Comprehensive analysis of edge detection in color image processing

Shu-Yu Zhu
University of Toronto
Department of Electrical and Computer Engineering
Toronto, Ontario M5S 3G4
Canada

Konstantinos N. Plataniotis
Ryerson Polytechnic University
Math, Physics and Computer Science Department
350 Victoria Street
Toronto, Ontario M5B 2K3
Canada
E-mail: kplatani@acs.ryerson.ca

Anastasios N. Venetsanopoulos
University of Toronto
Department of Electrical and Computer Engineering
Toronto, Ontario M5S 3G4
Canada

Abstract. Various approaches to edge detection for color images, including techniques extended from monochrome edge detection as well as vector space approaches, are examined. In particular, edge detection techniques based on vector order statistic operators and difference vector operators are studied in detail. Numerous edge detectors are obtained as special cases of these two classes of operators. The effect of distance measures on the performance of different color edge detectors is studied by employing distance measures other than the Euclidean norm. Variations are introduced to both the vector order statistic operators and the difference vector operators to improve noise performance. They both demonstrate the ability to attenuate noise with added algorithm complexity. Among them, the difference vector operator with adaptive filtering shows the most promising results. Other vector directional filtering techniques are also introduced and utilized for color edge detection. Both quantitative and subjective tests are performed in evaluating the performance of the edge detectors, and a detailed comparison is presented.
[S0091-3286(99)00904-6]

Subject terms: color image processing; edge detection; vector techniques; subjective evaluation.

Paper 980180 received May 4, 1998; revised manuscript received Oct. 28, 1998; accepted for publication Nov. 2, 1998.

1 Introduction

The subject of color image processing has gained increasing recent attention because color images convey more information about objects in a scene than gray-scale images and this information can be used to further refine the performance of an imaging system. The multichannel nature of a color image, however, adds considerable complexity to the processing system. One of the challenges facing color image processing is to extract the additional color information without incurring large complexity in the system.

One of the fundamental tasks in image processing is edge detection. High level image processing, such as object recognition, segmentation, image coding, and robot vision, depends on the accuracy of edge detection since edges contain essential image information. In a monochrome image, an edge usually corresponds to object boundaries or changes in physical properties such as illumination or reflectance. This definition is more elaborate in the case of multispectral (color) images since more detailed edge information is expected from color edge detection. According to psychological research on the human visual system,[1,2] color plays a significant role in the perception of boundaries. Monochrome edge detection may not be sufficient for certain applications since no edges will be detected in gray value images when neighboring objects have different hues but equal intensities.[3] Objects with such boundaries are treated as one big object in the scene. Since the capability to distinguish between different objects is crucial for applications such as object recognition and image segmentation, the additional boundary information provided by color is of paramount importance. Color edge detection also outperforms monochrome edge detection in low contrast images.[3] There is thus a strong motivation to develop efficient color edge detectors that provide high quality edge maps.

Despite the relatively short period of time, numerous approaches of different complexities to color edge detection have been proposed. It is important to identify their strengths and weaknesses in choosing the best edge detector for an application. The major performance issues concerning edge detectors are their ability to extract edges accurately, their robustness to noise, and their computational efficiency. To provide a fair assessment, it is necessary to have a set of effective performance evaluation methods. Though numerous evaluation methods for edge detection have been proposed, there has not been any standardized method. In image processing, the evaluation methods can usually be categorized into objective and subjective evaluation. While objective evaluation can provide analytical data for comparison purposes, it is not sufficient to represent the complexity of the human visual systems. In most image processing applications, human evaluation is the final step, as noted by Clinque et al.[4] The subjective evaluation, which takes into account human perception, seems to be very attractive in this perspective. The visual assessment method proposed by Heath et al.[5,6] is entirely based on subjective evaluation. In this paper, both types of evaluation methods are utilized for comparing various edge detectors.

An overview of the methodology for color edge detection is presented in Section 2, where early approaches extended from monochrome edge detection, as well as the more recent vector space approaches are addressed. The edge detectors illustrated in this section, among others, are the Sobel operator,[3,7] the Laplacian operator,[3] the Mexican hat operator,[3,8] the vector gradient operator,[9] the directional operator,[10] the entropy operator,[11] and the Cumani operator.[12] Two families of vector based edge detection operators, vector order statistic operators[13,14] and difference vector operators[15] are studied in detail in Section 3 and 4, respectively. A variety of edge detectors obtained as special cases of the two families are introduced and their performances are evaluated. The evaluation results from both objective and subjective tests are listed in Section 5 followed by our conclusions in Section 6.

2 Overview of Color Edge Detection Methodology

2.1 *Techniques Extended from Monochrome Edge Detection*

In a monochrome image, an edge is defined as an intensity discontinuity. In the case of color images, the additional variation in color must also be considered. Early approaches to color edge detection are extensions of monochrome edge detection. These techniques are applied to the three color channels independently and then the results are combined using certain logical operation.[3] Several standard techniques can be applied in this way. One of the representative classes of edge detection is the Sobel operator. It can be realized by convolving the image with the following two convolution masks:

$$\mathbf{M}_x=\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad \mathbf{M}_y=\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}. \tag{1}$$

These two masks are applied to each color channel independently and the sum of the squared convolution gives an estimated gradient in each channel. A pixel is regarded as an edge point if the maximum of the gradient magnitude values in the three channels exceeds a predetermined threshold.

Another operator that can also be used in a similar fashion is the eight-neighbor Laplacian operator. The Laplacian convolution mask is defined as follows:

$$\mathbf{M}=\begin{pmatrix} 10 & 22 & 10 \\ 22 & 128 & 22 \\ 10 & 22 & 10 \end{pmatrix}. \tag{2}$$

Again, the Laplacian mask is applied to the three color channels independently and edge points are located by thresholding the maximum gradient magnitude.

Another group of edge detectors commonly used in monochrome edge detection is based on second derivative operators, and they can also be extended to color edge detection in the same way. A second derivative method can be implemented based on the preceding operator. The Mexican hat operator uses convolution masks generated based on the negative Laplacian derivative of the Gaussian distribution:

$$-\nabla^2 G(x,y)=\frac{x^2+y^2-2\sigma^2}{2\pi\sigma^6}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{3}$$

Edge points are located if zero-crossing occur in any one color channel.

The gradient operators proposed for gray-scale images[16] can also be extended to color images by taking the vector sum of the gradients for individual components.[9,11] Similar to Sobel and Laplacian operators, the gradient operator also employs first derivative like mask patterns. Other approaches consider performing operations in an alternative color space. The Hueckel edge operator[17] operates in the luminance, chrominance color space. The edges in the three color components are also assumed to be independent under the constraints that they must have the same orientation. In studying the application of the compass gradient edge detection method to color images, Robinson[18] also utilized different color coordinates.

One common problem with the preceding approaches is that they failed to take into account the correlation among the color channels, and as a result, they are not able to extract certain crucial information conveyed by color. For example, they tend to miss edges that have the same strength but in opposite direction in two of their color components. Consequently, the approach to treat the color image as vector space has been proposed.

2.2 *Vector Space Approach*

Various approaches proposed consider the problem of color edge detection in vector space. Color images can be viewed as a 2-D three-channel vector field,[19] which can be characterized by a discrete integer function $\mathbf{f}(x,y)$. The value of this function at each point is defined by a 3-D vector in a given color space. In the RGB color space, the function can be written as $\mathbf{f}(x,y)=(R(x,y),G(x,y),B(x,y))$, where (x,y) refers to the spatial dimensions in the 2-D plane. Most existing edge detection algorithms use either first or second difference between neighboring pixels for edge detection. A significant change gives rise to a peak in the first derivative and zero-crossing in the second difference, both of which can be identified fairly easily. Some of these operators are considered in the following.

2.2.1 *Vector gradient operators*

The vector gradient operator employs the concept of a gradient operator, except that instead of scalar space the operator operates in a 2-D three-channel color vector space. There are several ways of implementing the vector gradient operator.[9] One simple approach is to employ a 3×3 window centered on each pixel and then obtain eight distance values $(D_1,D_2,\ldots,D_8)$ by computing the Euclidean distance between the center vector and its eight neighboring vectors. The vector gradient (g) is then chosen as

$$g=\max(D_1,D_2,\ldots,D_8). \tag{4}$$

Another approach employs directional operators. Let the image be a vector function $\mathbf{f}(x,y) = (R(x,y),G(x,y),B(x,y))$, and let $\mathbf{r}$, $\mathbf{g}$, and $\mathbf{b}$ be the unit vectors along the R, G, and B axes, respectively. The horizontal and vertical directional operators can be defined as:

$$\mathbf{u}=\frac{\partial R}{\partial x}\mathbf{r}+\frac{\partial G}{\partial x}\mathbf{g}+\frac{\partial B}{\partial x}\mathbf{b}, \tag{5}$$

$$\mathbf{v}=\frac{\partial R}{\partial y}\mathbf{r}+\frac{\partial G}{\partial y}\mathbf{g}+\frac{\partial B}{\partial y}\mathbf{b}, \tag{6}$$

$$g_{xx}=\mathbf{u}\cdot\mathbf{u}=\left|\frac{\partial R}{\partial x}\right|^2+\left|\frac{\partial G}{\partial x}\right|^2+\left|\frac{\partial B}{\partial x}\right|^2, \tag{7}$$

$$g_{yy}=\mathbf{v}\cdot\mathbf{v}=\left|\frac{\partial R}{\partial y}\right|^2+\left|\frac{\partial G}{\partial y}\right|^2+\left|\frac{\partial B}{\partial y}\right|^2, \tag{8}$$

$$g_{xy}=\frac{\partial R}{\partial x}\frac{\partial R}{\partial y}+\frac{\partial G}{\partial x}\frac{\partial G}{\partial y}+\frac{\partial B}{\partial x}\frac{\partial B}{\partial y}. \tag{9}$$

Then the maximum rate of change of $\mathbf{f}$ and the direction of the maximum contrast can be calculated as:

$$\theta=\frac{1}{2}\arctan\frac{2g_{xy}}{g_{xx}-g_{yy}}, \tag{10}$$

$$F(\theta)=\tfrac{1}{2}[(g_{xx}+g_{yy})+\cos 2\theta(g_{xx}-g_{yy})+2g_{xy}\sin\theta]. \tag{11}$$

Edge can be obtained by thresholding $[F(\theta)]^{1/2}$.

