Journal of Electronic Imaging 16(3), 033008 (Jul–Sep 2007)

Nonlinear vector filtering for impulsive noise removal from color images

M. Emre Celebi
Louisiana State University in Shreveport
Department of Computer Science
Shreveport, Louisiana 71115
E-mail: ecelebi@lsus.edu

Hassan A. Kingravi
Georgia Institute of Technology
Department of Computer Science
Atlanta, Georgia 30332

Y. Alp Aslandogan
Prairie View A&M University
Department of Computer Science
Prairie View, Texas 77446

Abstract. *A comprehensive survey of 48 filters for impulsive noise removal from color images is presented. The filters are formulated using a uniform notation and categorized into 8 families. The performance of these filters is compared on a large set of images that cover a variety of domains using three effectiveness and one efficiency criteria. In order to ensure a fair efficiency comparison, a fast and accurate approximation for the inverse cosine function is introduced. In addition, commonly used distance measures (Minkowski, angular, and directional-distance) are analyzed and evaluated. Finally, suggestions are provided on how to choose a filter given certain requirements.* © 2007 SPIE and IS&T. [DOI: 10.1117/1.2772639]

1 Introduction

The growing use of color images in diverse applications such as medical image analysis, content-based image retrieval, remote sensing, and visual quality inspection has led to an increasing interest in color image processing. These applications involve many of the same tasks as their gray-scale counterparts, such as edge detection, segmentation, and feature extraction.[1] However, color images are often contaminated with noise, which not only lowers their visual quality, but also complicates automated processing. Therefore, the removal of such noise is often a necessary preprocessing step for color image processing applications.[2]

Image noise can come from many sources and can be introduced into an image during either acquisition or transmission through sensors or communication channels, respectively.[3] "Impulsive noise" is noise of low duration and high energy that can be caused either by faulty sensors or by electrical disturbances such as lightning and the operation of high-voltage machinery corrupting the transmission signal.[4] The introduction of such noise into an image is often detrimental to its future usage. If the image is meant for human consumption, the presence of noise lowers its perceptual quality. On the other hand, if it is to be processed further, the noise can make complex tasks such edge detection and segmentation even more difficult.

Numerous filters have been proposed in the literature for impulsive noise removal from color images. Among these, nonlinear filters have proved successful in the preservation of edges and fine image details while removing the noise.[5] The early approaches to nonlinear filtering of color images often involved the application of a scalar filter to each color channel independently. However, since separate processing ignores the inherent correlation between the color channels, these methods often introduce color artifacts to which the human visual system is very sensitive.[6] Therefore, vector filtering techniques that treat the color image as a vector field and process color pixels as vectors are more appropriate.[7] An important class of nonlinear vector filters is the one based on robust order statistics, with the vector median filter (VMF)[8] being the most widely known example. These filters involve reduced ordering[9,10] of a set of input vectors within a window to compute the output vector. Recent applications of these include enhancement of cDNA microarray images,[11,12] virtual restoration of artwork,[13,14] and video filtering.[15–18]

The motivation of this study is twofold. First, a large number of nonlinear vector filters have been proposed in the literature since 1990. Therefore, a study that categorizes and presents these filters in a unified notation is desirable. Second, to the best of the authors' knowledge, no study to date has objectively compared the performance of these

Paper 06119R received Jul. 3, 2006; revised manuscript received Jan. 20, 2007; accepted for publication Feb. 5, 2007; published online Oct. 1, 2007.
1017-9909/2007/16(3)/033008/21/$25.00 © 2007 SPIE and IS&T.

filters on a large and diverse set of images. A similar study[19] presents a detailed survey of the nonlinear vector filters, noise models, filtering performance criteria, and applications; however, it does not provide an experimental comparison of these filters.

In this study, 48 impulsive noise removal filters are presented in a systematic fashion and categorized into 8 families. Furthermore, the performance of these filters in terms of both effectiveness and efficiency are compared on a set of 100 images that cover a multitude of domains. In order to ensure fairness in the efficiency comparisons, a fast and accurate approximation for the inverse cosine function (used in many of the filters) is introduced. In addition, the relative merits of commonly used distance measures (Minkowski, angular, and directional-distance) are analyzed and compared. Finally, suggestions are provided on how to choose a filter given certain requirements.

The rest of the paper is organized as follows. Section 2 introduces the notation and categorizes the filters. Section 3 describes the image set, the noise models, and the filtering performance criteria. Finally, Section 4 discusses the experimental results and gives the conclusions.

2 Categorization of the Filters

In this section, the 48 impulsive noise removal filters are categorized into 8 groups as follows:

1. Basic vector filters
2. Adaptive fuzzy vector filters
3. Hybrid vector filters
4. Adaptive center-weighted vector filters
5. Entropy vector filters
6. Peer group vector filters
7. Vector sigma filters
8. Miscellaneous vector filters

The notation used in the descriptions of these filters is shown in Table 1. Note that the author-recommended parameter values for each filter are indicated in the descriptions.

2.1 *Basic Vector Filters*

These are the earliest impulsive noise removal filters proposed in the literature. The subsequent, more advanced filters are more or less based on these basic filters. Table 2 shows the mathematical expressions for these filters.

2.1.1 *Vector median filter*

The vector median filter (VMF)[8] and its extensions[20,21] follow directly from the concept of the nonlinear order statistics in that the output of the filter is the lowest-ranked vector in the window. The VMF orders the color input vectors according to their relative magnitude differences using the Minkowski metric as a distance measure. The two most widely used such measures are the $L1$- (Manhattan distance) and the $L2$- (Euclidean distance) norms.[22]

2.1.2 *Alpha-trimmed vector median filter*

The alpha-trimmed vector median filter (ATVMF)[18] selects the lowest-ranked $1+\alpha$ vectors as input to an averaging filter. The trimming operation guarantees good performance in the presence of impulsive noise. In addition, the averaging operation helps the filter cope with Gaussian noise. The parameter α is set to $\lfloor n/2 \rfloor$.

2.1.3 *Basic vector directional filter*

Another method for detecting the outliers in a window is to rank the color vectors based on the orientation difference between them. In other words, vectors with atypical directions are considered to be outliers. The basic vector directional filter (BVDF)[23] uses this concept in a manner similar to the VMF, by using the angle between two color vectors as the distance criterion. Since a vector's direction corresponds to its chromaticity,[24] this filter preserves the chromaticity of the input vectors better than the VMF.

2.1.4 *Generalized vector directional filter*

The generalized vector directional filter (GVDF)[24] is a generalization of the BVDF in that its output is a superset of the single BVDF output. After the vectors are ranked according to the angular distance criterion, a set of low-rank vectors is selected as input to an additional filter to produce a single output vector. In the second step, only the magnitudes of the vectors are considered. Thus, any grayscale filter[25] such as the arithmetic mean filter (AMF), the multistage median filter, and various morphological filters can be used. In this study, the AMF is used for magnitude processing.

2.1.5 *Directional distance filter*

The directional distance filter (DDF)[26,27] is a combination of the VMF and the BVDF derived by the simultaneous minimization of their defining functions (see Table 2). The motivation behind this is to incorporate information about both a vector's magnitude (brightness) and its direction (chromaticity) in the calculation of the distance metric. The parameter γ in this case controls the relative importance of each component. This parameter is set to 0.5, which implies an equal consideration for both measures.

2.1.6 *Content-based ranked filter*

The content-based ranked filter (CBRF),[28] like the DDF, ranks the vectors according to a distance metric that incorporates more information about the vector as a whole than the criteria used by the VMF and the BVDF. The similarity between two vectors in this case can be expressed as the ratio of some function of what they share (commonality) to what they comprise (totality).[29] The numerator (commonality) and the denominator (totality) correspond to the vector difference and the vector sum, respectively.

2.2 *Adaptive Fuzzy Vector Filters*

These filters utilize data-dependent coefficients to adapt to local image characteristics.[11,30,31] The general form of an adaptive fuzzy vector filter is given as a nonlinear transformation of a fuzzy weighted average of the input vectors within a window W:

Table 1 Notations used in the study.

Notation	Meaning
N	Number of pixels in an image
W	Filtering window
n	Number of pixels in W
x_i	ith pixel in W
x_i^k	kth component of x_i (k=1: red, k=2: green, k=3: blue)
$x_{(i)}$	Pixel with the ith ranking according to a particular ordering scheme
x_f	Output of a particular filter 'f' within W
$C=(n+1)/2$	Index of the center pixel in W
$\|\|x_i\|\|=(x_i^1\cdot x_i^1+x_i^2\cdot x_i^2+x_i^3\cdot x_i^3)^{1/2}$	Euclidean norm of x_i
$\bar{x}=x_{\mathrm{AMF}}=\frac{1}{n}\sum_{i=1}^{n}x_i$	Mean vector within W. Also, the output of the arithmetic mean filter (AMF)
$\langle x_i,x_j\rangle=x_i^1\cdot x_j^1+x_i^2\cdot x_j^2+x_i^3\cdot x_j^3$	Inner product between x_i and x_j
$D(x_i,x_j)$	Distance between x_i and x_j according to a particular measure
$L_p(x_i,x_j)=\|\|x_i-x_j\|\|_p=\left(\sum_{k=1}^{3}\|x_i^k-x_j^k\|^p\right)^{1/p}$	Minkowski distance between x_i and x_j
$l(i)=l_p(i)=\sum_{j=1}^{n}L_p(x_i,x_j)$	Cumulative Minkowski distance associated with x_i
$A(x_i,x_j)=\cos^{-1}\left(\frac{\langle x_i,x_j\rangle}{\|\|x_i\|\|\cdot\|\|x_j\|\|}\right)$	Angular distance between x_i and x_j
$a(i)=\sum_{j=1}^{n}A(x_i,x_j)$	Cumulative angular distance associated with x_i
$d(i)=\left(\sum_{j=1}^{n}A(x_i,x_j)\right)^{\gamma}\cdot\left(\sum_{j=1}^{n}L_p(x_i,x_j)\right)^{1-\gamma}$	Cumulative directional distance associated with x_i

$$x_{afvf}=g\left(\frac{\sum_{i=1}^{n}w_i x_i}{\sum_{i=1}^{n}w_i}\right), \quad (1)$$

where x_{afvf} is the filter output, $g(.)$ is a nonlinear function, and $w_i \geq 0 \Leftrightarrow x_i$ are the fuzzy weights that correspond to each input vector. The weights provide the degree to which an input vector contributes to the filter output and are determined by fuzzy transformations of the cumulative distances associated with each input vector.

2.2.1 *Fuzzy-weighted average filters*

In the fuzzy-weighted average filters (FWAFs), the function $g(.)$ is the identity function:

Table 2 Basic vector filters.

Filter	Formulation
VMF	$x_{VMF}=\mathrm{argmin}_{x_i\in W}(l(i))$
ATVMF	$x_{ATVMF}=\dfrac{1}{1+\alpha}\sum_{i=1}^{1+\alpha} x_{(i)}, \quad \alpha\in[0,n-1]$
BVDF	$x_{BVDF}=\mathrm{argmin}_{x_i\in W}(a(i))$
DDF	$x_{DDF}=\mathrm{argmin}_{x_i\in W}(d(i))$
CBRF	$x_{CBRF}=\mathrm{argmin}_{x_i\in W}\sum_{j=1}^{n} G(x_i,x_j)$ $G(x_i,x_j)=\left(\dfrac{\|x_i\|^2+\|x_j\|^2-2\|x_i\|\|x_j\|\cos(\theta)}{\|x_i\|^2+\|x_j\|^2+2\|x_i\|\|x_j\|\cos(\theta)}\right)^{1/2}$

$$x_{fwaf}=\frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}. \tag{2}$$

Because of the averaging operation, the filter output x_{fwaf} is generally not included in the input vector set $\{x_1,x_2,\ldots,x_n\}$. This allows better performance in the presence of Gaussian noise when compared to pure order statistics-based filters that select the output vector from the set of input vectors. Note that depending on the distance criterion and the corresponding fuzzy transformation, various fuzzy filters can be derived from Eq. (2).

Fuzzy vector median filter. In the fuzzy vector median filter (FVMF),[30–32] the Minkowski metric is used as the distance function and the fuzzy membership function has an exponential form. In this case the fuzzy weights are given by

$$w_i=\exp(-l^{\gamma}(i)/\beta) \quad \text{for } i=1,2,\ldots,n, \tag{3}$$

where γ and β are parameters that control the amount of fuzziness in the weights.[33] The following values are used for these parameters: $\gamma=0.5$ and $\beta=1.0$.

Fuzzy vector directional filter. In the fuzzy vector directional filter (FVDF),[30,31] the vector angle metric is used as the distance function and the fuzzy membership function has a sigmoidal form. In this case the fuzzy weights are given by

$$w_i=\frac{\beta}{(1+\exp(a(i)))^{\gamma}} \quad \text{for } i=1,2,\ldots,n, \tag{4}$$

where γ is a parameter that can be used to adjust the weighting effect of the membership function and β is a weight-scale threshold. The following values are used for these parameters: $\gamma=1.0$ and $\beta=2.0$.

Adaptive nearest-neighbor filter. In the adaptive nearest-neighbor filter (ANNF),[34] the fuzzy weights are determined as follows:

$$w_i=\frac{a_{(n)}-a_{(i)}}{a_{(n)}-a_{(1)}} \quad \text{for } i=1,2,\ldots,n, \tag{5}$$

where $a_{(n)}$ and $a_{(1)}$ are the maximum and minimum cumulative angular distances, respectively. It should be noted that other distance measures such as the Minkowski and directional-distance functions can also be used in Eq. (5).

