Department of Automatic, Electronics and Computer Science
Silesian University of Technology, Gliwice, Poland

Color Image Filtering

by

Marek Kazimierz Szczepański

Contents

Chapter 1

Color representation and noise models

1.1 Color Representation

Color is a sensation created in response to excitation of our visual system by electromagnetic radiation known as light [14]. Color is how a human observer interprets light. Color cannot be defined in physical terms, since it is not a physical property, but once a behavioral definition is in place, it can be used to deduce the correlation between color sensation and the physical properties of the light. Color, in fact, is the perceptual result of light incident upon retina of the human eye.

The human retina, in general, has two types photo-receptor cells, so-called rods and cones. Rods are responsible for 'night vision' and they are active only at extremely low light levels, and do not contribute to color perception. Cones do not respond very well in dim lighting, their purpose is to handle vision during the daylight. Color perception is based on the cones activity. There are three different types of cones, which respond to radiation with somewhat different spectral response curves [14].

Since there are three types of human color photo-receptors, three numerical components are necessary and sufficient to describe a specific color. Therefore, a color can be specified by three component vector.

To utilize color as a visual cue in multimedia, image processing, graphics and computer vision applications, an appropriate method for representing the color signal is needed.

Through the years, different color models were proposed. Color spaces used in image processing are derived from visual system models (e.g. RGB, opponent color space, IHS etc.), technical domains (e.g. colorimetry: XYZ, television: YUV etc.) or

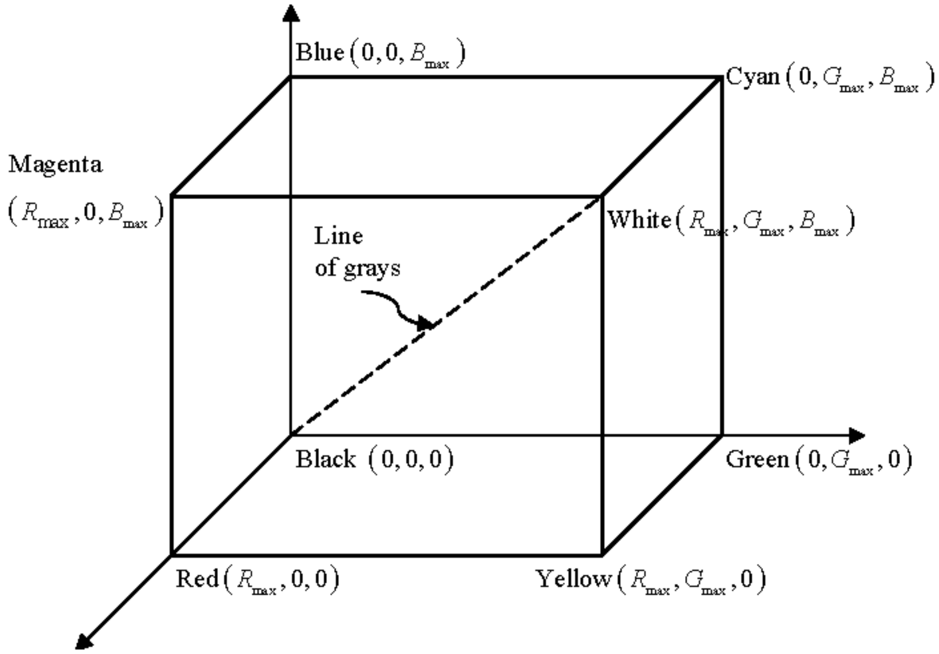

Blue $(0,0,B_{\max})$
Cyan $(0,G_{\max},B_{\max})$
Magenta $(R_{\max},0,B_{\max})$
White $(R_{\max},G_{\max},B_{\max})$
Line of grays
Black $(0,0,0)$
Green $(0,G_{\max},0)$
Red $(R_{\max},0,0)$
Yellow $(R_{\max},G_{\max},0)$

Figure 1.1: Representation of color in RGB color space.

developed especially for image processing (e.g. Ohta space, Kodak Photo YCC)

1.1.1 RGB Color Space

The RGB color space is the most frequently used one in image processing applications. Since color cameras, scanners and displays are most often provided with direct RGB signal input or output, this color space is the basic one, which is, if necessary, transformed into other color spaces. The color gamut [1] in the RGB color space forms a cube (Fig. 1.1). All grey (achromatic) colors are placed on the main diagonal of this cube from black ($R = G = B = 0$) to white ($R = G = B = 255$). The main disadvantages of RGB color space in applications involving natural images are:

- RGB components are highly correlated (correlation between $B - R$ is about 0.78, 0.98 for $R - G$ and 0.94 for $G - B$),
- it is not psychologically intuitive, it is hard to imagine a color based on R,G,B components,
- it is not uniform, i.e. it is impossible to evaluate the perceived differences between colors on the basis of distance in RGB space.

[1] gamut - set of all reproducible colors using color primaries

The RGB components for a given image are proportional to the amount of light incident on the scene represented by image. In order to eliminate the influence of illumination intensity, so-called *chromaticity coordinates* are used [36]:

$$\begin{aligned} r &= \tfrac{R}{R+G+B} \\ g &= \tfrac{G}{R+G+B} \\ b &= \tfrac{B}{R+G+B} = 1 - r - g \end{aligned} \tag{1.1}$$

1.1.2 XYZ Color Space

The XYZ color space, which was accepted by the *Commission Internationalle de l'Eclairage* (CIE) in 1931, is based on CIE colorimetric system. The CIE system is the one of most important color specification. It provides a standard method for specifying a color stimulus under controlled viewing conditions. It originated in 1931 an was further supplemented in 1964. These are known as the CIE 1931 and CIE 1964 supplementary systems [6]. The XYZ color space was designed to yield non-negative tristimulus values for each color. The tristimulus values XYZ are related to CIE RGB tristimulus values by the following transformation [18]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.2}$$

1.1.3 Opponent Color Space

Opponent color space, also called physiologically motivated color space, has been inspired by physiology of human visual system [31]. Opponent process theory says that our sensation of color is organized along two axes [14]. The first axis encodes the blueness or yellowness of a color. Figure 1.2 end equation 1.3 show how RGB 'cone' signals are transformed to three channels - one hypothesized achromatic and two opponent color channels.

$$\begin{aligned} RG &= R - G \\ YeB &= 2B - R - G \\ WhBl &= R + G + B \end{aligned} \tag{1.3}$$

Sometimes modified versions of opponent color spaces are used. For example opponent color space proposed by Yamaba and Miyake (1993) has modified second equation:

$$YeB = 0.4\,(R + G) - B \tag{1.4}$$

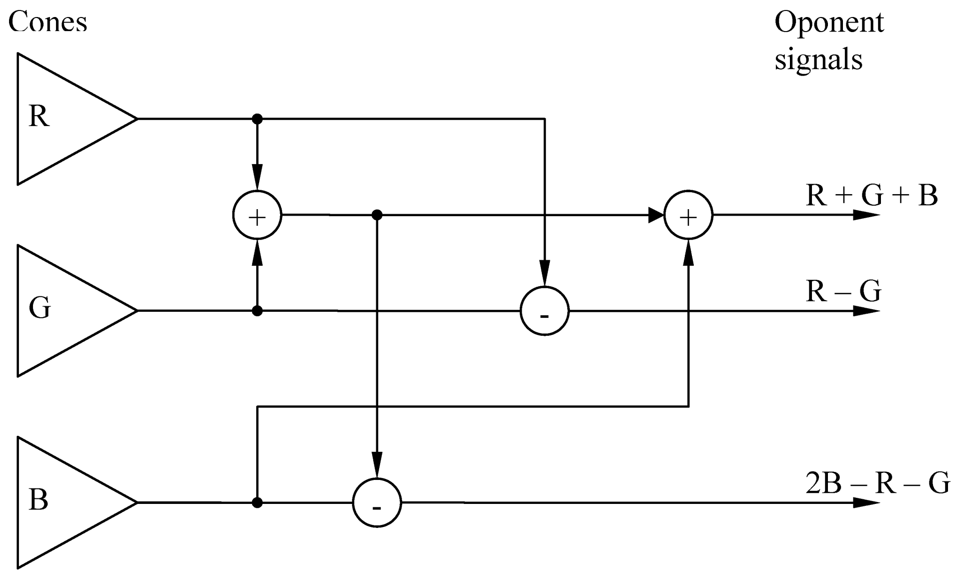


Figure 1.2: Schematic representation of opponent color stage of human visual system.

The human cones response is proportional to the stimulus intensity, thus logarithmic version of opponent color space was developed [7]:

$$\begin{aligned} RG &= \log R - \log G \\ YeB &= \log B - \tfrac{1}{2}\left(\log R + \log G\right) \\ WhBl &= \log G \end{aligned} \tag{1.5}$$

1.1.4 HSI Family of Perceptual Color Spaces

The human being cannot easily specify a desired color in the RGB model. On the other hand, perceptual features, such as intensity (brightness, lightness), hue and saturation correlate well withe human perception of color. Therefore, a color model in which these color attributes form the basis of the space is preferable from the users point of view. The hue H represents impression related to the dominant wavelength of the color stimulant. The saturation S corresponds to relative color purity (lack of white in the color) and in the case of pure colors $S = 100\%$, for gray levels the saturation is equal to zero. Maximum intensity is sensed as pure white, minimum intensity as pure black. The HSI color models family use approximately cylindrical coordinates. The Saturation S is proportional to radial distance, and the Hue H is function of the angle in the polar coordinate system. The intensity I is the distance along the axis perpendicular to the polar coordinate plane (Fig. 1.3).