The image derivatives along the x and y directions can be computed by convolving the vector function $\mathbf{f}$ with two spatial masks as follows:

$$\frac{\partial f_i}{\partial x}\simeq\frac{1}{6}\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}*f_i,$$

$$\frac{\partial f_i}{\partial y}\simeq\frac{1}{6}\begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 0\\ -1 & -1 & -1\end{bmatrix}*f_i. \tag{12}$$

Unlike the gradient operator extended from the mentioned monochrome edge detection, the vector gradient operator can extract more color information from the image because it considers the vector nature of the color image. On the other hand, the vector gradient operator is very sensitive to small texture variations.[14] This may be undesirable in some cases since it can cause confusion in identifying the real objects. The operator is also sensitive to Gaussian and impulse noise.

2.2.2 *Directional operators*

The direction of an edge in color images can be utilized in a variety of image analysis tasks.[20] A class of directional vector operators was proposed to detect the location and orientation of edges in color images.[10] In this approach, a color $c(r,g,b)$ is represented by a vector $\mathbf{c}$ in color space. Similar to the well known Prewitt operator[21] shown below,

$$\Delta H=\frac{1}{3}\begin{pmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{pmatrix},\quad \Delta V=\frac{1}{3}\begin{pmatrix}-1 & -1 & -1\\ 0 & 0 & 0\\ 1 & 1 & 1\end{pmatrix}, \tag{13}$$

the row and column directional operators (i.e., in the horizontal and vertical directions), each have one positive and one negative component. For operators of size $(2w+1)\times(2w+1)$ the configuration is the following:

$$\Delta\mathbf{H}=[\mathbf{H}_-\mathbf{0}\mathbf{H}_+],\quad \Delta\mathbf{V}=\begin{bmatrix}\mathbf{V}_-\\ \mathbf{0}\\ \mathbf{V}_+\end{bmatrix}, \tag{14}$$

where the parameter w is a positive integer. These positive and negative components are *convolution kernels*, denoted by $\mathbf{V}_-$, $\mathbf{V}_+$, $\mathbf{H}_-$ and $\mathbf{H}_+$, whose outputs are vectors corresponding to the local *average colors*. To estimate the color gradient at the pixel (x_o,y_o), the outputs of these components are calculated as follows:

$$\mathbf{H}_+(x_o,y_o)=\frac{1}{w(2w+1)}\sum_{y=y_o-w}^{y=y_o+w}\sum_{x=x_o+1}^{x=x_o+w}\mathbf{c}(x,y),$$

$$\mathbf{H}_-(x_o,y_o)=\frac{1}{w(2w+1)}\sum_{y=y_o-w}^{y=y_o+w}\sum_{x=x_o-1}^{x=x_o-w}\mathbf{c}(x,y),$$

$$\mathbf{V}_+(x_o,y_o)=\frac{1}{w(2w+1)}\sum_{y=y_o+1}^{y=y_o+w}\sum_{x=x_o-w}^{x=x_o+w}\mathbf{c}(x,y), \tag{15}$$

$$\mathbf{V}_-(x_o,y_o)=\frac{1}{w(2w+1)}\sum_{y=y_o-1}^{y=y_o-w}\sum_{x=x_o-w}^{x=x_o+w}\mathbf{c}(x,y),$$

where $\mathbf{c}(x,y)$ denotes the color (r,g,b) at the image location (x,y).

Local colors and local statistics affect the output of the operator components $[\mathbf{V}_+(x,y),\ \mathbf{V}_-(x,y),\ \mathbf{H}_+(x,y)$ and $\mathbf{H}_-(x,y)]$. To estimate the local variation in the vertical and horizontal directions, the following vector differences are calculated:

$$\Delta\mathbf{H}(x_o,y_o)=\mathbf{H}_+(x_o,y_o)-\mathbf{H}_-(x_o,y_o),$$

$$\Delta\mathbf{V}(x_o,y_o)=\mathbf{V}_+(x_o,y_o)-\mathbf{V}_-(x_o,y_o). \tag{16}$$

The scalars $\|\Delta\mathbf{H}(x_o,y_o)\|$ and $\|\Delta\mathbf{V}(x_o,y_o)\|$ give the variation rate at (x_o,y_o) in orthogonal directions (i.e., are the amounts of color contrast in the horizontal and vertical directions). The local changes in the color channels (i.e., R, G and B) can not be combined properly by simply adding the R, G and B components of $\Delta\mathbf{H}$ and $\Delta\mathbf{V}$. This approach leads to a mutual canceling out effect in several situations (e.g., when contrast is in phase opposition in different channels). Instead, the local changes in R, G and B are assumed to be independent (i.e., orthogonal), and the intensity of the local color contrast is obtained as the magnitude of the resultant vector in the RGB space (using the Euclidean norm), as shown in Equations (17) and (18). Therefore, the magnitude B of the maximum variation rate at (x_o,y_o) is estimated as the magnitude of the resultant vector:

$$B(x_o,y_o)=[\|\Delta\mathbf{V}(x_o,y_o)\|^2+\|\Delta\mathbf{H}(x_o,y_o)\|^2]^{1/2}, \tag{17}$$

and the direction θ of the maximum variation rate at (x_o,y_o) is estimated as:

$$\theta=\arctan\left[\frac{\Delta V'(x_o,y_o)}{\Delta H'(x_o,y_o)}\right]+k\pi, \tag{18}$$

where, k is an integer, and:

$$\Delta V'(x_o,y_o)=\begin{cases}\text{if } \|\mathbf{V}_+(x_o,y_o)\|\geqslant\|\mathbf{V}_-(x_o,y_o)\|\\ \text{then } \|\Delta\mathbf{V}(x_o,y_o)\|\\ \text{otherwise, } -\|\Delta\mathbf{V}(x_o,y_o)\|,\end{cases}$$

$$\Delta H'(x_o,y_o)=\begin{cases}\text{if } \|\mathbf{H}_+(x_o,y_o)\|\geqslant\|\mathbf{H}_-(x_o,y_o)\|\\ \text{then } \|\Delta\mathbf{H}(x_o,y_o)\|\\ \text{otherwise, } -\|\Delta\mathbf{H}(x_o,y_o)\|,\end{cases}$$

where $\|.\|$ denotes the *Euclidean* norm. In this formulation, the color contrast has no sign. To obtain the direction of maximal contrast, a convention is adopted to attribute signs to the quantities $\Delta V'(x_o,y_o)$ and $\Delta H'(x_o,y_o)$ in Equation (18). These quantities are considered positive if the luminance increases in the positive directions of the image coordinate system. The luminance quantities are estimated here by the norms: $\|\mathbf{H}_+\|$, $\|\mathbf{H}_-\|$, $\|\mathbf{V}_+\|$ and $\|\mathbf{V}_-\|$. Typically the luminance has been estimated by the *luminance* quantity, using the norm $\|\mathbf{c}\|_1=r+g+b$. However, the norm $\|\mathbf{c}\|_2=(r^2+g^2+b^2)^{1/2}$ also has been used to estimate luminance.[22] Here $\|.\|_2$ is denoted simply by $\|.\|$. Another possibility would be to consider the local color contrast with respect to a reference (e.g., the central portion of the operator $\mathbf{c}_o$), instead of the luminance quantity. However, this last possibility could present some ambiguities [e.g., in vertical ramp edges $\|\mathbf{H}_- - \mathbf{c}_o\|=\|\mathbf{H}_+ - \mathbf{c}_o\|$, then $\Delta H'(x_o,y_o)$ would have positive sign, irrespective of the actual sign of the ramp slope].[10]

Note the similarity between the color gradient formulated above and a Prewitt-type $(2w+1)\times(2w+1)$ monochromatic gradient.[21] The larger the parameter w, the smaller the operator sensitivity to noise, and also to sharp edges. This happens because there is a *smoothing* (low pass) *effect* associated with the convolution mask. Therefore, the larger the size of the convolution mask, the stronger the low pass effect and the less sensitivity the operator has to high spatial frequencies. Also, note that $\mathbf{H}_-$, $\mathbf{H}_+$, $\mathbf{V}_-$ and $\mathbf{V}_+$ are in fact convolution masks, and could easily implement the latest vector order statistics filtering approaches.

2.2.3 *Compound edge detectors*

The simple color gradient operator can also be used to implement compound gradient operators.[10] A well known example of a compound operator is the *derivative of Gaussian* (ΔG) operator.[21] In this case, each channel of the color image is initially convolved with a *Gaussian* smoothing function $G(x,y,\sigma)$, where σ is the *standard deviation*, and, after then this gradient operator is applied to the smoothed color image to detect edges. Torre and Poggio[23] stated that differential operations on sampled images require the image to first be smoothed by filtering. The Gaussian filtering has the advantage that it guarantees the bound-limitedness of the signal, so the derivative exists everywhere. This is equivalent to *regularizing* the signal using a low pass filter prior to the differentiation step.

