Rastislav Lukac, Bogdan Smolka, Karl Martin,
Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos

Vector Filtering for Color Imaging

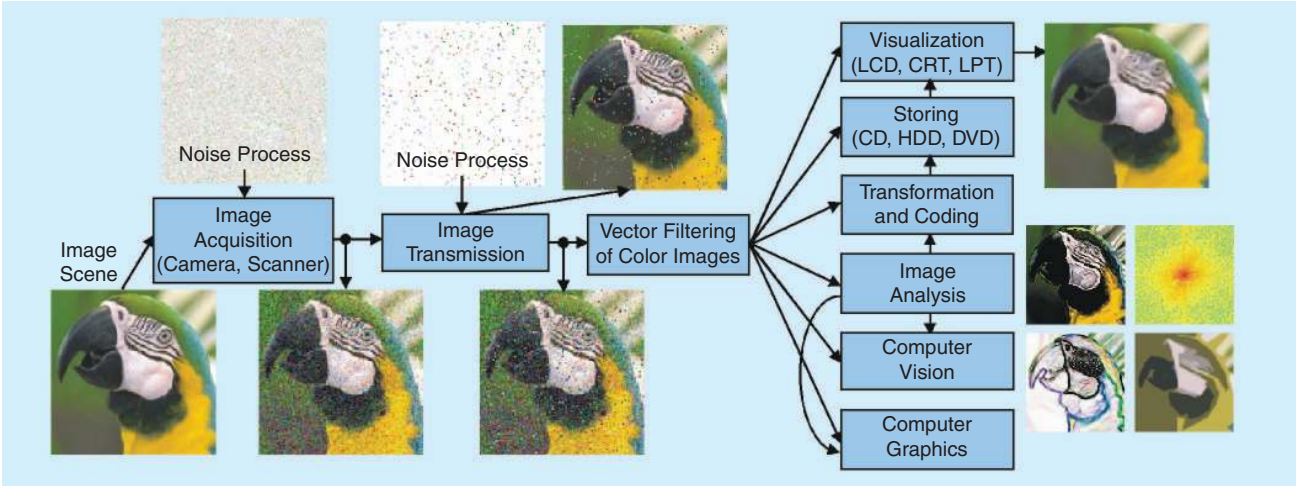
Opening a world of possibilities

Color image processing systems [1] are used for a variety of purposes, ranging from capturing scenes for posterity [2] to processing of imagery for feature extraction [3], [4]. These systems often rely on filtering operations to, for example, suppress noise that would inhibit the central functionality of the system or to actually perform the central task itself. This article describes some modern color image filters that rely on the trichromatic theory of color [5], [6]. Some of the varied applications for these filters will also be covered; namely, noise removal, edge detection, spatial interpolation, and spectral interpolation (demosaicking). This list, while certainly not exhaustive, provides a good indication of the usefulness and, often, necessity of color image filters in a color image processing system.

Noise removal is one of the main applications discussed in this article, since noise can arise from numerous sources and is present in almost any image processing system. Two fundamental types are considered: noise produced during image formation (i.e., sensor noise) and noise produced during transmission (i.e., channel noise). If destined for human consumption, noise will reduce the perceptual quality of an image, and its inherent value will thus be limited. Similarly, if an image is destined for numerical analysis, noise will usually limit the performance of the system, if not defeat it altogether. Therefore, noise filtering—the process of estimating the original image information from noisy data—is an essential part of many image processing systems (Figure 1).

Edges convey essential information about a scene. Thus, edge detection is also a common component in image processing systems. Numerous higher level functions, such as object recognition, robot vision, and segmentation, rely on accurate edge detection. While monochrome techniques may be applied in such situations, if the available trichromatic information is exploited, imaging systems may better mimic the human perception of a visual environment. Modern color image filters are thus ideal for this task.

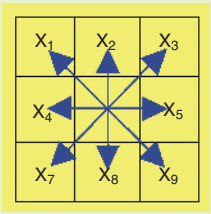
1053-5888/05/$20.00©2005IEEE



[FIG1] **Image processing chain.**

WINDOW-BASED FILTERING BASICS

Natural images are nonstationary. Filtering schemes operate on the premise that an image can be subdivided into small regions, each of which can be treated as stationary [1]. Small image regions are determined using the supporting window $W = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with the pixels $\mathbf{x}_i$ centered around $\mathbf{x}_{(N+1)/2}$, as shown in Figure A. Operating at the pixel level, spatial filtering operators replace $\mathbf{x}_{(N+1)/2}$ with the output pixel $\hat{\mathbf{x}}_{(N+1)/2} = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$.



[FIGA] **The 3×3 filtering mask with the window center $\mathbf{x}_{(N+1)/2} = \mathbf{x}_5$.**

The performance of a filtering scheme is generally influenced by the size of the local area inside W. Some applications may require larger support to read local image features and complete the task appropriately. On the other hand, a filter operating on a smaller spatial neighborhood can better match image details [2].

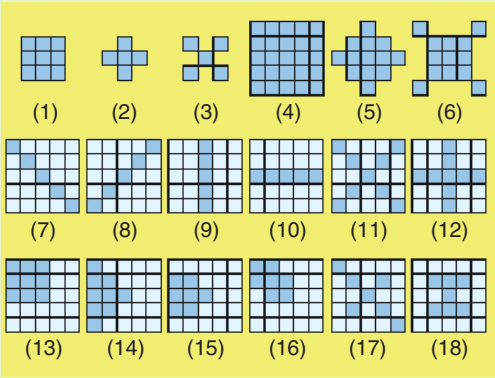
A particular window (Figure B) can be designed to preserve specifically oriented image edges. For example, unidirectional windows [Figure B(7)–(10)] preserve image details in one of the main four directions. Bidirectional windows [Figure B(11) and (12)] follow the structural content of the image in two directions. The most commonly used window is a rectangular shape [Figure B(1) and (4)] due to its versatility and performance. To completely process the input image, the filter window W is moved over the image domain to individually affect all the image pixels (Figure C). This concept is commonly known as a running (sliding) window.

Filtering schemes based on the ordering of the input samples within W are nonlinear in nature [1]. In the case of reduced ordering scheme, aggregated distances $D_i = \sum_{j=1}^{N} d(\mathbf{x}_i, \mathbf{x}_j)$ or aggregated similarities $D_i = \sum_{j=1}^{N} s(\mathbf{x}_i, \mathbf{x}_j)$ associated with the vectorial (color) inputs $\mathbf{x}_i$ serve as the ordering criterion.

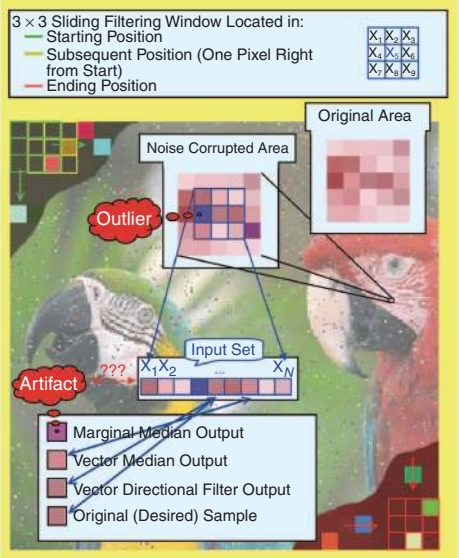
References

[1] I. Pitas and A.N. Venetsanopoulos, *Nonlinear Digital Filters, Principles and Applications*. Norwell, MA: Kluwer, 1990.

[2] W.K. Pratt, *Digital Image Processing*, 3rd ed. New York: Wiley, 2001.



[FIGB] **Most popular window shapes: (1)–(6) conventional spatial windows, (7)–(10) unidirectional windows, (11) and (12) bidirectional windows, (13)–(18) special window structures. Other structures can be obtained by rotating those shown in (13)–(18).**



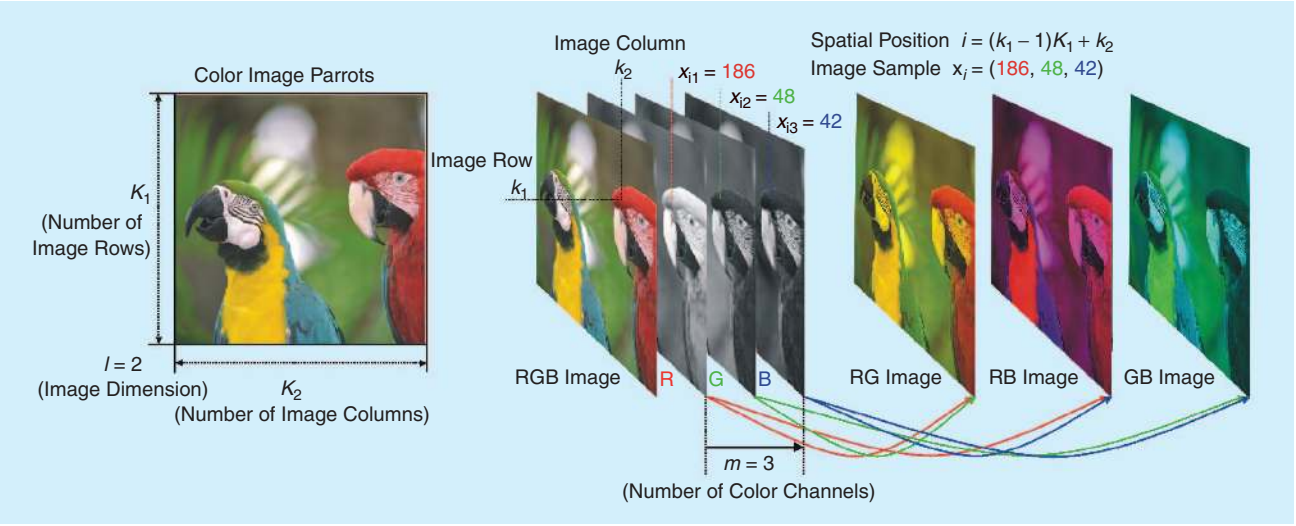
[FIGC] **Sliding filter window concept.**

Interpolation can be performed on color images in either the spatial domain or in the spectral domain. Spatial interpolation involves increasing the apparent resolution of an image by increasing the number of pixels representing the visual data. Spectral interpolation, on the other hand, involves estimating individual color components based on known components within a processing window. This is a required operation in most digital cameras, since the image is formed using a monochromatic sensor and a color filter array (CFA), hence producing only one of the three required color components at each spatial location.