Adaptive nearest-neighbor multichannel filter. The adaptive nearest-neighbor multichannel filter (ANNMF)[35] is a modification of the ANNF that uses a composite distance function rather than an angular one:

$$D(x_i,x_j)=1-\left(\frac{\langle x_i,x_j\rangle}{\|x_i\|\cdot\|x_j\|}\right)\left(1-\frac{|\|x_i\|-\|x_j\||}{\max(\|x_i\|,\|x_j\|)}\right). \tag{6}$$

2.2.2 *Fuzzy ordered vector filters*

The fuzzy ordered vector filters FOVF[31,36] are a fuzzy generalization of the alpha-trimmed filters in which the input vectors are ordered according to their fuzzy membership strengths and only those vectors with the largest fuzzy weights contribute to the output vector:

Table 3 Hybrid vector filters.

Filter	Formulation
EXVMF	$x_{\mathrm{EXVMF}}=\begin{cases} x_{\mathrm{AMF}} & \text{if } l(x_{\mathrm{AMF}})\le l(x_{\mathrm{VMF}}) \\ x_{\mathrm{VMF}} & \text{otherwise}\end{cases}$
HDF	$x_{\mathrm{HDF}}=\begin{cases} x_{\mathrm{VMF}} & \text{if } x_{\mathrm{VMF}}=x_{\mathrm{BVDF}} \\ \frac{\lVert x_{\mathrm{VMF}}\rVert}{\lVert x_{\mathrm{BVDF}}\rVert}\cdot x_{\mathrm{BVDF}} & \text{otherwise}\end{cases}$
AHDF	$x_{\mathrm{AHDF}}=\begin{cases} x_{\mathrm{VMF}} & \text{if } x_{\mathrm{VMF}}=x_{\mathrm{BVDF}} \\ x_{\mathrm{out1}} & \text{if } l(x_{\mathrm{out1}})\le l(x_{\mathrm{out2}}) \\ x_{\mathrm{out2}} & \text{otherwise}\end{cases}$ $x_{\mathrm{out1}}=\frac{\lVert x_{\mathrm{VMF}}\rVert}{\lVert x_{\mathrm{BVDF}}\rVert}\cdot x_{\mathrm{BVDF}},\quad x_{\mathrm{out2}}=\frac{\lVert x_{\mathrm{AMF}}\rVert}{\lVert x_{\mathrm{BVDF}}\rVert}\cdot x_{\mathrm{BVDF}}$
VMRHF	$x_{\mathrm{VMRHF}}=x_{\mathrm{CWVMF}}+\frac{\alpha_1\cdot x_{\mathrm{VMF}_1}+\alpha_2\cdot x_{\mathrm{CWVMF}}+\alpha_3\cdot x_{\mathrm{VMF}_2}}{\beta_1+\beta_2\cdot\lVert x_{VMF_1}-x_{\mathrm{VMF}_2}\rVert}$
FVMRHF	$x_{\mathrm{FVMRHF}}=x_{\mathrm{FCWVMF}}+\frac{\alpha_1\cdot x_{\mathrm{FVMF}_1}+\alpha_2\cdot x_{\mathrm{FCWVMF}}+\alpha_3\cdot x_{\mathrm{FVMF}_2}}{\beta_1+\beta_2\cdot\lVert x_{\mathrm{FVMF}_1}-x_{\mathrm{FVMF}_2}\rVert}$
FVDRHF	$x_{\mathrm{FVDRHF}}=x_{\mathrm{FCWVDF}}+\frac{\alpha_1\cdot x_{\mathrm{FVDF}_1}+\alpha_2\cdot x_{\mathrm{FCWVDF}}+\alpha_3\cdot x_{\mathrm{FVDF}_2}}{\beta_1+\beta_2\cdot A(x_{\mathrm{FVDF}_1},x_{\mathrm{FVDF}_2})}$ $\qquad \sum_{i=1}^{3}\alpha_i=0$
FDDRHF	$x_{\mathrm{FDDRHF}}=x_{\mathrm{FCWDDF}}+\frac{\alpha_1\cdot x_{\mathrm{FDDF}_1}+\alpha_2\cdot x_{\mathrm{FCWDDF}}+\alpha_3\cdot x_{\mathrm{FDDF}_2}}{\beta_1+\beta_2\cdot[A^{\gamma}(x_{\mathrm{FDDF}_1},x_{\mathrm{FDDF}_2})\cdot\lVert x_{\mathrm{FDDF}_1}-x_{\mathrm{FDDF}_2}\rVert^{1-\gamma}]}$
KVMF	$x_{\mathrm{KVMF}}=\mu(\lVert x_C-x_{\mathrm{VMF}}\rVert)\cdot x_C+(1-\mu(\lVert x_C-x_{\mathrm{VMF}}\rVert))\cdot x_{\mathrm{VMF}}$ $\mu(d)=\exp(-d/h),\quad h\cong\frac{\beta}{\left(\sum_{i=1}^{N}\lVert x_i-\bar{x}\rVert/8N^2\right)^{1/2}}$

$$x_{\mathrm{FOVF}}=\frac{\sum_{i=1}^{k} w_{(i)}x_{(i)}}{\sum_{i=1}^{k} w_{(i)}},\quad k\in[1,n],\tag{7}$$

where $x_{(k)}\le x_{(k-1)}\le\cdots\le x_{(1)}$, are the vectors with the k largest weights $w_{(k)}\le w_{(k-1)}\le\cdots\le w_{(1)}$ respectively.

The number of vectors (k) can be determined adaptively by considering only those input vectors with fuzzy weights greater than $1/n$.[30] Note that any fuzzy membership function such as (3), (4), or (5) can be used to determine the weights in (7). In this study, only the fuzzy ordered vector median filter (FOVMF) [Eqs. (3) and (7)] and the fuzzy ordered vector directional filter (FOVDF) [Eqs. (4) and (7)] are considered.

2.3 *Hybrid Vector Filters*

These filters utilize a number of subfilters of different types (hence the term "hybrid") and define the output as a linear or nonlinear combination of the input vectors.[37] Consequently, the output is often not included in the input set. Table 3 shows the mathematical expressions for these filters.

2.3.1 *Extended vector median filter*

The extended vector median filter (EXVMF)[8] combines the VMF with linear filtering to compensate for the deficiency of the VMF in dealing with Gaussian noise. Near edges this filter behaves like the VMF and preserves the details, while in smooth areas it behaves like the AMF, resulting in improved noise attenuation.

2.3.2 *Hybrid directional filter*

The hybrid directional filter (HDF)[38] is also based on the concept of independent vectorial attribute processing introduced in the DDF. It can be thought of as a nonlinear combination of the VMF and the BVDF filters.

2.3.3 *Adaptive hybrid directional filter*

The adaptive hybrid directional filter (AHDF)[38] is an extension of the HDF that utilizes the AMF in the filter structure. This is so the magnitude of the output vector will be that of the mean vector in smooth regions and that of the median operator near edges. Note that the criterion for the selection of the output vector in this filter is similar to the one used in the EXVMF.

2.3.4 *Vector median-rational hybrid filter*

The vector median-rational hybrid filter (VMRHF)[39–41] is a multichannel extension of the median-rational hybrid filter that combines the output of three subfilters (two vector median filters and a center weighted vector median filter*) in a rational function. It differs from a linear low-pass filter mainly due to the scaling, which is essentially an edge-sensing term characterized by the Euclidean distance between the two VMF outputs. The coefficient vector $\alpha = [\alpha_1 \alpha_2 \alpha_3]$ in the numerator is chosen a priori and serves to weight the outputs of the three subfilters. The parameters β_1 and β_2 in the denominator are positive constants. The former ensures numerical stability while the latter regulates the nonlinearity. The masks utilized by each subfilter are as follows:

$$\mathrm{VMF}_1: \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathrm{CWVMF}: \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \tag{8}$$

$$\mathrm{VMF}_2: \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Note that only those pixels with nonzero coefficients are considered in each of these masks. The parameter values are chosen as follows: $\alpha_1 = 1.0$, $\alpha_2 = -2.0$, $\alpha_3 = 1.0$, $\beta_1 = 3.0$, and $\beta_2 = 3.0$.

2.3.5 *Fuzzy rational hybrid filters*

The fuzzy rational hybrid filters[36,42,43] are a family of adaptive hybrid filters that are derived from the VMRHF. In the fuzzy vector median-rational hybrid filter (FVMRHF), one of the subfilters is a fuzzy center-weighted vector median filter (FCWVMF) and the other two are fuzzy vector median filters (FVMF). The fuzzy weights for these subfilters are given by

$$w_i = \frac{2}{1 + \exp(l^{\gamma}(i))} \quad \text{for } i = 1, 2, \ldots, n. \tag{9}$$

The fuzzy vector directional-rational hybrid filter (FVDRHF) and the fuzzy directional distance-rational hybrid filter (FDDRHF) are the angular and the directional-distance counterparts of the FVMRHF, respectively. The smoothing parameter γ is set to 1.0, and for the remaining parameters the VMRHF values are used.

2.3.6 *Kernel vector median filter*

The kernel vector median filter (KVMF)[44–48] outputs a vector that lies somewhere between the center pixel and the VMF output. In other words, the output vector is a linear combination of the two vectors. The weights are determined by the kernel μ for which several choices such as Laplacian, Gaussian, Cauchy, Epanechnikov, etc. are available. Table 3 gives the filter formulation for the Laplacian kernel with the normalization factor β and the kernel width h. The value of β depends on the kernel of choice ($\beta = 0.5$ for the Laplacian kernel). The parameter h can be estimated from the entire image as shown in Table 3.

The operation of this filter represents a compromise between the VMF and the identity operation. The kernel is a function of the distance between the center pixel and the VMF output; if the center pixel is not noisy, then the kernel function is close to 1, and the output will be close to the original value of the center pixel. Otherwise, the output will be close to the VMF output.

2.4 *Adaptive Center-Weighted Vector Filters*

The vector median filter can be generalized by associating with each pixel x_i a nonnegative integer-valued weight[18,49]:

$$x_{\mathrm{WVMF}} = \underset{x_i \in W}{\operatorname{argmin}} \left(\sum_{j=1}^{n} w_j \|x_i - x_j\| \right). \tag{10}$$

This filter is called the weighted vector median filter (WVMF). Note that by replacing the distance function in (10) with the angular or directional-distance functions, one can obtain the analogous weighted vector directional filter (WVDF) or weighted directional-distance filter (WDDF), respectively.[50,51]

The flexible form of the weighted vector filters allows one to design an optimal filter for a particular domain by adjusting the weights. The weights are often determined by an optimization procedure using a number of training images.[50,52–54] If only the center weight is varied while the others are fixed, the WVMF simplifies to the center-weighted vector median filter (CWVMF)[55,56]:

$$x_{\mathrm{CWVMF}^k} = \underset{x_i \in W}{\operatorname{argmin}} \left(\sum_{j=1}^{n} w_j(k) \cdot \|x_i - x_j\| \right),$$

$$w_j(k) = \begin{cases} n - 2k + 2 & \text{for } j = C, \\ 1 & \text{otherwise,} \end{cases}, \ k \in [1, C]. \tag{11}$$

When the smoothing parameter $k = 1$, the CWVMF is equivalent to the identity filter and thus no smoothing is performed. As the value of k is increased, the smoothing capability of the filter increases. Finally, when k attains its maximum value C, the filter becomes equivalent to the VMF, and the maximum amount of smoothing is per-

*See Section 2.4.

Table 4 Adaptive center-weighted vector filters.

Filter	Formulation
MCWVMF	$x_{\text{MCWVMF}} = \begin{cases} x_{\text{VMF}} & \text{if } I(x_{\text{VMF}}) < w \cdot I(C), \\ x_C & \text{otherwise}, \end{cases} \quad w \in [0,1]$
ACWVMF	$x_{\text{ACWVMF}} = \begin{cases} x_{\text{VMF}} & \text{if } \sum_{k=\lambda}^{\lambda+2} \lVert x_{\text{CWVMF}^k} - x_C \rVert > T, \\ x_C & \text{otherwise}, \end{cases} \quad \lambda \in [1, C-1]$
ACWVDF	$x_{\text{ACWVDF}} = \begin{cases} x_{\text{BVDF}} & \text{if } \sum_{k=\lambda}^{\lambda+2} A(x_{\text{CWVDF}^k}, x_C) > T, \\ x_C & \text{otherwise}, \end{cases} \quad \lambda \in [1, C-1]$
ACWDDF	$x_{\text{ACWDDF}} = \begin{cases} x_{\text{DDF}} & \text{if } \sum_{k=\lambda}^{\lambda+2} A^{\gamma}(x_{\text{CWDDF}^k}, x_C) \cdot \lVert x_{\text{CWDDF}^k} - x_C \rVert^{1-\gamma} > T, \\ x_C & \text{otherwise} \end{cases} \quad \lambda \in [1, C-1]$

Table 5 Entropy vector filters.

Filter	Formulation
EVMF	$x_{\text{EVMF}} = \begin{cases} x_{\text{VMF}} & \text{if } P_C > T_C \\ x_C & \text{otherwise} \end{cases}$ $P_i = \frac{\lVert x_i - \bar{x} \rVert}{\sum_{j=1}^{n} \lVert x_j - \bar{x} \rVert}, \quad T_i = \frac{-P_i \log P_i}{-\sum_{j=1}^{n} P_j \log P_j}$
EBVDF	$x_{\text{EBVDF}} = \begin{cases} x_{\text{BVDF}} & \text{if } P_C > T_C \\ x_C & \text{otherwise} \end{cases}$ $P_i = \frac{A(x_i, \bar{x})}{\sum_{j=1}^{n} A(x_j, \bar{x})}, \quad T_i = \frac{-P_i \log P_i}{-\sum_{j=1}^{n} P_j \log P_j}$
EDDF	$x_{\text{EDDF}} = \begin{cases} x_{\text{DDF}} & \text{if } P_C > T_C \\ x_C & \text{otherwise} \end{cases}$ $P_i = \frac{A(x_i, \bar{x})^{\gamma} \lVert x_i - \bar{x} \rVert^{1-\gamma}}{\sum_{j=1}^{n} A(x_j, \bar{x})^{\gamma} \lVert x_j - \bar{x} \rVert^{1-\gamma}}, \quad T_i = \frac{-P_i \log P_i}{-\sum_{j=1}^{n} P_j \log P_j}$

Table 6 Peer group vector filters.