The low pass filtering (*regularization*) step is done by the convolution of a *Gaussian* $G(x,y,\sigma)$ with the image signal. In a multispectral image, each pixel is associated to a vector $\mathbf{c}(x,y)$ whose components are denoted by $c_i(x,y)$, and $i=1,2,3$. This convolution is expressed as follows:

$$G(x,y,\sigma)\otimes\mathbf{I}=G(x,y,\sigma)\otimes I_i, \quad \forall i, \tag{19}$$

where $\mathbf{I}$ and I_i denote the image itself and the image component i. The image edges are then detected using the operator described before, and at each pixel the edge orientation $\theta(x,y)$ and magnitude $B(x,y)$ are obtained. The filtering operation introduces an arbitrary parameter, the *scale* of the filter, e.g., the standard deviation for the Gaussian filter. A number of authors have discussed the relationship existing between multi resolution analysis, Gaussian filtering and zero-crossings of filtered signals.[23,24]

The actual *edge locations* are detected by computing the zero-crossings of the second-order differences image, denoted by $\text{ZC}(x,y)$, obtained by applying first-order difference operators twice. Once the zero-crossings are found, they still must be tested for maximality of contrast. In practice, the image components are only known at the nodes of a rectangular grid of sampling points (x,y), and the zero-crossing condition $\text{ZC}(x,y)=0$ often does not apply. The simple use of *local minima* conditions leaves a margin for uncertainty. The zero-crossing image locations can be located by identifying how the sign of $\text{ZC}(x,y)$ varies in the direction of maximal contrast, near the zero-crossing location.[12] Therefore, the condition $\text{ZC}(x,y)=0$ must then be substituted by the more practical condition:

$$\text{ZC}(x_i,y_i)\cdot\text{ZC}(x_j,y_j)<0, \tag{20}$$

where the sampling points (x_i,y_i) and (x_j,y_j) are eight-adjacent, and the derivatives required for the computation of $\text{ZC}(x,y)$ are approximated by convolutions with the masks proposed by Beaudet.[25] Notice that (x_i, y_i) and (x_j, y_j) are in the direction of maximal contrast calculated at (x_o, y_o), the center of the eight-adjacent neighborhood. To improve the spatial location (mostly with larger operator sizes w), a *local minimum* condition is also used [i.e., $|\mathrm{ZC}(x_o, y_o)| < \tau$, $\tau \simeq 0$]. With the compound detector, the Gaussian noise can be reduced due to the Gaussian smoothing function. Though this operator improves performance in Gaussian noise, it is still sensitive to impulsive noise.

2.2.4 *Entropy operator*

The entropy operator is employed for both monochrome and color images. It yields a small value when the chromaticity in the local region is uniform, and a large value when there are drastic changes in the chromaticity. The entropy in a processing window (i.e., 3×3) centered on vector $\mathbf{v}_o = (r_o, g_o, b_o)$ is defined as:

$$H = q_R H_R + q_G H_G + q_B H_B \tag{21}$$

where H_R, H_G, and H_B denote the entropies in the R, G, and B directions, respectively, and

$$q_R = \frac{r_o}{r_o + g_o + b_o}, \quad q_G = \frac{g_o}{r_o + g_o + b_o}, \quad q_B = \frac{b_o}{r_o + g_o + b_o}. \tag{22}$$

Let $X_o, X_1, \ldots, X_N$, $(X = R, G, B)$ denote the values in each corresponding channel inside the processing window, and H_X is defined as:

$$H_X = -\frac{\Sigma_{i=1}^{N} p_{X_i} \log(p_{X_i})}{\log(N)}, \tag{23}$$

$$p_{X_i} = \frac{X_i}{\Sigma_{j=1}^{N} X_j}. \tag{24}$$

Edges can be extracted by detecting the change of entropy H in a window region. Since the presence of noise can disturb the local chromaticity in an image, the entropy operator is sensitive to noise.[14]

2.2.5 *Second derivative operators*

A more sophisticated approach which involves second derivative operator is suggested by Cumani. Given a vector field $\mathbf{f}(x,y)$ for a color image, the squared local contrast of $\mathbf{f}$ at point $P = (x,y)$ in the direction of the unit vector $\mathbf{n}(n_1, n_2)$ is defined as:

$$S(P, \mathbf{n}) = E n_1^2 + F n_1 n_2 + G n_2^2, \tag{25}$$

where

$$E = \frac{\partial \mathbf{f}}{\partial x} * \frac{\partial \mathbf{f}}{\partial x} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial x} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial x} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial x}, \tag{26}$$

$$F = \frac{\partial \mathbf{f}}{\partial x} * \frac{\partial \mathbf{f}}{\partial y} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}, \tag{27}$$

$$E = \frac{\partial \mathbf{f}}{\partial y} * \frac{\partial \mathbf{f}}{\partial y} = \frac{\partial R}{\partial y}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial y}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial y}\frac{\partial B}{\partial y}. \tag{28}$$

The eigenvalues of the 2×2 matrix $\left(\begin{smallmatrix} E & F \\ F & G \end{smallmatrix}\right)$ coincide with the extreme values of $S(P, \mathbf{n})$ and are attained when $\mathbf{n}$ is the corresponding eigenvector. The extreme values are

$$\lambda_{\pm} = \frac{E + G \pm [(E - G)^2 + 4F^2]^{1/2}}{2}, \tag{29}$$

and the two corresponding eigenvectors $\mathbf{n}_+$ and $\mathbf{n}_-$ are given as

$$\mathbf{n}_{\pm} = (\cos \theta_{\pm}, \sin \theta_{\pm}) \tag{30}$$

$$\theta_+ = \begin{cases} \dfrac{\pi}{4} & \text{if } (E - G) = 0 \text{ and } F > 0 \\ -\dfrac{\pi}{4} & \text{if } (E - G) = 0 \text{ and } F < 0 \\ \text{undefined} & \text{if } E = F = G = 0 \\ \dfrac{1}{2} \arctan \dfrac{2F}{E - G} + k\pi & \text{otherwise,} \end{cases} \tag{31}$$

$$\theta_- = \theta_+ \pm \frac{\pi}{2}. \tag{32}$$

Possible edge points are considered as point P where the first directional derivative $D_s(P, \mathbf{n})$ of maximal squared contrast $\lambda_+(P)$ is zero in the direction of maximal contrast $\mathbf{n}_+(P)$. The directional derivative is defined as:

$$D_s(P, \mathbf{n}) = \nabla \lambda_+ \cdot \mathbf{n}_+ \tag{33}$$

$$= \frac{\partial \lambda_+}{\partial x} n_1 + \frac{\partial \lambda_+}{\partial y} n_2 \tag{34}$$

$$= E_x n_1^3 + (2F_x + E_y) n_1^2 n_2 + (G_x + 2F_y) n_1 n_2^2 + G_y n_2^3. \tag{35}$$

The edge points are determined by computing zero-crossings of $D_s(P, \mathbf{n})$. Since the local directional contrast must be a maximum or minimum, the sign of D_s along a curve tangent at P in the direction of $\mathbf{n}_+$ is checked and the edge point is located if it is found to be a maximal point.

The ambiguity of the gradient direction in the above method causes some difficulties in locating edge points. Cumani suggested the subpixel technique with bilinear interpolation to solve the problem. Alshatti and Lambert[26] suggested a modification in solving the ambiguities by estimating the eigenvector $\mathbf{n}_+$, which can avoid the computational costly subpixel approximation. Other techniques[3] have also been proposed to reduce the complexity of the Cumani operator. Similar to the vector gradient operator, the second-order derivative operator is very sensitive to texture variations and impulsive noise, but it produces thinner edges. The regularizing filter applied in this operator causes a certain amount of blurriness in the edge map.

3 Vector Order Statistic Operators

3.1 *Introduction*

One important family of operators for image processing is based on order statistics.[27] It has played an important role in monochrome image processing and it was also extended to color image filtering and edge detection.[28] This approach is inspired by the morphological edge detectors[29,30] that have been proposed for the monochrome images. This class of color edge detectors is characterized by linear combinations of the sorted vector samples. Different sets of coefficients of the linear combination give rise to different edge detectors that vary in performance and efficiency. The primary step in order statistics is to arrange a set of random variables in ascending order according to certain criteria. In color space, since we are dealing with 2-D, multichannel variables, there is no universal way of defining an ordering. A number of ways have been proposed to perform multivariate data ordering[28] and they can be classified into marginal ordering (M-ordering), reduced aggregate ordering (R-ordering), partial ordering (P-ordering), and conditional ordering (C-ordering). In M-ordering, the ordered vectors do not correspond to the original vectors, and P-ordering is difficult to implement for digital image processing. C-ordering considers only one color component. R-ordering is hence more appropriate for color image processing. R-ordering reduces each multichannel variable to a scalar value according to a distance criterion. Let the image vectors in a window W denote $\mathbf{X}_i$, $i=1,2,\ldots,n$ and $D(\mathbf{X}_i,\mathbf{X}_j)$ be a measure of distance between vectors $\mathbf{X}_i$ and $\mathbf{X}_j$. The reduced scalar quantity associated with $\mathbf{X}_i$ is defined as

$$d_i=\sum_{k=1}^{n} D(\mathbf{X}_i,\mathbf{X}_j),\quad i=1,2,\ldots,n. \tag{36}$$

The arrangement of d_i in ascending order ($d^{(1)}\leqslant d^{(2)}\leqslant\ldots\leqslant d^{(n)}$) corresponds to the same ordering to the multivariate variables:

$$\mathbf{X}^{(1)}\leqslant\mathbf{X}^{(2)}\leqslant\ldots\leqslant\mathbf{X}^{(n)}. \tag{37}$$

In the ordered sequence, $\mathbf{X}^{(1)}$ is the vector median and vectors appearing at high ranks are referred to as outliers because they diverge the most from the data population.