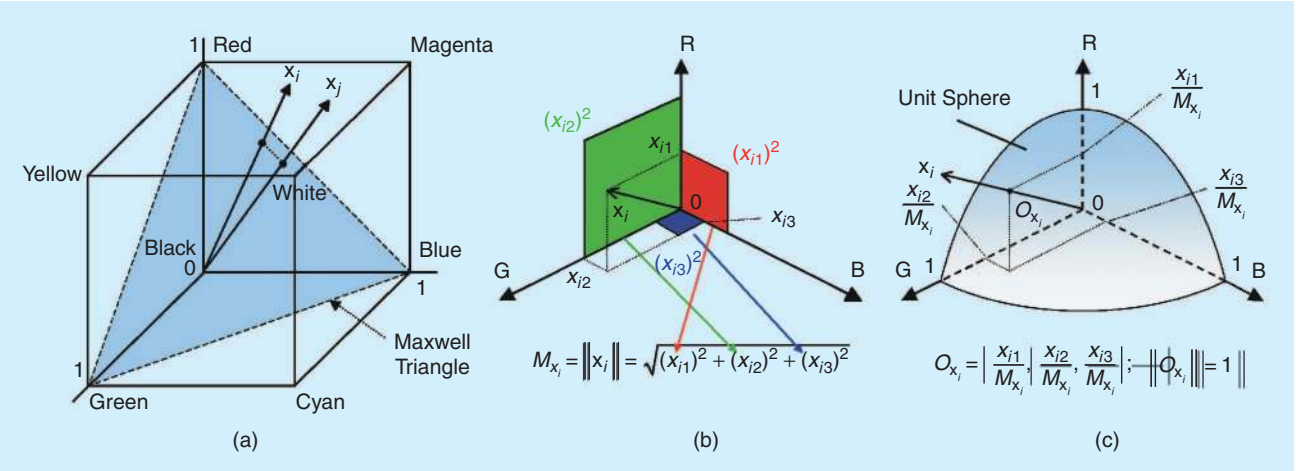
The basic filtering approach, described in "Window-Based Filtering Basics," is common to all the applications described here. Namely, a sliding window will pass over the image to capture information in a localized area that will be used to determine the output value of the pixel at the center of the window. This approach is chosen due to the localized nature of image features and the point-wise model of noise formation. The following will discuss the specific filtering frameworks.

COLOR FILTERING FRAMEWORKS

Since each individual channel of a color image can be considered a monochrome image, traditional image filtering techniques often involved the application of scalar filters on each channel separately. However, this disrupts the correlation that exists between the color components of natural images represented in a correlated color space, such as sRGB [7]. Since each processing step is usually accompanied by a certain inaccuracy, the formation of the output color vector from the separately processed color components usually produces color artifacts. Thus, vector filtering techniques that treat the color image as a vector field are more appropriate. With this approach, the filter output $\hat{\mathbf{x}}_{(N+1)/2}$ is a function of the vectorial inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ located within the supporting window W (see "Window-Based Filtering Basics"). Assuming a color red, green, blue (RGB) image $\mathbf{x}: Z^2 \rightarrow Z^3$ (Figure 2), each pixel $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ represents a three-component vector in a color space [Figure 3(a)]. Thus, the color image $\mathbf{x}$ is a vector



[FIG2] **Color image representation in the RGB color domain.**



[FIG3] **(a) RGB color cube and (b), (c) the basic parameters related to the RGB color vector $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$. (b) The magnitude $M_{\mathbf{x}_i}$. (c) The orientation defined as the point $O_{\mathbf{x}_i}$ on unit sphere.**

DISTANCE AND SIMILARITY MEASURES

The most commonly used measure to quantify the distance between two color vectors $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ and $\mathbf{x}_j = [x_{j1}, x_{j2}, x_{j3}]^T$ in the magnitude domain the generalized weighted Minkowski metric $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_L = c(\sum_{k=1}^{3} \xi_k |x_{ik} - x_{jk}|^L)^{\frac{1}{L}}$, described in [1]. The nonnegative scaling parameter c is a measure of the overall discrimination power. The exponent L defines the nature of the distance metric. The most popular members of this class are obtained when $L = 1$ (city-block distance) and when $L = 2$ (Euclidean distance). The chess-board distance corresponds to $L \to \infty$. In this case, the distance between the two 3-D vectors is considered equal to the maximum distance among their components. The parameter ξ_k measures the proportion of attention allocated to the dimensional component k and, therefore, $\sum_k \xi_k = 1$. Vectors having a range of values greater than a desirable threshold can be scaled down by the use of the weighting function ξ.

An alternative method to the Minkowski metric is the Canberra distance $d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{3}(|x_{ik} - x_{jk}|/(x_{ik} + x_{jk}))$. The summand is defined to be zero if both x_{ik} and x_{jk} are zero [2].

Opposite to distance measures, a similarity measure $s(\mathbf{x}_i, \mathbf{x}_j)$ is defined as a symmetric function whose value is large when the vectorial inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar. Similarity in orientation is expressed through the normalized inner product $s(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \mathbf{x}_j^T/(|\mathbf{x}_i||\mathbf{x}_j|)$, which corresponds to the cosine of the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$. Since similar colors have almost parallel orientations and significantly different colors point in different overall directions in a 3-D color space such as the RGB space, the normalized inner product, or equivalently the angular distance $\theta = \arccos(\mathbf{x}_i \mathbf{x}_j^T/(|\mathbf{x}_i||\mathbf{x}_j|))$, can be used instead of the Minkowski metric to quantify the dissimilarity (here the orientation difference) between the two vectors [2].

It is obvious that a generalized similarity measure model that can effectively quantify differences among color signals should take into consideration both the magnitude and orientation of the color vectors. Thus, a generalized measure based on both the magnitude and orientation of vectors should provide a robust solution to the problem of similarity quantification between two vectors. Such an idea is used in constructing the generalized content model family of measures that treat similarity between two vectors as the degree of common content in relation to the total content of the two vectors. Therefore, given the common quantity, commonality C_{ij}, and the total quantity, totality T_{ij}, the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as $s(\mathbf{x}_i, \mathbf{x}_j) = C_{ij}/T_{ij}$.

Based on this general framework, different similarity measures can be obtained by using different commonality and totality concepts [2]:

$$s(\mathbf{x}_i, \mathbf{x}_j) = w_i \left(\frac{\mathbf{x}_i \mathbf{x}_j^T}{|\mathbf{x}_i||\mathbf{x}_j|} \right) w_j \left(1 - \frac{||\mathbf{x}_i| - |\mathbf{x}_j||}{\max(|\mathbf{x}_i|, |\mathbf{x}_j|)} \right)$$

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{h_i + h_j}{|\mathbf{x}_i| + |\mathbf{x}_j|} = \frac{|\mathbf{x}_i|\cos(\theta) + |\mathbf{x}_i|\cos(\theta)}{|\mathbf{x}_i| + |\mathbf{x}_j|} = \cos(\theta)$$

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{h_i + h_j}{(|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 + 2|\mathbf{x}_i||\mathbf{x}_j|\cos(\theta))^{\frac{1}{2}}}$$

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{(|h_i|^2 + |h_j|^2 + 2|h_i||h_j|\cos(\theta))^{\frac{1}{2}}}{|\mathbf{x}_i| + |\mathbf{x}_j|}$$

$$s(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{(|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 - 2|\mathbf{x}_i||\mathbf{x}_j|\cos(\theta))^{\frac{1}{2}}}{(|\mathbf{x}_i|^2 + |\mathbf{x}_j|^2 + 2|\mathbf{x}_i||\mathbf{x}_j|\cos(\theta))^{\frac{1}{2}}}.$$

[1] R.M. Nosovsky, "Choice, similarity and the context theory of classification," *J. Exp. Psychol. Learn., Memory Cognit.*, vol. 10, no. 1, pp. 104–114, Jan. 1984.