Filter	Formulation
PGF	$c(i)=\lVert x_C-x_i\rVert$ for $i=1,2,\ldots,n$ $\delta(i)=c_{(i+1)}-c_{(i)}$ for $i=1,2,\ldots,m=(\sqrt{n}+1)/2$ $x_{\mathrm{PGF}}=\begin{cases}x_{\mathrm{VMF}} & \text{if } \exists\, i\in[1,m]\ s.t.\ \delta(i)>T\\ x_C & \text{otherwise}\end{cases}$
FPGF	$x_{\mathrm{FPGF}}=\begin{cases}x_{\mathrm{VMF}} & \text{if } \lvert\{x_{i\neq C}\in W\ s.t.\ \lVert x_C-x_i\rVert\leq T\}\rvert<m\\ x_C & \text{otherwise}\end{cases}$

formed. Similar formulations can be derived for the angular and directional-distance functions.

2.4.1 *Adaptive center-weighted vector filters*

The adaptive center-weighted vector filters,[55,57] i.e., ACWVMF, ACWVDF, and ACWDDF, employ a user-specified threshold to determine whether the center pixel is noisy or not. If the center pixel is noisy, it is replaced by the output of one of the three basic vector order statistics filters, the VMF, the BVDF, or the DDF. Otherwise, it remains unchanged. The mathematical expressions for these filters are given in Table 4. The thresholds are set to 80, 0.19, and 10.8 for the ACWVMF, ACWVDF, and ACWDDF, respectively. The λ parameter is set to 2.

An alternative design for the adaptive center-weighted filters is proposed in Ref. 58. Extensions of these filters for image sequence processing and efficient hardware implementations can be found in Refs. 15 and 17.

2.4.2 *Modified center-weighted vector median filter*

The modified center-weighted vector median filter (MCWVMF)[59,60] is a modification of the CWVMF in which only the cumulative distance associated with the center pixel is weighted. In contrast, in the CWVMF the center weight contributes to all of the cumulative distance values except for that associated with the center pixel. This allows the MCWVMF to be faster than the CWVMF since fewer multiplications are involved in the former. Table 4 shows the mathematical expression of the MCWVMF. Note that the center weight w in the MCWVMF is a real number between 0 and 1, whereas the one in the CWVMF is a nonnegative integer. The w parameter is set to 0.5.

2.5 *Entropy Vector Filters*

Entropy vector filters[61,62] are a family of adaptive switching filters that are multichannel extensions of the grayscale local contrast entropy filter.[63] For the grayscale case, the contrast of a pixel x_i within a window W can be expressed as

$$C_i=\frac{|x_i-\bar{x}|}{\bar{x}}=\frac{\Delta_i}{\bar{x}}, \tag{12}$$

where $\bar{x}$ denotes the mean gray level. The local contrast probability P_i and local contrast entropy H_i associated with pixel x_i are given by

$$P_i=\frac{\Delta_i}{\sum_{j=1}^{n}\Delta_j},$$
$$H_i=-P_i\log P_i. \tag{13}$$

Noisy pixels heavily contribute to the total local contrast entropy, which is given by

$$H=\sum_{i=1}^{n}H_i. \tag{14}$$

Extensions of this formulation for the multichannel case are given in Table 5. These filters, i.e., EVMF, EBVDF, and EDDF, employ an adaptive threshold (the fraction of local contrast entropy contributed by the center pixel) to determine whether the center pixel is noisy or not. If the center pixel is noisy, it is replaced by the output of one of the three basic vector filters, the VMF, the BVDF, or the DDF. Otherwise, it remains unchanged. An extension of the entropy filters for color video sequence enhancement can be found in Ref. 64.

2.6 *Peer Group Vector Filters*

These are adaptive switching filters based on the peer group concept.[65] Essentially, the peer group of a pixel in a given window represents the set of neighboring pixels that are sufficiently similar to it according to a particular measure. Table 6 shows the mathematical expressions for these filters.

2.6.1 *Peer group filter*

In the peer group filter (PGF),[65] the pixels in the window are sorted in ascending order according to their distances to the center pixel. The peer group of the center pixel is then determined as the $m=(\sqrt{n}+1)/2$ pixels that rank the lowest in this sorted sequence. Next, in order to remove the effect of the impulsive noise, the first-order differences $\delta(i)$ are calculated. Finally, the center pixel is considered noisy if one of these difference values is greater than a user-

Table 7 Vector sigma filters.

Filter	Formulation
SVMF_mean	$x_{\mathrm{SVMF\_mean}} = \begin{cases} x_{\mathrm{VMF}} & \text{if } l(C) \geq (1+\lambda/n)\cdot l(\bar{x}) \\ x_C & \text{otherwise} \end{cases}$
SVMF_rank	$x_{\mathrm{SVMF\_rank}} = \begin{cases} x_{\mathrm{VMF}} & \text{if } l(C) \geq (1+\lambda/(n-1))\cdot l(x_{\mathrm{VMF}}) \\ x_C & \text{otherwise} \end{cases}$
SBVDF_mean	$x_{\mathrm{SBVDF\_mean}} = \begin{cases} x_{\mathrm{BVDF}} & \text{if } a(C) \geq (1+\lambda/n)\cdot a(\bar{x}) \\ x_C & \text{otherwise} \end{cases}$
SBVDF_rank	$x_{\mathrm{SBVDF\_rank}} = \begin{cases} x_{\mathrm{BVDF}} & \text{if } a(C) \geq (1+\lambda/(n-1))\cdot a(x_{\mathrm{BVDF}}) \\ x_C & \text{otherwise} \end{cases}$
SDDF_mean	$x_{\mathrm{SDDF\_mean}} = \begin{cases} x_{\mathrm{DDF}} & \text{if } d(C) \geq (1+\lambda/n)\cdot d(\bar{x}) \\ x_C & \text{otherwise} \end{cases}$
SDDF_rank	$x_{\mathrm{SDDF\_rank}} = \begin{cases} x_{\mathrm{DDF}} & \text{if } d(C) \geq (1+\lambda/(n-1))\cdot d(x_{\mathrm{DDF}}) \\ x_C & \text{otherwise} \end{cases}$
ASVMF_mean	$x_{\mathrm{ASVMF\_mean}} = \begin{cases} x_{\mathrm{VMF}} & \text{if } \lVert x_C-\bar{x}\rVert \geq \sigma \\ x_C & \text{otherwise} \end{cases}$ $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} \lVert x_i-\bar{x}\rVert^2$
ASVMF_rank	$x_{\mathrm{ASVMF\_rank}} = \begin{cases} x_{\mathrm{VMF}} & \text{if } \lVert x_C-x_{\mathrm{VMF}}\rVert \geq \sigma \\ x_C & \text{otherwise} \end{cases}$ $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n} \lVert x_i-x_{\mathrm{VMF}}\rVert^2$
ASBVDF_mean	$x_{\mathrm{ASBVDF\_mean}} = \begin{cases} x_{\mathrm{BVDF}} & \text{if } A(x_C,\bar{x}) \geq \sigma \\ x_C & \text{otherwise} \end{cases}$ $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} A^2(x_i,\bar{x})$
ASBVDF_rank	$x_{\mathrm{ASBVDF\_rank}} = \begin{cases} x_{\mathrm{BVDF}} & \text{if } A(x_C,x_{\mathrm{BVDF}}) \geq \sigma \\ x_C & \text{otherwise} \end{cases}$ $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n} A^2(x_i,x_{\mathrm{BVDF}})$
ASDDF_mean	$x_{\mathrm{ASDDF\_mean}} = \begin{cases} x_{\mathrm{DDF}} & \text{if } A^{\gamma}(x_C,\bar{x})\cdot\lVert x_C-\bar{x}\rVert^{1-\gamma} \geq \sigma \\ x_C & \text{otherwise} \end{cases}$ $\sigma^2 = \left(\frac{1}{n}\sum_{i=1}^{n} A^2(x_i,\bar{x})\right)^{\gamma}\left(\frac{1}{n}\sum_{i=1}^{n}\lVert x_i-\bar{x}\rVert^2\right)^{1-\gamma}$
ASDDF_rank	$x_{\mathrm{ASDDF\_rank}} = \begin{cases} x_{\mathrm{DDF}} & \text{if } A^{\gamma}(x_C,x_{\mathrm{DDF}})\lVert x_C-x_{\mathrm{DDF}}\rVert^{1-\gamma} \geq \sigma \\ x_C & \text{otherwise} \end{cases}$ $\sigma^2 = \left(\frac{1}{n-1}\sum_{i=1}^{n} A^2(x_i,x_{\mathrm{DDF}})\right)^{\gamma}\left(\frac{1}{n-1}\sum_{i=1}^{n}\lVert x_i-x_{\mathrm{DDF}}\rVert^2\right)^{1-\gamma}$

specified threshold. In this case, the center pixel is replaced with the VMF output; otherwise it remains unchanged. The threshold T is set to 45.

2.6.2 *Fast peer group filter*

The fast peer group filter (FPGF)[4] is a fast modification of the PGF in which the center pixel is considered to be noise-free as soon as m pixels in the window are determined to be sufficiently similar to it. If m is low, and the amount of noise in the image is not very high, the number of distance computations that need to be performed can be dramatically reduced. The parameters m and T are set to 3 and 45, respectively.

2.7 *Vector Sigma Filters*

Vector sigma filters[66–70] are a family of adaptive switching filters that are multichannel extensions of the gray scale sigma filter.[71] These filters utilize approximations of the multivariate variance within a window to determine whether the center pixel is noisy or not. If the center pixel is noisy, it is replaced by the output of one of the three basic vector filters, the VMF, the BVDF, or the DDF. Otherwise, it remains unchanged.

The concept of variance can be extended to the multivariate case using the covariance matrix. Scalar measures for multivariate variance can be calculated from this matrix as the sum or product of the eigenvalues.[72] However, computing the variance within each window in this manner is computationally very expensive. Therefore, vector sigma filters employ approximations of the multivariate variance based on either the mean vector or the lowest-ranked vector.

The members of the vector sigma filter family are given in Table 7. The nonadaptive vector sigma filters (SVMF, SBVDF, and SDDF) require a tuning parameter λ to determine the switching threshold, while the adaptive vector sigma filters (ASVMF, ASBVDF, and ASDDF) determine this threshold adaptively. The parameter λ is set to 4.0.

2.8 *Miscellaneous Vector Filters*

This section contains the filters that do not fit into any of the categories described earlier. Table 8 shows the mathematical expressions for these filters. Some of these have commonalities with certain filters in other categories. For example, the adaptive multichannel nonparametric filters resemble the KVMF in that they are based on similarity rather than dissimilarity (distance). However, they are not included in the hybrid vector filters category since they do not utilize multiple subfilters of different types.

2.8.1 *Vector signal-dependent rank order mean filter*

The vector signal-dependent rank order mean filter (VSDROMF)[73] is an extension of the grayscale SDROM filter.[74] In this filter, the pixels in the window are first sorted according to their cumulative distances to all other pixels. The distances between the center pixel and each of the lowest-ranked 4 (for the general case $\lfloor n/2 \rfloor$) pixels are then compared against increasing thresholds. If any of these distances exceeds its respective threshold, the center pixel is considered to be noisy and is replaced by the lowest-ranked pixel, i.e., the VMF output. Otherwise, the center pixel remains unchanged. The thresholds are set to 35, 40, 45, and 50.

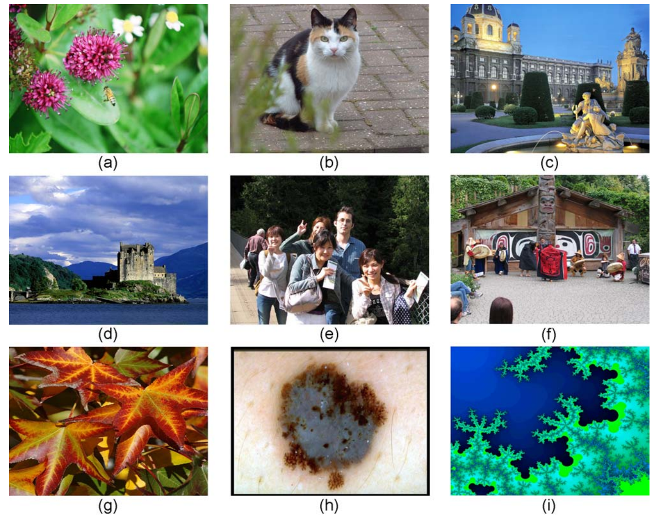

Fig. 1 Representative images from the image set. (a) flowerbee; (b) cat; (c) Austria; (d) Scotland; (e) Capilano Suspension Bridge; (f) Native American; (g) sweetgum; (h) dermoscopy; (i) fractal.

2.8.2 *Adaptive multichannel nonparametric filters*

The adaptive multichannel nonparametric filters (AMNFs)[75,76] approach the filtering problem from an estimation theoretic perspective. Specifically, these filters are based on nonparametric kernel density estimation.[77] The general form of the AMNFs is given in Table 8. Two possible choices for the kernel function are the multivariate exponential $K(z)=e^{-|z|}$ (AMNFE) and the multivariate Gaussian $K(z)=e^{-0.5z^Tz}$ (AMNFG) functions. The k parameter in the kernel width calculation is set to 0.33.