3.2 *Edge Detectors*

The vector range (VR) edge detector is the simplest color edge detector based on order statistics. It expresses the deviation of the vector outlier in the highest rank from the vector median in W as follows:

$$\mathrm{VR}=D(\mathbf{X}^{(n)},\mathbf{X}^{(1)}), \tag{38}$$

where VR is small in a uniform area since all vectors are close together, and it gives large output when discontinuities exist. Edges can be obtained by thresholding the VR outputs.

The VR detector, though simple and efficient, is sensitive to noise, especially to impulsive noise. It will respond to a noise pixel at the center of W with n pixels. To improve noise performance, a more general class of operators, vector dispersion edge detector (VDED), is defined as a linear combination of the ordered vectors:

$$\mathrm{VDED}=\left\|\sum_{i=1}^{n}\alpha_i\mathbf{X}^{(i)}\right\|, \tag{39}$$

where $\|\cdot\|$ denotes the appropriate norm. Note that VR is a special case of VDED with $\alpha_1=-1$, $\alpha_n=1$, and $\alpha_i=0$, $i=2,\ldots,n-1$. The preceding equation can be further generalized by employing several sets of coefficients and combining the resulting vector magnitude in a suitable way. The coefficients can be chosen in a way to attenuate noise. One proposed class of operator is the minimum vector dispersion (MVD) detector, and it is defined as:

$$\mathrm{MVD}=\min_j\left\{D\left[\mathbf{X}^{(n-j+1)},\sum_{i=1}^{l}\frac{\mathbf{X}^{(i)}}{l}\right]\right\},\qquad j=1,2,\ldots,k,k,l<n. \tag{40}$$

The choice of k and l depend on n, the size of W. These two parameters control the trade-off between complexity and noise attenuation. This more computationally involved operator can improve edge detection performance in the presence of both impulsive and Gaussian noise. It can eliminate[14] up to $k-1$ impulse noise pixels in W. Let there be $k-1$ impulsive noise pixels in a window of n pixels. By their nature, impulsive noise differ from the rest of the pixels by a large amount. Therefore, after ordering, the impulsive noise pixels have the highest ranks, $\mathbf{X}^{(n-k+2)},\mathbf{X}^{(n-k+3)},\ldots,\mathbf{X}^{(n)}$. Since the distance between these noise pixels and the rest of the pixels are large, Equation (40) can be reduced to $\mathrm{MVD}=D[\mathbf{X}^{(n-k+1)},\Sigma_{i=1}^{l}\mathbf{X}^{(i)}/l]$. Notice that none of the noise pixels appears at this equation, and thus would not affect the edge detection process. The MVD is also robust in Gaussian noise due to the l-points average term.

An alternative design of the generalized VDED operators utilizes the adaptive nearest-neighbor (NN) filter.[31] The coefficients are chosen to adapt to local image characteristics. Instead of constants, the coefficients are determined by an adaptive weight function for each window W. The operator is defined as the distance between the outlier and the weighted sum of all the ranked vectors:

$$\mathrm{NNVR}=D\left[\mathbf{X}^{(n)},\sum_{i=1}^{n} w_i\mathbf{X}^{(i)}\right]. \tag{41}$$

The weight function w_i is determined adaptively using transformations of a distance criterion at each image location and it is not uniquely defined. There are two constraints on the weight function:

1. Each weight coefficient is positive, $w_i \geqslant 0$.
2. The weight function is normalized, $\Sigma_{i=1}^{n} w_i = 1$.

Since the operator should also attenuate noise, it is important to assign a small weight to the pixels with high ranks (i.e., outliers). A possible weight function can be defined as follows:

$$w_i=\frac{d^{(n)}-d^{(i)}}{n\cdot d^{(n)}-\Sigma_{j=1}^{n} d^{(j)}}. \tag{42}$$

One special case for this weight function occurs in highly uniform areas where all pixels have the same distance. The preceding weight function can not be used since the denominator is zero. Since no edge exists in this area, the difference measure NNVR is set to zero.

The MVD operator can also be incorporated with the NNVR operator to further improve its performance in the presence of impulse noise as follows:

$$\mathrm{NNMVD}=\min_j\left\{D\left[\mathbf{X}^{(n-j+1)}-\sum_{i=1}^{n} w_i\mathbf{X}^{(i)}\right]\right\},$$
$$j=1,2,\ldots,k,k<n. \tag{43}$$

A final annotation on the class of vector order statistic operators concerns the distance measure $D(\mathbf{X}_i,\mathbf{X}_j)$. By convention, the Euclidean distance measure (L_2 norm) is adopted. The use of L_1 norm is also considered because it reduces the computational complexity by computing the absolute values instead of squares and square root, and it shows no notable deviation in performance. A few other distance measures are also considered in the attempt to locate an optimal measure, namely, the Canberra metric implementation, the Czekanowski coefficient, and the angular distance measure. Their performances are addressed later. The Canberra metric implementation[32] is defined as:

$$D(\mathbf{X}_i,\mathbf{X}_j)=\sum_{k=1}^{m}\frac{|\mathbf{X}_{i,k}-\mathbf{X}_{j,k}|}{\mathbf{X}_{i,k}+\mathbf{X}_{j,k}}. \tag{44}$$

The Czekanowski coefficient[32] is defined as:

$$D(\mathbf{X}_i,\mathbf{X}_j)=1-\frac{2\Sigma_{k=1}^{m}\min(\mathbf{X}_{i,k}+\mathbf{X}_{j,k})}{\Sigma_{k=1}^{m}(\mathbf{X}_{i,k}+\mathbf{X}_{j,k})}, \tag{45}$$

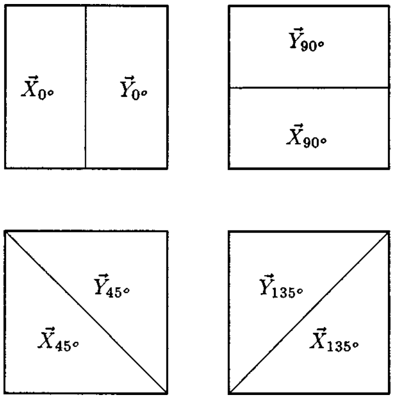

Fig. 1 Subwindow configurations.

where the m is the number of vector components, in the case of color image, $m=3$, corresponding to the three channels (R, G, and B). The angular distance measure[13] is defined as:

$$D(\mathbf{X}_i,\mathbf{X}_j)=\arccos\left(\frac{\mathbf{X}_i\cdot\mathbf{X}_j}{\|\mathbf{X}_i\|\|\mathbf{X}_j\|}\right), \tag{46}$$

where $\|\cdot\|$ denotes the magnitude of the color vector $\mathbf{x}_i$. Based on these three distance measures, a variety of color edge detectors can be established.

4 Difference Vector Operators

The class of difference vector (DV) operators can be viewed as first derivative like operators. This group of operators is extremely effective from the point of view of the computational aspects of the human visual system. In this approach, each pixel represents a vector in the RGB color space, and a gradient is obtained in each of the four possible directions (0, 45, 90, and 135 deg) by applying convolution kernels to the pixel window. Then a threshold can be applied to the maximum gradient vector to locate edges. The gradients are defined as:

$$|\nabla f|_{0\ \mathrm{deg}}=\|\mathbf{Y}_{0\ \mathrm{deg}}-\mathbf{X}_{0\ \mathrm{deg}}\|, \tag{47}$$

$$|\nabla f|_{90\ \mathrm{deg}}=\|\mathbf{Y}_{90\ \mathrm{deg}}-\mathbf{X}_{90\ \mathrm{deg}}\|, \tag{48}$$

$$|\nabla f|_{45\ \mathrm{deg}}=\|\mathbf{Y}_{45\ \mathrm{deg}}-\mathbf{X}_{45\ \mathrm{deg}}\|, \tag{49}$$

$$|\nabla f|_{135\ \mathrm{deg}}=\|\mathbf{Y}_{135\ \mathrm{deg}}-\mathbf{X}_{135\ \mathrm{deg}}\|, \tag{50}$$

$$\mathrm{DV}=\max(|\nabla f|_{0\ \mathrm{deg}},|\nabla f|_{90\ \mathrm{deg}},|\nabla f|_{45\ \mathrm{deg}},|\nabla f|_{135\ \mathrm{deg}}), \tag{51}$$

where $\|\cdot\|$ denotes the L_2 norm, and $\mathbf{X}$ and $\mathbf{Y}$ are three dimensional vectors used as convolution kernels. The variation in the definitions of these convolution kernels give rise to a number of operators. Figure 1 shows the partition of the pixel window into two subwindows within which each convolution kernel is calculated in all four directions.