[2] K.N. Plataniotis, D. Androutsos, and A.N. Venetsanopoulos, "Adaptive fuzzy systems for multichannel signal processing," *Proc. IEEE*, vol. 87, no. 9, pp. 1601–1622, Sept. 1999.

array or a two-dimensional (2-D) matrix of three component samples $\mathbf{x}_i$ with x_{ik} denoting the R ($k = 1$), G ($k = 2$), or B component ($k = 3$). The chromatic properties of $\mathbf{x}_i$ as depicted in Figure 3(b) and (c) relate to its magnitude $M_{\mathbf{x}_i} = \|\mathbf{x}_i\| = \sqrt{(x_{i1})^2 + (x_{i2})^2 + (x_{i3})^2}$ and direction (orientation in the vector space) $O_{\mathbf{x}_i} = \mathbf{x}_i/\|\mathbf{x}_i\| = \mathbf{x}_i/M_{\mathbf{x}_i}$, with $\|O_{x_i}\| = 1$. Both the magnitude and the direction can be used in classifying the differences between two vectorial inputs.

One of the most popular families of nonlinear filtering operators is based on robust order statistics [8]. These filters utilize algebraic ordering of a windowed set of data to compute the output signal. However, there is no universal method for defining the ordering of multivariate data. Hence, a number of possibilities arise, with the most common summarized as follows [9]–[11].

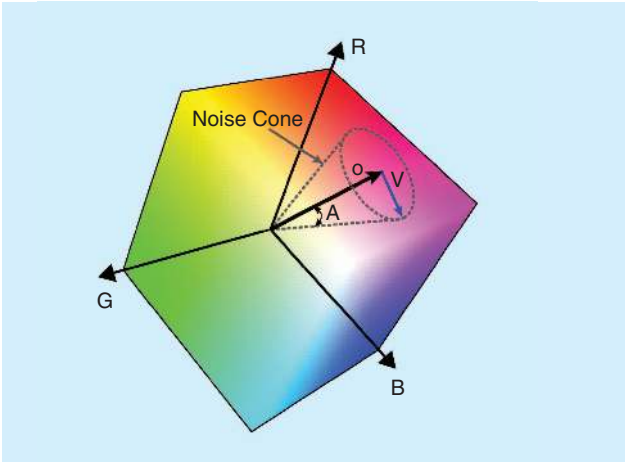
- *Marginal ordering (M-ordering)*: the vector's components are ordered along each dimension independently. Since the M-ordering approach often produces output vectors that differ from the set of vectorial inputs, application of M-ordering to natural color images often results in color artifacts.
- *Conditional ordering (C-ordering)*: the vector samples are ordered based on the marginal ordering of one component. C-ordering fails to take into consideration the full vectorial nature of the input.
- *Partial ordering (P-ordering)*: the samples are partitioned into smaller groups that are then ordered. P-ordering is difficult to perform in more than 2-D, and therefore it is not appropriate for three-component signals such as RGB color images.
- *Reduced or aggregated ordering (R-ordering)*: each vector is reduced to a scalar representative, and then the vectorial inputs are ordered in coincidence with the ranked scalars. The R-ordering approach is the most attractive, since it involves an overall ranking of the original set of input samples, and the output is selected from the same set.

To order the color vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ located inside the supporting window W, the R-ordering based vector filters (Figure 4) use the aggregated distances or the aggregated similarities (see "Distance and Similarity Measures")

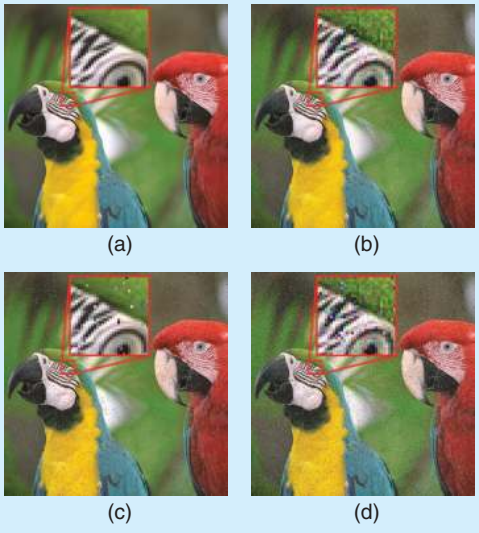
```
Inputs:  NumberOfRows × NumberOfColumns image
         Window size N
         Moving window spawning the input set {x_1, x_2, ..., x_N}
Output:  NumberOfRows × NumberOfColumns image

For a = 1 to NumberOfRows
  For b = 1 to NumberOfColumns
    Determine the input set W(a,b) = {x_1, x_2, ..., x_N}
    For i = 1 to N
      Let the aggregated distance D_i = d(x_i, x_1) + d(x_i, x_2) + ... + d(x_i, x_N)
    End
    Sort scalars D_1, D_2, ..., D_N to the ordered set D_(1) ≤ D_(2) ≤ ... ≤ D_(N)
    Apply the same ordering scheme to the vectors x_1, x_2, ..., x_N
    Store the ordered sequence as x_(1) ≤ x_(2) ≤ ... ≤ x_(N)
    Let the filter output y(a,b) = x_(1)
  End
End
```

[FIG4] Algorithm of the vector filters (e.g., VMF or BVDF) outputting the lowest ranked vector.



[FIG5] Angular noise margins for a color signal.



[FIG6] Test image Parrots (512 × 512) corrupted by different kinds of noise: (a) original image, (b) additive Gaussian noise with σ = 20, (c) 5% impulsive noise, (d) mixed noise (additive Gaussian noise of σ = 20 followed by 5% impulsive noise).

$$D_i = \sum_{j=1}^{N} d(\mathbf{x}_i, \mathbf{x}_j) \quad \text{or} \quad D_i = \sum_{j=1}^{N} s(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

associated with the vectorial input $\mathbf{x}_i$, for $i = 1, 2, \ldots, N$. Thus, the ordered sequence of scalars $D_{(1)} \leq D_{(2)} \leq \cdots \leq D_{(\tau)} \leq \cdots \leq D_{(N)}$, implies the same ordering of the corresponding vectors $\mathbf{x}_i$: $\mathbf{x}_{(1)}; \mathbf{x}_{(2)}; \ldots; \mathbf{x}_{(\tau)}; \ldots; \mathbf{x}_{(N)}$.

Many filters use the lowest ranked sample (lowest order statistic) $\mathbf{x}_{(1)}$ as the output. This selection is due to the fact that vectors that diverge greatly from the data population usually appear in higher indexed locations in the ordered sequence [12], [13].

VECTOR MEDIAN FILTERS

The most popular vector filter is the vector median filter (VMF) [14], [15]. The VMF is a vector processing operator that has been introduced as an extension of the scalar median filter. The VMF can be derived either as a maximum likelihood estimate (MLE) when the underlying probability densities are double exponential or by using vector order–statistics techniques.

In both definitions, the generalized Minkowski metric $\|\mathbf{x}_i - \mathbf{x}_j\|_L$ is used to quantify the distance between two color pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ in the magnitude domain. The VMF output is the sample $\mathbf{x}_{(1)} \in W$ that minimizes the distance to the other samples inside W:

$$\mathbf{x}_{(1)} = \arg\min_{\mathbf{x}_i \in W} \sum_{j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\|_L. \tag{2}$$

The vector median of a population can be also defined as the minimal vector $\mathbf{x}_{(1)} \in W$ according to the vector ordering scheme shown in Figure 4. Since the ordering can be used to determine the positions of the different input vectors without any a priori information regarding the signal distributions, vector order-statistics filters, such as the VMF, are robust estimators.

Since the impulse response of the VMF is zero, it excellently suppresses impulsive noise [14]. To improve its performance in the suppression of additive Gaussian noise, the VMF has been combined with linear filters [14]. Other important VMF extensions include the weighted VMF [16] and VMF modifications following the properties of color spaces [17]. To speed up the calculation of the distances between the color vectors, the VMF based on the linear approximation of the Euclidean norm has been proposed [15].

VECTOR DIRECTIONAL FILTERS

The family of vector directional filters (VDFs) [18] represents a different type of vector processing filter. These filters operate on the directions of image vectors, aiming at eliminating vectors with atypical directions in the color space. Similar to the VMF scheme, the basic vector directional filter (BVDF) [18] can be derived either as the MLE or by using vector order-statistics techniques.

