

Observations on Adaptive Vector Filters for Noise Reduction in Color Images

Michael J. Cree, *Member, IEEE*

Abstract—In a series of papers, Plataniotis *et al.* proposed a number of filters for noise reduction in color images where the noise type is unknown. In this letter, those filters with a unified notation are summarized, and it is shown that they are essentially variants of the same filtering procedure. It is also shown that the class of adaptive vector filters can be considered as interpolants between the arithmetic mean filter and the vector median filter. Results are presented of numerical computations with the filters on test images corrupted with noise. It is found that the adaptive vector filters perform well with general applicability.

Index Terms—Adaptive filters, image enhancement, image processing, median filters, multidimensional digital filters.

I. INTRODUCTION

PLATANIOTIS *et al.* [1]–[8] propose a number of different vector filters for the processing of color images for the purpose of noise reduction. Their aim is to develop a noise-reduction filter that performs well in the presence of any type of noise. In this letter, the theory behind the color vector filters described by Plataniotis *et al.* is summarized with consistent notation, and it is shown that the filters are essentially variants of the same filtering procedure. Some of these filters have parameters that can be varied; however, Plataniotis *et al.* only try them for a few select values of the parameters. If, instead, the parameters are considered as continuously variable, as is done below, then it can be shown that the filters interpolate between the well-known arithmetic mean filter (AMF) and the vector median filter (VMF). A number of numerical simulations have been run to find the optimal settings for the parameters and to test the efficacy of the filters for restoring images contaminated with Gaussian distributed, impulse or speckle noise.

II. BASIC THEORY AND NOTATION

All filters described use a connected neighborhood Ω of N pixels around the pixel to be calculated. The color value, which is a vector in three-dimensional color space, of the i th pixel in the noisy image is represented by \mathbf{f}_i . The pixel \mathbf{f}_k of the resultant filtered image is calculated in some manner from all pixels \mathbf{f}_i in Ω . It is typical to use a square neighborhood centered on the location of \mathbf{f}_k .

Manuscript received February 12, 2003; revised April 27, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ricardo L. De Queiroz.

The author is with the Department of Physics and Electronic Engineering, University of Waikato, Hamilton, New Zealand (e-mail: m.cree@ieee.org).

Digital Object Identifier 10.1109/LSP.2003.821692

The simplest filter for smoothing color images is the AMF given by

$$\mathbf{f}_k = \frac{1}{N} \sum_{i \in \Omega} \mathbf{f}_i. \quad (1)$$

The AMF performs well on images contaminated with Gaussian distributed noise and poorly on images contaminated with spike noise. The AMF often blurs images unacceptably.

To apply a median filter to color images a function that maps the color pixel values to rankable scalars is required. Here we use a distance function $D(\mathbf{f}_i, \mathbf{f}_j)$ which returns a scalar value corresponding to the “distance” apart in color space of the color values of the i th and j th pixels in the noisy image. Then, for each pixel in Ω , the dissimilarity measure d_j is calculated according to

$$d_j = \sum_{i \in \Omega, i \neq j} D(\mathbf{f}_i, \mathbf{f}_j) \quad \forall j \in \Omega. \quad (2)$$

If the j th pixel is most like all other pixels in Ω , then the dissimilarity measure d_j will be small, whereas if the j th pixel is most unlike all other pixels in Ω , then d_j will be relatively large. The dissimilarity measure gives a ranking of pixels, and the vector median filter (VMF) chooses the pixel in Ω that is the most similar to all the others to replace the center pixel, namely that \mathbf{f}_k is given by [9]

$$\mathbf{f}_k = \{\mathbf{f}_j | d_j \leq d_i, \quad \forall i, j \in \Omega\}. \quad (3)$$

If more than one pixel in Ω has the minimal dissimilarity measure, then one of the minimal pixels’ color value is arbitrarily chosen to replace the central pixel.