2.8.3 *Fast modified vector median filter*

In the fast modified vector median filter (FMVMF),[78,79] the center pixel is replaced with the window pixel that minimizes the cumulative distance to all others (excluding the center pixel), provided that the difference between the cumulative distance associated with the center pixel and the minimum cumulative distance is greater than a threshold. Otherwise, the center pixel remains unchanged. Note that this scheme privileges the center pixel since its cumulative distance calculations involve $n-1$ terms, whereas the calculations associated with the other pixels involve $n-2$ terms. The distance threshold parameter is set to 0.75.

2.8.4 *Adaptive vector median filter and adaptive basic vector directional filter*

In the adaptive vector median filter (AVMF),[80] the center pixel is considered to be noisy if the distance between itself and the mean of the lowest-ranked k vectors is greater than a threshold. In this case, the center pixel is replaced by the VMF output. Otherwise, it remains unchanged.

The adaptive basic vector directional filter (ABVDF)[81] is the angular counterpart of the AVMF. The thresholds are

Table 8 Miscellaneous vector filters.

Filter	Formulation
VSDROMF	$x_{\text{VSDROMF}} = \begin{cases} x_{\text{VMF}} & \text{if } \exists\, i \in \{1,2,3,4\}\ s.t.\ \lVert x_C - x_{(i)} \rVert > T_i \\ x_C & \text{otherwise} \end{cases}$ $T_1 \le T_2 \le T_3 \le T_4$
AMNF	$x_{\text{AMNF}} = \sum_{i=1}^{n} x_i \left(\frac{h_i^{-c} K\left(\frac{x_C - x_i}{h_i}\right)}{\sum_{j=1}^{n} h_j^{-c} K\left(\frac{x_C - x_j}{h_j}\right)} \right)$ $h_i = n^{-k/c} \sum_{j=1}^{n} \lVert x_i - x_j \rVert_1$
FMVMF	$x_{\text{FMVMF}} = \begin{cases} x_{k^*} & \text{if } \left(\sum_{i=1}^{n} \lVert x_C - x_i \rVert - \sum_{i=1, i \ne C}^{n} \lVert x_{k^*} - x_i \rVert \right) > T \\ x_C & \text{otherwise} \end{cases}$ $x_{k^*} = \underset{x_k \in W}{\operatorname{argmin}} \sum_{i=1, i \ne C}^{n} \lVert x_k - x_i \rVert$
AVMF	$x_{\text{AVMF}} = \begin{cases} x_{\text{VMF}} & \text{if } \left\lVert x_C - \frac{1}{k} \sum_{i=1}^{k} x_{(i)} \right\rVert > T \\ x_C & \text{otherwise} \end{cases}$
ABVDF	$x_{\text{ABVDF}} = \begin{cases} x_{\text{BVDF}} & \text{if } A\left(x_C, \frac{1}{k} \sum_{i=1}^{k} x_{(i)} \right) > T \\ x_C & \text{otherwise} \end{cases}$
FFNRF	$x_{\text{FFNRF}} = \begin{cases} x_{k^*} & \text{if } \sum_{i=1}^{n} M(x_C, x_i) < \sum_{i=1, i \ne C}^{n} M(x_{k^*}, x_i) \\ x_C & \text{otherwise} \end{cases}$ $x_{k^*} = \underset{x_k \in W}{\operatorname{argmax}} \sum_{i=1, i \ne C}^{n} M(x_k, x_i), \quad M^{\alpha}(x_i, x_j) = \prod_{k=1}^{3} \left(\frac{\min(x_i^k, x_j^k) + K}{\max(x_i^k, x_j^k) + K} \right)^{\alpha}$

set to 100 and 0.16 for the AVMF and ABVDF, respectively. The k parameters in these filters are both set to $\lceil n/2 \rceil$.

2.8.5 *Fast fuzzy noise reduction filter*

In the fast fuzzy noise reduction filter (FFNRF),[82,83] the center pixel is replaced with the window pixel that maximizes the cumulative similarity to all others excluding the center pixel. Note that this center exclusion scheme is the same as in the FMVMF. The similarity between two pixels is determined using a special fuzzy metric[84] (see Table 8). An interesting property of this metric is that the value of each term in the product can be precomputed as

$$Q^{\alpha}(a,b) = \left(\frac{\min(a,b)+K}{\max(a,b)+K} \right)^{\alpha}. \tag{15}$$

Using the precomputed values, the fuzzy similarity between two pixels x_i and x_j can be computed as

$$M^{\alpha}(x_i,x_j) = \prod_{k=1}^{3} Q^{\alpha}(x_i,x_j). \tag{16}$$

It's empirically demonstrated that the computation of the fuzzy metric M using the precomputed values is even faster than that of the L_1-norm. The K and α parameters are set to 1024 and 3.5, respectively.

3 Experimental Setup

In this section, the image set that will be used in the experiments is first described. The impulsive noise models that are used to artificially corrupt the images for evaluation purposes are then presented. Finally, the filtering performance criteria that will be considered in the comparisons are detailed.

3.1 *Image Set Description*

In order to compare the performance of the filters on a wide variety of images, a set of 100 high-quality RGB images was collected from the Internet. These included images of people, animals, plants, buildings, aerial maps, manmade objects, natural scenery, paintings, and sketches, as well scientific, biomedical, and synthetic images and test images commonly used in the color image processing literature. Figure 1 shows representative images from this set.

3.2 *Noise Models*

Various simplified color image noise models have been proposed in the literature.[3,5,18] In this study, the following two impulsive noise models are considered:

1. *Uncorrelated impulsive noise*

$$x^k = \begin{cases} r^k & \text{with probability } \varphi, \\ o^k & \text{with probability } 1-\varphi, \end{cases}$$

where $o=\{o^1,o^2,o^3\}$ and $x=\{x^1,x^2,x^3\}$ represent the original and noisy color vectors, respectively, φ denotes the channel corruption probability, and $r=\{r^1,r^2,r^3\}$ is a random vector that represents the impulsive noise such that $r^k \in [0,10]$ or $r^k \in [245,255]$ with equal probability.

2. *Correlated impulsive noise*

$$x = \begin{cases} o & \text{with probability } 1-\varphi, \\ \{r^1,o^2,o^3\} & \text{with probability } \varphi_1 \cdot \varphi, \\ \{o^1,r^2,o^3\} & \text{with probability } \varphi_2 \cdot \varphi, \\ \{o^1,o^2,r^3\} & \text{with probability } \varphi_3 \cdot \varphi, \\ \{r^1,r^2,r^3\} & \text{with probability } (1-(\varphi_1+\varphi_2+\varphi_3)) \cdot \varphi, \end{cases}$$

where φ is the sample corruption probability and φ_1, φ_2, and φ_3 are the channel corruption probabilities. In this study, the following values are used: $\varphi_1=\varphi_2=\varphi_3=0.25$.

In the following discussion, a particular combination of a noise model and a noise level such as "5% correlated noise" will be referred to as a "noise configuration".

3.3 *Filtering Performance Criteria*

In order to evaluate the performance of the filters, three effectiveness and one efficiency criteria are employed. The effectiveness criteria are[5]:

1. *Mean absolute error (MAE)*

$$\text{MAE} = \frac{1}{3 \cdot M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} [|R(i,j)-\hat{R}(i,j)| + |G(i,j)-\hat{G}(i,j)| + |B(i,j)-\hat{B}(i,j)|], \tag{17}$$

where M and N represent the image dimensions, $\{R(i,j),G(i,j),B(i,j)\}$ and $\{\hat{R}(i,j),\hat{G}(i,j),\hat{B}(i,j)\}$ are the RGB coordinates of the pixel (i,j) in the original and the filtered images, respectively. MAE is a measure of the detail preservation capability of a filter.

2. *Mean squared error (MSE)*

$$\text{MSE} = \frac{1}{3 \cdot M \cdot N} \sum_{i=1}^{M} \sum_{j=1}^{N} [(R(i,j)-\hat{R}(i,j))^2 + (G(i,j) - \hat{G}(i,j))^2 + (B(i,j)-\hat{B}(i,j))^2], \tag{18}$$

MSE is a measure of the noise suppression capability of a filter.

3. *Normalized Color Distance (NCD)*

$$\text{NCD} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sqrt{[(L^*_{ab}(i,j)-\hat{L}^*_{ab}(i,j))^2 + (a^*(i,j)-\hat{a}^*(i,j))^2 + (b^*(i,j)-\hat{b}^*(i,j))^2]}}{\sum_{i=1}^{M} \sum_{j=1}^{N} \sqrt{L^{*\,2}_{ab}(i,j)+a^{*2}(i,j)+b^{*2}(i,j)}}, \tag{19}$$

where $\{L^*_{ab}(i,j),a^*(i,j),b^*(i,j)\}$ and $\{\hat{L}^*_{ab}(i,j),\hat{a}^*(i,j),\hat{b}^*(i,j)\}$ are the *CIE* $L^*a^*b^*$ coordinates of the pixel (i,j) in the original and the filtered images, respectively. NCD is a perceptually oriented metric that measures the color preservation capability of a filter.

Table 9 Comparison of the filters based on the MAE measure (AR: average ranking).

MAE	Uncorrelated Noise						Correlated Noise					
	5%		10%		15%		5%		10%		15%	
Rank	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR
0	acwddf	1.56	acwddf	1.71	acwddf	1.78	acwddf	1.75	acwddf	1.84	acwddf	2.35
1	pgf	2.29	pgf	3.31	sddf_rank	3.27	pgf	1.96	pgf	2.52	pgf	3.32
2	mcwvmf	3.31	mcwvmf	5.07	pgf	5.16	mcwvmf	3.30	acwvmf	4.58	sddf_rank	3.95
3	acwvdf	5.03	sddf_rank	5.08	acwvmf	5.34	acwvmf	4.75	sddf_rank	5.23	acwvmf	3.98
4	acwvmf	5.06	acwvmf	5.30	sddf_mean	6.82	acwvdf	5.89	mcwvmf	5.25	svmf_rank	6.98
5	abvdf	7.07	acwvdf	6.21	acwvdf	7.82	avmf	6.75	sddf_mean	8.07	sddf_mean	7.85
6	sddf_rank	7.66	sddf_mean	7.74	svmf_rank	8.75	sddf_rank	7.68	acwvdf	8.08	ffnrf	8.75
7	avmf	7.85	abvdf	9.84	asddf_rank	10.34	ffnrf	8.13	ffnrf	8.88	svmf_mean	9.51
8	ffnrf	8.70	ffnrf	10.32	ffnrf	10.63	abvdf	8.61	svmf_rank	9.66	acwvdf	10.30
9	sddf_mean	9.50	svmf_rank	11.20	mcwvmf	11.04	sddf_mean	9.63	avmf	9.98	fmvmf	11.23
10	sbvdf_rank	11.65	sbvdf_rank	11.51	svmf_mean	11.49	fpgf	11.72	svmf_mean	11.52	mcwvmf	11.48
11	fpgf	12.12	asddf_rank	11.97	sbvdf_rank	11.83	sbvdf_rank	12.28	abvdf	12.39	avmf	12.13
12	fmvmf	12.91	avmf	12.81	abvdf	12.23	svmf_rank	12.39	fmvmf	12.40	asddf_rank	12.70
13	svmf_rank	13.31	fmvmf	13.15	fmvmf	12.47	fmvmf	12.51	asddf_rank	12.85	fpgf	13.98
14	asddf_rank	13.69	svmf_mean	13.17	asddf_mean	13.15	asddf_rank	13.75	sbvdf_rank	13.36	sbvdf_rank	15.11
15	sbvdf_mean	14.29	asddf_mean	14.20	eddf	14.47	svmf_mean	13.89	fpgf	13.55	evmf	15.14
16	svmf_mean	14.81	fpgf	14.57	fpgf	15.77	sbvdf_mean	15.81	asddf_mean	16.16	abvdf	15.20
17	asddf_mean	15.83	sbvdf_mean	16.32	avmf	16.49	asddf_mean	16.37	eddf	16.85	asvmf_mean	15.43
18	eddf	18.34	eddf	16.34	evmf	16.93	eddf	18.41	asvmf_rank	17.25	asvmf_rank	15.58
19	asbvdf_rank	18.89	asvmf_mean	18.62	asvmf_mean	17.50	asvmf_rank	18.95	evmf	17.58	eddf	15.88
20	asvmf_rank	19.90	evmf	18.79	asvmf_rank	17.63	asbvdf_rank	19.60	asvmf_mean	18.73	asddf_mean	16.71
21	vsdromf	20.43	asbvdf_rank	18.80	sbvdf_mean	18.75	evmf	19.94	sbvdf_mean	19.47	vsdromf	19.05
22	evmf	20.86	asvmf_mean	20.18	asbvdf_rank	19.22	vsdromf	20.22	vsdromf	20.20	asbvdf_rank	21.94
23	ebvdf	21.62	vsdromf	21.16	vsdromf	20.64	asvmf_mean	21.43	asbvdf_rank	20.62	vmrhf	22.27
24	asvmf_mean	22.38	ebvdf	23.31	vmrhf	23.51	ebvdf	23.40	vmrhf	23.85	sbvdf_mean	22.30
25	asbvdf_mean	22.70	asbvdf_mean	23.35	asbvdf_mean	24.03	asbvdf_mean	24.26	asbvdf_mean	25.05	fvmrhf	24.67
26	vmrhf	25.30	vmrhf	24.52	ebvdf	24.84	vmrhf	24.98	ebvdf	25.50	kvmf	24.80
27	kvmf	25.59	kvmf	26.29	fvmrhf	25.64	kvmf	25.52	kvmf	25.73	asbvdf_mean	26.73
28	fvmrhf	26.88	fvmrhf	26.43	kvmf	25.78	fvmrhf	26.64	fvmrhf	25.94	fddrhf	27.60
29	fddrhf	28.77	fddrhf	28.49	fddrhf	28.13	fddrhf	28.64	fddrhf	28.26	ebvdf	27.89
30	vmf	30.97	vmf	30.79	vmf	30.57	vmf	30.82	vmf	30.51	vmf	30.13
31	cbrf	32.31	ddf	32.60	ddf	32.31	cbrf	32.29	exvmf	32.27	ddf	32.01
32	ddf	32.67	exvmf	32.74	exvmf	32.54	exvmf	32.58	ddf	32.55	exvmf	32.12
33	exvmf	32.85	cbrf	32.97	fovmf	32.81	ddf	32.76	cbrf	33.07	fovmf	32.28
34	amnfe	34.02	fovmf	33.80	fvmf	32.91	fovmf	34.37	fovmf	33.46	fvmf	32.28
35	fovmf	34.55	amnfe	34.41	cbrf	33.79	amnfe	34.60	fvmf	33.90	cbrf	34.03
36	ahdf	35.06	fvmf	34.41	amnfe	35.30	ahdf	34.91	ahdf	35.01	ahdf	34.82
37	fvmf	35.57	ahdf	35.27	ahdf	35.45	fvmf	35.40	amnfe	35.52	hdf	35.74
38	hdf	36.19	hdf	36.34	hdf	36.39	hdf	36.12	hdf	36.06	amnfe	36.64
39	amnfg	38.56	amnfg	38.51	amnfg	38.63	amnfg	38.57	amnfg	38.67	atvmf	38.40
40	atvmf	39.83	atvmf	39.58	atvmf	39.28	atvmf	39.73	atvmf	39.22	amnfg	39.04
41	annmf	41.10	annmf	41.70	fovdf	41.56	annmf	41.12	annmf	41.79	fovdf	41.46
42	fovdf	43.15	fovdf	42.27	gvdf	42.07	fovdf	42.80	fovdf	42.15	gvdf	41.64
43	annf	43.86	gvdf	42.82	annmf	42.67	gvdf	43.49	gvdf	42.27	annmf	42.61
44	fvdrhf	43.86	bvdf	44.72	bvdf	44.39	annf	44.14	bvdf	44.30	bvdf	43.66
45	gvdf	44.15	annf	44.83	fvdf	44.71	fvdrhf	44.34	fvdrhf	45.26	fvdf	44.67
46	bvdf	44.87	fvdrhf	44.84	fvdrhf	45.38	bvdf	44.66	annf	45.32	fvdrhf	45.77
47	fvdf	46.13	fvdf	45.64	annf	45.79	fvdf	46.14	fvdf	45.32	annf	45.89