Since the BVDF operates in the directional domain of a color image, its output is the color vector $\mathbf{x}_{(1)} \in W$ whose direction is the MLE of directions of the input vectors [19]. The BVDF output $\mathbf{x}_{(1)}$ minimizes the angular ordering criteria to other samples inside the sliding filtering window W:

$$\mathbf{x}_{(1)} = \arg\min_{\mathbf{x}_i \in W} \sum_{j=1}^{N} \theta(\mathbf{x}_i, \mathbf{x}_j), \tag{3}$$

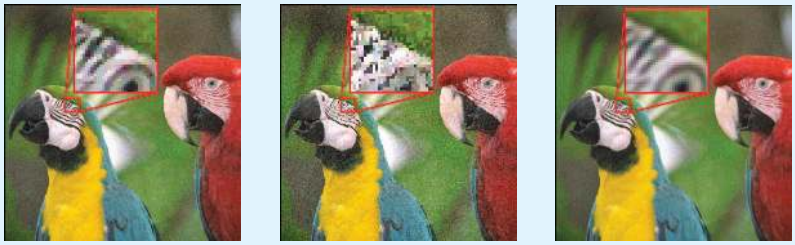
where $\theta(\mathbf{x}_i, \mathbf{x}_j)$ represents the angle between two color vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Using the R-ordering scheme (Figure 4) based on the similarity measure $\theta(\mathbf{x}_i, \mathbf{x}_j)$, the BVDF output can be equivalently expressed as the lowest vector order-statistic.

The angular minimization approach is useful for directional data such as color pixels. Since a vector's direction corresponds to its color chromaticity, minimizing the angular distances between vector inputs may produce better performance than the VMF-based approaches in terms of direction preservation. On the other hand, the VDF filters do not take into account the brightness of color vectors. To utilize both features in color image filtering [18], the generalized vector directional filters (GVDFs) and double window GVDF first eliminate the color vectors with atypical directions in the vector space, and subsequently process the vectors with the most similar orientation according to their magnitude. Thus, the GVDF splits the color image processing into directional processing and magnitude processing.

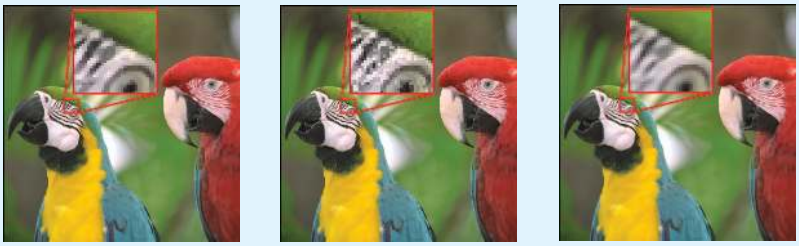
Another approach, the directional-distance filter (DDF) [20], combines both ordering criteria used in the VMF and the BVDF schemes. Spherical medians [18] minimize the angular criteria (3), excluding the constraint requiring the output sample, and lie within the filtering window W. An orientation-based, image enhancement technique for color images has been developed as well [21]. Selection weighted VDFs [22] minimize the aggregated weighted angles between the color vectors and improve the detail-preserving filtering characteristics of the conventional VDF schemes. Another class of filters, the generalized weighted vector filters [23], allow for the simultaneous utilization of both the magnitude and directional characteristics of the color vectors. These filters generalize a number of previous filtering techniques including VMF, BVDF, DDF, and their weighted modifications.

DATA ADAPTIVE FILTERS

Since standard color filters, such as VMF or BVDF, operating on the supporting window are designed to perform a fixed amount of smoothing, they cannot follow varying image statistics. Such a filter may exhibit excessive smoothing, affecting texture and blurring fine image details. To avoid such problems, local correlation in the data is used by applying fuzzy rules directly on the signal elements that lie within the processing window.

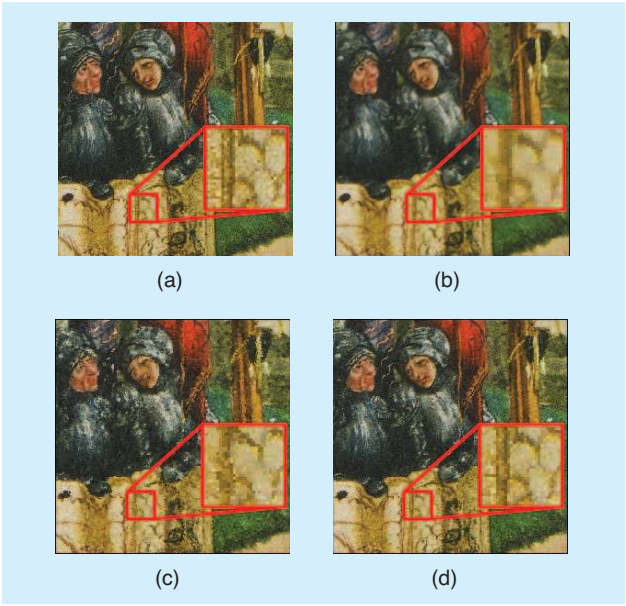


[FIG7] **Additive Gaussian noise (σ = 20) filtered output. (a) VMF, (b) BVDF, and (c) data adaptive filter utilizing the angular distance measure. See Figure 6(a) for original image and 6(b) for noisy input image.**

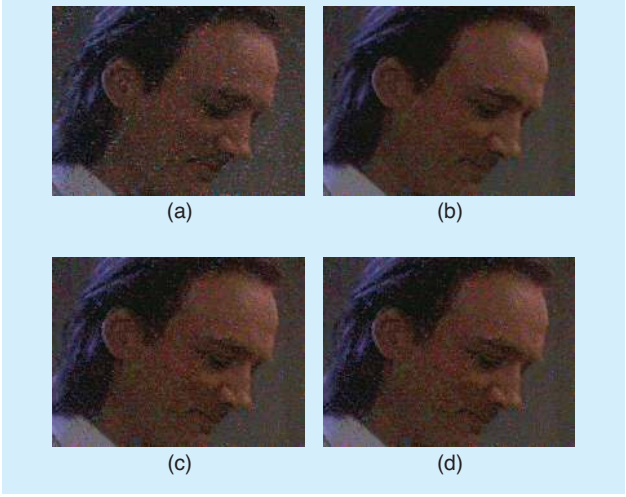


[FIG8] **5% impulsive noise filtered output. (a) VMF, (b) BVDF, and (c) data adaptive filter utilizing the angular distance measure. See Figure 6(a) for original image and 6(c) for noisy input image.**

[FIG9] **Mixed noise filtered output (Gaussian with σ = 20 and 5% impulsive noise. (a) VMF, (b) BVDF, and (c) data adaptive filter utilizing the angular distance measure. See Figure 6(a) for original image and 6(d) for noisy input image.**



[FIG10] (a) Input (real noisy), (b) VMF and (c) BVDF output digitized artwork images (256 × 256). (d) The adaptive filter uses aggregated weighted angular distances as the ordering criterion [22].



[FIG11] (a) Input (real noisy) and output television images (720 × 480) [(b)VMF and (c) BVDF.]. (d) The adaptive filter utilizes the angular distance measure.

Since the antecedents of fuzzy rules can be composed of several local characteristics, it is possible for such a filter to adapt to local data [24].

The most commonly used method to smooth high-frequency variations and transitions is averaging. Therefore, the general form of the data-dependent filter is given as a fuzzy weighted average [24] of the input vectors inside the supporting window W:

$$\hat{\mathbf{x}}_{(N+1)/2} = f\left(\sum_{i=1}^{N} w_i^* \mathbf{x}_i\right) = f\left(\sum_{i=1}^{N} w_i \mathbf{x}_i \Big/ \sum_{i=1}^{N} w_i\right) \quad (4)$$

where $f(\cdot)$ is a nonlinear function that operates over the weighted average of the input set, and w_i is the filter weight equivalent to the fuzzy membership function associated with the input color vector $\mathbf{x}_i$. Note that the two constraints $w_i^* \geq 0$ and $\sum_{i=1}^{N} w_i^* = 1$ are necessary to ensure that the filter output is an unbiased estimator and produces the samples within the desired intensity range.

Operating on the vectorial inputs $\mathbf{x}_i$,the weights w_i of (4) are determined adaptively using functions of a distance criterion between the input vectors. Using a distance $d(\mathbf{x}_i, \mathbf{x}_j)$ between input vectors and the sigmoidal membership function, the weight adaptation in (4) can be performed [24] by $w_i = \beta(1 + \exp\{\sum_{j=1}^{N} d(\mathbf{x}_i, \mathbf{x}_j)\})^{-r}$, where r is a parameter adjusting the weighting effect of the membership function, and β is a normalizing constant. Note that the distance measure $d(\mathbf{x}_i, \mathbf{x}_j)$ can be replaced with any similarity measure $s(\mathbf{x}_i, \mathbf{x}_j)$ discussed in "Distance and Similarity Measures."

Within the general fuzzy adaptive filter framework (4), numerous filters may be constructed [24] by changing the form of the nonlinear function $f(\cdot)$, as well as the way the fuzzy weights are calculated. The choice of these two parameters determines the filter characteristics.

APPLICATION OF THE FILTERING FRAMEWORKS

NOISE REMOVAL

In real-world scenarios, noise may result from many sources, such as electronic interference, flaws in the data transmission procedure, sensor malfunction, and simply the underlying physics of the imaging sensor itself. Based on the difference between the observation (noisy) color vector $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ and the original (desired) sample $\mathbf{o}_i = [o_{i1}, o_{i2}, o_{i3}]^T$, the noise corruption is modeled via the additive noise model [14] defined as follows:

$$\mathbf{x}_i = \mathbf{o}_i + \mathbf{v}_i, \quad (5)$$

where $\mathbf{v}_i = [v_{i1}, v_{i2}, v_{i3}]^T$ is the vector describing the noise process and i denotes the spatial position of the samples in the image. Note that $\mathbf{v}_i$ can describe either signal-dependent or independent noise.