The basic operator of this group employs a 3×3 window involving a center pixel and eight neighboring pixels. Let each pixel denotes $\mathbf{v}(x,y)$, and the convolution kernels for the center pixel $\mathbf{v}(x_0,y_0)$ in all four directions are defined as:

$$\mathbf{X}_{0\ \mathrm{deg}}=\mathbf{v}(x_{-1},y_0),\quad \mathbf{Y}_{0\ \mathrm{deg}}=\mathbf{v}(x_1,y_0),\tag{52}$$

$$\mathbf{X}_{45\ \mathrm{deg}}=\mathbf{v}(x_{-1},y_1),\quad \mathbf{Y}_{45\ \mathrm{deg}}=\mathbf{v}(x_1,y_{-1}),\tag{53}$$

$$\mathbf{X}_{90\ \mathrm{deg}}=\mathbf{v}(x_0,y_{-1}),\quad \mathbf{Y}_{90\ \mathrm{deg}}=\mathbf{v}(x_0,y_1),\tag{54}$$

$$\mathbf{X}_{135\ \mathrm{deg}}=\mathbf{v}(x_1,y_1),\quad \mathbf{Y}_{135\ \mathrm{deg}}=\mathbf{v}(x_{-1},y_{-1}).\tag{55}$$

This operator requires the least amount of computation among the edge detectors considered so far. However, as with the VR operator, the DV operator is also sensitive to impulsive and Gaussian noise.[15] As a result, more complex operators with subfiltering are designed. A larger window size is required in this case to allow more data for processing. Although there is no upper limit on the size of the window, usually a 5×5 window is preferred since the computational complexity is directly linked to the size of the window. In addition, when the window becomes too large it can no longer represent the characteristics of the local region. For an $n\times n$ window ($n=2k+1$, $k=2,3,\ldots$), the number of pixels in each of the subwindows illustrated in Figure 1 is $N=(n^2-1)/2$. A filter function can be applied to these N pixels in each subwindow to obtain the respective convolution kernels:

$$\mathbf{X}_{d\ \mathrm{deg}}=f(\mathbf{v}_{d\ \mathrm{deg},1}^{\mathrm{sub}_1},\mathbf{v}_{d\ \mathrm{deg},2}^{\mathrm{sub}_1},\ldots,\mathbf{v}_{d\ \mathrm{deg},N}^{\mathrm{sub}_1}),\tag{56}$$

$$\mathbf{Y}_{d\ \mathrm{deg}}=f(\mathbf{v}_{d\ \mathrm{deg},1}^{\mathrm{sub}_2},\mathbf{v}_{d\ \mathrm{deg},2}^{\mathrm{sub}_2},\ldots,\mathbf{v}_{d\ \mathrm{deg},N}^{\mathrm{sub}_2}),\tag{57}$$

$$\text{where}\quad d=0,45,90,135.\tag{58}$$

Depending on the type of noise one wishes to attenuate, different filters can be utilized. Four types of nonlinear image filters based on order statistics are employed in our work.

The first type of color edge detector incorporates the vector median filter.[33] This filter outputs the vector median of the N vector samples in the subwindow using the concept of vector order statistics introduced earlier, where the N vector samples are arranged in ascending order using R-ordering, $\mathbf{v}^{(1)}\leqslant\mathbf{v}^{(2)}\leqslant\ldots\mathbf{v}^{(N)}$, and the vector with the lowest rank, $\mathbf{v}^{(1)}$, is the vector median. We can make this operator more efficient by locating the vector with the minimum reduced distance calculated in R-ordering instead of ordering all N samples since only the vector median is of importance here. The output of the filter is:

$$f_{\mathrm{VM}}(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_N)=\mathbf{v}^{(1)}.\tag{59}$$

The vector median filter is very effective for reducing impulse noise because it can reject up to $N-1$ impulse noise pixels in a subwindow. However, since only the median vectors are used for edge detection, some edges may be rejected as noise and not able to be detected.

The second type of filter is the vector mean filter. This filter reduces the effect of Gaussian noise by averaging all the vector samples:

$$f_{\overline{\mathrm{VM}}}(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_N)=\frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i\,.\tag{60}$$

Due to the simplicity of the averaging operation, the vector mean operator is much more efficient than the vector median operator. The vector mean operator may cause certain false edges since the pixels used for edge detection are no longer the original pixels.

The third type of filter, the α-trimmed mean filter, is a compromise between the preceding two filters. It is defined as:

$$f_{\alpha-\mathrm{trim}}(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_N)=\frac{1}{N(1-2\alpha)}\sum_{i=1}^{N(1-2\alpha)}\mathbf{v}^{(i)},\tag{61}$$

where α is in the range $[0,0.5)$. When α is 0, no vector is rejected and the filter reduces to vector mean filter. When α is 0.5, all vectors except vector median are rejected and the filter reduces to vector median filter. For other α values, this operator can reject $200\alpha\%$ of impulse noise pixels, and it outputs the average of the remaining vector samples. Therefore the α-trimmed mean filter can improve noise performance in the presence of both Gaussian and impulse noise.

The last type of filter to be addressed is the adaptive nearest-neighbor filter.[34] As introduced in the last section, the output of this filter is a weighted vector sum with a weight function that varies adaptively for each subwindow:

$$f_{\mathrm{adap}}(\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_N)=\sum_{i=1}^{N}w_i\mathbf{v}_i\,,\tag{62}$$

where the weight function w_i was given in Eq. (42), and it assigns a higher weight to the vectors with lower ranks, and a lower weight to the outliers. This filter is also effective with mixed Gaussian and impulsive noise and it bears approximately the same complexity as the α-trimmed mean filter since they both need to perform the R-ordering. Again since edge detection is performed on the outputs of the filter instead of the original pixels, there may be a reduction in resulting edge qualities.

Another group of operators denotes a similar concept as the subfiltering operators where prefiltering is used instead. Any one of the preceding filters can be used to perform prefiltering on an image with a 3×3 window, and then the DV operator with the same window size is used for edge detection. Unlike the previous group, in this family the pixel window is not divided into subwindows during filtering, and the filter is applied only once to the whole window. The advantage with this group of operators is that it is considerably more efficient than the previous group since the filtering operation, which accounts for most of the complexity, is performed only once instead of eight times (two for each of the four directions) for each pixel.

One last proposed variation for the DV operators considers edge detection in only two directions, horizontal and vertical, instead of four directions:

$$\mathrm{DV\_}hv = \max(|\nabla f|_{0\ \mathrm{deg}}, |\nabla f|_{90\ \mathrm{deg}}). \qquad (63)$$

It is anticipated that such a design will be as powerful as the other DV operator due to the facts that (1) human vision is more sensitive to horizontal and vertical edges than to others and (2) the horizontal and vertical difference vectors are able to detect most of the diagonal edges as well, which in turn can reduce the thickness of these edges by eliminating the redundancy from the diagonal detectors. In addition, the amount of computation involved with this operator is slightly reduced.

5 Evaluation Procedures and Results

To investigate further the performance of the vector order statistic operators and the DV operators, we must determine how these two classes of operator compare to each other and how the individual edge detectors in each class rank among themselves. Both quantitative and qualitative measures are used to evaluate the performance of the edge detectors in terms of accuracy in edge detection and robustness to noise. The quantitative performance measures can be grouped into two types, probabilistic measures and distance measures. The first type is based on statistics of correct edge detection and false edge rejection. The second type is based on edge deviation from true edges. The first type of measure can be adopted to evaluate the accuracy of edge detection by measuring the percentage of correctly and falsely detected edges. Since a predefined edge map (ground truth) is required, synthetic images are used for this experiment. The second type of measure can be adopted to evaluate the noise performance by measuring the deviation of edges caused by noise from the true edges. Since numerical measures are not sufficient to model the complexity of human visual systems, qualitative evaluation using subjective tests is required in most image processing applications. Also, evaluation based on synthetic images has limited value because they can not be extrapolated to real images easily.[35] As a result, real images are also used in the evaluation process. All the images used for evaluation are defined in the RGB color space.

A total of 24 edge detectors from the class of the vector order statistic operators and the difference vector operators are implemented and their performance are evaluated along with the Sobel edge detector. Table 1 provides a list of the 25 edge detectors for later reference.

Table 1 Edge detectors for evaluation.

Edge Detector	Description
Sobel	Sobel edge detector extended from monochrome edge detection
	Vector Order Statistic Operators
VR0	VR operator (W:3×3) with L_1 norm
VR1	VR operator (W:3×3) with Canberra metric implementation
VR2	VR operator (W:3×3) with Czekanowski coefficient
VR3	VR operator (W:3×3) with angular distance measure
MVD_3	MVD operator (W:3×3) with $k=3$, $l=4$
MVD_5a	MVD operator (W:5×5) with $k=3$, $l=4$
MVD_5b	MVD operator (W:5×5) with $k=6$, $l=9$
NNVR_3	NNVR operator (W:3×3)
NNVR_5	NNVR operator (W:5×5)
NNMVD_3	NNMVD operator (W:3×3) with $k=3$
NNMVD_5a	NNMVD operator (W:5×5) with $k=3$
NNMVD_5b	NNMVD operator (W:5×5) with $k=6$
	Difference Vector Operators
DV	DV operator (W:3×3) in four directions
DV_hv	DV operator (W:3×3) in only horizontal and vertical directions
DVadap	DV operator (W:5×5) with adaptive subfilter
DVadap_hv	Same as DVadap except in only two directions
DVαtrim	DV operator (W:5×5) with αtrim subfilter
DVmean	DV operator (W:5×5) with vector mean subfilter
DVmedian	DV operator (W:5×5) with vector median subfilter
fDVadap	DV operator (W:3×3) with adaptive prefilter on entire window
fDVadap_hv	same as fDVadap except in only two directions
fDVαtrim	DV operator (W:3×3) with αtrim prefilter on entire window
fDVmean	DV operator (W:3×3) with vector mean prefilter on entire window
fDVmedian	DV operator (W:3×3) with vector median prefilter on entire window

5.1 *Probabilistic Evaluation*

Several artificial images with prespecified edges are created for accessing the probabilistic performance of selected edge detectors. To analyze the responses of the edge detectors to different types of edge, these images contain vertical, horizontal, and diagonal edges; round and sharp edges; edges caused by variation in only one, only two or all three components; isoluminant and nonisoluminant areas. In this experiment, noise is not added to the images. The resulting edge maps from each detector are compared with the predefined edge maps, and the number of correct and false edges detected are computed and are represented as hit and fault ratio, as shown in Table 2. The hit ratio is defined as the percentage of correctly detected edges and the fault ratio is the ratio between the number of false edges detected and the number of true edges in the predefined edge map. These two parameters are selected for this evaluation because they characterize the accuracy of an edge detector.