Table 10 Comparison of the filters based on the MSE measure (AR: average ranking).

MSE	Uncorrelated Noise						Correlated Noise					
	5%		10%		15%		5%		10%		15%	
Rank	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR
0	acwddf	3.26	acwddf	3.96	acwddf	4.29	pgf	2.56	pgf	4.13	acwvmf	4.32
1	pgf	3.82	sddf_rank	4.37	sddf_rank	5.00	acwddf	3.88	sddf_rank	4.51	fvmrhf	5.76
2	sddf_rank	5.50	acwvmf	6.86	acwvmf	6.72	sddf_rank	5.38	acwvmf	4.68	sddf_rank	5.88
3	acwvmf	6.64	pgf	7.43	fvmrhf	6.94	acwvmf	5.48	acwddf	4.89	acwddf	5.99
4	acwvdf	7.37	sddf_mean	9.15	vmrhf	9.04	mcwvmf	7.12	svmf_rank	6.93	pgf	6.49
5	mcwvmf	7.76	acwvdf	9.47	fddrhf	9.07	sddf_mean	8.45	fvmrhf	8.98	fddrhf	7.73
6	sddf_mean	7.95	svmf_rank	9.95	svmf_rank	11.27	svmf_rank	9.04	vmrhf	10.65	vmrhf	7.85
7	svmf_rank	11.22	fvmrhf	10.50	acwvdf	12.43	svmf_mean	11.03	svmf_mean	10.77	svmf_rank	9.05
8	abvdf	11.35	vmrhf	12.37	pgf	12.99	acwvdf	12.30	fddrhf	11.20	fpgf	10.65
9	asddf_rank	12.91	asddf_rank	12.87	fmvmf	13.17	fvmrhf	12.78	sddf_mean	11.41	fmvmf	10.93
10	svmf_mean	13.07	fddrhf	13.33	fpgf	13.85	ffnrf	13.37	fpgf	12.90	ffnrf	12.50
11	sbvdf_rank	13.68	svmf_mean	13.35	sddf_mean	14.24	vmrhf	13.68	ffnrf	13.02	kvmf	14.35
12	fvmrhf	14.35	sbvdf_rank	14.48	asddf_rank	15.60	fpgf	13.88	fmvmf	13.92	svmf_mean	15.46
13	vmrhf	15.31	abvdf	15.13	kvmf	16.34	asddf_rank	15.01	asvmf_mean	16.77	vsdromf	16.11
14	fpgf	15.33	fpgf	15.51	ffnrf	16.68	fddrhf	15.50	evmf	16.89	asvmf_mean	18.10
15	ffnrf	15.59	fmvmf	16.02	abvdf	16.85	avmf	16.11	kvmf	17.73	sddf_mean	18.55
16	asddf_mean	15.71	eddf	16.64	svmf_mean	17.12	fmvmf	16.63	acwvdf	18.41	fvmf	18.64
17	eddf	16.22	ffnrf	16.69	asvmf_mean	17.54	eddf	17.47	asvmf_rank	19.06	amnfe	19.24
18	fddrhf	17.37	asddf_mean	17.63	sbvdf_rank	18.60	sbvdf_rank	17.74	asddf_rank	19.10	evmf	19.61
19	fmvmf	18.11	evmf	18.36	evmf	18.95	evmf	18.12	vsdromf	19.46	fovmf	20.67
20	evmf	20.03	asvmf_mean	18.80	vsdromf	18.99	asvmf_mean	19.40	eddf	20.93	asvmf_rank	20.92
21	avmf	20.40	kvmf	19.54	eddf	20.20	asddf_mean	19.63	avmf	21.33	amnfg	22.74
22	kvmf	21.30	asvmf_rank	21.11	amnfe	21.13	abvdf	19.70	mcwvmf	22.25	avmf	23.09
23	asvmf_mean	21.41	mcwvmf	21.98	asddf_mean	22.21	asvmf_rank	19.96	amnfe	22.33	acwvdf	23.25
24	sbvdf_mean	21.47	vsdromf	22.15	fvmf	22.30	kvmf	20.09	sbvdf_rank	23.37	asddf_rank	24.59
25	asvmf_rank	21.78	amnfe	24.66	asvmf_rank	22.31	vsdromf	22.30	fvmf	23.53	exvmf	25.09
26	asbvdf_rank	23.01	asbvdf_rank	26.11	fovmf	24.26	amnfe	25.94	fovmf	25.31	ahdf	25.57
27	vsdromf	24.10	fvmf	26.51	amnfg	24.36	fvmf	27.85	amnfg	25.65	atvmf	26.14
28	amnfe	27.17	amnfg	27.24	exvmf	28.33	sbvdf_mean	28.30	asddf_mean	26.40	vmf	26.16
29	fvmf	29.35	avmf	27.84	ahdf	28.69	asbvdf_rank	28.35	abvdf	26.57	eddf	26.71
30	amnfg	29.58	fovmf	28.11	vmf	29.76	amnfg	28.47	exvmf	29.07	hdf	28.41
31	asbvdf_mean	30.87	sbvdf_mean	28.27	atvmf	30.11	fovmf	29.51	ahdf	29.72	ddf	28.94
32	ebvdf	30.94	exvmf	31.37	avmf	30.48	exvmf	32.38	atvmf	30.82	abvdf	29.19
33	fovmf	31.06	ahdf	32.18	asbvdf_rank	30.60	ahdf	33.13	vmf	30.87	sbvdf_rank	30.32
34	exvmf	33.86	vmf	33.33	hdf	31.66	vmf	34.71	hdf	32.58	cbrf	30.78
35	ahdf	34.66	atvmf	33.70	ddf	32.95	annmf	35.21	ddf	33.67	asddf_mean	32.34
36	fvdrhf	35.51	hdf	34.99	cbrf	34.24	atvmf	35.44	cbrf	33.85	fovdf	32.39
37	vmf	36.37	fvdrhf	35.58	sbvdf_mean	35.10	fvdrhf	35.57	annmf	34.81	annmf	33.85
38	annmf	36.42	cbrf	36.23	fvdrhf	35.29	hdf	36.02	asbvdf_rank	34.97	fvdf	33.95
39	atvmf	37.11	ddf	36.66	fovdf	35.37	cbrf	36.59	fvdrhf	35.39	gvdf	34.31
40	hdf	37.44	asbvdf_mean	36.90	mcwvmf	35.71	asbvdf_mean	37.93	sbvdf_mean	36.82	fvdrhf	34.32
41	cbrf	37.88	annmf	37.15	fvdf	36.06	ebvdf	37.93	fovdf	37.21	mcwvmf	36.16
42	ddf	39.55	ebvdf	37.94	annmf	37.47	ddf	38.07	fvdf	37.54	annf	38.72
43	annf	40.23	fvdf	39.16	gvdf	38.03	annf	39.84	gvdf	38.56	asbvdf_rank	39.45
44	fvdf	41.66	fovdf	39.50	annf	39.67	fvdf	40.85	annf	39.76	sbvdf_mean	41.49
45	fovdf	42.30	annf	40.26	asbvdf_mean	40.30	fovdf	41.27	asbvdf_mean	41.92	bvdf	41.70
46	gvdf	43.61	gvdf	41.15	ebvdf	41.85	gvdf	42.17	ebvdf	43.13	asbvdf_mean	44.11
47	bvdf	46.41	bvdf	45.51	bvdf	43.88	bvdf	45.86	bvdf	44.23	ebvdf	45.43

Table 11 Comparison of the filters based on the NCD measure (AR: average ranking).

NCD	Uncorrelated Noise						Correlated Noise					
	5%		10%		15%		5%		10%		15%	
Rank	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR
0	acwddf	1.79	acwddf	2.31	acwddf	2.54	acwddf	1.80	acwddf	2.22	acwddf	2.34
1	pgf	4.18	sddf_rank	3.29	sddf_rank	3.04	pgf	3.28	sddf_rank	3.62	sddf_rank	3.46
2	acwvdf	4.41	acwvdf	5.05	acwvdf	5.28	acwvmf	4.25	acwvmf	4.37	acwvmf	4.43
3	acwvmf	4.60	acwvmf	5.63	acwvmf	6.16	acwvdf	5.03	pgf	5.10	acwvdf	6.07
4	mcwvmf	5.35	sddf_mean	7.19	fmvmf	7.29	mcwvmf	5.33	acwvdf	5.63	pgf	6.32
5	sddf_rank	5.69	pgf	7.44	svmf_rank	7.78	sddf_rank	5.94	svmf_rank	7.82	fmvmf	6.58
6	sddf_mean	8.39	sbvdf_rank	8.10	sbvdf_rank	7.83	ffnrf	8.87	sddf_mean	8.05	svmf_rank	7.33
7	abvdf	8.48	svmf_rank	8.65	sddf_mean	8.54	abvdf	8.95	fmvmf	8.90	sbvdf_rank	9.52
8	ffnrf	10.09	fmvmf	9.44	pgf	10.02	sddf_mean	8.96	sbvdf_rank	9.35	sddf_mean	10.09
9	sbvdf_rank	10.33	abvdf	10.05	abvdf	10.30	fmvmf	10.33	abvdf	10.66	abvdf	10.69
10	fmvmf	10.44	svmf_mean	11.60	svmf_mean	12.03	svmf_rank	10.70	ffnrf	10.73	ffnrf	10.76
11	svmf_rank	11.08	asddf_rank	11.94	fpgf	12.75	sbvdf_rank	11.21	svmf_mean	11.52	fpgf	11.25
12	fpgf	12.23	ffnrf	12.93	asddf_rank	12.89	fpgf	11.80	fpgf	12.71	svmf_mean	12.37
13	asddf_rank	13.33	fpgf	13.64	ffnrf	13.52	svmf_mean	12.90	asddf_rank	12.94	asddf_rank	14.33
14	svmf_mean	13.35	asddf_mean	14.34	eddf	14.47	avmf	13.32	mcwvmf	14.93	vsdromf	14.96
15	sbvdf_mean	15.03	eddf	15.03	asddf_mean	15.16	asddf_rank	13.93	eddf	16.07	eddf	15.98
16	avmf	15.93	mcwvmf	15.19	asvmf_mean	15.57	sbvdf_mean	16.53	asddf_mean	16.09	asvmf_mean	16.04
17	asddf_mean	16.01	sbvdf_mean	17.02	vsdromf	15.88	asddf_mean	16.78	evmf	17.28	evmf	17.12
18	eddf	18.06	evmf	17.24	evmf	16.40	vsdromf	18.44	vsdromf	17.47	vmrhf	17.33
19	vsdromf	18.90	asvmf_mean	17.91	vmrhf	18.15	eddf	18.66	asvmf_mean	17.62	asddf_mean	17.94
20	asvmf_rank	19.11	asbvdf_rank	18.13	asvmf_rank	18.82	asvmf_rank	18.69	asvmf_rank	18.52	asvmf_rank	18.64
21	asbvdf_rank	19.12	vsdromf	18.31	asbvdf_rank	19.09	evmf	19.20	sbvdf_mean	19.66	kvmf	21.07
22	evmf	19.53	asvmf_rank	18.89	sbvdf_mean	20.86	asbvdf_rank	20.01	asbvdf_rank	19.97	asbvdf_rank	21.28
23	asvmf_mean	20.89	vmrhf	22.36	kvmf	21.96	asvmf_mean	20.57	avmf	20.09	fvmrhf	21.92
24	ebvdf	23.48	avmf	23.09	fvmrhf	22.69	vmrhf	24.46	vmrhf	21.99	avmf	23.02
25	asbvdf_mean	23.79	kvmf	24.33	mcwvmf	23.70	ebvdf	24.52	kvmf	23.78	sbvdf_mean	23.65
26	vmrhf	24.67	asbvdf_mean	24.67	fddrhf	26.47	asbvdf_mean	24.81	fvmrhf	25.33	mcwvmf	24.16
27	kvmf	25.22	fvmrhf	25.48	asbvdf_mean	27.11	kvmf	24.83	asbvdf_mean	26.59	fddrhf	26.08
28	fvmrhf	26.80	ebvdf	25.69	avmf	27.39	fvmrhf	26.57	ebvdf	27.79	fvmf	29.74
29	fddrhf	28.95	fddrhf	28.19	ebvdf	29.29	fddrhf	28.96	fddrhf	28.09	ddf	30.08
30	ddf	31.37	ddf	31.15	ddf	30.12	ddf	31.50	ddf	31.08	vmf	30.24
31	vmf	32.80	fvmf	31.87	fvmf	30.33	vmf	32.77	fvmf	31.69	asbvdf_mean	30.28
32	exvmf	33.08	vmf	32.03	vmf	30.69	exvmf	32.91	vmf	31.74	fovmf	30.43
33	fvmf	33.11	fovmf	32.50	fovmf	30.83	fvmf	33.01	fovmf	32.30	ebvdf	32.06
34	fovmf	33.97	exvmf	33.21	exvmf	32.79	fovmf	33.80	exvmf	32.78	exvmf	32.08
35	cbrf	34.38	cbrf	34.27	ahdf	33.55	ahdf	34.41	ahdf	34.18	ahdf	33.20
36	ahdf	34.48	ahdf	34.40	cbrf	33.64	cbrf	34.47	cbrf	34.47	cbrf	33.84
37	amnfe	34.98	hdf	35.75	hdf	34.98	amnfe	35.28	hdf	35.42	hdf	34.49
38	hdf	35.98	amnfe	36.29	atvmf	37.16	hdf	35.76	amnfe	37.34	atvmf	36.37
39	amnfg	38.23	atvmf	38.69	amnfe	38.01	amnfg	38.43	atvmf	37.84	amnfe	39.05
40	atvmf	39.86	amnfg	39.14	fovdf	39.79	atvmf	39.76	amnfg	39.74	fovdf	39.17
41	fovdf	41.87	fovdf	40.96	amnfg	39.96	fovdf	41.80	fovdf	40.70	gvdf	40.26
42	annmf	42.49	gvdf	41.63	gvdf	40.92	annmf	42.46	gvdf	41.36	amnfg	40.57
43	gvdf	42.78	bvdf	42.42	bvdf	41.27	gvdf	42.56	bvdf	42.03	bvdf	40.71
44	bvdf	43.42	annmf	43.88	fvdf	44.00	bvdf	43.16	annmf	43.95	fvdf	43.85
45	annf	44.11	fvdf	44.79	annmf	44.61	annf	44.35	fvdf	44.37	annmf	44.55
46	fvdrhf	45.91	annf	45.33	annf	45.67	fvdf	45.88	annf	45.53	annf	45.67
47	fvdf	45.95	fvdrhf	46.56	fvdrhf	46.70	fvdrhf	46.06	fvdrhf	46.59	fvdrhf	46.63