A charge-coupled device (CCD) is commonly used as the sensor in many imaging devices [6] and is accompanied by a whole host of noise sources such as photon shot noise, dark current shot noise, on-chip and off-chip amplifier noise, and fixed pattern noise [25]. Through appropriate system design choices and the availability of necessary resources, many sources of noise can be significantly reduced. However, shot noise resulting from the photo-electric process can never be removed through camera design.

The photo-electric process is governed by Poisson statistics [25]. However, considering the likely presence of many noise sources, it is reasonable to assume that the overall noise process can be modeled as a zero mean white Gaussian, affecting each color component and pixel position independently [24], [26]. It is

also assumed that the noise variance σ is the same for all three color components in a correlated color space, such as RGB.

Given this model, the noise can be reduced to a scalar perturbation, the magnitude of the noise vector $p_i = \|\mathbf{v}_i\| = \sqrt{v_{i1}^2 + v_{i2}^2 + v_{i3}^2}$. It follows that the distribution of the p_is is [26]:

$$Pr(p) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^3 4\pi p^2 e^{-\frac{p^2}{2\sigma^2}}. \tag{6}$$

As shown in Figure 5, this perturbation results in a "noise cone" in the RGB color space. Using basic geometry, this vector magnitude perturbation can be translated into an angular perturbation A. Assuming that $\|\mathbf{o}_i\| \gg \sigma$ for most pixels, A can be approximated to have the distribution [26] $Pr(A) \approx A(\|\mathbf{o}\|^2/\sigma^2) \exp\{-(\|\mathbf{o}\|^2 A^2/(2\sigma^2))\}$. This is in the form of a Rayleigh distribution with mean $\overline{A} \approx \sqrt{\sigma^2\pi/(2\|\mathbf{o}\|^2)}$.

Using this concept of color noise as an angular perturbation of the original color vector represented in a correlated vector color space, the effect of the median operator can be roughly derived. Assuming that the median vector $\mathbf{x}_{(1)}$ chosen from a window of size N is the sample that is closest to the original signal $\mathbf{o}_i$, the average angular perturbation is reduced to $\overline{A}_N \approx \sqrt{\sigma^2\pi/(2N\|\mathbf{o}\|^2)}$. This reduction by a factor of $\sqrt{N}$ represents a rough upper bound on the smoothing capability of the median operator. Of course, N should not be too large so as to corrupt the nonstationary features of the original image.

TWO FUNDAMENTAL TYPES OF NOISE ARE CONSIDERED: NOISE PRODUCED DURING IMAGE FORMATION AND NOISE PRODUCED DURING TRANSMISSION.

Aside from acquisition noise, noise generated during image transmission must also be considered. Transmission noise has been found to be mostly impulsive in nature with sources ranging from human-made (e.g., switching and interference) to natural (e.g., lightning) [24], [27], [28]. This noise can be simply modeled by replacing the original color vector $\mathbf{o}_i$ with the impulse noise vector $\mathbf{v}_i$ with probability p (typically expressed as a percentage). The impulse vector $\mathbf{v}_i$ is independent from pixel to pixel and generally has significantly deviating characteristics (e.g., amplitude in at least one of the components) compared to those of the neighboring samples.

Figure 6 shows some simulated examples of both individual and mixed noise corruption. The results obtained from applying the VMF, BVDF, and a data adaptive filter to various noisy images (both simulated and real) are shown in Figures 7–11. Note that the data adaptive filter uses the angular distance criterion and a sigmoidal membership function. The typical 3×3 filter window (Figure 15) was used in all cases. For the Gaussian, impulsive, and mixed noise (Figures 7–9), it can be generally said that the VMF and BVDF perform best under impulsive noise, while the data adaptive filter is able to remove both types of noise at the cost of some reduced sharpness. Figure 10 shows output from applying the filtering schemes to digitized artwork for the purposes of restoration [29]. Figure 11 shows the results where the input is a television image. Finally, "Vector Processing of cDNA Microarray Images" shows results from applying the filters to a cDNA microarray image.

EDGE DETECTION

In monochrome images, edges can be simply modeled as discontinuities in intensity. However, the situation is much more complex for color images. Although it is not common, the edge of an object may be characterized by a change in chromaticity with the intensity remaining constant. Numerous operators have been devised, such as the vector gradient operator, for the detection of edges in color images [24], [30]. Following the theme of this article, the vector order statistics operators for color edge detection are discussed.

The simplest edge detector designed within this framework is the vector range (VR) detector [24]:

$$VR = d\left(\mathbf{x}_{(N)}, \mathbf{x}_{(1)}\right). \tag{7}$$

The vectors $\mathbf{x}_{(1)}$ and $\mathbf{x}_{(N)}$ correspond to the extremes of the R-ordered list and $d(\cdot,\cdot)$ is some distance measure corresponding to the same measure used to order the vectors. Thus, VR quantitatively expresses the deviation of values within the processing window. By thresholding VR, the presence of edges may be detected. However, VR is rather sensitive to noise. In order to resolve this, the concept is generalized by using a linear combination of the ordered vectors. As a result, the vector dispersion edge detectors (VDEDs) are created as follows [24]:

$$\text{VDED} = \left\| \sum_{i=1}^{N} \alpha_i \mathbf{x}_{(i)} \right\|, \tag{8}$$

where $\|\cdot\|$ is some appropriate norm. Using $\alpha_1 = -1$, $\alpha_N = 1$, and $\alpha_i = 0$, $i = 2, 3, \ldots, N-1$ the special case of the VR detector is obtained. A further generalization is achieved by considering k sets of coefficients α_i and then processing the set of resulting vector magnitudes. One useful example is the minimum vector dispersion (MVD) operator defined as follows [24]:

$$\text{MVD} = \min_j \left(\left\| \mathbf{x}_{(N-j+1)} - \sum_{i=1}^{l} \frac{\mathbf{x}_{(i)}}{l} \right\| \right), \quad j = 1, 2, \ldots, k, \quad k, l < N, \tag{9}$$

where the parameters k and l control the tradeoff between complexity and noise attenuation.

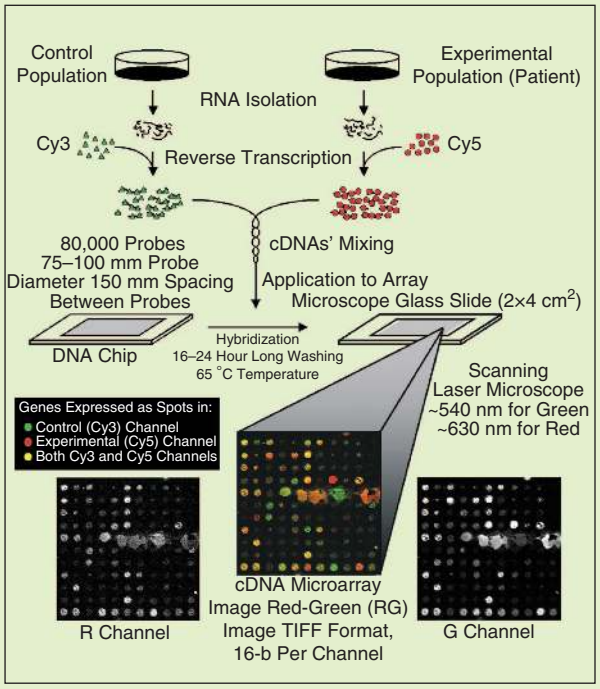
For demonstration purposes, the simple VR detector is applied under noiseless conditions. This is shown in Figure 12 with (b) and (c) corresponding to the use of the Euclidean and

VECTOR PROCESSING OF cDNA MICROARRAY IMAGES

cDNA Microarray Technology

Microarray technology [1], [2] is the most popular functional genomic tool used to effectively analyze changes in human genome, which are caused by carcinogens, reproductive toxins, and genetic diseases. Using a Cy3/Cy5 system, complementary deoxyribonucleic acid (cDNA) microarrays are formed as red-green (RG) images indicating particular genes expressed as spots [2]. The vast number of spots contained in microarray images necessitates the development of an automated image processing pipeline for parallel analysis of thousands of genes [1]–[3].

cDNA Microarray Image Formation



Two-color (Cy3/Cy5) system [2], [3]. Control (Cy3) and experimental (Cy5) channels.

Problem Formulation

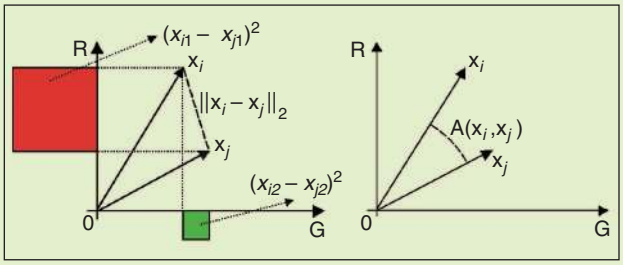
Due to variations in the image background and the spot sizes and positions, microarray image analysis task is complicated and challenging. Noise contamination caused by photon noise, electronic noise, laser light reflection, and dust on the glass slide significantly contributes to the overall noise contributions ν_i, [2]:

$$\mathbf{x}_i = \mathbf{o}_i + \mathbf{v}_i$$

Impairments present in microarray images prohibit the extraction of absolute or relative intensity values from each spot and identification of the genes expressed in a particular cell type. Since each error is propagated down through analysis steps, image filtering prior to subsequent analysis is necessary [2].