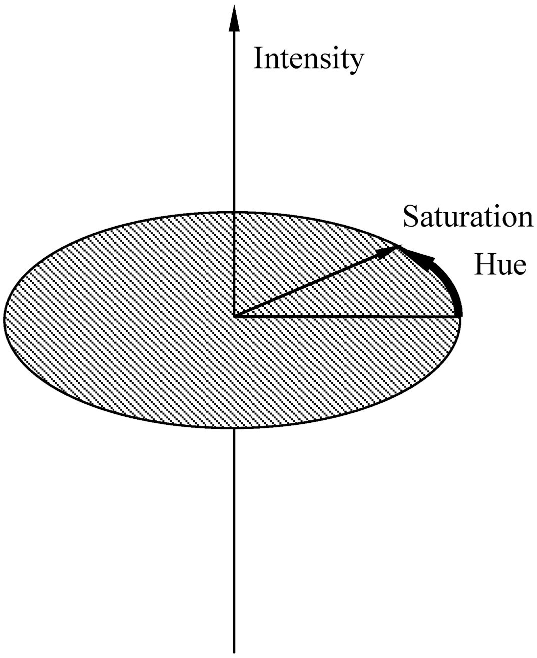


Figure 1.3: Cylindrical nature of HSI color space.

In the literature different transformation between the RGB and HSI color space are given. The original transformation, nowadays called ’basic’, was proposed by Tenembaum, Garvey and Wolf in [30]. Conversion from RGB to HSI color space is highly nonlinear and considerably complicated, for ’basic’ transformation [18, 30] we have:

$$\begin{aligned} &H_1 = \cos^{-1} \tfrac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}, && \\ &H = H_1 && \text{if } B \leq G \\ &H = 360^\circ - H_1 && \text{if } B > G \end{aligned} \tag{1.6}$$

$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B} \tag{1.7}$$

$$I = \frac{R+G+B}{3} \tag{1.8}$$

There are some simplified versions of this transformation, (e.g. Bajon’s transformation [4, 18]).

Two other HSI color spaces are very often applied in computer graphics, where they are called *color models*: HSV and HLS [24, 22, 18]. Figure 1.4 illustrates the geometric interpretation of these models.

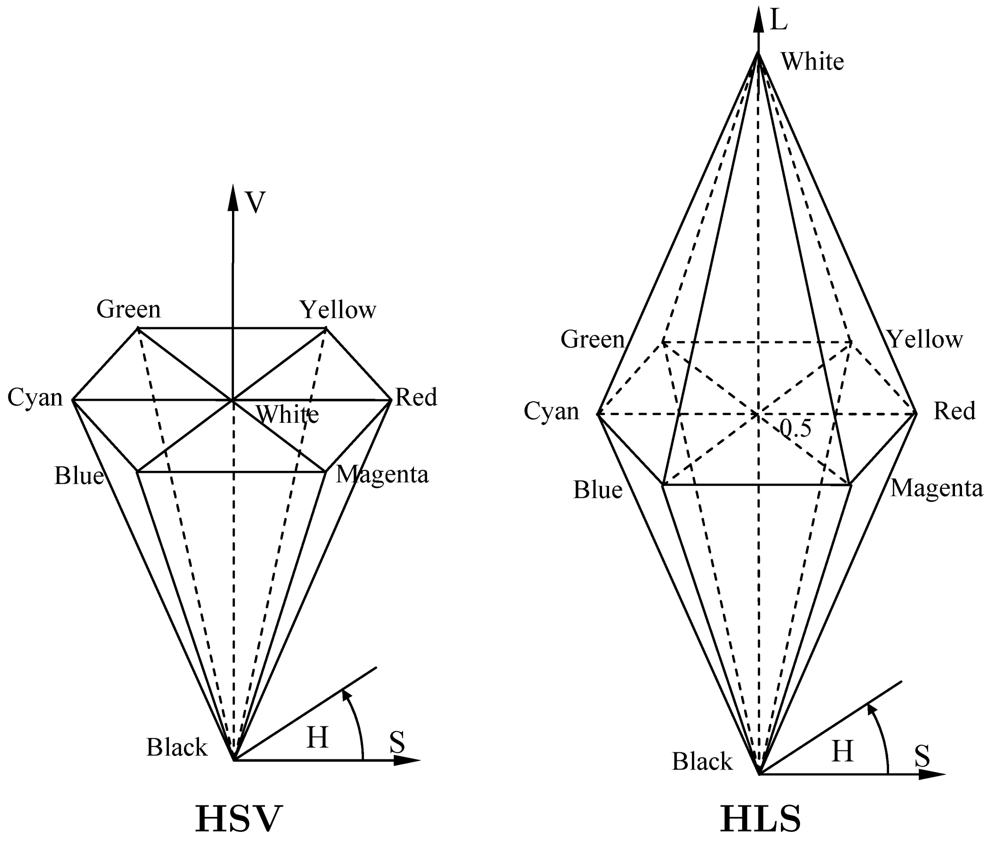


Figure 1.4: The HSV and HLS color models.

The HSV (Hue Saturation and Value) color model proposed originally by Smith in [24] is usually represented by the hexcone model (Fig. 1.4). The set of equation below can be used to transform a point in the RGB space to the appropriate value in the HSV coordinates.

$$\begin{aligned} &H_1 = \cos^{-1} \tfrac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}} && \\ &H = H_1 && \text{if } B \leq G \\ &H = 360^\circ - H_1 && \text{if } B > G \end{aligned} \tag{1.9}$$

$$S = \frac{\max{(R,G,B)} - \min{(R,G,B)}}{\max{(R+G+B)}} \tag{1.10}$$

$$V = \frac{\max{(R,G,B)}}{255} \tag{1.11}$$

The HLS (Hue Lightness and Saturation) color model is represented as double hexcone (Fig. 1.4). In the HLS model lightness L is defined as:

$$L = \frac{\max(R,G,B) + \min(R,G,B)}{2} \tag{1.12}$$

If the maximum and the minimum value coincide then $S = 0$ and the hue is undefined. Otherwise saturation is defined on the basis of lightness:

$$S = \begin{cases} \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)+\min(R,G,B)}, & \text{if } L \leq 0.5 \\ \frac{\max(R,G,B)-\min(R,G,B)}{2-\max(R,G,B)-\min(R,G,B)}, & \text{if } L > 0.5 \end{cases} \tag{1.13}$$

Hue in HLS color model is defined as follows:

$$H = \begin{cases} \frac{G-B}{\max(R,G,B)-\min(R,G,B)}, & \text{if } \max(R,G,B) = R \\ \frac{B-R}{\max(R,G,B)-\min(R,G,B)}, & \text{if } \max(R,G,B) = G \\ 4 + \frac{R-G}{\max(R,G,B)-\min(R,G,B)}, & \text{if } \max(R,G,B) = B \end{cases} \tag{1.14}$$

The HSI color spaces are compatible with human intuition, chromatic and achromatic values can be separated, moreover they give possibility to segment objects of the scene using only hue component. However this group of color spaces has severe drawbacks:

- the irreducible singularities of the RGB to HSI transformation (H for all achromatic colors and S for black),
- sensitivity to small deviation of RGB values near singularities,
- problems with some operation on angular values of hue (e.g. averaging)

1.1.5 Perceptually Uniform Color Spaces

Perceptual color spaces are derived from the idea to construct a color space in which perceived color differences recognized as equal by the human eye would correspond to equal Euclidean distances. As perceptually uniform color space two spaces recommended by CIE in 1976 are mainly used. These are so-called CIELAB space with $L^*a^*b^*$ values (for reflected light) and CIELUV with $L^*u^*v^*$ values (mainly for emitted light).

CIELAB Color Space

The CIELAB space is approximately uniform, it is defined by the following expression, where (X_0, Y_0, Z_0) represents reference white:

$$\begin{aligned} L^* &= 116 f\left(\tfrac{Y}{Y_0}\right) - 16 \\ a^* &= 500\left[f\left(\tfrac{X}{X_0}\right) - f\left(\tfrac{Y}{Y_0}\right)\right] \\ b^* &= 200\left[f\left(\tfrac{Y}{Y_0}\right) - f\left(\tfrac{Z}{Z_0}\right)\right] \end{aligned} \tag{1.15}$$

where

$$f(x) = \begin{cases} x^{\frac{1}{3}}, & x > 0.008856 \\ 7.787x + \frac{16}{116} & \text{otherwise,} \end{cases} \tag{1.16}$$

L^* stands for lightness value and is orthogonal to a^* and b^*. The following formula is used to determine color difference between sets of $L^*a^*b^*$ coordinates (L_1^*, a_1^*, b_1^*) and (L_2^*, a_2^*, b_2^*):

$$\Delta E_{\mathrm{ab}} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \tag{1.17}$$

where $\Delta L^* = L_2^* - L_1^*$, $\Delta a^* = a_2^* - a_1^*$ and $\Delta b^* = b_2^* - b_1^*$ The ΔL_{ab}^* color difference is widely used for the evaluation of color reproduction quality in image processing system.

