noise (*σ* = 30) (*n* = 2, 2 iterations).

# Conclusions

In this paper, a new class of filters for noise reduction in color images has been presented. Experimental results indicate that the new filtering tech- nique outperforms the standard procedures used to reduce mixed impulsive and Gaussian noise in color images. The new methods have some interest- ing features useful for segmentation of noisy color images. Especially, filters based on the escaping path models (*EPP*), have the ability of segmentation of strongly disturbed images (Fig. 8). The efficiency of the new filtering techniques are shown in Tabs. 2 and 3 and in Figs. 7, 8.

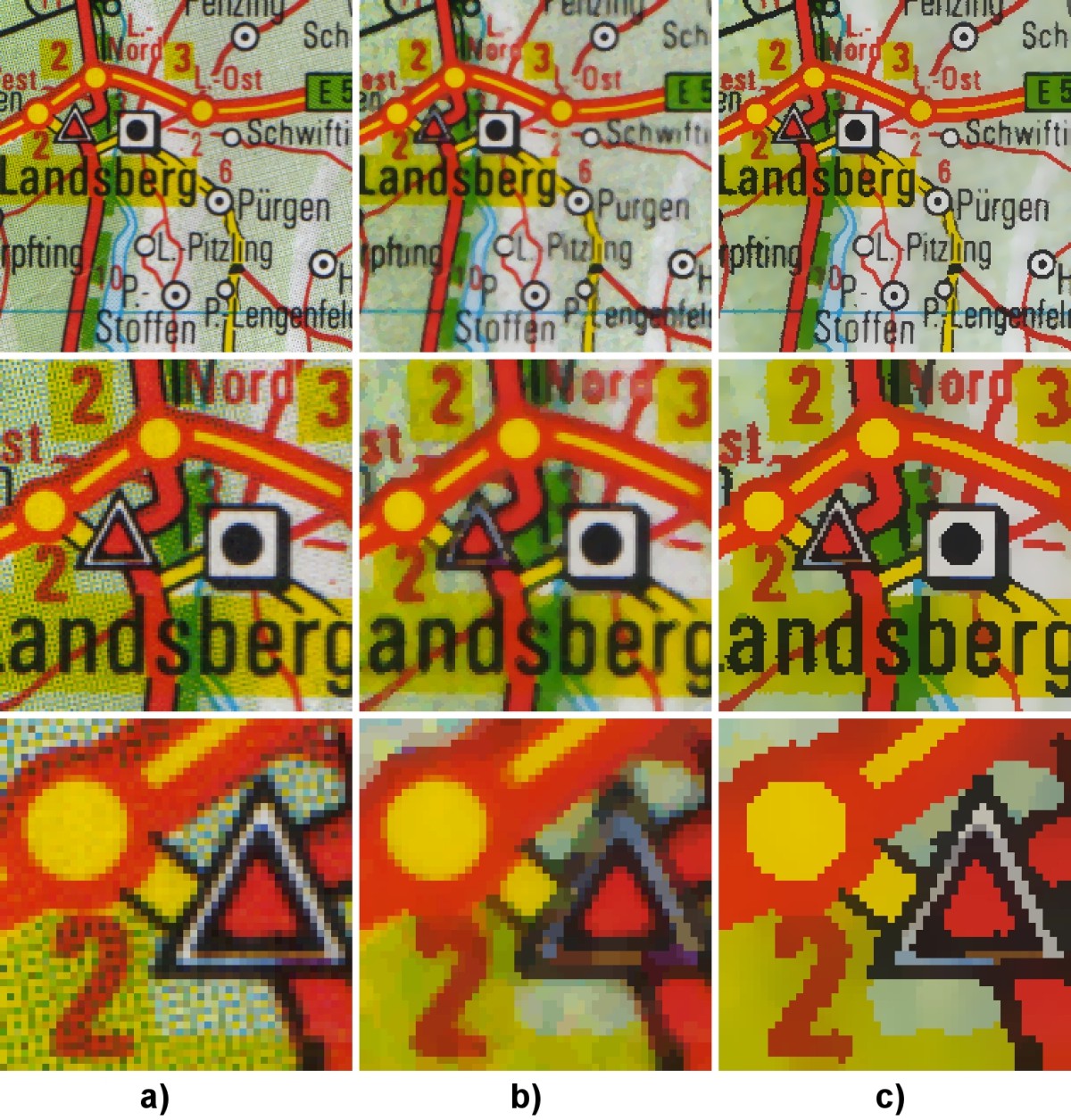


Fig. 7. Comparison of efficiency of the vector median with the new filter proposed in this paper **a**) test image (part of a scanned map), **b**) result of the standard vector median filtration (3 *×* 3 mask), **c**) result of the filtration with the new filter using SAP (*β* = 20*,α* = 1*.*25*,n* = 2, 3 iterations)