Table 12 Comparison of the filters based on execution time (AR: average ranking).

TIME	Uncorrelated Noise						Correlated Noise					
	5%		10%		15%		5%		10%		15%	
Rank	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR	Filter	AR
0	fpgf	0.08	fpgf	0.20	fpgf	0.50	fpgf	0.06	fpgf	0.27	asvmf_mean	0.47
1	asvmf_mean	0.92	asvmf_mean	0.81	asvmf_mean	0.54	asvmf_mean	0.94	asvmf_mean	0.75	fpgf	0.54
2	pgf	2.01	pgf	2.02	pgf	2.11	pgf	2.02	pgf	2.07	pgf	2.26
3	svmf_mean	3.00	svmf_mean	2.97	svmf_mean	2.86	svmf_mean	2.99	svmf_mean	2.91	svmf_mean	2.73
4	ffnrf	3.99	ffnrf	4.00	ffnrf	3.99	ffnrf	3.99	ffnrf	4.00	ffnrf	4.00
5	mcwvmf	5.44	mcwvmf	5.33	mcwvmf	5.10	mcwvmf	5.39	mcwvmf	5.29	mcwvmf	5.16
6	asbvdf_mean	6.88	asbvdf_mean	6.73	svmf_rank	7.44	asbvdf_mean	7.04	asbvdf_mean	7.09	svmf_rank	7.32
7	svmf_rank	7.57	svmf_rank	7.66	asbvdf_mean	7.52	fmvmf	7.78	svmf_rank	7.41	asbvdf_mean	7.61
8	fmvmf	8.06	fmvmf	7.71	fmvmf	7.72	svmf_rank	7.83	fmvmf	7.91	fmvmf	7.73
9	vmf	8.48	vmf	8.47	vmf	8.15	vmf	8.33	vmf	8.21	vmf	8.02
10	asvmf_rank	9.85	asvmf_rank	9.56	asvmf_rank	9.24	asvmf_rank	9.74	asvmf_rank	9.46	asvmf_rank	9.34
11	asddf_mean	10.23	asddf_mean	10.85	asddf_mean	11.27	asddf_mean	10.33	asddf_mean	10.92	asddf_mean	11.27
12	sbvdf_mean	12.46	exvmf	12.38	exvmf	11.83	exvmf	12.41	exvmf	11.89	exvmf	11.68
13	exvmf	12.60	sbvdf_mean	12.83	sbvdf_mean	13.32	sbvdf_mean	12.78	sbvdf_mean	13.54	kvmf	13.95
14	kvmf	14.53	kvmf	14.78	kvmf	14.80	kvmf	14.52	kvmf	14.68	avmf	14.40
15	avmf	15.05	avmf	14.98	avmf	14.99	avmf	14.86	avmf	14.88	atvmf	14.65
16	atvmf	15.46	atvmf	15.38	atvmf	15.18	atvmf	15.58	atvmf	15.30	sbvdf_mean	15.55
17	vsdromf	17.39	vsdromf	17.17	vsdromf	17.17	vsdromf	17.32	vsdromf	17.30	vsdromf	17.24
18	vmrhf	18.12	vmrhf	17.67	vmrhf	17.68	vmrhf	17.85	vmrhf	17.62	vmrhf	17.55
19	sddf_mean	19.01	acwvmf	19.22	acwvmf	19.01	acwvmf	19.44	acwvmf	19.07	acwvmf	19.02
20	acwvmf	19.72	sddf_mean	20.17	sddf_mean	20.58	sddf_mean	19.87	annmf	20.71	annmf	20.03
21	annmf	21.13	annmf	21.02	annmf	20.91	annmf	21.11	sddf_mean	20.92	amnfe	22.00
22	amnfe	22.18	amnfe	22.36	amnfe	22.30	amnfe	22.19	amnfe	22.37	cbrf	22.21
23	cbrf	23.50	cbrf	23.35	cbrf	23.11	evmf	23.24	cbrf	23.09	evmf	22.84
24	evmf	23.56	evmf	23.54	evmf	23.88	cbrf	23.54	evmf	23.39	sddf_mean	23.73
25	amnfg	24.33	amnfg	24.38	amnfg	24.29	amnfg	24.44	amnfg	24.56	amnfg	24.25
26	bvdf	26.00	bvdf	25.98	bvdf	25.98	bvdf	25.92	bvdf	25.95	bvdf	25.98
27	sbvdf_rank	26.96	sbvdf_rank	26.88	sbvdf_rank	26.89	sbvdf_rank	26.98	sbvdf_rank	26.85	sbvdf_rank	26.83
28	ebvdf	28.50	ebvdf	28.47	ebvdf	28.66	ebvdf	28.61	ebvdf	28.67	ebvdf	28.66
29	asbvdf_rank	29.04	asbvdf_rank	29.00	asbvdf_rank	28.78	asbvdf_rank	28.85	asbvdf_rank	28.81	asbvdf_rank	28.82
30	gvdf	29.47	gvdf	29.51	gvdf	29.56	gvdf	29.59	gvdf	29.52	gvdf	29.51
31	abvdf	31.04	abvdf	31.00	abvdf	30.95	abvdf	31.01	abvdf	31.02	abvdf	30.96
32	eddf	32.46	eddf	32.47	annf	32.68	eddf	32.47	eddf	32.61	annf	32.48
33	annf	32.81	annf	32.80	eddf	32.98	annf	32.88	annf	32.78	eddf	33.13
34	ddf	33.58	ddf	33.65	ddf	33.28	ddf	33.48	ddf	33.49	ddf	33.38
35	sddf_rank	34.98	sddf_rank	35.05	sddf_rank	35.05	sddf_rank	34.93	sddf_rank	34.93	sddf_rank	34.87
36	acwvdf	35.96	acwvdf	35.95	acwvdf	36.05	acwvdf	36.07	acwvdf	36.25	acwvdf	36.34
37	asddf_rank	36.85	asddf_rank	36.88	asddf_rank	36.83	asddf_rank	36.82	asddf_rank	36.72	asddf_rank	36.68
38	hdf	37.94	hdf	37.94	hdf	37.94	hdf	37.94	hdf	37.94	hdf	37.94
39	ahdf	39.10	ahdf	39.01	ahdf	38.99	ahdf	39.03	ahdf	38.96	ahdf	38.98
40	acwddf	39.93	acwddf	40.01	acwddf	40.01	acwddf	40.01	acwddf	40.07	acwddf	40.05
41	fvdf	40.93	fvdf	40.95	fvdf	40.96	fvdf	40.92	fvdf	40.92	fvdf	40.94
42	fovdf	41.96	fovdf	41.96	fovdf	41.96	fovdf	41.96	fovdf	41.96	fovdf	41.96
43	fvmrhf	43.01	fvmrhf	43.01	fvmrhf	43.02	fvmrhf	43.01	fvmrhf	43.01	fvmrhf	43.01
44	fvdrhf	43.96	fvdrhf	43.96	fvdrhf	43.96	fvdrhf	43.96	fvdrhf	43.96	fvdrhf	43.96
45	fddrhf	45.03	fddrhf	45.03	fddrhf	45.00	fddrhf	45.03	fddrhf	45.02	fddrhf	45.00
46	fvmf	45.97	fvmf	45.97	fvmf	45.99	fvmf	45.97	fvmf	45.97	fvmf	45.98
47	fovmf	46.97	fovmf	46.98	fovmf	46.99	fovmf	46.97	fovmf	46.98	fovmf	46.99

The efficiency of a filter is measured by the execution time in seconds. In order to ensure a fair comparison, all of the filters were implemented in the same style in the C language and compiled with the *gcc* v3.4 compiler. The experiments were performed on an Intel Pentium D 2.66-GHz machine.

An issue in the comparison of the execution times is the cost of the inverse cosine (acos) function that is utilized in the angular and directional-distance filters. Standard library implementations of this function are computationally very expensive, causing angular distance computations to be much slower than the Minkowski distance computations. For example, on a typical 512×512 image, the VMF takes about 0.36 s, while the BVDF takes approximately 10.0 s. A solution to mitigate this problem is to use an approximation for the acos function over the interval [0, 1]. However, this is not easy because of the singularity of this function at arguments very close to 1. This can be circumvented by using the following numerically more stable identity for $x \geq 0.5$[85]:

$$\mathrm{acos}(x) = 2 \cdot \mathrm{asin}(\sqrt{(1-x)/2}). \tag{20}$$

In Eq. (20), the inverse sine (asin) function receives its arguments from the interval [0, 0.5]. Fortunately, this function is almost linear in this interval and can be accurately approximated using a third-degree minimax polynomial[86]:

$$\begin{aligned}\mathrm{asin}(x) \cong & -0.67921302\mathrm{e}-4 + (1.003729762 \\ & + (-0.309031329\mathrm{e}-1 \\ & + .2356774247 \cdot \mathrm{x}) \cdot \mathrm{x}) \cdot \mathrm{x}.\end{aligned} \tag{21}$$

The approximation error is $\varepsilon = 0.00006792131489$. Similarly, the acos function is almost linear in the interval [0, 0.5] and can be approximated by

$$\begin{aligned}\mathrm{acos}(x) \cong & \, 1.570864248 + (-1.003729768 \\ & + (0.309031763\mathrm{e}-1 - .2356774861 \cdot \mathrm{x}) \cdot \mathrm{x}) \cdot \mathrm{x}.\end{aligned} \tag{22}$$

The approximation error is $\varepsilon = 0.00006792158693$. This piecewise approximation of the acos function gives virtually the same numerical results, i.e., MAE, MSE, and NCD, as the standard acos function when used in the implementation of an angular or a directional-distance filter. This is because the error propagation is not very significant considering the small size of a filter window (only 9 pixels in a 3×3 window).

In order to demonstrate the effect of the approximation on the running time of an actual filter, the BVDF implementation that uses the standard acos function and the one that uses the approximation were both executed on the entire image set (100 images). The standard implementation took 1428 s, while the approximate one took 102 s. Similar gains in the execution times ($\approx 13-14$X) were observed for the other angular and directional-distance filters as well. Note that methods used to speed up the VMF operation itself[87,88] are beyond the scope of this study.

Table 13 Most effective filters at each noise level.