Differences Between two cDNA Vectors

- Effective quantification of the differences between cDNA vectors:
  - differences in magnitude
  - differences in orientation (direction)



- Euclidean metric (to quantify the magnitude difference)

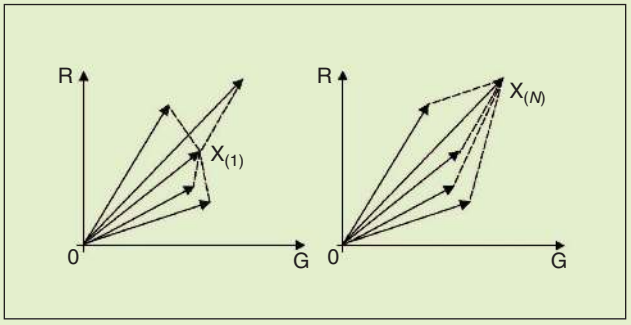
$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2}$$

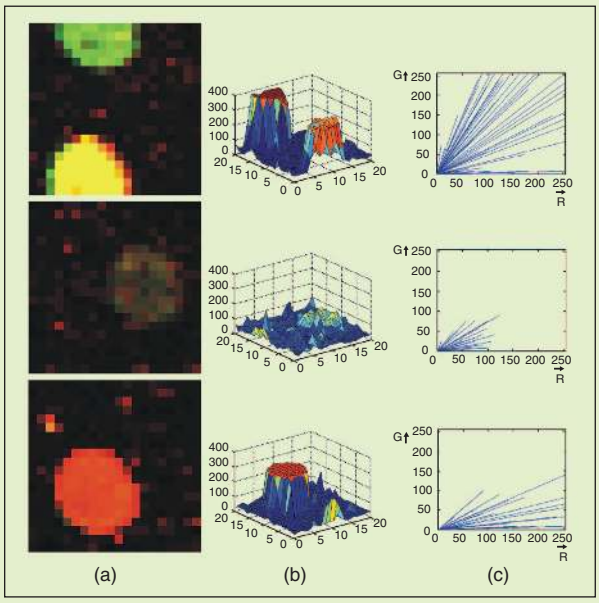
- Angular distance (to quantify the difference in vector directionality)

$$\mathbf{A}(\mathbf{x}_i, \mathbf{x}_j) = \arccos\left(\frac{\mathbf{x}_i \mathbf{x}_j^T}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}\right).$$

Aggregated Distances (AD)

- Used to determine centroid (associated with the minimum AD) or outlier (associated with the maximum AD) in the population of cDNA vectorial inputs



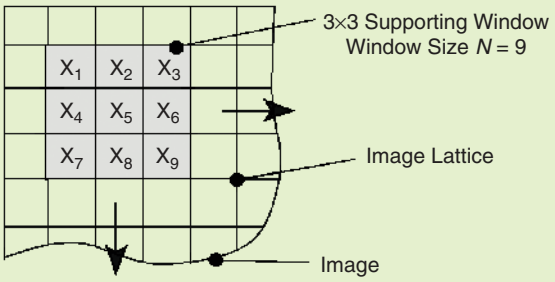


cDNA image characteristics (a) cDNA image vectors. (b) Magnitude characteristics. (c) Directional characteristics.

Both magnitude and directional information can be used to process the cDNA image vectors [2].

Vector Processing: A Framework

- Supporting running window based vector filtering scheme input set of cDNA vectors $W = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$



- Each vector $\mathbf{x}_i$, for $i = 1, 2, \ldots, N$, is associated with the AD [2]

$$D_i = \sum_{j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\|_2$$

- Noise can be removed using the vector median filter [2]

$$\mathbf{y} = \min \arg_{\mathbf{x}_i \in W} D_i \qquad (1)$$

- or data-dependent adaptive filtering [4]

$$\mathbf{y} = f\left(\sum_{i=1}^{N} w_i \mathbf{x}_i \Big/ \sum_{i=1}^{N} w_i\right) = f\left(\sum_{i=1}^{N} w_i^* \mathbf{x}_i\right) \qquad (2)$$

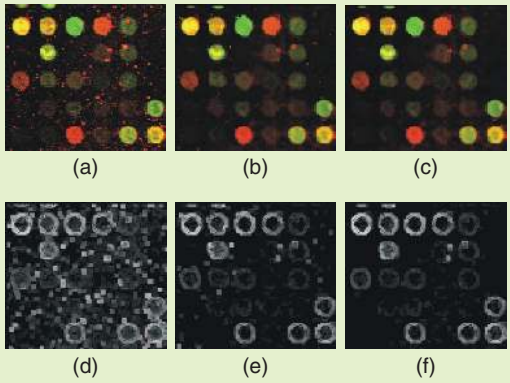
with the weights adapted as follows [4]:

$$w_i = \beta(1 + \exp(D_i))^{-r}$$

- Microarray spots are localized using vector range detector [2]

$$d_{VR} = \|\mathbf{x}_{(N)} - \mathbf{x}_{(1)}\|_2 \qquad (3)$$

Experimental Results



Achieved results: (a) input; (b) output of (1); (c) output of (2); (d)–(f) edge maps corresponding to (a)–(c) are produced using (3).

Conclusions

Vector processing operations [2], [4] use essential spectral and spatial information to remove noise and localize microarray spots. The proposed fully automated vector technique can be easily implemented in either hardware or software; and incorporated in any existing microarray image analysis and gene expression tool.

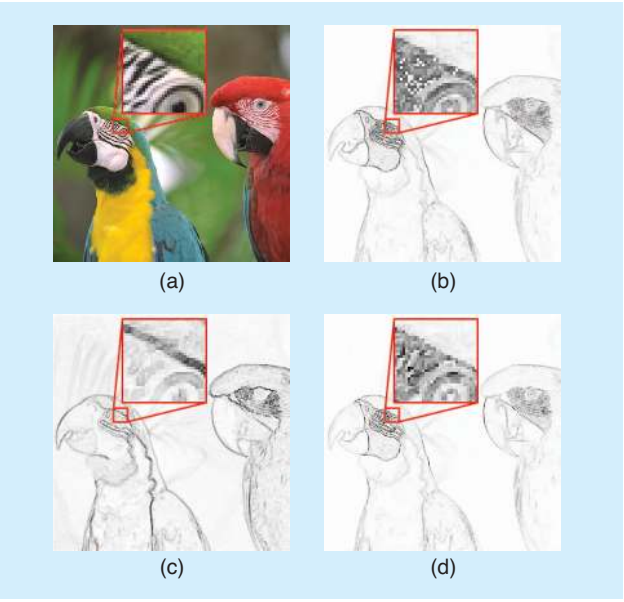
References

[1] S.K. Moore, "Making chips to probe genes," *IEEE Spectr.*, vol. 38, no. 3, pp. 54–60, Mar. 2001.

[2] R. Lukac, K.N. Plataniotis, B. Smolka, and A.N. Venetsanopoulos, "A multichannel order-statistic technique for cDNA microarray image processing," *IEEE Trans. Nanobioscience*, vol. 3, no. 4, pp. 272–285, Dec. 2004.

[3] A.K. Whitchurch, "Gene expression microarrays," *IEEE Potentials*, vol. 21, no. 1, pp. 30–34, Feb.–Mar. 2002.

[4] R. Lukac, K.N. Plataniotis, B. Smolka, and A.N. Venetsanopoulos, "cDNA microarray image processing using fuzzy vector filtering framework," *J. Fuzzy Sets Syst.*, to appear 2005.



[FIG12] Edge detection input and output (512 × 512). (a) Input, (b) VR (Euclidean), (c) VR (angular), and (d) MVD (Euclidean) detector using $k = 3$ and $l = 4$.

angular distance measures, respectively. Figure 12(d) shows the output using the MVD detector using the Euclidean distance and parameters $k = 3$ and $l = 4$.

SPATIAL INTERPOLATION

Spatial interpolation of a digital image is the process of increasing the number of pixels representing the image data. It is required for tasks such as zooming and displaying on high resolution display devices [31]. Increasing the spatial resolution of digital camera output is an important application of spatial interpolation, which is currently of great interest [32]. Due to the commercial proliferation of single-sensor digital cameras, spatial interpolation closely relates to demosaicking [33], and therefore, some new solutions combine both demosaicking and zooming concepts to provide the enlarged, demosaicked camera output [32]. As with most other color imaging applications, scalar techniques performed on the individual color channels are insufficient, since the correlation between the channels is not considered. Also, many conventional methods such as nearest neighbor pixel replication, bilinear interpolation, and spline based techniques often cause excessive blurring or geometric artifacts [31]. This begs for the more sophisticated vector based, nonlinear approaches discussed in this article.