# References

1. Astola J., Haavisto P., Neuovo Y., *Vector median filters*, IEEE Proc., **78** (1990), 678-689
2. Borgefors G., *Distances transformations in digital images*, Computer Vision,

Graphics and Image Processing, **34** (1986) 334-371

1. Cuisenaire O., ”Distance transformations: fast algorithms and applications to medical image processing”, PhD Thesis, Universite Catholique de Louvain, Oct. 1999
2. Gabbouj M., Cheickh F.A., *Vector median - vector directional hybrid filter for colour image restoration*, Proceedings of EUSIPCO, 879-881, 1996
3. Karakos D.,Trahanias P.E., *Generalized multichannel image filtering structures*,

IEEE Trans. on Image Processing, 6, (7), 1038-1045, 1997

1. Pitas I., Venetsanopoulos A. N, ”Nonlinear Digital Filters : Principles and Applications”, Kluwer Academic Publishers, Boston, MA, 1990
2. Plataniotis K.N., Venetsanopoulos A.N., ”Color Image Processing and Applications”, Springer Verlag, (June 2000)
3. Plataniotis K.N., Androutsos D., Venetsanopoulos A.N.V., *Fuzzy adaptive filters for multichannel image processing*, Signal Processing Journal, 55, (1), 93-106, 1996
4. Plataniotis K.N., Androutsos D., Sri V.,Venetsanopoulos A.N.V., *A nearest neighboor multichannel filter*, Electronic Letters, 1910-1911, 1995
5. Plataniotis K.N., Androutsos D., Vinayagamoorthy S., Venetsanopoulos A.N.V., *Color image processing using adaptive multichannel filters*,IEEE Trans. on Image Processing, 6 (7), 933-950, 1997
6. Schmitt M., *Lecture Notes on Geodesy and Morphological Measurements*, Proceedings of the Summer School on Morphological Image and Signal Processing, Zakopane, Poland (1995)
7. Smolka B., Wojciechowski K., *Random walk approach to image enhancement*,

Signal Processing, Vol. 81, No. 3, 465-482, 2001

1. Toivanen P.J., *New geodesic distance transforms for gray scale images*, Pattern Recognition Letters, **17** (1996) 437-450
2. Trahanias P.E., Venetsanopoulos A.N., *Vector directional filters: A new class of multichannel image processing filters*, IEEE Trans. on Image Processing, **2, 4** (1993) 528-534
3. Trahanias P.E., Karakos D., Venetsanopoulos A.N., *Directional processing of color images : theory and experimental results*, IEEE Trans. on Image Processing, 5, (6),868-880, 1996