Noise Level	Most Effective Filters
5% noise	ACWDDF, PGF, SDDF_rank, ACWVMF, ACWVDF, MCWVMF, SDDF_mean
10% noise	ACWDDF, PGF, SDDF_rank, ACWVMF, SDDF_mean, SVMF_rank
15% noise	ACWDDF, PGF, SDDF_rank, ACWVMF, SVMF_rank

4 Discussion and Conclusions

This section discusses the experimental results and presents the conclusions. First, the filters are compared based on the previously described measures of effectiveness and efficiency. Second, the filters that achieve a good compromise between effectiveness and efficiency are identified. Finally, the three commonly used distance measures are compared.

4.1 *Discussion*

Tables 9–12 show the rankings[†] of the filters based on the following criteria: MAE, MSE, NCD, and execution time, respectively. The results[‡] are presented for the two noise models (uncorrelated and correlated impulsive noise) and three noise levels (5%, 10%, and 15%). The average rankings are obtained by averaging the individual filter rankings over the entire image set.

In order to determine the most effective filters at each noise level, we select the 10 best filters with respect to each effectiveness measure (MAE, MSE, NCD) for each noise model. Based on this selection, at each noise level, the filters that perform well regardless of the noise model and the effectiveness measure are determined (see Table 13). It can be observed that two filter families are particularly prominent in effectiveness: the adaptive center-weighted vector filters and the vector sigma filters. This can be attributed to the effectiveness of the noise detection criteria used in these families. By varying the smoothing parameter, the adaptive center-weighted vector filters employ a computationally expensive but robust iterative scheme to determine whether the center pixel is noisy or not. On the other hand, the vector sigma filters utilize approximations of the multivariate variance within a window in their noise detection criteria. Interestingly, in general, the nonadaptive vector sigma filters perform better than their adaptive counterparts.

The filters that are effective under any circumstances are those that appear in every row of Table 13. These are the ACWDDF, PGF, SDDF_rank, and ACWVMF. Among these filters, the ACWDDF consistently ranks the highest under different noise configurations. The PGF and ACWVMF have relatively stable rankings, whereas the SDDF_rank exhibits somewhat fluctuating behavior. Figures 2 and 3 show the results of these filters on two images corrupted by 10% and 15% correlated noise, respectively.

[†]Note that the rankings start from 0 rather than 1.

[‡]For comparison purposes, the window size for each filter is set to 3×3 and the $L2$-norm is used whenever the Minkowski distance is involved.

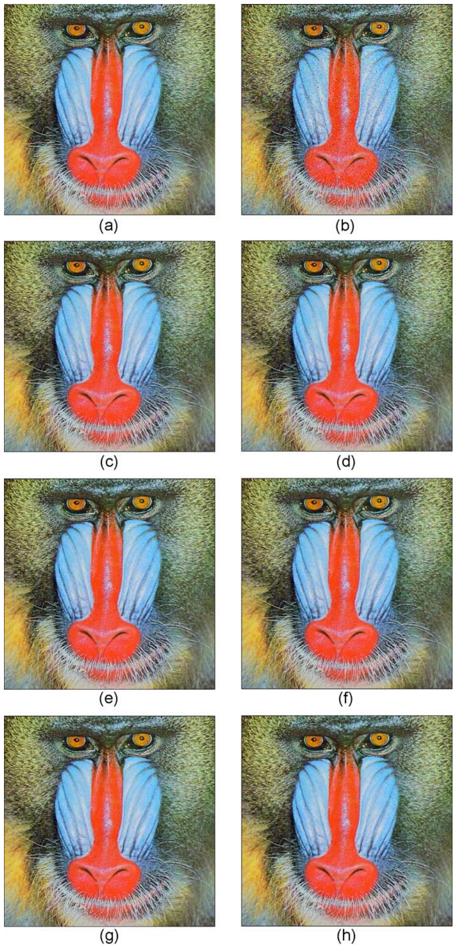

Fig. 2 Sample filtering results for the baboon image. (a) Original; (b) 10% correlated noise; MAE: 6.058; MSE: 893.707; NCD: 0.101; (c) ACWDDF: MAE: 1.902; MSE: 76.956; NCD: 0.012; (d) ACWVDF: MAE: 2.182; MSE: 102.892; NCD: 0.014; (e) PGF: MAE: 2.293; MSE: 98.825; NCD: 0.015; (f) SDDF_mean: MAE: 3.017; MSE: 124.358; NCD: 0.019; (g) SDDF_rank: MAE: 3.031; MSE: 123.020; NCD: 0.019; (h) ACWVMF: MAE: 3.726; MSE: 171.007; NCD: 0.023.

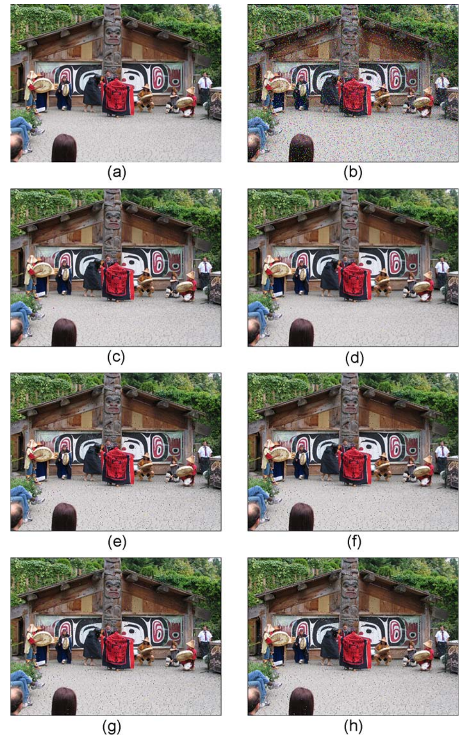

Fig. 3 Sample filtering results for the Native American image. (a) Original; (b) 15% correlated noise; MAE: 9.600; MSE: 1558.290; NCD: 0.182; (c) ACWDDF: MAE: 1.453; MSE: 49.316; NCD: 0.015; (d) PGF: MAE: 1.593; MSE: 54.189; NCD: 0.021; (e) SDDF_rank: MAE: 1.594; MSE: 50.284; NCD: 0.016; (f) ACWVMF: MAE: 1.643; MSE: 53.992; NCD: 0.019; (g) SDDF_mean: MAE: 1.776; MSE: 74.073; NCD: 0.021; (h) ACWVDF: MAE: 2.030; MSE: 109.360; NCD: 0.018.

The execution time is also a very important factor that determines the practicality of a noise removal filter. As Table 12 shows, the ordering of the filters with respect to execution time remains almost unchanged across different noise configurations. The 10 most efficient filters are FPGF, ASVMF_mean, PGF, SVMF_mean, FFNRF, MCWVMF, ASBVDF_mean, SVMF_rank, FMVMF, and VMF. The following observations are in order:

- Except for the VMF, every filter in the list is based on the concept of switching (alternating between the identity and the filter operations).
- The FPGF is clearly the most efficient filter.
- The PGF is the only filter that ranks very high in terms of both effectiveness and efficiency. This is significant, considering the most effective filter, i.e., ACWDDF, is actually among the slowest.

It should be emphasized that some filters that appear in the 10 most efficient filters list but not in Table 13 still achieve a good compromise between effectiveness and efficiency. These include MCWVMF, FMVMF, FFNRF, SVMF_rank, SVMF_mean, and FPGF.

An examination of the distance measures (Minkowski, angular, directional-distance) with respect to effectiveness and efficiency shows that no distance measure completely outperforms the other two. However, it is interesting to note that among the 4 most effective filters, 2 are based on directional-distance (ACWDDF, SDDF_rank). Considering that only 8 of the 48 filters are based on directional-distance, the idea of combining the Minkowski and angular distance functions proves to be quite advantageous. On the other hand, as explained in Section 3.3, the filters based on the Minkowski distance are inherently more efficient than their angular and directional-distance counterparts. In fact, it can be seen from Table 12 that, except for the FFNRF and ASBVDF_mean, the 10 most efficient filters are all based on the Minkowski distance. In contrast, the most efficient angular filter (ASBVDF_mean) appears at the 7th rank, whereas the most efficient directional-distance filter (ASDDF_mean) ranks 12th. This shows that if execution time is of prime importance, filters based on the Minkowski distance are the most obvious choice.

The unsatisfactory performance of the hybrid and adaptive fuzzy filters can be attributed to the fact that these filters introduce color artifacts by determining the output in a window as a linear or nonlinear combination of the input vectors. However, it should be noted that these filters are known to be more effective in the presence of Gaussian noise due to their averaging nature.

The reader should note that due to time constraints some filters in the literature were omitted from this study. Notable examples include the fast adaptive similarity-based noise reduction filter (FANRF),[89], the fuzzy inference-based vector filter (FIVF),[90] and the vector rank M-type K-nearest neighbor (VRMKNNF).[16] The FANRF is based on the notion of similarity rather than distance. The similarity between two pixels can be calculated using various kernel functions, which allows for more flexibility when designing filters tailored for particular applications. The FIVF employs a novel fuzzy inference system for noise detection and involves switching between the identity operation and the L-filter, whose coefficients are determined using a fast constrained least-mean-squares approach. The VRMKNNF is based on combined RM-estimators with different influence functions. It employs an adaptive nonparametric approach that determines the functional form of the probability density of the noise to improve the filtering performance.

4.2 *Conclusions*

This study presented a systematic survey of 48 impulsive noise removal filters using a unified notation. The filters were categorized into families and compared on a large image set in order to ensure an objective appraisal of their effectiveness and efficiency. A fast approximation for the inverse cosine function was introduced to allow for a more even comparison of efficiency. Furthermore, commonly used distance measures were compared and contrasted. Finally, recommendations for selecting filters that meet certain criteria were provided.

The implementations of the filters described in this article have been made publicly available as part of the Fourier image processing and analysis library, which can be downloaded from http://sourceforge.net/projects/fourier-ipal.

Acknowledgments

This work was supported by grants from the NSF (#0216500-EIA), Texas Workforce Commission (#3204600182), and James A. Schlipmann Melanoma Cancer Foundation. Sources for the images in Fig. 1 are as follows: http://pics.tech4learning.com (a–d, g), EDRA Interactive Atlas of Dermoscopy (h), and Dr. Peter Alfeld (i).

References

1. H. J. Trussell, E. Saber, and M. J. Vrhel, "Color image processing: basics and special issue overview," *IEEE Signal Process. Mag.* **22**(1), 14–22 (2005).
2. R. Lukac, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "Selection weighted vector directional filters," *Comput. Vis. Image Underst.* **94**(1/3), 140–167 (2004).
3. B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "Nonlinear techniques for color image processing," in *Nonlinear Signal and Image Processing: Theory, Methods, and Applications*, edited by K. E. Barner and G. R. Arce, pp. 445–505, CRC Press, Boca Raton, FL (2004).
4. B. Smolka and A. Chydzinski, "Fast detection and impulsive noise removal in color images," *Real-Time Imag.* **11**(5/6), 389–402 (2005).
5. K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*, Springer-Verlag, New York (2000).
6. G. Sharma and H. J. Trussell, "Figures of merit for color scanners," *IEEE Trans. Image Process.* **6**(7), 990–1001 (1997).
7. R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis, and A. N. Venetsanopoulos, "Vector filtering for color imaging," *IEEE Signal Process. Mag.* **22**(1), 74–86 (2005).
8. J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE* **78**(4), 678–689 (1990).
9. V. Barnett, "The ordering of multivariate data," *J. Am. Stat. Assoc.* **139**(3), 318–354 (1976).
10. I. Pitas and P. Tsakalides, "Multivariate ordering in color image filtering," *IEEE Trans. Circuits Syst. Video Technol.* **1**(3), 247–259 (1991).
11. R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "cDNA microarray image processing using fuzzy vector filtering framework," *Fuzzy Sets Syst.* **152**(1), 17–35 (2005).
12. R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "A multichannel order-statistic technique for cDNA microarray image processing," *IEEE Trans. Nanobiosci.* **3**(4), 272–285 (2004).
13. M. Barni, F. Bartolini, and V. Cappellini, "Image processing for virtual restoration of artworks," *IEEE Multimedia* **7**(2), 34-37 (2000).
14. L. Lucchese and S. K. Mitra, "A new class of chromatic filters for color image processing: theory and applications,"*IEEE Trans. Image Process.* **13**(4), 534–548 (2004).
15. R. Lukac, V. Fischer, G. Motyl, and M. Drutarovsky, "Adaptive video filtering framework," *Int. J. Imaging Syst. Technol.* **14**(6), 223–237 (2004).
16. V. I. Ponomaryov, F. J. Gallegos-Funes, and A. Rosales-Silva, "Real-time color imaging based on RM-filters for the impulsive noise reduction," *J. Imaging Sci. Technol.* **49**(3), 205–219(2005).
17. V. Fischer, R. Lukac, and K. Martin, "Cost-effective video filtering solution for real-time vision systems," *EURASIP J. Appl. Signal Process.* **13**, 2026–2042 (2005).
18. T. Viero, K. Oistamo, and Y. Neuvo, "Three-dimensional median-related filters for color image sequence filtering," *IEEE Trans. Circuits Syst. Video Technol.* **4**(2), 129–142 (1994).
19. R. Lukac and K. N. Plataniotis, "A taxonomy of color image filtering and enhancement solutions," *in Advances in Imaging & Electron Physics*, Vol. 140, edited by P. W. Hawkes, pp. 187–264, Academic Press, San Diego, CA (2006).
20. C. S. Regazzoni and A. Teschioni, "A new approach to vector median filtering based on space filling curves," *IEEE Trans. Image Process.* **6**(7), 1025–1037 (1997).
21. M. I. Vardavoulia, I. Andreadis, and P. Tsalides, "A new vector median filter for colour image processing," *Pattern Recogn. Lett.* **22**(6/7), 675–689 (2001).
22. M. Barni, V. Cappellini, and A. Mecocci, "The use of different metrics in vector median filtering: application to fine arts and paintings," *Proc. EUSIPCO'92*, pp. 1485–1488 (1992).
23. P. E. Trahanias and A. N. Venetsanopoulos, "Vector directional filters: a new class of multichannel image processing filters," *IEEE Trans. Image Process.* **2**(4), 528–534 (1993).
24. P. E. Trahanias, D. Karakos, and A. N. Venetsanopoulos, "Directional processing of color images: theory and experimental results," *IEEE Trans. Image Process.* **5**(6), 868–880 (1996).
25. J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC Press, Boca Raton, FL (1997).
26. D. G. Karakos and P. E. Trahanias, "Combining vector median and vector directional filters: the directional distance filters," *Proc. IEEE ICIP Conf.*, pp. 171–174 (1995).
27. D. G. Karakos and P. E. Trahanias, "Generalized multichannel image filtering structures," *IEEE Trans. Image Process.* **6**(7), 1038–1045 (1997).
28. K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, "Content-based color image filters," *Electron. Lett.* **33**(3), 202–203 (1997).
29. G. Ekman, "A direct method for multidimensional ratio scaling," *Psychometrika* **28**, 33–41 (1963).
30. K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, "Fuzzy adaptive filters for multichannel image processing," *Signal Process.* **55**(1), 93–106 (1996).
31. K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, "Adaptive fuzzy systems for multichannel signal processing," *Proc. IEEE* **87**(9), 1601–1622 (1999).
32. V. Chatzis and I. Pitas, "Fuzzy scalar and vector median filters based on fuzzy distances," *IEEE Trans. Image Process.* **8**(5), 731–734 (1999).
33. J. C. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition*, IEEE Press, New York (1992).
34. K. N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, and A. N. Venetsanopoulos, "A nearest neighbor multichannel filter," *Electron. Lett.* **31**(22), 1910–1911 (1995).
35. K. N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, and A. N. Venetsanopoulos, "An adaptive nearest neighbor multichannel filter," *IEEE Trans. Circuits Syst. Video Technol.* **6**(6), 699–703 (1996).
36. L. Khriji and M. Gabbouj, "Adaptive fuzzy order statistics-rational hybrid filters for color image processing," *Fuzzy Sets Syst.* **128**(1), 35–46 (2002).