It remains to choose a suitable distance function $D(\cdot, \cdot)$. A common choice for D is the L_1 -norm

$$D(\mathbf{x}, \mathbf{y}) = D_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^3 |x_i - y_i| \quad (4)$$

where x_i is the i th component of the color value \mathbf{x} , the summation is over the three components of a color value and $|x|$ is the absolute value of x . Other natural contenders for the distance function are the L_2 -norm (or Euclidean distance)

$$D(\mathbf{x}, \mathbf{y}) = D_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^3 (x_i - y_i)^2} \quad (5)$$

and the angle between the color pixel values

$$D(\mathbf{x}, \mathbf{y}) = D_a(\mathbf{x}, \mathbf{y}) = \cos^{-1} \left(\frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|} \right) \quad (6)$$

where $|\mathbf{x}|$ is the magnitude of the vector \mathbf{x} . Using (6) with the VMF constitutes the basic vector directional filter (BVDF)

[10]. The VMF performs poorly on images contaminated with Gaussian distributed noise and best on images contaminated with spike noise. Neither the AMF nor the VMF lend themselves to be used as a general-purpose filter when the noise type is unknown.

III. ADAPTIVE VECTOR FILTERS

To develop a general-purpose filter, Plataniotis *et al.* use adaptive vector filters, in which the pixel \mathbf{f}_k of the filtered image is calculated from the linear-combination of all the pixels in Ω [1], [2]

$$\mathbf{f}_k = \frac{\sum_{i \in \Omega} W(d_i) \mathbf{f}_i}{\sum_{i \in \Omega} W(d_i)} \quad (7)$$

where $W(x)$ is a weighting function chosen to be a nonnegative monotonically decreasing function of its argument. Therefore, pixels with low dissimilarity index are weighted highly in the linear combination, and pixels with high dissimilarity index are given a small weighting toward forming the new pixel value. This is based on the assumption that pixels that are most dissimilar to all other pixels in the neighborhood are most likely to be in error. It remains to choose a suitable weighting function.

The adaptive nearest neighbor filter (ANNF) uses the angular distance measure (6) and the weight function given by [1], [6]

$$W(d_i) = \frac{d_+ - d_i}{d_+ - d_-} \quad (8)$$

with $d_- = \min(\{d_i | i \in \Omega\})$ and $d_+ = \max(\{d_i | i \in \Omega\})$. The adaptive nearest neighbor multichannel filter (ANNMF) generalizes the ANNF with a scalar parameter $\alpha \in [0, 1)$ introduced into the weight function by [2]

$$W(d_i) = \frac{1}{1 + \alpha} \left(\frac{d_+ - d_i}{d_+ - d_-} \right) + \frac{\alpha}{1 + \alpha}. \quad (9)$$

With $\alpha = 0$, it reduces to the ANNF. Plataniotis *et al.* [2] require α to be between the limits of 0 and 1, but should we ignore the upper limit and let α approach infinity, then the ANNMF reduces to the AMF. The ANNMF can, therefore, be considered as interpolating between the AMF and the ANNF.

Plataniotis *et al.* [3]–[7] introduce a class of filters including the fuzzy vector filter (FVF), the fuzzy vector directional filter (FVDF), the fuzzy vector median filter (FVMF), fuzzy vector mean filter (FVMEF), the adaptive vector filter (AVF), and the adaptive vector directional filter (AVDF). The way they were introduced serves to hide the fact these filters are all of the same class and should be given a unified treatment. They are all based on (7), and all are referred to as adaptive vector filters (AVF) in this letter. The three primary weight functions proposed by Plataniotis *et al.* for use in the AVF are [3]–[7]

$$W_1(x) = \frac{1}{(1 + e^{\beta x})^r} \quad (10)$$

$$W_2(x) = e^{-\frac{x^r}{\beta}} \quad (11)$$

$$W_3(x) = \left(\frac{1}{x} \right)^r \quad (12)$$

with β and r parameters that are chosen to give a good general-purpose filter. Our version of W_1 has the parameter β in the exponential, whereas Plataniotis *et al.* had it in the numerator of

TABLE I
LIMITS OF DIFFERENT WEIGHT FUNCTIONS FOR THE AVF

	$r \rightarrow 0$	$r \rightarrow \infty$	$\beta \rightarrow 0$	$\beta \rightarrow \infty$
W_1	AMF	VMF	AMF	VMF
W_2	AMF	VMF	VMF	AMF
W_3	AMF	VMF		

W_1 where it will serve no purpose, since a constant multiplier gets cancelled out by the normalization process in (7).