From the results in Table 2, a few conclusions can be drawn:

Table 2 Numerical evaluation with synthetic images.

Edge Detector	Hit (%)	Fault Ratio
Sobel	97.9	1.21
VR0	99.4	1.55
VR1	93.9	1.49
VR2	92.9	1.48
VR3	91.3	1.46
MVD_3	88.7	0.95
MVD_5a	99.2	3.33
MVD_5b	98.3	1.53
NNVR_3	99.4	1.55
NNVR_5	99.6	4.01
NNMVD_3	87.5	0.95
NNMVD_5a	94.4	3.3
NNMVD_5b	93.6	1.51
DV	99.4	1.55
DV_hv	99.1	1.14
DVadap	4.6	0.06
DVαtrim	60.5	0.65
fDVadap	98.4	2
fDVadap_hv	97.7	1.58
fDVαtrim	98.4	1.99

- Detectors such as the Sobel operator, the VR operator with L_1 norm, and the DV operator without any filtering all give good performance for images free of noise contamination.
- MVD with 3×3 window size has a lower hit ratio, but it also gives less false edges. The MVD operators with larger window size (i.e, 5×5) are able to provide high hit ratio.
- The NNVR operators also show good performance but the NNMVD operators give slightly lower hit ratio than those achieved by the MVD operators.
- The L_1 norm used in VR operators shows superior performance than other distance measures.
- For the DV operators, the detectors with only horizontal and vertical direction detection have almost the same hit ratio as the DV operator with all four directions, but they detect considerably less false edges.
- The DV operator with adaptive and α-trimmed subfiltering show very poor results. Note that this is not the case with real images, as we see later. The subfiltering seems to have undesirable effects on synthetic images. When prefiltering is performed (fDVadap, fDVα-trim), this undesirable effect does not exist, and these operators show good performances.

5.2 *Noise Performance*

Real images with corrupted mixed noise are used for this experiment. The mixed noise contain 4% impulsive noise and 30% Gaussian noise, with 5% variance for both types of noise. The edge maps of the images corrupted with noise are compared with the edge maps of the original image for each edge detector. The noise performance is measured in terms of the peak SNR (PSNR) values, and the results are given in Table 3. The PSNR is a easily quantifiable measure of image quality, although it provides only a rough evaluation of the actual visual quality the eye may perceive in an edge map.

Table 3 Noise performance.

Edge Detector	PSNR (dB)
Sobel	30.9
VR0	24.4
DV	29.4
MVD_3	26.3
MVD_5a	33.6
MVD_5b	35.4
NNVR_3	23.2
NNVR_5	28.6
NNMVD_3	25.9
NNMVD_5a	33.5
NNMVD_5b	35.2
DVadap	52.4
DVadap_hv	52.2
DVα-trimmed	45.5
fDVadap	62.6
fDVadap_hv	62.3
fDVα-trimmed	59.6

A few observations can be made from the results:

- The simple operators such as Sobel, VR and DV are sensitive to both impulsive and Gaussian noise. The noise performance can be improved with added complexity.
- In the case of vector order statistic operators, the MVD and NNMVD operators show more robustness in the presence of noise. We can also confirmed that the noise performance improves with the increase complexity of the operators, which are controlled by the two parameters k and l.
- For the class of DV operators, the added filtering improve the performance drastically. Since mixed noise is present, we used adaptive and α-trimmed filter for this experiment. The use of adaptive filter as prefilters on the whole window demonstrates the best performance in noise suppression. Hence we can conclude that the adaptive filter outperforms the α-trimmed filter and the prefiltering method is better than the subfiltering method in terms of noise suppression. Operators in only the horizontal and vertical directions show very slight deviation in PSNR values from ones in all four directions.

5.3 *Subjective Evaluation*

Since subjective evaluation is very important in image processing, we have applied the aforementioned operators to a collection of real and artificial images ranging from face features to outdoor scenes. The subjective evaluation enables us to investigate further the characteristics of the obtained edge maps through the involvement of human factors. In the subjective evaluation, images to be compared are viewed simultaneously, under identical viewing conditions by a set of observers. The operators are rated in terms of several criterion: ease in recognizing objects; continuity of edges; thinness of edges; performance in suppressing noise. The results obtained are in good agreement in all cases with our selected criterion.

After examining large quantities of edge maps produced by each edge detector, the following conclusions can be drawn:

- As suggested in the quantitative tests, the performance of Sobel, VR, and DV operators are very similar in that they all produce good edge maps for noiseless images.
- The MVD and NNMVD operators produce thinner edges and are less sensitive to small texture variations because of the averaging operation, which smooths out small variations. Also as expected, these two groups of operators are able to extract edges even in noise corrupted images.
- The two groups of DV operators with subfiltering and prefiltering all demonstrate excellent performance for noise corrupted images. The vector mean operator performs best in impulsive noise, vector median operator performs best in Gaussian noise, and adaptive and α-trimmed operators perform best in mixed noise. The subfiltering operator with adaptive filter is able to produce fair edge maps for real images despite its unsuccessful attempts with synthetic images during the numerical evaluation. However, the visual assessments are in agreement with the numerical tests in that the group of prefiltering operators outperform the group of subfiltering operators of the same filter.
- One last note on the difference vector operators is that the operators with only horizontal and vertical directions produce thinner diagonal edges than those in all four directions.

Figure 2 shows one of the images used in the experiment, ‘‘Lenna.’’ This image is corrupted by 4% of impulse noise and 30% of Gaussian noise. Figures 3–6 provide the edge maps produced by four selected operators. Figure 7 is another real image used, ‘‘Flower,’’ and four of the edge maps are shown in Figures 8–11. One of the synthetic images, ‘‘Ellipse,’’ is shown in Fig. 12 and Figs. 13–16 show a set of edge maps of ‘‘Ellipse.’’

6 Conclusion

Recently, many effective methods for color edge detection have been proposed and this paper presented a comparative study of some of the representative edge detectors. Two classes of operators, vector order statistic operators and VD operators were studied in detail because both of these are effective with multivariate data and are computationally efficient. Several variations were introduced to these two classes of operators to achieve better noise suppression and higher efficiencies. We discovered that both classes offer a

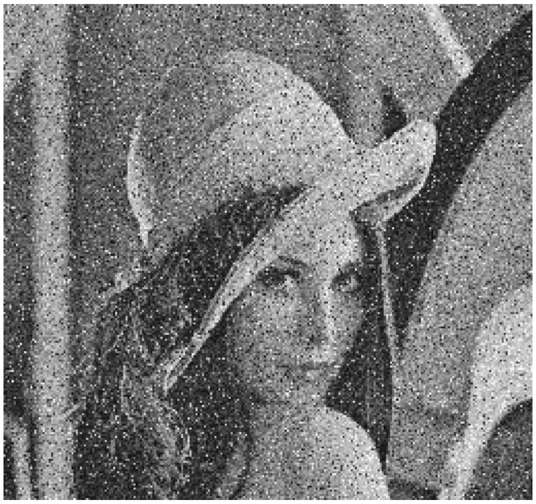

Fig. 2 Test image ‘‘Lenna.’’

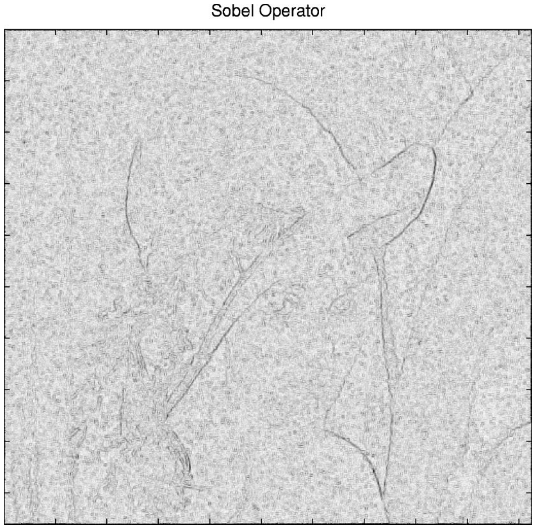

Fig. 3 Sobel detector: edge map of ‘‘Lenna.’’

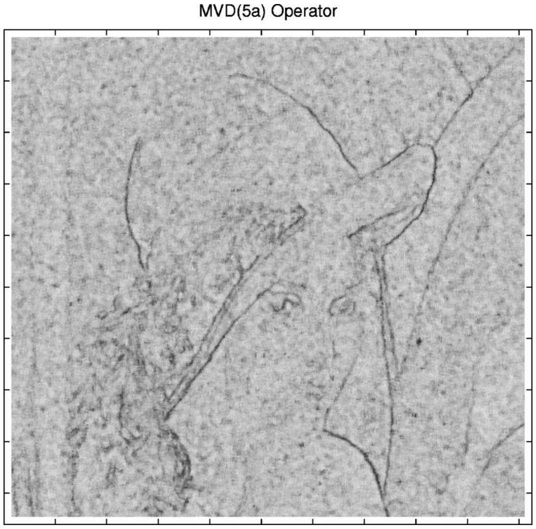

Fig. 4 MVD$_3$ detector: edge map of ‘‘Lenna.’’