CIELUV Color Space

The CIELUV color space is recommended by CIE for applications in additive light source conditions (e.g. displays). The definition of L^* is the same as for the CIELAB space given in 1.15 The u^* and v^* coordinates are defined as follows:

$$\begin{aligned} u^* &= 13L^*\left(u' - u_0'\right) \\ v^* &= 13L^*\left(v' - v_0'\right) \end{aligned}, \tag{1.18}$$

where:

$$\begin{aligned} u' &= \tfrac{4X}{X+15Y+3Z} \\ v' &= \tfrac{9Y}{X+15Y+3Z} \end{aligned}. \tag{1.19}$$

Values with superscripts 0 correspond to reference white. Similarly to CIELAB color space the perceptual color difference between two luminous sources is defined as follows:

$$\Delta E_{\mathrm{uv}} = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2} \tag{1.20}$$

1.2 Color Noise

Image noise reduction without structure degradation is perhaps the most important low-level image processing task [20, 22]. Noise introduces random variations into sensor readings, making them different from the ideal values, and thus introducing errors and undesirable side effects in subsequent stages of the image processing process. Faulty sensors, optic imperfectness, electronics interference, or data transmission errors may introduce noise to digital images. In considering the signal-to-noise ratio over practical communication media, such as microwave or satellite links, there can be degradation in quality due to low received signal power. Image quality degradation can be also a result of processing techniques, such as aperture correction, which amplifies both high frequency signals and noise.

The noise encountered in digital image processing applications cannot always be described in terms of commonly assumed Gaussian model. Very often, it can be characterized in terms of impulsive sequences which occur in the form of short duration, high energy spikes attaining large amplitudes with probability higher that predicted by Gaussian density model. Thus image filters should be robust to impulsive or generally heavy-tailed noise [20, 22]. In addition, when color images are processed, care must be taken to preserve image chromaticity, edges and fine image structures.

Based on trichromatic color theory, color pixels are encoded as three scalar values, namely, red green and blue. Although it is relatively easy to treat noise in three chromatic channels separately and apply existing gray-scale filtering techniques to reduce scalar noise magnitudes, however a different treatment of noise in the context of color images is needed. Color noise can be viewed as color fluctuations in the the certain color signal. As such the color the color noise signal should be considered as a 3-channel perturbation vector in color space, affecting the spread of the actual color vectors in the space.

1.2.1 Sensor Noise

Image sensors can be divided into two categories, photochemical and photoelectronic sensors. The positive and negative photographic films are typical photochemical sensors. In photochemical sensors noise is mainly due to silver grain in the active film surface. This kind of noise, often called film grain noise, can be modelled as a Poisson or Gaussian process [20]. In addition to the film grain noise, photographic noise is due to dust that collects on the optics and on the films during the developing process.

Photoelectronic sensors have the advantage over the film that they can be used to drive an image digitizer directly. Among the several photoelectronic sensors, such as vidicon tubes, Charge Injection Devices (CID), Charge Coupled Devices (CCD), CCD sensors are the most extensively used.

Sensors in CCD cameras consist of two-dimensional arrays of solid-state light sensing elements, the so-called cells. Photons reaching sensor surface induce electric charges in each cell. These charges are shifted to the right from cell to cell using two-phase clock and they come to read-out register. The rows of cells are scanned sequentially during a vertical scan. In CCD sensors two types of noise appear, namely

- thermal noise which is usually modelled as additive *white*, zero-mean Gaussian noise,
- photoelectronic noise, which is produced by the random fluctuation of the number of photons on the light sensitive surface of the sensor. Assuming low fluctuation level, it has a Bose-Einstein statistics and is modelled by Poisson-like statistics.

When photoelectronic noise level is low, the CCD sensor noise can be modelled using Gaussian distribution. In particular case of CCD cameras, transfer loss noise is also present. In CCD matrices charges are transferred from one cell to another. However, in practical case, this process is not perfect. Some fraction of charges is not transferred and it represents the transfer noise. The transfer noise occurs along the sensor rows of cells and thus it has strong horizontal correlation. This kind of noise usually appears as white smear located on one side of bright image spots. In the CCD cameras other types of noise due to capacitance coupling of clock lines or due to noisy cell recharging are also present. One of classical example of such kind of noise is a halo surrounding bright object in the image.

1.2.2 Transmission Noise (Impulsive Noise)

In many practical applications images are corrupted by noise caused either by faulty image sensors (thermal noise) or due to transmission corruption resulting from man-made phenomena such as ignition transients in the vicinity of the receivers or even natural phenomena such as lightning in the atmosphere and the ice cracking in the Arctic region. Transmission noise, also known as salt-and-pepper noise in grey-scale imaging, is modelled by an impulsive distribution. However, a problem in the study of the effect of the noise in the image processing community is the lack of commonly accepted multivariate impulsive noise model.

A number of simplified models have been introduced recently, to assist the performance evaluation of the different color image filters.

The impulsive noise model considered here is as follows [34, 22]:

$$\mathbf{F}_I = \begin{cases} (F_1, F_2, F_3)^T & \text{with probability } (1-p) \\ (d, F_2, F_3)^T & \text{with probability } p_1 \cdot p \\ (F_1, d, F_3)^T & \text{with probability } p_2 \cdot p \\ (F_1, F_2, d)^T & \text{with probability } p_3 \cdot p \\ (d, d, d)^T & \text{with probability } p_4 \cdot p \end{cases} \tag{1.21}$$

with $\mathbf{F}_I$ is the noisy signal, $\mathbf{F} = (F_1, F_2, F_3)^T$ is the noise-free color vector, d is the impulse value and

$$p_4 = 1 - p_1 - p_2 - p_3, \text{where } \sum_{i=1}^{3} p_i \leq 1 \tag{1.22}$$

Impulse d can have either positive or negative values. We further assume that $d \gg F_1, F_2, F_3$ and that the d values are situated at $(-255, +255)$. Thus, when an impulse is added or subtracted, forcing the pixel value outside the $[0, 255]$ range, clipping is applied to force the corrupted noise value into the integer range specified by the 8-bit arithmetic.

Another possibility to model impulsive noise is to use perceptual color spaces such as HSV. The noisy image pixel is generated according to the following rule:

$$\mathbf{F}_{I-HSV} = \begin{cases} \{H, S, V\} & \text{with probability } (1-p) \\ \{\delta, S, V\} & \text{with probability } p_1 \cdot p \\ \{H, \delta, V\} & \text{with probability } p_2 \cdot p \\ \{H, S, \delta\} & \text{with probability } p_3 \cdot p \\ \{\delta, \delta, \delta\} & \text{with probability } p_4 \cdot p \end{cases}, \tag{1.23}$$

p denotes the degree of impulsive noise distortion, $p_4 = 1 - (p_1 + p_2 + p_3)$, and δ is a positive or negative impulse. In this noise model δ is a random variable in a small range very close to the upper or lower bound of a HSV component, according to its original value[33].

1.2.3 Uniform Noise Models

There are a number of possibilities to define uniformly distributed types of noise:

$$\mathbf{F}_{u1} = \begin{cases} \{d_1, F_2, F_3\} & \text{with probability } p \\ \{F_1, d_2, F_3\} & \text{with probability } p \\ \{F_1, F_2, d_3\} & \text{with probability } p \end{cases}, \tag{1.24}$$

where d_1, d_2, d_3 are random integer values with uniform distribution over the interval $[0, \ldots, 255]$

$$\mathbf{F}_{u2} = \{d_1, d_2, d_3\} \quad \text{with probability } p, \tag{1.25}$$

where $d_1, d_2, d_3 \in [0, \ldots, 255]$ are uniformly distributed random integer values .

$$\mathbf{F}_{u3} = \begin{cases} \{d, F_2, F_3\} & \text{with probability } p \\ \{F_1, d, F_3\} & \text{with probability } p \\ \{F_1 F_2, d\} & \text{with probability } p \end{cases}, \tag{1.26}$$

where d i a random integer variable with uniform distribution over the interval $\in [0, 1, \ldots, 255]$

1.2.4 Mixed Noise

In many practical situations an image is often corrupted by both additive Gaussian noise due to faulty sensors (thermal-noise) and impulsive transmission noise introduced by environmental interference or faulty communication channels. An image can therefore be thought of as being corrupted by mixed noise according to the following model:

$$\mathbf{F}_M = \begin{cases} \mathbf{F} + \mathbf{F}_G & \text{with probability } (1 - p_I) \\ \mathbf{F}_I & \text{otherwise,} \end{cases} \tag{1.27}$$

where $\mathbf{F}$ is the noise free color signal with the additive noise $\mathbf{F}_G$ modelled as zero mean *white* Gaussian noise and $\mathbf{F}_I$ transmission noise modelled as multivariate impulsive noise with $\mathbf{p}_I = (p, p_1, p_2, p_3)$ the degree of impulsive noise contamination [22].