IV. OBSERVATIONS

Plataniotis *et al.* [2] describe a version of the ANNMF that replaces α of (9) with an adaptively calculated α_i given by

$$\alpha_i = \frac{1}{d_+ - d_i}. \quad (13)$$

This is a mystifying choice, for it gives $W(d_i) = 1$ if $d_i = d_+$ or if $d_i = d_-$, and a value less than one otherwise; thus, (9) coupled with (13) does not lead to a monotonically decreasing weight function. This filter is, therefore, unlikely to perform as well as the plain ANNF, particularly for images contaminated with spike noise, since the error pixel (the spike noise) will most likely get the maximal weighting! Our numerical simulations confirm this observation (results not shown here).

Plataniotis *et al.* only report simulations for a few select choices of the parameters β and r for the AVF type filters. If we relax that restriction and consider the parameters as continuously variable, then we can make some general observations about the behavior of the AVF filters. For example, consider

$$\lim_{r \rightarrow 0} W_1 = \lim_{r \rightarrow 0} \frac{1}{(1 + e^{\beta x})^r} = 1. \quad (14)$$

Since the weights are all the same, this is identical to the AMF. In the other extreme

$$\lim_{r \rightarrow \infty} W_1 = \lim_{r \rightarrow \infty} \frac{1}{(1 + e^{\beta x})^r} \quad (15)$$

$$= \begin{cases} \text{very small (relatively)} & \text{for } x \text{ large} \\ \text{very large (relatively)} & \text{for } x \text{ small.} \end{cases} \quad (16)$$

Of course, the limit above is ultimately zero for all the weights, but by (16) above, we mean that for x large the limit tends to zero very much faster than for x small. This effect is so marked that only the smallest dissimilarity value leads to a significant weight. The filter, therefore, takes the pixel value corresponding to the smallest dissimilarity measure to replace the center pixel of the neighborhood Ω . This is the definition of the VMF. One can continue the process for the other parameter β and for the other weight functions W_2 and W_3 . The results are listed in Table I. Because these weight functions can continuously be varied from the AMF to the VMF, we can consider them as interpolating between the AMF and the VMF.

V. EXPERIMENTAL METHODOLOGY

The above filters have been implemented and tested for their efficacy at reducing noise in contaminated color images. The filters tested are the AMF (1), VMF (3), ANNF (8), AVF1 (10), AVF2 (11), and the AVF3 (12). The filters (except the AMF) were tested with each of the distance measures listed above;

TABLE II
NOISE TYPES ADDED TO TEST IMAGES

Description	Label	Parameter	SNR (dB)
Gaussian	g15	0.01384	19.5
Gaussian	g30	0.0346	13.7
Spike	i2	2%	17.1
Spike	i4	4%	14.0
Gaussian and Spike	g15i2	0.01384, 2%	15.1
Gaussian and Spike	g30i4	0.0346, 4%	11.0
Speckle	i11	0.05536	21.7
Speckle	i23	0.1384	16.0

however, results are only reported for the D_a and D_1 distance measures as the D_2 distance measure gave very similar results to the D_1 measure.

The results of testing on the well-known ‘‘Lena’’ image are reported here. The original and resultant filtered images are in 24-bit RGB format, and the filtering process was performed in the RGB color space using IEEE double-precision floating-point (approximately 16 digits precision) for internal calculations. A square 3×3 pixel neighborhood was used in all filters. The test images were contaminated using the Matlab routine ‘‘imnoise’’ [11]. Table II lists the types and amount of noise added. The ‘‘labels’’ in this table are used to identify the noise types in the tables of the Results section, and the parameter column refers to the parameter passed to the Matlab ‘‘imnoise’’ routine. The SNR of each of the contaminated ‘‘Lena’’ images is also listed in the table.

A filtered image \mathbf{f}^* was compared to its corresponding un-contaminated image \mathbf{f} and the SNR calculated by

$$\text{SNR} = 10 \log \frac{\sum_i |\mathbf{f}_i|^2}{\sum_i |\mathbf{f}_i - \mathbf{f}_i^*|^2} \quad (\text{dB}) \quad (17)$$

where the summations are over all pixels in the image less a two-pixel-wide boundary around the edges of the image. As well as calculating the SNR in RGB space (the color space the images were filtered in), the original and filtered images were converted to CIE 1976 $L^*a^*b^*$ (CIELab) space [12], wherein the SNR ratio was also calculated. The CIELab color space has the property that the Euclidean distance between two color vectors gives the human perceived color error between the two vectors.