A 2-D factor K interpolation involves the estimation of $K^2 - 1$ pixels in the region of every one original pixel. Multiple options are open in terms of the arrangement of the original pixels in the new, high-resolution image. For the sake of brevity, the simple case of $K = 2$ and a uniform upsampling pattern is considered here. Using a 3×3 processing window (Figure A in "Window-Based Filtering Basics") on the upsampled image, the following three pixel configurations are obtained when the window is centered on an empty pixel position: 1) $\{x_1, 0, x_3, 0, 0, 0, x_7, 0, x_9\}$, 2) $\{0, 0, 0, x_4, 0, x_6, 0, 0, 0\}$, and 3) $\{0, x_2, 0, 0, 0, 0, 0, x_8, 0\}$, where 0 represents an empty position. Configurations 2) and 3) provide an insufficient number of original pixels for the estimation of the unknown vector at the center of the window. Consequently, a two-pass procedure is employed. The first pass requires positioning the window to achieve all possible type 1) configurations and estimating the unknown vector using some operator on the four known values. After the first pass, all configurations become $\{0, x_2, 0, x_4, 0, x_6, 0, x_8, 0\}$, with the center of the window to be interpolated. Thus, the second pass is performed on all remaining positions with unknown pixels by using an operator on the two known and two estimated pixels within the window.

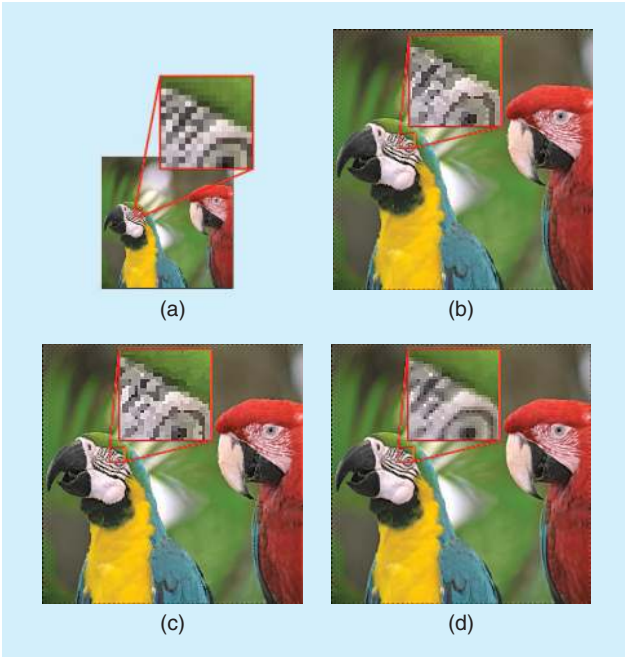
Using this methodology, the VMF, BVDF, and a data adaptive filter are applied with the input and output shown in Figure 13. The data adaptive filter uses the Euclidean distance measure and a sigmoidal membership function. During each pass, the four color vectors available at each window position are used as input to the filters. As can be seen in Figure 13(b) and (c), the VMF and BVDF are able to maintain sharp edges. The data adaptive output in Figure 13(d) is not as sharp but provides an excellent tradeoff between sharpness and the reduction of artifacts.

SPECTRAL INTERPOLATION

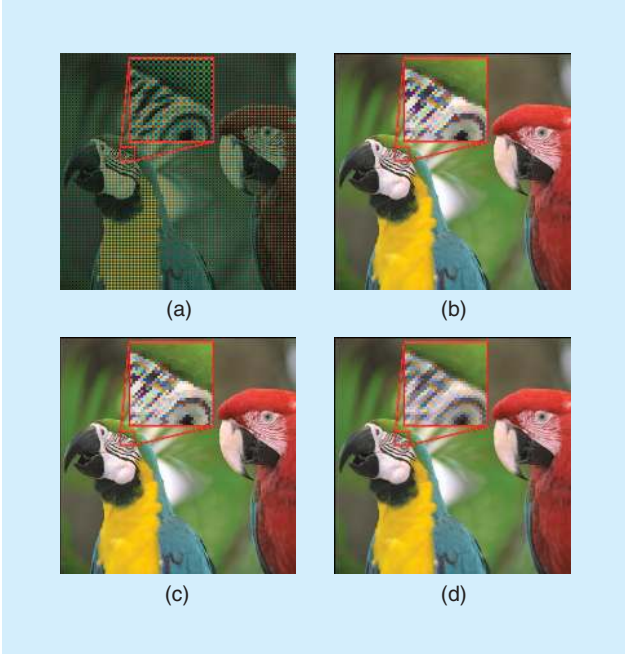
In the domain of digital color imagery, spectral interpolation or demosaicking refers to the process of estimating individual color components [32], [33]. Since the imaging sensor is a monochrome device, most consumer level digital cameras use either a CCD or CMOS sensor in conjunction with a CFA in order to obtain a mosaic of color components [32], [33]. The sensor values constitute the so-called CFA image, which is a low-resolution color image due to the fact that only one spectral component is available at each spatial location [32]. These individual color components must be combined to produce an image with a full color vector at each position.

The common Bayer pattern CFA [34] is used here to demonstrate how the vector-based filtering approach may be applied to create the interpolated full color image. Following the principles of window based filtering, it is assumed that the image is stationary within the window, and so adjacent color components can be combined into color vectors. Thus, the vector filtering frameworks are not directly applied to a Bayer CFA image [Figure 14(a)], but instead, a set of "pseudopixels" is used as the vector filters' input [35]. For example, having a 3×3 window with the CFA inputs 1) $\{g_1, r_1, g_2, b_1, g_3, b_2, g_4, r_2, g_5\}$ or 2) $\{r_1, g_1, r_2, g_2, b_1, g_3, r_3, g_4, r_4\}$, the procedure results in four color vectors $\{[r_1, g_3, b_1]^T, [r_1, g_3, b_2]^T, [r_2, g_3, b_1]^T, [r_2, g_3, b_2]^T\}$ or eight color vectors $\{[r_1, g_1, b_1]^T, [r_1, g_2, b_1]^T, [r_2, g_1, b_1]^T, [r_2, g_3, b_1]^T, [r_3, g_2, b_1]^T, [r_3, g_4, b_1]^T, [r_4, g_3, b_1]^T, [r_4, g_4, b_1]^T\}$, respectively. If a G CFA component is located at the window center, then the population of four vectors is used as the input for the vector filtering frameworks. Otherwise, the window center is an R (or B) CFA component, and eight vectors are used to estimate the color pixel at the center of the window.

The images shown in Figure 14(b)–(d) were generated using the VMF, BVDF, and a data adaptive filter based on the angular



[FIG13] Spatial interpolation input (256 × 256) and output (512 × 512). (a) Input, (b) VMF, (c) BVDF, and (d) adaptive filter utilizing the Euclidean distance measure.



[FIG14] Spectral interpolation input and output (512 × 512). (a) Input (color version of the Bayer CFA sensor image), (b) VMF, (c) BVDF, and (d) adaptive filter utilizing the angular distance measure.

distance, respectively. The VMF and BVDF generate sharp results with some color artifacting around sharp, high contrast edges. The data adaptive method is able to reduce some of the color artifacts at the cost of some minor blurring.

CONCLUSION

Vector filtering of color images is a rich and expanding field. Noise removal and the correct perception of desired color is of paramount importance in emerging applications related to biomedical science, earth science, cultural heritage preservation, video communications, image postprocessing, robotic inspection and surveillance. From the presented examples, it is clear that vector image filtering techniques open up a world with almost endless possibilities.

Selected sets of vector filtering techniques are being introduced as standard tools in sophisticated graphics/image processing software tools available to practitioners as well as end-users. A growing expanse of Internet applications will include color image processing tools in the form of active scripts automatically running when browsing the Web. Images appearing in the Web will be automatically filtered and displayed in high resolution quality. Other processing steps (e.g., edge detection, image segmentation, and adjusting operations such as histogram modification and changes of color appearance) will be performed in the same manner.

It is no overstatement to claim that color image filtering techniques have an extremely valuable position in modern color image science, communication, and multimedia applications. As our insatiable thirst for rich, meaningful graphical information is not likely to subside any time soon, neither will the need for effective filtering techniques.

AUTHORS

Rastislav Lukac received the Ph.D. degree in telecommunications from the Technical University of Kosice, Slovak Republic, in 2001. Since 2002, he has been a researcher in Slovak Image Processing Center, Dobsina. From January 2003 to March 2003 he was a postdoctoral fellow at the Artificial Intelligence and Information Analysis Lab at the Aristotle University of Thessaloniki, Greece. Since May 2003, he has been a postdoctoral fellow with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto in Toronto, Canada. His research interests include digital camera image processing, microarray image processing, multimedia security, and nonlinear filtering and analysis techniques for color image and video processing. He is a Member of the IEEE Circuits and Systems, Consumer Electronics, and Signal Processing Societies. In 2003, he was awarded the NATO/NSERC Science Award.