Chapter 2

Noise Removal Techniques

2.1 Linear Filters

Linear smoothing is a filtering process in which the value of the output pixel is a linear combination of the values of the pixels in the input pixel neighborhood. The simplest noise removal techniques are linear filters.

Linear filtering of an image is accomplished through an operation called *convolution.* In convolution, the value of the output pixel is computed as a weighted sum of neighboring pixels. The matrix of weights is called the convolution kernel [23, 29].

For example if we use convolution mask H of size $M \times N$ then the result of convolution of grayscale image F with mask H is:

$$\hat{F}(i,j) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} H(i,j) \cdot F(i-k, j-l) = H(i,j) * F(i,j) \tag{2.1}$$

In case of color images convolution filters are applied to each color component separately.

2.1.1 Moving Average Filter

The simplest linear filter is the moving average filter. For a 3×3 neighborhood the convolution mask H for moving average filter is presented in Fig. 2.1

The main advantage of this filter is strong noise suppression effect and low computational burden due to separability of its kernel. However this kind of filters blur edges and image details.

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1\\ \hline 1 & 1 & 1\\ \hline 1 & 1 & 1\\ \hline\end{array} = \frac{1}{3}\begin{array}{|c|c|c|}\hline 1 & 1 & 1\\ \hline\end{array} \cdot\frac{1}{3}\begin{array}{|c|}\hline 1\\ \hline 1\\ \hline 1\\ \hline\end{array}$$

Figure 2.1: Moving Average convolution mask H of size 3×3

$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1\\ \hline 2 & 4 & 2\\ \hline 1 & 2 & 1\\ \hline\end{array} = \frac{1}{4}\begin{array}{|c|c|c|}\hline 1 & 2 & 1\\ \hline\end{array} \cdot\frac{1}{4}\begin{array}{|c|}\hline 1\\ \hline 2\\ \hline 1\\ \hline\end{array}$$

Figure 2.2: Gaussian mask H of size 3×3

2.1.2 Gaussian Filter

The significance of the central pixel may be increased, as it approximates the properties of noise with a Gaussian probability distribution. Coefficients of the Gaussian kernel are calculated using formula 2.2.

$$^{n}H_{k,l} = \frac{(n!)^2}{2^{2n}\left(\frac{n}{2}-k\right)!\left(\frac{n}{2}+k\right)!\left(\frac{n}{2}-l\right)!\left(\frac{n}{2}+l\right)!}, \quad k,l = -\frac{n}{2},\dots\frac{n}{2} \qquad (2.2)$$

Mask obtained for $n = 2$ (3×3) is presented in Fig 2.2. The Gaussian filter kernel is separable, thus images can be convolved separately with one-dimensional filter masks. Figure 2.3 presents a 5×5 Gaussian kernel as a product of two vectors multiplication. Elements of such vectors can be calculated from the following formula $^{n}H\left(i\right) = \left(\begin{array}{c} 2n \\ i \end{array}\right)$

2.2 Nonlinear Filters

Several techniques have been proposed over the years. Among them are linear processing methods, whose mathematical simplicity and the existence of unifying theory

$$\frac{1}{256}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 6 & 24 & 36 & 24 & 6 \\ \hline 4 & 16 & 24 & 16 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array} = \frac{1}{16}\begin{array}{|c|c|c|c|c|}\hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array} \cdot \frac{1}{16}\begin{array}{|c|}\hline 1 \\ \hline 4 \\ \hline 6 \\ \hline 4 \\ \hline 1 \\ \hline \end{array}$$

Figure 2.3: Gaussian mask H of size 5 × 5

make their design and implementation easy.

However, not all filtering problems can be efficiently solved by using linear techniques. For example, conventional linear techniques cannot cope with nonlinearities of the image formation model and fail to preserve edges and image details.

To this end, nonlinear image processing techniques are introduced. Nonlinear techniques, theoretically, are able to suppress non-Gaussian noise and preserve important image elements, such as edges and details, and eliminate degradations occurring during image formation or transmission through nonlinear channels.

2.2.1 Linear Filter Modification

Averaging with Data Validity

Methods that average with limited data validity try to avoid blurring by averaging only those pixels which satisfy some criterion, the aim being to prevent involving pixels that are part of a separate feature. One of possible criterion is to use only pixels in the original image with brightness in a predefined interval $[\alpha, \beta]$. Considering the point (i, j) in the image, the convolution mask is calculated in the neighborhood $\mathcal{N}$, which is usually defined by processing window W, from the nonlinear formula:

$$H(k,l) = \begin{cases} 1, & \text{for } F(i+k, j+l) \in [\alpha, \beta] \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

Next method performs the averaging only if the computed brightness change of a pixel is in some predefined interval. This method permits repair to large-area errors resulting from slowly changing brightness of the background without affecting the rest of the image.

Another possibility to improve performance of averaging filters is to uses edge strength (i.e., magnitude of a gradient) as a validating criterion. The magnitude of some gradient operator is first computed for the entire image, and only pixels in the input image with a gradient magnitude smaller than a predefined threshold are used in averaging.

Averaging According to Inverse Gradient

Another possibility to enhance averaging filters is to calculate convolution mask at each pixel according to the inverse gradient. The idea is that brightness change within a region is usually smaller than between neighboring regions [11].

Let (k, l) be the central pixel of a convolution mask with odd size; the inverse gradient at the point (i, j) with respect to (k, l) is then

$$D\left(k,l,i,j\right)=\begin{cases}\frac{1}{\left|F(i,j)-F(k,l)\right|}, & \text{if } F\left(i,j\right)\neq F\left(k,l\right)\\ \gamma, & \text{if } F\left(i,j\right)=F\left(k,l\right)\end{cases} \tag{2.4}$$

In this definition inverse gradient is in the interval $(0, \gamma]$, and is smaller on the edge than in the interior of a homogeneous region. Weight coefficients in the convolution mask H are then normalized by the inverse gradient:

$$H\left(k,l,i,j\right)=\frac{D\left(k,l,i,j\right)}{\sum\limits_{(i',j')\in\mathcal{N}} D\left(k,l,i',j'\right)} \tag{2.5}$$

Presented method assumes sharp edges. Isolated noise points within homogeneous regions have small values of the inverse gradient. Points from the neighborhood take part in averaging and the noise is removed. When the convolution mask is close to an edge, pixels from the region have larger coefficients than pixels near the edge, and are not blurred.

When color signal is considered, assuming that RGB color space is used, image pixels are represented with vectors $\mathbf{F}(i,j) = \{F_1, F_2, F_3\}$ Gradient magnitude may be calculated using proper vector norm, and then the inverse gradient takes the form:

$$D\left(k,l,i,j\right)=\begin{cases}\frac{1}{\left\|\mathbf{F}(i,j)-\mathbf{F}(k,l)\right\|}, & \text{if } \mathbf{F}\left(i,j\right)\neq \mathbf{F}\left(k,l\right)\\ \gamma, & \text{if } \mathbf{F}\left(i,j\right)=\mathbf{F}\left(k,l\right)\end{cases}, \tag{2.6}$$

where $\|\cdot\|$ denotes proper vector norm.

2.2.2 Order-statistics Filters

One of the most popular families of nonlinear filters for noise removal are order-statistics filters. Their theoretical framework is based on robust statistics. These filters utilize algebraic ordering of a windowed set of data to compute the output signal.

The early approaches to color image processing usually comprise extensions of the scalar filters to color images. Ordering of scalar data, such as gray-scale images is well defined and it was extensively studied [20].

However, the concept of input ordering, initially applied in scalar quantities is not easily extended to multichannel data, since there is no universal way to define ordering in vector spaces. A number of different ways to order multivariate data has been proposed. These techniques are generally classified into:

- *marginal ordering* (M-ordering), where the multivariate samples are ordered along each of their dimension independently,
- *reduced* or *aggregate ordering* (R-ordering), where each multivariate observation is reduced to a scalar value according to a distance metric,
- *partial ordering* (P-ordering), where the input data are partitioned into smaller groups which are then ordered,
- *conditional ordering* (C-ordering), where multivariate samples are ordered conditional on one of its marginal sets of observations [5, 21].

R-ordering Filters

Let $\mathbf{F}(\mathbf{x})$ be a multichannel image and let W be a window of finite size k (filter length). The noisy image vectors inside the filtering window W are denoted as $\mathbf{F}_j$, $j = 0, 1, ..., k-1$. If the distance between two vectors $\mathbf{F}_i, \mathbf{F}_j$ is denoted as $\rho(\mathbf{F}_i, \mathbf{F}_j)$ then the scalar quantity

$$R_i = \sum_{j=0}^{k-1} \rho(\mathbf{F}_i, \mathbf{F}_j), \tag{2.7}$$

is the aggregated distance associated with the noisy vector $\mathbf{F}_i$ inside the processing window. Assuming a reduced ordering of the R_i's

$$R_{(0)} \leq R_{(1)} \leq \ldots \leq R_{(\tau)} \leq \ldots, \leq R_{(k-1)}, \tag{2.8}$$

implies the same ordering of the corresponding vectors $\mathbf{F}_i$

$$\mathbf{F}_{(0)}; \mathbf{F}_{(1)}; \ldots ; \mathbf{F}_{(\tau)}; \ldots ; \mathbf{F}_{(k-1)}. \tag{2.9}$$

Nonlinear ranked type multichannel filters define the vector $\mathbf{F}_{(0)}$ as the result of the filtering operation. 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 [19, 10].