The AVF1, AVF2, and AVF3 filters have parameters yet to be set. These filters were run on the contaminated ‘‘Lena’’ images with different values of r and β . The SNR only changed slowly for varying r and β over reasonable ranges, so a near optimal value of r and β was easily chosen by inspection of the results. Table III lists the SNR (calculated in RGB space) for running AVF1, using D_1 as the distance measure, on the Lena image contaminated with Gaussian distributed noise (g30), contaminated with spike noise (i4) and contaminated with both Gaussian and spike noise (g30i4). It can be seen how the AVF1 tends to the AMF for low r and β and performs well on the Gaussian contaminated image, and how the AVF1 tends to the VMF for large r and β and then performs best on the image contaminated with spike noise. It can also be seen how the filter starts to fail for extreme values of r and β due to numerical overflows/underflows. The values of $r = 0.75$, $\beta = 0.005$ were chosen as

TABLE III
BEHAVIOR (SNR CALCULATED IN RGB SPACE) OF AVF1 FOR DIFFERENT VALUES OF r AND β WHEN RUN ON ‘‘LENA’’ IMAGE USING THE D_1 DISTANCE MEASURE

		β				
		g30	0.001	0.005	0.02	0.25
r	0.10	21.8	21.8	21.6	19.6	19.2
	0.25	21.8	21.7	21.0	18.9	18.5
	0.50	21.8	21.5	20.4	18.4	14.7
	0.75	21.8	21.2	20.0	15.1	11.3
	1.00	21.8	21.0	19.8	13.8	11.0
		β				
		g30i4	0.001	0.005	0.02	0.25
r	0.10	19.6	20.4	21.0	19.2	18.5
	0.25	19.9	20.9	20.6	18.5	14.7
	0.50	20.3	20.9	20.0	14.7	11.3
	0.75	20.5	20.8	19.6	11.3	11.0
	1.00	20.7	20.6	19.4	11.0	11.0
		β				
		i4	0.001	0.005	0.02	0.25
r	0.10	22.3	24.3	27.5	28.6	28.5
	0.25	22.9	26.5	28.4	28.5	26.2
	0.50	23.8	27.8	28.6	26.2	21.3
	0.75	24.6	28.2	28.6	21.3	18.0
	1.00	25.2	28.4	28.6	18.0	18.0

TABLE IV
NEAR-OPTIMAL VALUES OF r AND β FOR THE THREE AVF FILTERS

Filter	AVF1		AVF2		AVF3	
	D_1	D_a	D_1	D_a	D_1	D_a
r	0.75	0.75	0.5	0.5	1	2
β	0.005	0.75	5	0.5		

being near optimal for unknown noise type. Table IV lists the values of r and β chosen as near optimal for each filter under the assumption of unknown noise type.

VI. RESULTS AND DISCUSSION

The results, given as SNRs calculated in RGB space, of running the AMF, VMF, ANNF, AVF1, AVF2, and AVF3 on each of the contaminated images is presented in Table V. All the filters (except the AMF) were tested with both the D_1 and D_a distance measures, but for the AVF filters only test results using the D_1 measure are presented here. The noise-type labels used in Table V are defined in Table II.

As expected the AMF performs best on Gaussian and speckle type noise and the VMF on the images contaminated with spike noise. The VMF when using the D_a distance measure, performs poorly compared to the VMF using the D_1 measure if the SNR is calculated in RGB color space. When the SNR is calculated in CIELab color space the VMF using D_a is generally the better measure (results not shown). Visual inspection of the images leads this investigator to prefer the VMF calculated with the D_1 measure and there appears to be little reason to prefer any distance measure other than the relatively computationally inexpensive D_1 measure.