Bogdan Smolka received the diploma in physics degree from the Silesian University in Katowice, Poland, in 1986 and the Ph.D. degree in automatic control from the Department of Automatic Control, Silesian University of Technology, Gliwice, Poland. From 1986 to 1989, he was a teaching assistant at the Department of Biophysics, Silesian Medical University, Katowice, Poland, and from 1992 to 1994, at the Technical University of Goeppingen, Germany. In 1994, he joined the Silesian University of Technology, where he has been an associate professor, since 1998, in the Department of Automatic Control. He is also an associate researcher at Multimedia Laboratory of the University of Toronto, Canada. His current research interests include low-level color image processing, human-computer interaction, wavelet based methods of image compression, and the visual aspects of image quality.

Karl Martin received the B.A.Sc. degree in engineering science (electrical specialty) and the M.A.Sc. degree in electrical engineering from the University of Toronto, Canada, in 2001 and 2003, respectively. He is currently pursuing a Ph.D. at the University of Toronto. His research interests include multimedia security, multimedia processing, wavelet-based image coding, object-based coding, and CFA processing. He is a Member of both the IEEE Signal Processing Society and IEEE Communications Society. Since 2003, he has held the position of vice-chair of the Signals and Applications Chapter, IEEE Toronto Section.

Konstantinos N. Plataniotis is an assistant professor with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto in Toronto, Canada, a Nortel Institute for Telecommunications associate, a member of the Knowledge Media Design Institute, and an adjunct professor in the Department of Mathematics, Physics and Computer Science at Ryerson University, Toronto, Canada. His research interests include multimedia signal processing, intelligent and adaptive systems, and wireless communication systems. He is a registered professional engineer in Ontario, a member of the Technical Chamber of Greece, and the IEEE Toronto Section chair.

Anastasios N. Venetsanopoulos received the diploma in engineering degree from the National Technical University of Athens, Greece, in 1965 and the M.S., M.Phil., and Ph.D. degrees in electrical engineering from Yale University in 1966, 1968, and 1969, respectively. He joined the Department of Electrical and Computer Engineering of the University of Toronto in 1968. Since 1997, he has been associate chair of graduate studies of the Edward S. Rogers Sr. Department of Electrical and Computer Engineering. He is a contributor and coauthor of many books. He has served on numerous IEEE boards, councils, and technical conference committees. He was president of the Canadian Society for Electrical Engineering and vice president of the Engineering Institute of Canada (EIC). He was a guest editor or associate editor for several IEEE journals and the editor of the *Canadian Electrical Engineering Journal*. He was the technical program cochair of ICIP-2001. He is a member of the IEEE Communications, Circuits and Systems, Computer, and Signal Processing Societies, as well as a member of Sigma Xi, the Technical Chamber of Greece, the European Association of Signal Processing, the Association of Professional Engineers of Ontario (APEO) and Greece. He is a Fellow of the IEEE and the EIC and was awarded an Honorary Doctorate from the National Technical University of Athens and the 1994 "Excellence in Innovation Award" of the Information Technology Research Centre of Ontario and Royal Bank of Canada, for innovative work in color image processing.

REFERENCES

[1] M.J. Vrhel, E. Saber, and H.J. Trussell, "Color image generation and display technologies," *IEEE Signal Processing Mag.*, vol. 22, no. 1, pp. 23–33, Jan. 2005.

[2] R. Ramanath, W.E. Snyder, Y. Yoo, and M.S. Drew, "Color image processing pipeline," *IEEE Signal Processing Mag.*, vol. 22, no. 1, pp. 34–43, Jan. 2005.

[3] R. Cucchiara and C. Grana, "Color analysis, segmentation and retrieval in medical imaging," submitted.

[4] A. Koschan and M. Abidi, "Detection and classification of edges in color images," *IEEE Signal Processing Mag.*, vol. 22, no. 1, pp. 64–73, Jan. 2005.

[5] G. Wyszecki and W.S. Stiles, *Color Science, Concepts and Methods, Quantitative Data and Formulas,* 2nd ed. New York: Wiley, 1982.

[6] G. Sharma and H.J. Trussell, "Digital color imaging," *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 901–932, July 1997.

[7] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, "A standard default color space for the internet—sRGB," Hewlett-Packard and Microsoft Tech. Rep, 1996 [Online]. Available: http://www.w3.org/Graphics/Color/sRGB.html)

[8] I. Pitas and A.N. Venetsanopoulos, "Order statistics in digital image processing," *Proc. IEEE*, vol. 80, no. 12, pp. 1892–1919, Dec. 1992.

[9] V. Barnett, "The ordering of multivariate data," *J. Roy. Statist. Soc. A*, vol. 139, no. 3, pp. 318–354, 1976.

[10] K. Tang, J. Astola, and Y. Neuvo, "Nonlinear multivariate image filtering techniques," *IEEE Trans. Image Processing*, vol. 4, no. 6, pp. 788–798, June 1995.

[11] K. Plataniotis and A.N. Venetsanopoulos, "Vector processing," in *Colour Image Processing*, S. Sangwine, Ed. London: Chapman & Hall, 1998, pp. 188–209.

[12] I. Pitas and P. Tsakalides, "Multivariate ordering in color image filtering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 3, pp. 247–259, Sept. 1991.

[13] R. Hardie and G. Arce, "Ranking in R^P and its use in multivariate image estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 2, pp. 197–208, June 1991.

[14] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, vol. 78, no. 4, pp. 678–689, Apr. 1990.

[15] M. Barni, V. Cappellini, and A. Mecocci, "Fast vector median filter based on Euclidean norm approximation," *IEEE Signal Processing Lett.*, vol. 1, no. 6, pp. 92–94, June 1994.

[16] T. Viero, K. Oistamo, and Y. Neuvo, "Three-dimensional median related filters for color image sequence filtering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 2, pp. 129–142, Apr. 1994.

[17] C. Regazoni and A. Teschioni, "A new approach to vector median filtering based on space filling curves," *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 1025–1037, July 1997.

[18] P. Trahanias, D. Karakos, and A. Venetsanopoulos, "Directional processing of color images: Theory and experimental results," *IEEE Trans. Image Processing*, vol. 5, no. 6, pp. 868–881, June 1996.

[19] N. Nikolaidis and I. Pitas, "Nonlinear processing and analysis of angular signals," *IEEE Trans. Signal Processing*, vol. 46, no. 12, pp. 3181–3194, Dec. 1998.

[20] D. Karakos and P. Trahanias, "Generalized multichannel image-filtering structure," *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 1038–1045, July 1997.

[21] B. Tang, G. Sapiro, and V. Caselles, "Color image enhancement via chromaticity difussion," *IEEE Trans. Image Processing*, vol. 10, no. 5, pp. 701–707, May 2001.

[22] R. Lukac, "Adaptive color image filtering based on center-weighted vector directional filters," *Multidimensional Syst. Signal Processing*, vol. 15, no. 2, pp. 169–196, Apr. 2004.

[23] R. Lukac, K.N. Plataniotis, B. Smolka, and A.N. Venetsanopoulos, "Generalized selection weighted vector filters," *EURASIP J. Appl. Signal Processing: Special Issue Nonlinear Signal Image Processing,* vol. 2004, no. 12, pp. 1870–1885, Oct. 2004.

[24] K. Plataniotis and A.N. Venetsanopoulos, *Color Image Processing and Applications*. New York: Springer Verlag, 2000.

[25] G.C. Holst, *CCD Arrays, Cameras, and Displays*, 2nd ed. Bellingham, WA: SPIE, 1998.

[26] K.K. Sung, "A vector signal processing approach to color," M.S. thesis, MIT, 1992.

[27] Y. Neuvo and W. Ku, "Analysis and digital realization of a pseudorandom gaussian and impulsive noise source," *IEEE Trans. Commun.*, vol. 23, no. 9, pp. 849–858, Sept. 1975.

[28] W. Henkel, T. Kessler, and H.Y. Chung, "Coded 64-CAP ADSL in an impulse-noise environment—modeling of impulse noise and first simulation results," *IEEE J. Select. Areas Commun.*, vol. 13, no. 9, pp. 1611–1621, Dec. 1995.

[29] M. Barni, F. Bartolini, and V. Cappellini, "Image processing for virtual restoration of artworks," *IEEE Multimedia*, vol. 7, no. 2, pp. 34–37, Apr.–Jun. 2000.

[30] J. Scharcanski and A. Venetsanopoulos, "Edge detection of color images using directional operators," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 397–401, Apr. 1997.

[31] N. Herodotou and A.N. Venetsanopoulos, "Colour image interpolation for high resolution acquisitions and display devices," *IEEE Trans. Consumer Electron.*, vol. 41, no. 4, pp. 1118–1126, Nov. 1995.

[32] R. Lukac, K. Martin, and K.N. Plataniotis, "Digital camera zooming based on unified CFA image processing steps," *IEEE Trans. Consumer Electron.*, vol. 50, no. 1, pp. 15–24, Feb. 2004.

[33] B.K. Gunturk, J. Glotzbach, Y. Altunbasak, R.M. Mersereau, and R.W. Schafer, "Demosaicking: Color filter array interpolation," *IEEE Signal Processing Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.

[34] B.E. Bayer, "Color imaging array," U.S. Patent 3 971 065, July 1976.

[35] M.R. Gupta and T. Chen, "Vector color filter array demosaicing," in *Proc. SPIE vol. 4306: Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications II*, 2001, pp. 374–382.

SP