Vector Median Filter (VMF)

The best known member of the family is the so called *Vector Median Filter* (VMF). The definition of the multichannel median is a direct extension of the ordinary scalar median definition with the L_1 or L_2 norm utilized to order vectors according to their relative magnitude differences [3]. The output of the VMF is the pixel $\mathbf{F}_{MED} \in W$ for which the following condition is satisfied:

$$\sum_{j=0}^{k-1} \rho(\mathbf{F}_{MED}, \mathbf{F}_j) \leq \sum_{j=0}^{k-1} \rho(\mathbf{F}_i, \mathbf{F}_j), \quad i = 0, \ldots, k-1\,. \tag{2.10}$$

In this way the VMF consists of computing and comparing the values of R_i (2.7) and the output is the vector $\mathbf{F}_l$ for which R_l minimizes the function R in (2.7). In other words if for some l the value

$$R_l = \sum_{j=0}^{k-1} \rho(\mathbf{F}_l, \mathbf{F}_j)\,, \tag{2.11}$$

is smaller than R_0:

$$R_0 = \sum_{j=0}^{k-1} \rho(\mathbf{F}_0, \mathbf{F}_j)\,, \tag{2.12}$$

and minimizes the function R, then the original pixel $\mathbf{F}_0$ in the filter window W is being replaced by $\mathbf{F}_l$ which satisfies the condition (2.10), which means that $l = \arg\min_i R_i$.

The construction of the VMF is shown in Fig. 2.4, where the Euclidean distance is used. However different norms can be applied for noise suppression using the VMF concept. Most commonly the Minkowski L_p metric

$$\rho(\mathbf{F}_i, \mathbf{F}_j) = \left\{ \sum_{k=1}^{L} (F_i^k - F_j^k)^p \right\}^{\frac{1}{p}}\,, \tag{2.13}$$

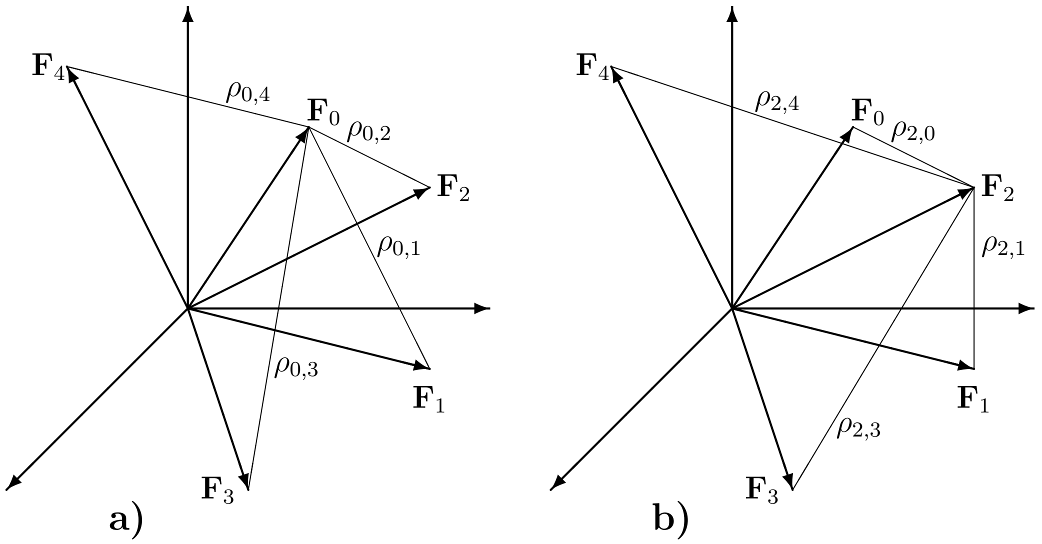


Figure 2.4: The distance R_0 associated with the vector $\mathbf{F}_0$ equals $R_0 = \rho(0,1) + \rho(0,2) + \rho(0,3) + \rho(0,4)$ **a**), and similarly the distance R_2 associated with the vector $\mathbf{F}_2$ equals $R_2 = \rho(2,0) + \rho(2,1) + \rho(2,3) + \rho(2,4)$ **b**).

is used for the construction of the *Vector Median Filter.* Choosing a specific value of p, we can obtain the following popular distance measures:

$$\rho^1(\mathbf{F}_i, \mathbf{F}_j) = \sum_{k=1}^{L} |F_i^k - F_j^k| \,, \quad p = 1 \,, \tag{2.14}$$

$$\rho^2(\mathbf{F}_i, \mathbf{F}_j) = \sqrt{\sum_{k=1}^{L} (F_i^k - F_j^k)^2} \,, \quad p = 2 \,, \tag{2.15}$$

$$\rho^\infty(\mathbf{F}_i, \mathbf{F}_j) = \max_k |F_i^k - F_j^k| \,, \quad p = \infty \,, \tag{2.16}$$

where L is the vector dimension.

Extended Vector Median Filter (EVMF)

The VMF concept may be combined with linear filtering for the case where the median is inadequate in filtering out noise (such as in the case of additive Gaussian noise). The filter based on this idea, so-called *Extended Vector Median Filter* (EVMF) has been presented in [3] and [9].

If the output of the *Arithmetic Mean Filter*, (AMF) is denoted as F_{AMF} then

$$\mathbf{F}_{EVMF} = \begin{cases} \mathbf{F}_{AMF} & \text{if} \quad \sum\limits_{j=0}^{n-1} ||\mathbf{F}_{AMF} - \mathbf{F}_j|| < \sum\limits_{j=0}^{n-1} ||\mathbf{F}_{VMF} - \mathbf{F}_j|| \\ \mathbf{F}_{VMF} & \text{otherwise} \end{cases}, \tag{2.17}$$

α-trimmed Vector Median Filter (αVMF)

Another filtering technique, so-called *α-trimmed Vector Median Filter* (αVMF) has been proposed. In this filter $k(1-2\alpha)$ samples closest to the vector median value are selected as inputs to averaging-type filter. The output of the α-trimmed VMF can be defined as follows [34, 21].:

$$\mathbf{F}_{\alpha VMF} = \sum_{i=0}^{k(1-2\alpha)-1} \frac{1}{k\,(1-2\alpha)} \mathbf{F}_i \tag{2.18}$$

with ordering defined in (2.9). The trimming operation guarantees good performance in presence of long-tailed or impulsive noise and helps in the preservation of sharp edges, while the averaging operation causes the filter to perform well in presence of short-tailed noise.

Crossing Level Median Mean Filter (CLMMF)

On the basis of vector ordering another efficient technique combining the idea of the VMF and the AMF can be proposed. The main idea of this technique is to consider in the filter output not only first order statistics but all samples from processing window W.

Let w_i be a weight associated with i^{th} element of the ordered vectors $\mathbf{F}_{(0)}; \mathbf{F}_{(1)}; \ldots; \mathbf{F}_{(\tau)}; \ldots; \mathbf{F}_{(k-1)}$, then the filter output is:

$$\mathbf{F}_{CLMMF} = \sum_{i=0}^{k-1} w_i \cdot \mathbf{F}_i \tag{2.19}$$

One of the simplest possibilities of weights selection is

$$w_i = \begin{cases} 1 - \frac{k-1}{\sqrt{k(k+\gamma)}} & \text{for } i = 0 \\ \frac{1}{\sqrt{k(k+\gamma)}} & \text{for } i = 1 \ldots k-1 \end{cases}, \tag{2.20}$$

where γ is the filter parameter. For $\gamma \rightarrow \infty$ we obtain the standard vector median filter, and for $\gamma = 0$ we obtain the arithmetic mean.