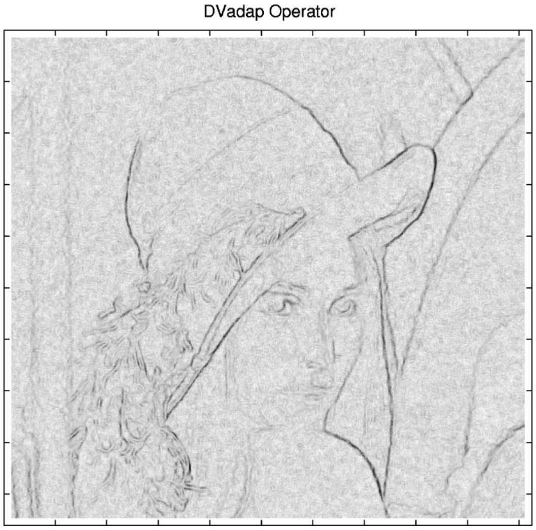

Fig. 5 DV detector: edge map of ''Lenna.''

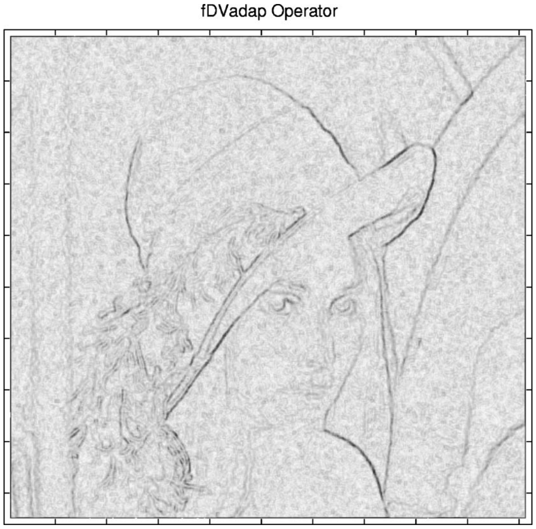

Fig. 6 Filtered-based DV (fDV) detector: edge map of ''Lenna.''

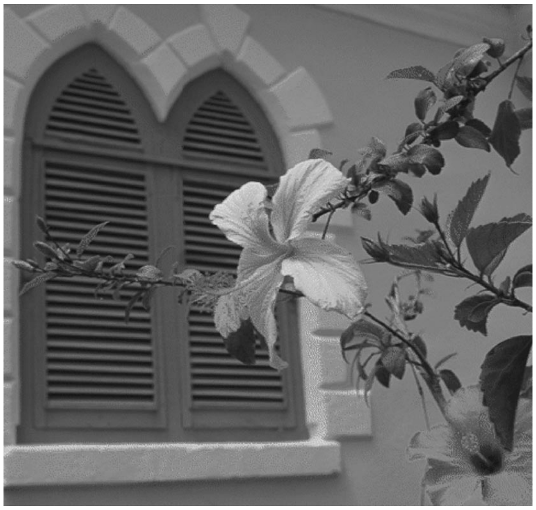

Fig. 7 Test image ''Flower.''

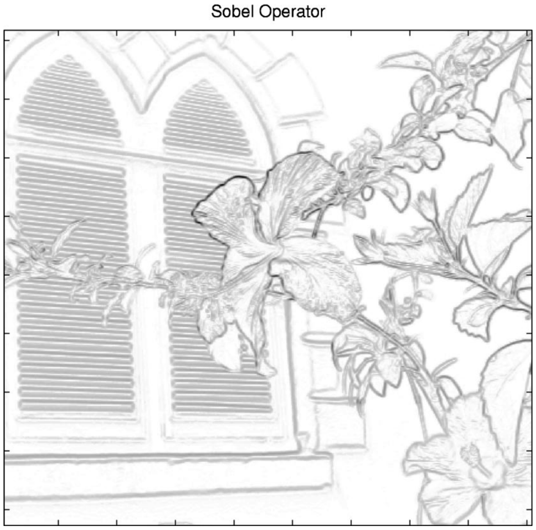

Fig. 8 Sobel detector: edge map of ''Flower.''

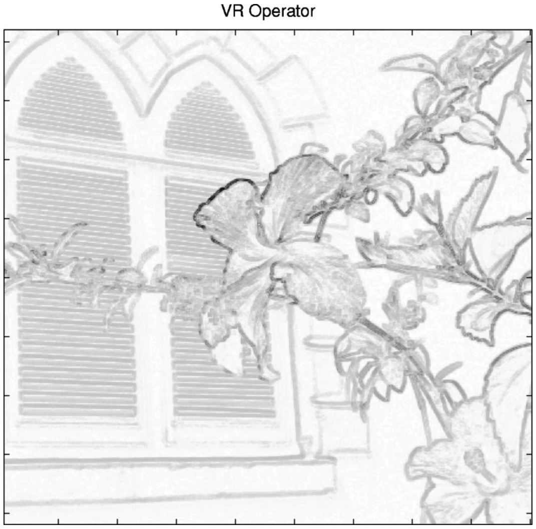

Fig. 9 VR detector: edge map of ''Flower.''

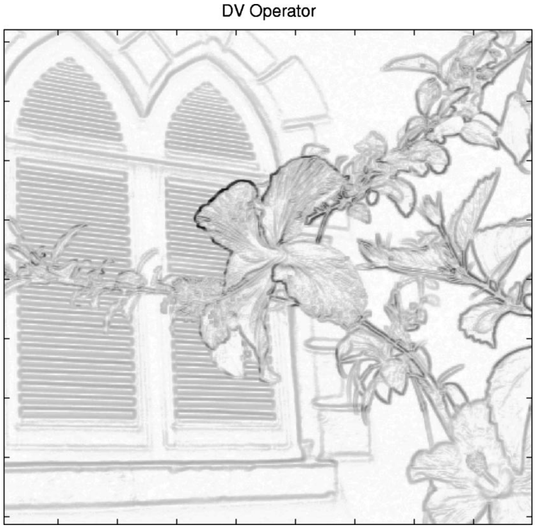

Fig. 10 Distance vector detector: edge map of ''Flower.''

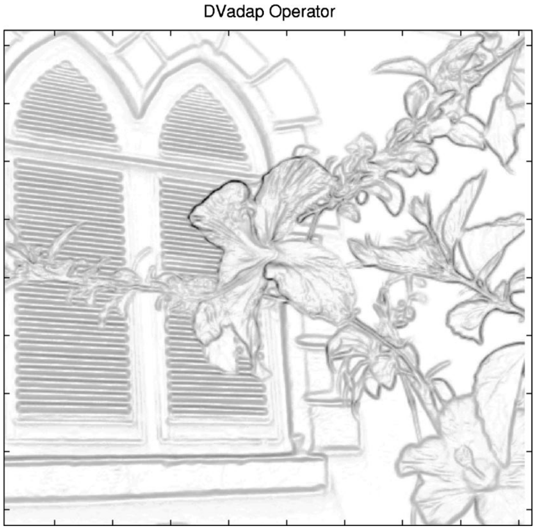

Fig. 11 The fDV detector: edge map of ''Flower.''

Fig. 12 Synthetic image ''Ellipse.''

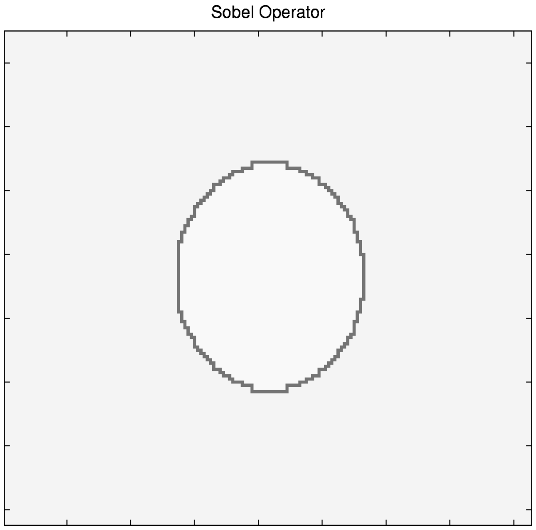

Fig. 13 Sobel detector: edge map of ''Ellipse.''

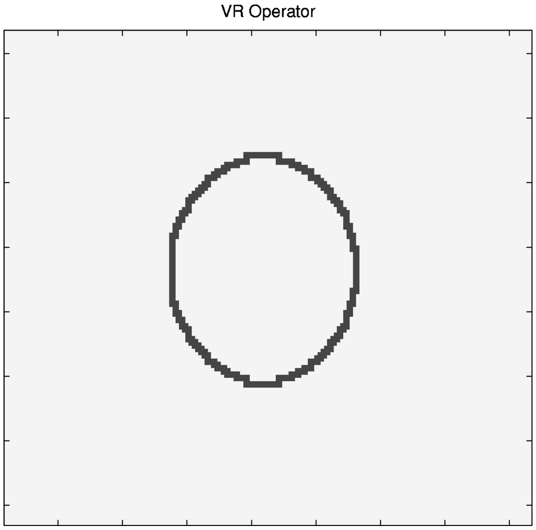

Fig. 14 VR detector: edge map of ''Ellipse.''

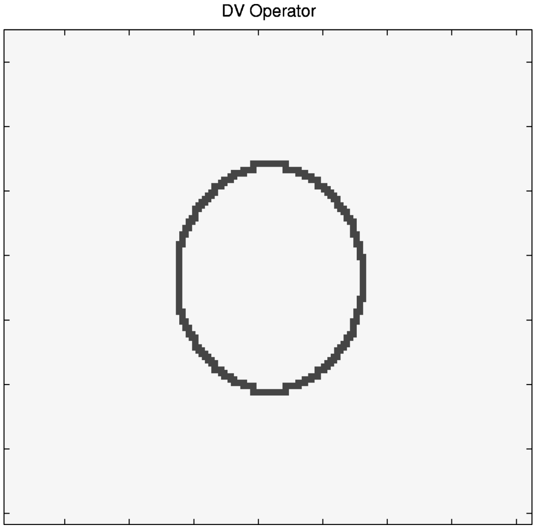

Fig. 15 VR detector: edge map of ''Ellipse.''