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| METHOD*N* | **NMSE** [10*−*3] | **RMSE** | **SNR [dB]** | **PSNR [dB]** | **NCD** [10*−*4] |
| NONE | 420.550 | 29.075 | 13.762 | 18.860 | 250.090 |
| AMF1 | 66.452 | 11.558 | 21.775 | 26.873 | 95.347 |
| AMF3 | 69.307 | 11.803 | 21.592 | 26.691 | 76.286 |
| AMF5 | 91.911 | 13.592 | 20.366 | 25.465 | 75.566 |
| VMF1 | 136.560 | 16.568 | 18.647 | 23.745 | 153.330 |
| VMF3 | 93.440 | 13.705 | 20.295 | 25.393 | 123.500 |
| VMF5 | 87.314 | 13.248 | 20.589 | 25.688 | 117.170 |
| BVDF1 | 289.620 | 24.128 | 15.382 | 20.480 | 143.470 |
| BVDF3 | 279.540 | 23.705 | 15.536 | 20.634 | 117.400 |
| BVDF5 | 281.120 | 23.772 | 15.511 | 20.610 | 114.290 |
| GVDF1 | 112.450 | 15.035 | 19.490 | 24.589 | 119.890 |
| GVDF3 | 76.988 | 12.440 | 21.136 | 26.234 | 89.846 |
| GVDF5 | 76.713 | 12.418 | 21.151 | 26.250 | 84.876 |
| DDF1 | 150.830 | 17.412 | 18.215 | 23.314 | 143.530 |
| DDF3 | 106.900 | 14.659 | 19.710 | 24.809 | 114.770 |
| DDF5 | 100.500 | 14.213 | 19.979 | 25.077 | 108.960 |
| HDF1 | 119.100 | 15.473 | 19.241 | 24.339 | 131.190 |
| HDF3 | 72.515 | 12.073 | 21.396 | 26.494 | 99.236 |
| HDF5 | 66.584 | 11.569 | 21.766 | 26.865 | 92.769 |
| AHDF1 | 105.480 | 14.561 | 19.768 | 24.867 | 129.710 |
| AHDF3 | 64.519 | 11.388 | 21.903 | 27.002 | 97.873 |
| AHDF5 | 60.166 | 10.997 | 22.206 | 27.305 | 91.369 |
| FVDF1 | 78.927 | 12.596 | 21.028 | 26.126 | 101.950 |
| FVDF3 | 57.466 | 10.748 | 22.406 | 27.504 | 77.111 |
| FVDF5 | 62.269 | 11.188 | 22.057 | 27.156 | 74.235 |
| ANNF1 | 86.497 | 13.186 | 20.630 | 25.729 | 107.130 |
| ANNF3 | 63.341 | 11.284 | 21.983 | 27.082 | 82.587 |
| ANNF5 | 66.054 | 11.523 | 21.801 | 26.900 | 78.677 |
| ANP-E1 | 66.082 | 11.525 | 21.799 | 26.898 | 95.237 |
| ANP-E3 | 60.396 | 11.018 | 22.190 | 27.288 | 76.896 |
| ANP-E5 | 73.416 | 12.148 | 21.342 | 26.441 | 75.456 |
| ANP-G1 | 66.095 | 11.526 | 21.798 | 26.897 | 95.244 |
| ANP-G3 | 60.443 | 11.023 | 22.187 | 27.285 | 76.890 |
| ANP-G5 | 73.497 | 12.155 | 21.337 | 26.436 | 75.458 |
| ANP-D1 | 81.306 | 12.784 | 20.899 | 25.997 | 104.980 |
| ANP-D3 | 58.389 | 10.834 | 22.337 | 27.435 | 78.486 |
| ANP-D5 | 63.136 | 11.265 | 21.997 | 27.096 | 75.442 |
| **SAP-2**1 | **51.869** | **10.211** | **22.851** | **27.950** | **80.682** |
| **SAP-2**2 | **45.203** | **9.532** | **23.448** | **28.547** | **69.149** |
| **SAP-2**3 | **48.918** | **9.916** | **23.105** | **28.204** | **68.676** |
| **SAP-3**1 | **50.600** | **10.085** | **22.958** | **28.057** | **74.218** |
| **SAP-3**2 | **51.396** | **10.164** | **22.891** | **27.989** | **68.796** |
| **SAP-3**3 | **58.020** | **10.799** | **22.364** | **27.463** | **70.046** |
| **SAP-Ad**1 | **52.631** | **10.286** | **22.788** | **27.886** | **81.282** |
| **SAP-Ad**2 | **45.018** | **9.513** | **23.466** | **28.565** | **68.579** |
| **SAP-Ad**3 | **48.158** | **9.839** | **23.173** | **28.272** | **67.573** |
| **EPM**1 | **60.028** | **10.985** | **22.216** | **27.315** | **79.352** |
| **EPM**2 | **55.030** | **10.518** | **22.594** | **27.693** | **68.265** |
| **EPM**3 | **60.099** | **10.991** | **22.211** | **27.310** | **68.914** |