Weighted Vector Median Filter (WVMF)

In [35] the vector median concept has been generalized and the so-called *Weighted Vector Median Filter* has been proposed. Using the weighted vector median approach, the filter output is the vector $\mathbf{F}_{WVMF}$, for which the following condition holds:

$$\sum_{j=0}^{k-1} w_j \cdot \rho(\mathbf{F}_{WVMF}, \mathbf{F}_j) < \sum_{j=0}^{k-1} w_j \cdot \rho(\mathbf{F}_i, \mathbf{F}_j), \quad i = 0, \ldots, k-1, \tag{2.21}$$

Central Weighted Vector Median Filter (WVMF)

Another weighted modification of the vector median filter was introduced in [28]. The construction of this filter is very similar to that of the WVMF proposed in [35] and in [13]. Let the distance associated with the center pixel be as in the VMF definition

$$R_0 = w_0 \cdot \sum_{j=0}^{k-1} \rho(\mathbf{F}_0, \mathbf{F}_j), \tag{2.22}$$

where w_0 is a weight assigned to the sum of distances between the central vector $\mathbf{F}_0$ and its neighbors. Then let the other sum of distances be

$$R_i = \sum_{j=0}^{k-1} \rho(\mathbf{F}_i, \mathbf{F}_j), \quad i = 1, \ldots, k-1. \tag{2.23}$$

Then for the CWVMF following condition should be fulfilled:

$$\sum_{j=0}^{k-1} \rho(\mathbf{F}_{CWVMF}, \mathbf{F}_j) < w_0 \cdot \sum_{j=0}^{k-1} \rho(\mathbf{F}_0, \mathbf{F}_j) \tag{2.24}$$

Above equation could be rewritten as follows:

$$\frac{\sum\limits_{j=0}^{k-1} \rho(\mathbf{F}_{CWVMF}, \mathbf{F}_j)}{\sum\limits_{j=0}^{k-1} \rho(\mathbf{F}_0, \mathbf{F}_j)} < w_0 \tag{2.25}$$

For $w_0 = 0$ no changes are introduced to the image, and for $w_0 = 1$ we obtain the standard vector median filter as proposed by Astola. If $\mathrm{w}_0 \in (0, 1)$, then the CWVMF filter has the ability of noise removal, while preserving fine image details (lines, edges, corners, texture).

Thresholded Vector Median Filter (TVMF)

The VMF gives good results for impulsive noise removal, but it has severe shortcoming - it changes more image pixels than it is necessary, and thus causes excessive oversmoothing.

One of the possibilities to reduce this effect is to introduce a threshold value which reduces the amount of changes introduced to the filtered image. This concept is used in the *Thresholded Vector Median Filter* (TVMF) which is defined as

$$\mathbf{F}_{TVMF} = \begin{cases} \mathbf{F}_{VMF} & \text{if } \|\mathbf{F}_{VMF} - \mathbf{F_0}\| > \beta \\ \mathbf{F}_0 & \text{otherwise} \end{cases} , \tag{2.26}$$

where $\mathbf{F}_{TVMF}$ is the output of the TVMF filter, $\mathbf{F}_{VMF}$ is the output of the VMF and $\mathbf{F_0}$ denotes original unchanged image vector, while $\|\cdot\|$ denotes the norm of the vector.

Fast Modified Vector Median Filter (FMVMF)

Another efficient technique for impulsive noise removal based on vector ordering is, the so-called *Fast Modified Vector Median Filter* (FMVMF) introduced in [27, 26, 25], which excludes the center pixel from the calculations of the aggregated distances associated with pixels from its neighborhood. The construction of the FMVMF is very similar to that of the VMF. Let the distance associated with the center pixel be

$$R_0 = -\beta + \sum_{j=1}^{k-1} \rho(\mathbf{F}_0, \mathbf{F}_j) , \tag{2.27}$$

where β is a threshold parameter and let the distance associated with the neighbors of $\mathbf{F}_0$ be

$$R_i = \sum_{j=1}^{k-1} \rho(\mathbf{F}_i, \mathbf{F}_j) , i = 1, \dots, k-1 . \tag{2.28}$$

Then, if R_τ is the minimal value of the function R and is smaller than R_0

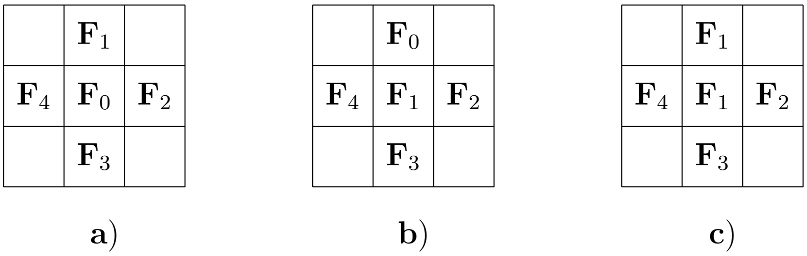
$$R_\tau = \sum_{j=1}^{k-1} \rho(\mathbf{F}_\tau, \mathbf{F}_j) \ < R_0 , \tag{2.29}$$

then $\mathbf{F}_0$ is being replaced by $\mathbf{F}_\tau$. Thus the FMVMF output fulfills following condition:

$$\sum_{j=1}^{k-1} \rho(\mathbf{F}_{FMVMF}, \mathbf{F}_j) < -\beta \sum_{j=1}^{k-1} \rho(\mathbf{F}_0, \mathbf{F}_j) , \tag{2.30}$$

which could be rewritten as:

$$\beta < \sum_{j=1}^{k-1} \left\{ \rho(\mathbf{F}_0, \mathbf{F}_j) - \rho(\mathbf{F}_{FMVMF}, \mathbf{F}_j) \right\} . \tag{2.31}$$



	$\mathbf{F}_1$	
$\mathbf{F}_4$	$\mathbf{F}_0$	$\mathbf{F}_2$
	$\mathbf{F}_3$	

a)

	$\mathbf{F}_0$	
$\mathbf{F}_4$	$\mathbf{F}_1$	$\mathbf{F}_2$
	$\mathbf{F}_3$	

b)

	$\mathbf{F}_1$	
$\mathbf{F}_4$	$\mathbf{F}_1$	$\mathbf{F}_2$
	$\mathbf{F}_3$	

c)

Figure 2.5: Illustration of the construction of the FMVMF for the 4-neighborhood system, as compared with the VMF. In the case of the vector median the center pixel is replaced by its neighbor and the sum of distances between the center and neighbor pixel is being calculated a) and b). So the sum of distances is $R_0 = \rho\{\mathbf{F}_0, \mathbf{F}_1\} + \rho\{\mathbf{F}_0, \mathbf{F}_2\} + \rho\{\mathbf{F}_0, \mathbf{F}_3\} + \rho\{\mathbf{F}_0, \mathbf{F}_4\}$ a) and $R_1 = \rho\{\mathbf{F}_1, \mathbf{F}_0\} + \rho\{\mathbf{F}_1, \mathbf{F}_2\} + \rho\{\mathbf{F}_1, \mathbf{F}_3\} + \rho\{\mathbf{F}_1, \mathbf{F}_4\}$ b). If $R_1 < R_0$ then the temporary output of the vector median operator is $\mathbf{F}_1$, then other pixels are being checked if they minimize the sum of distances. In the FMVMF, if the center pixel $\mathbf{F}_0$ is replaced by its neighbor $\mathbf{F}_1$, then the pixel $\mathbf{F}_0$ is rejected from W **c**) and the total distance $R_1 = \rho\{\mathbf{F}_1, \mathbf{F}_2\} + \rho\{\mathbf{F}_1, \mathbf{F}_3\} + \rho\{\mathbf{F}_1, \mathbf{F}_4\}$ between $\mathbf{F}_1$ (new center pixel) and its neighbors is calculated. If the total distance R_1 is smaller than $R_0 = \rho\{\mathbf{F}_0, \mathbf{F}_1\} + \rho\{\mathbf{F}_0, \mathbf{F}_2\} + \rho\{\mathbf{F}_0, \mathbf{F}_3\} + \rho\{\mathbf{F}_0, \mathbf{F}_4\} - \beta$ and gives the minimal value of the R function, then the center pixel is replaced by $\mathbf{F}_1$.

The major difference between the VMF and the FMVMF is presented in Fig. 2.5 c) and 2.6.

The omitting of the central pixel $\mathbf{F}_0$, when calculating R_τ, $\tau > 0$, is the most important feature of the new algorithm. As the central pixel is suspected to be noisy, it is not taken into consideration, when calculating the distances associated with the neighbors of $\mathbf{F}_0$. This is illustrated in Fig. 2.6.