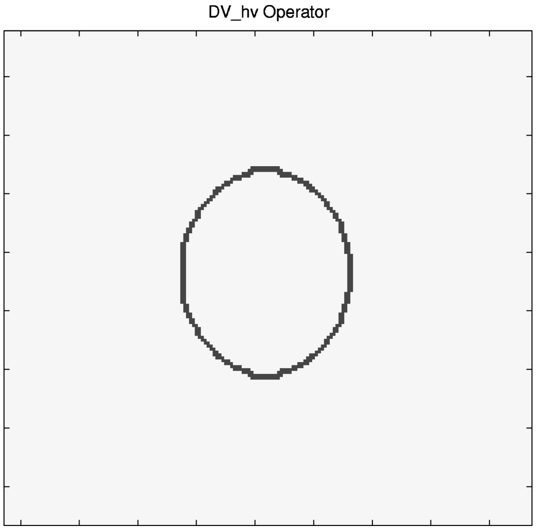

Fig. 16 DV-hv detector (horizontal/vertical processing only): edge map of ''Ellipse.''

mean of improving noise performance at the cost of increasing complexity. The performance of all edge detectors was evaluated both numerically and subjectively. The results presented demonstrate a superiority of the DV operator with adaptive prefiltering over other detectors. This operator scores high points in numerical tests and the edge maps it produces are perceived favorably by human eyes. Note that different applications have different requirements on the edge detectors, and though some of the general characteristics of various edge detectors were addressed, it is still better to select edge detectors that are optimum for the particular application.

References

1. A. Treisman and G. Gelade, ''A feature integration theory of attention,'' *Cogn. Psychol.* **12**, 97–136 (1980).
2. A. Treisman, ''Features and objects in visual processing,'' *Sci. Am.* **255**, 114B–125 (1986).
3. A. Koschan, ''A comparative study on color edge detection,'' in *Proc. 2nd Asian Conf. on Computer Vision-ACCV'95*, Vol. III, pp. 574–578 (1995).
4. L. Clinque, C. Guerra and S. Levialdi, ''Reply: on the paper by R. M. Haralick,'' *CVGIP: Image Understand.* **60**(2), 250–252 (1994).
5. M. Heath, S. Sarkar, T. Sanocki and K. Bowyer, ''Comparison of edge detectors: a methodology and initial study,'' in *1996 IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco (1996).
6. M. D. Heath, S. Sarkar, T. Sanocki and K. W. Bowyer, ''A robust visual method for assessing the relative performance of edge-detection algorithms,'' *IEEE Trans. Pattern. Anal. Mach. Intell.* **19**(12), 1338–1359 (1997).
7. L. E. Sobel, ''Camera models and machine perception,'' PhD Dissertation, Stanford University (1970).
8. D. Marr and E. Hildreth, ''Theory of edge detection,'' *Proc. R. Soc. London, Ser. B* **207**, 187–217 (1980).
9. S. D. Zenzo, ''A note on the gradient of a multiimage,'' *Comput. Vis. Graph. Image Process.* **36**, 1–9 (1986).
10. J. Scharcanski and A. N. Venetsanopoulos, ''Colour image edge detection using directional operators,'' in *Proc. IEEE Workshop on Nonlinear Signal and Image Processing*, Vol. II, pp. 511–515 (1995).
11. A. Shiozaki, ''Edge extraction using entropy operator,'' *Comput. Vis. Graph. Image Process.* **33**, 116–126 (1986).
12. A. Cumani, ''Edge detection in multispectral images,'' *CVGIP: Graph. Models Image Process.* **53**, 40–51 (1991).
13. P. E. Trahanias and A. N. Venetsanopoulos, ''Color edge detection using vector order statistics,'' *IEEE Trans. Image Process.* **2**(2), 259–264 (1993).
14. P. E. Trahanias and A. N. Venetsanopoulos, ''Vector order statistics operators as color edge detectors,'' *IEEE Trans. Syst. Man Cybern.* **26**(1), 135–143 (1996).
15. Y. Yang, ''Color edge detection and segmentation using vector analysis,'' MASc Thesis, University of Toronto (1994).
16. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed., Academic Press, New York (1982).
17. R. Nevatia, ''A color edge detector and its use in scene segmentation,'' *IEEE Trans. Syst. Man Cybern.* **7**(11), 820–825 (1977).
18. G. S. Robinson, ''Color edge detection,'' *Opt. Eng.* **16**(5), 479–484 (1987).
19. R. Machuca and K. Phillips, ''Applications of vector fields to image processing,'' *IEEE Trans. Pattern. Anal. Mach. Intell.* **PAMI-5**(3), 316–329 (1983).
20. J. Scharcanski, ''Color Texture representation and classification,'' PhD Thesis, University of Waterloo, Ontario, Canada (1993).
21. W. K. Pratt, *Digital Image Processing*, Wiley, New York (1991).
22. G. Healey, ''Segmenting images using normalized color,'' *IEEE Trans. Syst. Man Cybern.* **22**(1), 64–73 (1992).
23. T. Poggio, V. Torre and C. Koch, ''Computational vision and regularization theory,'' *Nature (London)* **317**, (1985).
24. A. Witkin, ''Scale-space filtering,'' in *Proc. 8th. Int. Joint Conf. on Artificial Intelligence*, Vol. 2, pp. 1019–1022 (1983).
25. P. Beaudet, ''Rotationally invariant image operators,'' in *Proc. Int. Joint Conf. on Pattern Recognition*, pp. 579–583 (1987).
26. W. Alshatti and P. Lambert, ''Using eigenvectors of a vector field for deriving a second directional derivative operator for color images,'' in *Proc. 5th Int. Conf. CAIP'93*, pp. 149–156 (1993).
27. H. A. David, *Order Statistics*, Wiley, New York (1980).
28. V. Barnett, ''The ordering of multivariate data,'' *J. R. Stat. Soc. A* **139**(3), 318–343 (1976).
29. R. J. Feechs and G. R. Arce, ''Multidimensional morphologic edge detection,'' in *Conf. Visual Communication and Image Processing, Proc. SPIE* **845**, 285–292 (1987).
30. J. S. J. Lee, R. M. Haralick and L. G. Shapiro, ''Morphologic edge detection,'' *IEEE J. Robot Automat.* **RA-3**(2), 142–156 (1987).
31. I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic, (1990).
32. K. Krzanowski, *Multivariate Analysis I: Distributions, Ordination and Inference*, Halsted Press, New York (1994).
33. J. Astola, P. Haavisto and Y. Neuvo, ''Vector median filters,'' *Proc. IEEE* **78**(4), 678–689 (1990).
34. K. N. Plataniotis, D. Androutsos, A. N. Venetsanopoulos, ''Color image filters: the vector directional approach,'' *Opt. Eng.* **36**(9), 2375–2383 (1997).
35. Y. T. Zhou, V. Venkateshwar and R. Chellappa, ''Edge detection and linear feature extraction using a 2D random field model,'' *IEEE Trans. Pattern. Anal. Mach. Intell.* **11**(1), 84–95 (1989).
36. S. Grossberg, *Neural Networks and Natural Intelligence*, MIT Press (1988).

Shu-Yu Zhu received the BS degree in electrical engineering from the University of Toronto in 1998, and she is currently working towards her MASc in the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada. From 1996 to 1997, she was with the Information Laboratory of NEC, Japan, and she participated in the Low-Power Single-Chip MPEG-2 encoder project. Her current research interests are in the area of perceptually lossless image and video compression techniques and their applications to the entertainment industry.

Konstantinos N. Plataniotis received the BEng degree in computer engineering and information science from the University of Patras, Patras, Greece, in 1988, and the MS and PhD degrees in electrical engineering from the Florida Institute of Technology (Florida Tech), Melbourne, Florida, in 1992 and 1994, respectively. He was with the Computer Technology Institute (CTI), Patras, Greece, from 1989 to 1991. He was a postdoctoral fellow at the Digital Signal & Image Processing Laboratory, Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada, from November 1994 to May 1997. Since September 1997 he has been with the School of Computer Science, Ryerson Polytechnic University, Toronto, Canada. He is also an adjunct professor in the Department of Electrical and Computer Engineering at the University of Toronto. His research interests include multimedia processing, multimedia databases, fuzzy and adaptive systems, and stochastic signal processing. Dr. Plataniotis is a member of the Technical Chamber of Greece.

Anastasios N. Venetsanopoulos received the BEng degree from the National Technical University of Athens (NTU), Greece, in 1965, and the MS, MPhil, and PhD degrees in electrical engineering from Yale University in 1966, 1968 and 1969, respectively. He joined the University of Toronto in September 1968 as a lecturer and has been a professor in the Department of Electrical and Computer Engineering since 1981. He has served as chairman of the Communications Group, associate chairman of the Department of Electrical Engineering, and associate chair of Graduate Studies for the Department of Electrical and Computer Engineering. He was on research leave at the Federal University of Rio de Janeiro, the Imperial College of Science and Technology, the National Technical University of Athens, Swiss Federal Institute of Technology, and the University of Florence, and was adjunct professor at Concordia University. He has served as lecturer in 130 short courses to industry and continuing education programs and as consultant to several organizations; he is a contributor to 24 books, co-author of two books, and has published more than 600 papers on digital signal and image processing and digital communications. He has served as chairman on numerous boards, councils and technical conference committees. He has been a guest editor or associate editor for several journals and the editor of the *Canadian Electrical Engineering Journal* from 1981 to 1983. He is a fellow of the IEEE and EIC and was awarded an honorary doctorate from the National Technical University of Athens. Present research interests include multimedia signal processing, multimedia databases, linear M-D and nonlinear filters, processing of multispectral (color) images and image sequences, and image and video coding.