Table 2

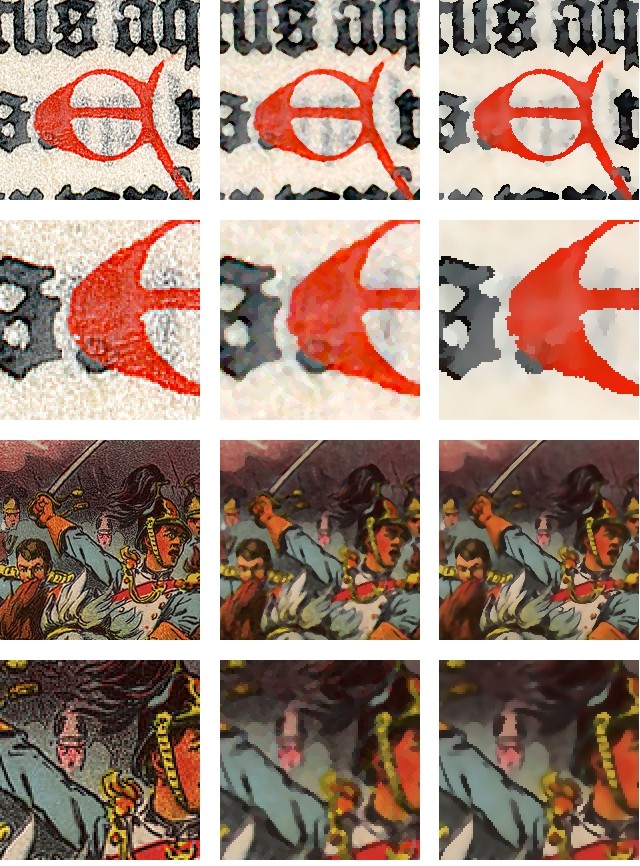
Comparison of the new algorithms with the standard techniques (Tab. 1) using the *LENA* standard image corrupted by Gaussian noise *σ* = 30. SAP-2, 3 denote the self avoiding path with 2 and 3 steps (*β* = 8*,α* = 1*.*2), SAP-Ad denotes adaptive version of SAP (*γ* = 4, 2 steps) and EPM denotes the filter based on a model of escaping path (3 steps, *β* = 11*,α* = 2). 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 |
| AMF1 | 128.940 | 16.099 | 18.896 | 23.995 | 122.880 |
| AMF3 | 97.444 | 13.996 | 20.112 | 25.211 | 95.800 |
| AMF5 | 113.760 | 15.122 | 19.440 | 24.539 | 92.312 |
| VMF1 | 161.420 | 18.013 | 17.920 | 23.019 | 161.700 |
| VMF3 | 104.280 | 14.478 | 19.818 | 24.916 | 128.620 |
| VMF5 | 96.464 | 13.925 | 20.156 | 25.255 | 121.790 |
| BVDF1 | 354.450 | 26.692 | 14.504 | 19.603 | 152.490 |
| BVDF3 | 336.460 | 26.006 | 14.731 | 19.829 | 123.930 |
| BVDF5 | 338.940 | 26.102 | 14.699 | 19.797 | 118.500 |
| GVDF1 | 140.970 | 16.833 | 18.509 | 23.607 | 126.820 |
| GVDF3 | 93.444 | 13.705 | 20.294 | 25.393 | 94.627 |
| GVDF5 | 91.118 | 13.534 | 20.404 | 25.503 | 89.277 |
| DDF1 | 176.670 | 18.845 | 17.528 | 22.627 | 152.050 |
| DDF3 | 119.330 | 15.488 | 19.232 | 24.331 | 119.940 |
| DDF5 | 110.620 | 14.912 | 19.561 | 24.660 | 113.390 |
| HDF1 | 143.190 | 16.966 | 18.441 | 23.539 | 139.360 |
| HDF3 | 82.413 | 12.871 | 20.840 | 25.939 | 104.620 |
| HDF5 | 74.487 | 12.236 | 21.279 | 26.378 | 97.596 |
| AHDF1 | 132.710 | 16.333 | 18.771 | 23.869 | 138.180 |
| AHDF3 | 75.236 | 12.298 | 21.236 | 26.334 | 103.410 |
| AHDF5 | 68.563 | 11.740 | 21.639 | 26.738 | 96.327 |
| FVDF1 | 108.760 | 14.786 | 19.635 | 24.734 | 111.220 |
| FVDF3 | 73.796 | 12.179 | 21.320 | 26.418 | 83.629 |
| FVDF5 | 76.274 | 12.382 | 21.176 | 26.275 | 80.081 |
| ANNF1 | 110.720 | 14.919 | 19.558 | 24.656 | 113.560 |
| ANNF3 | 75.652 | 12.332 | 21.212 | 26.310 | 86.836 |
| ANNF5 | 76.757 | 12.421 | 21.149 | 26.247 | 82.825 |
| ANP-E1 | 128.590 | 16.077 | 18.908 | 24.007 | 122.890 |
| ANP-E3 | 90.509 | 13.488 | 20.433 | 25.532 | 97.621 |
| ANP-E5 | 96.930 | 13.959 | 20.135 | 25.234 | 94.131 |
| ANP-G1 | 128.600 | 16.078 | 18.908 | 24.006 | 122.900 |
| ANP-G3 | 90.523 | 13.489 | 20.432 | 25.531 | 97.603 |
| ANP-G5 | 96.990 | 13.963 | 20.133 | 25.231 | 94.134 |
| ANP-D1 | 113.900 | 15.131 | 19.435 | 24.533 | 115.230 |
| ANP-D3 | 74.203 | 12.213 | 21.296 | 26.394 | 85.026 |
| ANP-D5 | 76.265 | 12.381 | 21.177 | 26.275 | 81.202 |
| **SAP-2**1 | **65.777** | **11.499** | **21.819** | **26.918** | **87.297** |
| **SAP-2**2 | **50.597** | **10.085** | **22.959** | **28.057** | **73.008** |
| **SAP-2**3 | **51.459** | **10.170** | **22.885** | **27.984** | **71.474** |
| **SAP-3**1 | **58.580** | **10.851** | **22.322** | **27.421** | **79.256** |
| **SAP-3**2 | **54.324** | **10.450** | **22.650** | **27.749** | **71.808** |
| **SAP-3**3 | **58.441** | **10.839** | **22.333** | **27.431** | **71.510** |
| **SAP-Ad**1 | **67.069** | **11.611** | **21.735** | **26.833** | **87.987** |
| **SAP-Ad**2 | **51.018** | **10.127** | **22.923** | **28.021** | **72.650** |
| **SAP-Ad**3 | **52.101** | **10.234** | **22.832** | **27.930** | **70.954** |
| **EPM**1 | **69.468** | **11.817** | **21.582** | **26.681** | **84.310** |
| **EPM**2 | **60.087** | **10.990** | **22.212** | **27.311** | **71.362** |
| **EPM**3 | **65.590** | **11.482** | **21.832** | **26.9300** | **72.118** |