Basic Vector Directional Filter (BVDF)

Within the framework of ranked type nonlinear filters the orientation difference between color vectors can also be used to remove vectors with atypical directions. The *Basic Vector Directional Filter* (BVDF) is a ranked order filter, similar to the VMF, which uses the angle between two color vectors as the distance criterion. This criterion is defined as the scalar measure

$$A_i = \sum_{j=0}^{k-1} \alpha(\mathbf{F}_i, \mathbf{F}_j), \quad \text{with} \quad \alpha(\mathbf{F}_i, \mathbf{F}_j) = \cos^{-1}\left(\frac{\mathbf{F}_i \cdot \mathbf{F}_j^T}{|\mathbf{F}_i|\,|\mathbf{F}_j|}\right), \tag{2.32}$$

the corresponding aggregated distance associated with the noisy vector $\mathbf{F}_i$ inside

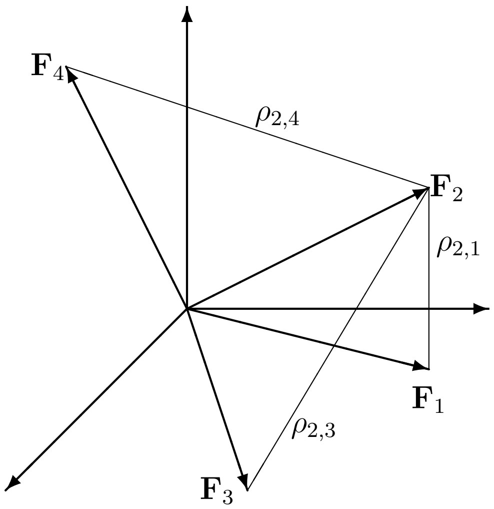


Figure 2.6: In the FMVMF, the distance R_0 is the same as shown in Fig. 2.4, however, for instance the distance R_2 equals now $R_2 = \rho(2,1) + \rho(2,3) + \rho(2,4)$.

the processing window W. As in the case of vector median filter, an ordering of the A_i 's

$$A_{(0)} \leq A_{(1)} \leq \ldots \leq A_{(\tau)} \leq \ldots, \leq A_{(k-1)}, \tag{2.33}$$

implies the same ordering of the corresponding vectors $\mathbf{F}_i$

$$\mathbf{F}_{(0)}; \mathbf{F}_{(1)}; \ldots; \mathbf{F}_{(\tau)}; \ldots; \mathbf{F}_{(k-1)}. \tag{2.34}$$

The BVDF outputs the vector $\mathbf{F}_{(0)}$ that minimizes the sum of angles with all the other vectors within the processing window. Since the BVDF uses only information about vector directions (chromaticity information) it cannot remove achromatic noisy pixels.

Generalized Vector Directional Filter (GVDF)

To overcome the deficiencies of the BVDF, the *Generalized Vector Directional Filter* (GVDF) was introduced [32]. The GVDF generalizes BVDF in the sense that its output is a superset of the single BVDF output. The first vector in (2.34) constitutes the output of the Basic Vector Directional Filter, whereas the first τ vectors constitute the output of the *Generalized Vector Directional Filter* (GVDF). In this way

$$BVDF\{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{k-1}\} = \mathbf{F}_0 \tag{2.35}$$

$$GVDF\{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_{k-1}\} = \{\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_\tau\}, \quad 1 \leq \tau \leq k-1 \tag{2.36}$$

The output of GVDF is subsequently passed through an additional filter in order to produce a single output vector. In this step the designer may only consider the magnitudes of the vectors $\mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_\tau$ since they have approximately the same direction in the vector space. As a result the GVDF separates the processing of color vectors into directional processing and then magnitude processing (the vector's direction signifies its chromaticity, while its magnitude is a measure of its brightness). The resulting cascade of filters is usually complex and the implementations may be slow since they operate in two steps.

Directional Distance Filter (DDF)

To overcome the deficiencies of the directional filters, a new method called *Directional - Distance Filter* (DDF) was proposed [12]. DDF constitutes a combination of VMF and BVDF and is derived by simultaneous minimization of their defining functions. Specifically, in the case of the DDF the distance inside the processing window is defined as:

$$B_i = \left(\sum_{j=0}^{k-1} \alpha\left(\mathbf{F}_i, \mathbf{F}_j\right) \right)^{\kappa} \left(\sum_{j=0}^{k-1} \rho\left(\mathbf{F}_i, \mathbf{F}_j\right) \right)^{1-\kappa} \tag{2.37}$$

where $\alpha\left(\mathbf{F}_i, \mathbf{F}_j\right)$ is the directional (angular) distance defined in 2.32 and distance $\rho\left(\mathbf{F}_i, \mathbf{F}_j\right)$ could be calculated using L_p norm. The parameter κ regulates the influence of angle and distance components. As for any other ranked-order filter, an ordering of the B_i's

$$B_{(0)} \leq B_{(1)} \leq \ldots \leq B_{(\tau)} \leq \ldots, \leq B_{(k-1)}, \tag{2.38}$$

implies the same ordering of the corresponding vectors $\mathbf{F}_i$

$$\mathbf{F}_{(0)}; \mathbf{F}_{(1)}; \ldots; \mathbf{F}_{(\tau)}; \ldots; \mathbf{F}_{(k-1)}. \tag{2.39}$$

Thus, DDF defines vector $\mathbf{F}_0$ as its output: $\mathbf{F}_{DDF} = \mathbf{F}_0$. For $\kappa = 0$ we obtain the VMF and for $\kappa = 1$ the BVDF. The DDF is defined for $\kappa = 0.5$ and its usefulness stems from the fact that it combines both the criteria used in BVDF and VMF [32].

Hybrid Directional Filter (HDF)

Another efficient rank-ordered operation called *Hybrid Directional Filter* was proposed in [8]. This filter operates on the directional and the magnitude of the color vectors independently and then combines them to produce a final output. This hybrid filter, which can be viewed as a nonlinear combination of the VMF and BVDF filters, produces an output according to the following rule:

$$\mathbf{F}_{HyF} = \begin{cases} \mathbf{F}_{VMF} & \text{if } \mathbf{F}_{VMF} = \mathbf{F}_{BVDF} \\ \left(\frac{||\mathbf{F}_{VMF}||}{||\mathbf{F}_{BVDF}||}\right) \cdot \mathbf{F}_{BVDF} & \text{otherwise} \end{cases}, \tag{2.40}$$

where $\mathbf{F}_{BVDF}$ is the output of the BVDF filter, $\mathbf{F}_{VMF}$ is the output of the VMF and $|| \cdot ||$ denotes the norm of the vector.

Signal Dependent Rank-Ordered Mean Filter (SD-ROM)

The majority of median filter modifications are implemented uniformly across the image, thus they modify pixels that are undisturbed by noise. As a result, they still tend to remove details from the image or leave too much impulsive noise. To avoid excessive blurring of images during filtering process, the *signal dependent rank-ordered mean* (SD-ROM) filter has been proposed by E. Abreu in 1995 [2, 1]. In the SD-ROM approach, the filtering operation is conditioned on the differences between the input pixels and remaining rank-ordered pixels in the sliding window [17, 16]. However the original formulation of SD-ROM filter was designed for gray-scale images, it could be extended to the multichannel case [15].

Let $F(\mathbf{x})$ be a gray-scale image and consider 3×3 window W with the center pixel denoted as F_0 (Fig. 2.7).

F_1	F_2	F_3
F_8	F_0	F_4
F_7	F_6	F_5

Figure 2.7: Processing window and notation of elements in the SD-ROM

Then we define vector containing all elements F_i of set W excluding the center pixel F_0. Assuming ordering of elements F_i:

$$F_{(1)} \leq F_{(2)} \leq, \ldots, \leq F_{(8)}, \tag{2.41}$$

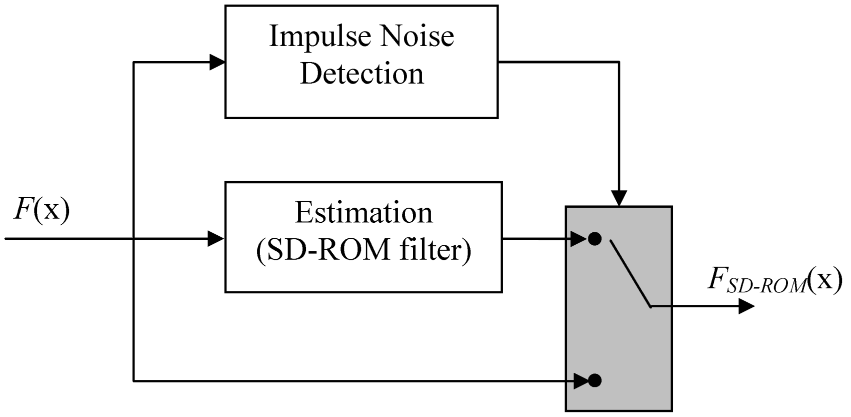


Figure 2.8: The SD-ROM filter structure.

we could define rank-ordered mean:

$$M_r = \frac{F_{(4)} + F_{(5)}}{2}, \tag{2.42}$$

and finally, we define rank ordered differences $\mathbf{D}_r$ as:

$$\mathbf{D}_r = [D_1, D_2, D_3, D_4], \tag{2.43}$$

where

$$D_i = \begin{cases} F_{(i)} - F_0 & \text{for } F_0 \leq M_r \ , \\ F_0 - F_{(n-i)} & \text{for } F_0 > M_r \ , \end{cases} \text{, for } i = 1, \ldots, 4 \tag{2.44}$$

The rank-ordered differences provide information about the likehood of corruption for the current pixel.

Figure 2.8 shows a block diagram of the SD-ROM filter structure. The purpose of the impulse noise detector is to determine whether the current pixel is corrupted. If a signal sample is detected as corrupted sample it is replaced with an estimation of the true value, based of the order statistics of the remaining pixels in the processing window W; otherwise, it is kept unchanged. Following equations describe SD-ROM filter output:

$$F_{SD-ROM} = \begin{cases} M_r & \text{if } D_i > T_i, \ i = 1, \ldots 4 \\ F_0 & \text{otherwise,} \end{cases} \tag{2.45}$$

where T_1, T_2, T_3, T_4 are threshold values, with $T_1 < T_2 < T_3 < T_4$. In other words, if the algorithms detects F_0 as a noisy sample, F_0 is replaced by the rank-ordered mean M_r, otherwise, it is kept unchanged. Computer simulation using variety of test images have shown that good results are obtained using thresholds selected from the following set of values [2, 1]:

$$T_1 \in \{4, 8, 12\}, T_2 \in \{15, 25\}, T_3 = 40, T_4 = 50.$$

For most cases, this algorithm performs well for subobtimally selected thresholds:

$$T_1 = 8, T_2 = 20, T_3 = 40, T_4 = 50.$$

2.3 Quality measures

The noise attenuation properties of the different filters can be examined by utilizing the different test images. The most common used are *LENA* and *PEPPERS*.