The results of running the ANNF with the D_a and D_1 distance measures on each of the contaminated versions of the Lena

TABLE V
SNR OF RESTORED IMAGES USING VARIOUS FILTERS ON THE LENA IMAGE. THE NOISE TYPE LABELS ARE AS GIVEN IN TABLE II

	None	AMF	VMF		ANNF		AVF1	AVF2	AVF3
			D_1	D_a	D_1	D_a	D_1	D_1	D_1
original	∞	27.5	28.4	27.3	28.7	28.5	28.7	28.7	28.6
g15	19.5	25.2	22.7	20.4	25.0	24.7	25.5	25.4	25.4
g30	13.7	21.8	18.1	15.9	20.8	20.7	21.2	21.3	21.5
g15i2	15.1	22.7	22.5	20.2	24.9	24.5	25.3	25.2	25.0
g30i4	11.0	19.4	17.7	15.4	20.7	20.2	20.8	20.8	20.8
i2	17.1	23.9	28.2	26.8	28.3	27.9	28.5	28.5	28.3
i4	14.0	21.8	28.0	26.4	27.6	26.9	28.2	28.2	27.7
s11	21.7	25.9	23.3	21.0	25.4	25.6	26.2	26.0	26.0
s23	16.0	23.2	18.7	16.5	21.3	21.9	22.2	22.2	22.4

image are also presented in Table V. The ANNF performs significantly better than the VMF on Gaussian noise and the AMF on speckle noise. It is comparable to the VMF on speckle noise and the AMF on Gaussian noise. The choice of distance function has relatively little bearing on the performance of the ANNF. These conclusions are the same as those that can be deduced from the SNR calculated in CIELab color space (CIELab results not shown here). We suggest that one may exclusively use D_1 and take advantage of its simplicity and relatively low computational overhead.

The results of the AVF filters (AVF1, AVF2, AVF3) run with the D_1 distance measure is also listed in Table V. They perform equivalently to each other and provide a slight improvement over the ANNF. Notably the AVF filters perform a little better than the AMF for low amounts of Gaussian distributed noise and the VMF for spike noise. Visual inspection of the filtered images reveals that the AVF filters do not blur images to the extent that the AMF does. Therefore, these results suggest that the AVF filters may be a better choice than the AMF or the VMF even when the noise type is known *a priori*!

VII. CONCLUSION

The adaptive vector filters of Plataniotis *et al.* [1]–[8] are essentially variants of the same filtering procedure and were summarized with a unified notation. It was shown that the filters can be considered to interpolate between the arithmetic mean filter and the vector median filter. The choice of distance function for these filters has little bearing on the performance of the filter;

therefore, the computationally inexpensive L_1 norm is a good choice. The filters perform well as general-purpose filters when the noise type is unknown *a priori*.

REFERENCES

- [1] K. N. Plataniotis, D. Androustos, V. Sri, and A. N. Venetsanopoulos, "Nearest-neighbor multichannel filter," *Electron. Lett.*, vol. 31, pp. 1910–1911, 1995.
- [2] K. N. Plataniotis, V. Sri, D. Androustos, and A. N. Venetsanopoulos, "An adaptive nearest neighbor multichannel filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 699–703, Dec. 1996.
- [3] K. N. Plataniotis, D. Androustos, and A. N. Venetsanopoulos, "Fuzzy adaptive filters for multichannel image processing," *Signal Process.*, vol. 55, pp. 93–106, 1996.
- [4] —, "Multichannel filter for image processing," *Signal Process. Image Commun.*, vol. 9, pp. 143–158, 1997.
- [5] —, "Color image filters: the vector directional approach," *Opt. Eng.*, vol. 36, pp. 2375–2383, 1997.
- [6] K. N. Plataniotis and A. N. Venetsanopoulos, "Vector filtering," in *The Color Image Processing Handbook*, S. J. Sangwine and R. E. N. Horne, Eds. London, U.K.: Chapman & Hall, 1998, pp. 188–209.
- [7] K. N. Plataniotis, D. Androustos, and A. N. Venetsanopoulos, "Color image processing using adaptive vector directional filters," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 1414–1415, Oct. 1998.
- [8] —, "Adaptive fuzzy systems for multichannel signal processing," *Proc. IEEE*, vol. 87, pp. 1601–1622, Sept. 1999.
- [9] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, vol. 78, pp. 678–689, Apr. 1990.
- [10] P. E. Trahanias and A. N. Venetsanopoulos, "Vector directional filters—A new class of multichannel image processing filters," *IEEE Trans. Image Processing*, vol. 2, pp. 528–534, Oct. 1993.
- [11] The MathWorks Inc., *Image Processing Toolbox*, 3rd ed., The MathWorks Inc., Natick, MA, 2001.
- [12] CIE, "Colorimetry," *Centr. Bureau CIE, Vienna, Austria, CIE Pub. 15.2*, 1986.