Table 3

Comparison of new algorithms with standard techniques using *LENA* image corrupted by 4% impulse and Gaussian noise *σ* = 30. SAP-2, 3 denote the SAP with 2 and 3 steps (*β* = 8*,α* = 1*.*2), SAP-Ad denotes adaptive version of SAP

(*γ* = 4, 2 steps) and EPM denotes the filter based on a model of escaping path (3 steps, *β* = 10*,α* = 1*.*4), subscripts denote the iteration number.



## a) b) c)

Fig. 8. Comparison of the efficiency of the vector median with the proposed noise reduction technique when Escaping Path Model is used: **a**) test images (parts of an old manuscript and a poster), **b**) result of the standard vector median filtration (3 *×* 3 mask), **c**) result of the EPM filtration (*β* = 20*,α* = 1*.*2*,n* = 3, 3 iterations).



Fig. 9. Comparison of the efficiency of the vector median with the proposed noise reduction technique for *LENA* and *PEPPERS* test images. **a**) test images, **b**) result of the standard vector median filtration (3 *×* 3 mask, five iterations), **c**) result of the Adaptive SAP filtration (*γ* = 4*,n* = 2, 5 iterations).



Fig. 10. Comparison of the efficiency of the vector median with the proposed noise reduction technique for *LENA* and *PEPPERS* test images: **a**) test images corrupted by 4% impulse and Gaussian noise *σ* = 30, **b**) result of the standard vector median filtration (3 *×* 3 mask, five iterations), **c**) result of the Adaptive SAP filtration (*γ* = 4*,n* = 2, 5 iterations).