37. Z. Ma, H. R. Wu, and B. Qiu, "A robust structure-adaptive hybrid vector filter color image restoration," *IEEE Trans. Image Process.* **14**(12), 1990–2001 (2005).
38. M. Gabbouj and F. A. Cheikh, "Vector median-vector directional hybrid filter for color image restoration," *Proc. EUSIPCO'96*, 2, 879–882 (1996).
39. L. Khriji and M. Gabbouj, "A class of multichannel image processing filters," *Electron. Lett.* **35**(4), 285–287 (1999).
40. L. Khriji and M. Gabbouj, "Vector median-rational hybrid filters for multichannel image processing," *IEEE Signal Process. Lett.* **6**(7), 186–190 (1999).
41. L. Khriji and M. Gabbouj, "A new class of multichannel image processing filters: vector median-rational hybrid filters," *IEICE Trans. Inf. Syst.* **E82**(12), 1589–1596 (1999).
42. L. Khriji and M. Gabbouj, "Multichannel image processing using fuzzy vector median-rational hybrid filters," *Proc. EUSIPCO'00*, pp. 1345–1348 (2000).
43. L. Khriji and M. Gabbouj, "Rational-based adaptive fuzzy filters," *Int. J. Comput. Cognition* **2**(1), 113–132 (2004).
44. B. Smolka and K. N. Plataniotis, "Soft-switching adaptive technique of impulsive noise removal in color images," *Proc. 2nd Int. Conf. Image Analysis and Recog. (ICIAR 2005)*, *Lect. Notes Comput. Sci.* **3656**, 686–693 (2005).
45. B. Smolka, R. Bieda, K. N. Plataniotis, and R. Lukac, "Adaptive soft-switching filter for impulsive noise suppression in color images," *Proc. EUSIPCO'05* (2005).
46. B. Smolka, K. N. Plataniotis, R. Lukac, and A. N. Venetsanopoulos, "New class of impulsive noise reduction filters based on kernel density estimation," *Proc. of the 28th IEEE Int. Conf. on Acoustics, Speech & Signal Process. (ICASSP'03)* 3, 721–724 (2003).
47. B. Smolka, R. Lukac, K. N. Plataniotis, and A. N. Venetsanopoulos, "Application of kernel density estimation for color image filtering," *Proc. Vis. Commun. and Image Process. (VCIP'03)*, SPIE Vol. 5150, 1650–1656 (2003).
48. B. Smolka, K. N. Plataniotis, R. Lukac, and A. N. Venetsanopoulos, "Kernel density estimation based impulsive noise reduction filter," *Proc. IEEE Int. Conf. Image Process. (ICIP'03)* 2, 137–140 (2003).
49. K. Oistamo, Q. Liu, M. Grundstrom, and Y. Neuvo, "Weighted vector median operation for filtering multispectral data," *Proc. IEEE Int. Conf. Syst. Eng.*, pp. 16–19 (1992).
50. R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "Generalized selection weighted vector filters," *EURASIP J. Appl. Signal Process.* **12**, 1870–1885 (2004).
51. R. Lukac, K. N. Plataniotis, and A. N. Venetsanopoulos, "Color image denoising using evolutionary computation," *Int. J. Imaging Syst. Technol.* **15**(5), 236–251 (2005).
52. L. Lucat, P. Siohan, and D. Barba, "Adaptive and global optimization methods for weighted vector median filters," *Signal Process. Image Commun.* **17**(7), 509–524 (2002).
53. Y. Shen and K. E. Barner, "Fast adaptive optimization of weighted vector median filters," *IEEE Trans. Signal Process.* **54**(7), 2497–2510 (2006).
54. R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "Weighted vector median optimization," *Proc. 4th EURASIP Conf. Focused on Video/Image Processing and Multimedia Commun. (EC-VIP-MC 2003)*, 1, 227–232 (2003).
55. R. Lukac, "Adaptive color image filtering based on center-weighted vector directional filters," *Multidimens. Syst. Signal Process.* **15**(2), 169–196 (2004).
56. Z. Ma, H. R. Wu, and D. Feng, "Partition-based vector filtering technique for suppression of noise in digital color images," *IEEE Trans. Image Process.* **15**(8), 2324–2342 (2006).
57. R. Lukac, "Optimised directional distance filter," *Mach. Graphics Vision* **11**(2/3), 311–326 (2002).
58. R. Lukac and S. Marchevsky, "Adaptive vector LUM smoother," *Proc. IEEE Int. Conf. on Image Processing. (ICIP'01)* 1, 878–881 (2001).
59. B. Smolka, "Efficient modification of the central weighted vector median filter," *Proc. 24th DAGM Sym. Patt. Recogn.*, *Lect. Notes Comput. Sci.* **2449**, 166–173 (2002).
60. B. Smolka, R. Lukac, and K. N. Plataniotis, "New algorithm for noise attenuation in color images based on the central weighted vector median filter," *Proc. 9th Int. Workshop on Systems, Signals and Image Process. (IWSSIP'02)*, pp. 544–548 (2002).
61. R. Lukac, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "Entropy vector median filter," *Proc. 1st Iberian Conf. Patt. Recogn. Image Analy. (IbPRIA)*, *Lect. Notes Comput. Sci.* **2652**, 1117–1125 (2003).
62. R. Lukac, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "Generalized entropy vector filters," *Proc. 4th EURASIP EC-VIP-MC Video Image Processing and Multimedia Commun. Conf.*, pp. 239–244 (2003).
63. A. Beghdadi and A. Khellaf, "A noise-filtering method using a local information measure," *IEEE Trans. Image Process.* **6**(6), 879-882 (1997).
64. R. Lukac, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "Three-dimensional entropy vector median filter for color video filtering," *Proc. Visual Commun. Image Process. (VCIP'03)*, SPIE Vol. 5150, 1642–1649 (2003).
65. C. Kenney, Y. Deng, B. S. Manjunath, and G. Hewer, "Peer group image enhancement," *IEEE Trans. Image Process.* **10**(2), 326–334 (2001).
66. R. Lukac, B. Smolka, K. N. Plataniotis, A. N. Venetsanopoulos, and P. Zavarsky, "Angular multichannel sigma filter," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP'03)* 3, 745–748 (2003).
67. R. Lukac, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "A variety of multichannel sigma filters," *Proc. SPIE* **5146**, 244–253 (2003).
68. R. Lukac, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "Generalized adaptive vector sigma filters," *Proc. Int. Conf. Multimedia and Expo (ICME'03)* 1, 537–540 (2003).
69. R. Lukac, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopoulos, "Vector sigma filters for noise detection and removal in color images," *J. Visual Commun. Image Represent* **17**(1), 1–26 (2006).
70. R. Lukac, K. N. Plataniotis, A. N. Venetsanopoulos, and B. Smolka, "A statistically-switched adaptive vector median filter," *J. Intell. Robotic Syst.* **42**(4), 361–391 (2005).
71. J. S. Lee, "Digital image smoothing and the sigma filter," *Comput. Vis. Graph. Image Process.* **24**(2), 255–269 (1983).
72. S. S. Wilks, "Certain generalizations in the analysis of variance," *Biometrika* **24**(3/4), 471–494 (1932).
73. M. S. Moore, M. Gabbouj, and S. K. Mitra, "Vector SD-ROM filter for removal of impulse noise from color images," *Proc. 2nd EURASIP Conf. Focused on DSP for Multimedia Commun. Services (ECMCS'99)* (1999).
74. E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Trans. Image Process.* **5**(6), 1012–1025 (1996).
75. K. N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, and A. N. Venetsanopoulos, "Color image processing using adaptive multichannel filters," *IEEE Trans. Image Process.* **6**(7), 933–949 (1997).
76. K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, "Adaptive multichannel filters for colour image processing," *Signal Process. Image Commun.* **11**(3), 171–177 (1998).
77. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification* (2nd ed.), Wiley-Interscience, New York (2000).
78. B. Smolka, M. Szczepanski, K. N. Plataniotis, and A. N. Venetsanopoulos, "Fast modified vector median filter," *Proc. 9th Int. Conf. Compu. Analy. Images and Patterns*, *Lect. Notes Comput. Sci.* **2124**, 570–580 (2001).
79. B. Smolka, M. Szczepanski, K. N. Plataniotis, and A. N. Venetsanopoulos, "On the fast modification of the vector median filter," *Proc. 16th Int. Conf. Patt. Recog. ICPR'02* 3, 931–934 (2002).
80. R. Lukac, "Adaptive vector median filtering," *Pattern Recogn. Lett.* **24**(12), 1889–1899 (2003).
81. R. Lukac, "Color image filtering by vector directional order-statistics," *Patt. Recog. Image Anal.* **12**(3), 279–285 (2002).
82. S. Morillas, V. Gregori, G. Peris-Fajarnes, and P. Latorre, "A new vector median filter based on fuzzy metrics," *Proc. 2nd Int. Conf. on Image Anal. Recog. (ICIAR'05)*, *Lect. Notes Comput. Sci.* **3656**, 82–91 (2005).
83. S. Morillas, V. Gregori, G. Peris-Fajarnes, and P. Latorre, "A fast impulsive noise color image filter using fuzzy metrics," *Real-Time Imag.* **11**(5/6), 417–428 (2005).
84. A. George and P. Veeramani, "On some results in fuzzy metric spaces," *Fuzzy Sets Syst.* **64**(3), 395–399 (1994).
85. W. J. Cody and W. Waite, *Software Manual for the Elementary Functions*, Prentice-Hall, Englewood Cliffs, NJ (1980).
86. J.-M. Muller, *Elementary Functions: Algorithms and Implementation* 2nd ed., Birkhäuser, Boston (2006).
87. M. Barni, "A fast algorithm for 1-norm vector median filtering," *IEEE Trans. Image Process.* **6**(10), 1452–1455 (1997).
88. M. Barni, F. Buti, F. Bartolini, and V. Cappellini, "A quasi-Euclidean norm to speed up vector median filtering," *IEEE Trans. Image Process.* **9**(10), 1704–1709 (2000).
89. B. Smolka, R. Lukac, A. Chydzinski, K. N. Plataniotis, and K. Wojciechowski, "Fast adaptive similarity based impulsive noise reduction filter," *Real-Time Imag.* **9**(4), 261–276 (2003).
90. E. S. Hore, B. Qiu, and H. R. Wu, "Improved vector filtering for color images using fuzzy noise detection," *Opt. Eng.* **42**(6), 1656–1664 (2003).

M. Emre Celebi received his BS degree in computer engineering from Middle East Technical University in Ankara, Turkey, in 2002. He received his MSc and PhD degrees in computer science and engineering from the University of Texas at Arlington in 2003 and 2006, respectively. He is currently an assistant professor in the Department of Computer Science at the Louisiana State University in Shreveport. His research interests include medical image analysis, color image processing, content-based image retrieval, and open-source software development.

Hassan A. Kingravi received his BS degree in computer science and engineering from the University of Texas at Arlington in 2006. He worked in the INFOLAB at UTA from 2005 to 2006. He is currently pursuing a master's degree in computer science at Georgia Institute of Technology. His research interests include pattern recognition and machine learning.

Y. Alp Aslandogan received his BS degrees in computer science and mathematics from Bogazici University in Istanbul, Turkey. He received his master's degree in computer science from Case Western Reserve University in Cleveland, Ohio, and his PhD degree in electrical engineering and computer science from the University of Illinois in Chicago. He has served on the technical program committees of the IEEE International Conference on Information Technology, International Conference on Information Reuse and Integration and Emerging Technologies Conference. Dr. Aslandogan currently serves on the faculty of the Department of Computer Science at Prairie View A&M University. His research interests include biomedical informatics, databases and data mining, multimedia information retrieval, and information visualization.