Different noise models should be taken into consideration in order to assess the performance of the filters under different noise scenarios.

There are standard *Root Mean Squared Error* (RMSE), the *Signal to Noise Ratio* (SNR), the *Peak Signal to Noise Ratio* (PSNR), the *Normalized Mean Square Error* (NMSE) and the *Normalized Color Difference* (NCD) [22] are used for the analysis.

All those objective quality measures could be defined using following formulas:

$$RMSE = \sqrt{\frac{1}{NML}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\sum_{l=1}^{L}\left(F^l\left(i,j\right) - \hat{F}^l(i,j)\right)^2} \tag{2.46}$$

$$NMSE = \frac{\sum\limits_{i=0}^{N-1}\sum\limits_{j=0}^{M-1}\sum\limits_{l=1}^{L}\left(F^l\left(i,j\right) - \hat{F}^l(i,j)\right)^2}{\sum\limits_{i=0}^{N-1}\sum\limits_{j=0}^{M-1}\sum\limits_{l=1}^{L}F^l\left(i,j\right)^2} \tag{2.47}$$

$$SNR = 10\log\left[\frac{\sum\limits_{i=0}^{N-1}\sum\limits_{j=0}^{M-1}\sum\limits_{l=1}^{L}F^l\left(i,j\right)^2}{\sum\limits_{i=0}^{N-1}\sum\limits_{j=0}^{M-1}\sum\limits_{l=1}^{L}\left(F^l\left(i,j\right) - \hat{F}^l(i,j)\right)^2}\right] \tag{2.48}$$

$$PSNR = 20\log\left(\frac{2^{nbb-1}}{RMSE}\right), \tag{2.49}$$

where M, N are the image dimensions, and $F^l(i,j)$ and $\hat{F}^l(i,j)$ denote the l^{th} component of the original image vector and its estimation at pixel $\left(i,j\right)$, respectively. The value *nbb*, in the PSNR definition, denotes number of bits per channel, so 2^{nbb-1} is a maximum value of the signal.

The NCD perceptual measure is evaluated over the uniform $L^*u^*v^*$ color space. The difference measure is given as follows:

$$NCD = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\Delta E_{Luv}}{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}E^*_{Luv}}, \tag{2.50}$$

where $\Delta E_{Luv} = [(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2]^{\frac{1}{2}}$ is the perceptual color error and $E^*_{Luv} = \left[(L^*)^2 + (u^*)^2 + (v^*)^2\right]^{\frac{1}{2}}$ is the *norm* or *magnitude* of the uncorrupted original image pixel vector in the $L^*u^*v^*$ space.

Bibliography

[1] E. Abreu. Signal-dependent rank-ordered mean (SD-ROM) filter. In G.L. Sicuranza S.K. Mitra, editor, *Nonlinear Image Processing*, pages 111–134. Academic Press, 2000.

[2] E. Abreu and S.K. Mitra. A signal-dependent rank-ordered mean (SD-ROM) filter -a new approach for removal of impulses from highly corrupted images. In *proc. Intl. Conf. on Accoustic, Speech, and Signal Processing*, volume 4, pages 2371–2374, 1995.

[3] J. Astola, P. Haavisto, and Y. Neuovo. Vector median filters. In *IEEE Proc.*, volume 78, pages 678–689, 1990.

[4] J. Bajon, M. Cattoen, and L. Liang. Identification of multicoloured objects using a vision module. In *Proceedings of of the 6th RoViSeC*, pages 21–30, 1986.

[5] V. Barnett. The ordering multivariate data. *Journal of Royal Statistical Society A*, 139(3):318–355, 1976.

[6] CIE. Colorimetry, second edn, viena, 1986. Publication No. 15.2-1986.

[7] M.M. Fleck, D.A. Forsyth, and C. Bregler. Finding naked people. In *Procc. of 4th European Conference on Computer Vision, Cambridge*, 1996.

[8] M. Gabbouj and F.A. Cheickh. Vector median - vector directional hybrid filter for colour image restoration. In *Proceedings of EUSIPCO*, pages 879–881, 1996.

[9] P. Haavisto, P. Heinonen, and Y. Neuovo. Vector fir-median hybrid filters for multispectral signals. *Electronics Letters*, 24(1):7–8, 1988.

[10] R.C. Hardie and G.R. Arce. Ranking in R^p and its use in multivariate image estimation. *IEEE Trans. on Circuits and Systems for Video Technology*, 1(2):197 –209, 1991.

[11] Ming Jiang. course on digital image processing, 2001. Department of Information Science, School of Mathematics, Peking University.

[12] D. Karakos and P.E. Trahanias. Generalized multichannel image filtering structures. *IEEE Trans. on Image Processing*, 6(7):1038–1045, 1997.

[13] R. Lukac. Adaptive impulse noise filtering by using center-weighted directional information. In *Proceedings of CGIV 2002*, pages 86–89, France, 2002.

[14] W. McIlhagga. Colour vision. In S.J. Sangwine and R.E.N. Horne, editors, *The Colour Image Processing Handbook*, pages 7–25. Chapman & Hall, 1998.

[15] M. S. Moore, M. Gabbouj, and S.K. Mitra. Vector SD-ROM filter for removal of impulse noise from color images. In *Proceedings of EURASIP, DSP for Multimedia Communications and Services (ECMCS), Krakow, Poland*, June 1999.

[16] M.S. Moore and S.K. Mitra. Performance analysis of the two-state signal-dependent rank order mean filter. *Nonlinear Image Processing X*, 3646:56–66, 1999.

[17] M.S. Moore and S.K. Mitra. Statistical threshold design for the two-state signal-dependent rank order mean filter. In *Proceedings of IEEE International Conference on Image Processing (ICIP), Vancouver, Canada*, volume 1, pages 904–907, Sept. 2000.

[18] H. Palus. Colour spaces. In S.J. Sangwine and R.E.N. Horne, editors, *The Colour Image Processing Handbook*, pages 67–89. Chapman & Hall, Cambridge, Great Britain, 1998.

[19] I. Pitas and P.Tsakalides. Multivariate ordering in color image processing. *IEEE Trans. on Circuits and Systems for Video Technology*, 1(3):247–256, 1991.

[20] I. Pitas and A.N. Venetsanopoulos. *Nonlinear Digital Filters: Principles and Applications.* Kluwer Academic Publishers, Boston, MA, 1990.

[21] K.N. Plataniotis and A.N. Venetsanopoulos. Vector filtering. In S.J. Sangwine and R.E.N. Horne, editors, *The Colour Image Processing Handbook*, pages 188–209. Chapman & Hall, Cambridge, Great Britain, 1998.

[22] K.N. Plataniotis and A.N. Venetsanopoulos. *Color Image Processing and Applications.* Springer Verlag, August 2000.

[23] W. Pratt. *Digital Image Processing.* John Willey & Sons, 1991.

[24] A.R. Smith. Color gammut transform pairs. *Computer Graphics (SIGGRAPH'78 Proceedings)*, 12(3):12–19, 1978.

[25] B. Smolka, A. Chydzinski, M. Szczepanski, K. N. Plataniotis, and A. N. Venetsanopoulos. On a new class of filters for the impulsive noise reduction in color images. In S. Loncaric and H. Babic, editors, *Proceedings of the 2nd International Symposium on Image and Signal Analysis, Pula, Croatia*, pages 114–118, June 2001.

[26] B. Smolka, A. Chydzinski, K. Wojciechowski, K. Plataniotis, and A.N. Venetsanopoulos. On the reduction of impulsive noise in multichannel image processing. *Optical Engineering*, 40(6):902–908, 2001.

[27] B. Smolka, M. Szczepanski, K.N. Plataniotis, and A. N. Venetsanopoulos. Fast modified vector median filter. In W. Skarbek, editor, *Computer Analysis of Images and Patterns, LNCS*, volume 2124, pages 570–580. Springer-Verlag, 2001.

[28] B. Smolka, M.K. Szczepanski, K. N. Plataniotis, and A. N. Venetsanopoulos. On the modified weighted vector median filter. In *Proceedings of Digital Signal Processing DSP2002, Santorini, Greece*, volume 2, pages 939–942, 2002.

[29] R. Tadeusiewicz. *Systemy wizyjne robotów przemysłowych.* WNT, 1992.

[30] J.M. Tenenbaum, T.D. Garvey, S. Weyl, and H. Wolf. An interactive facility for scene analysis research. Technical Report 87, Stanford Research Institute, AI Centre, 1974.

[31] T.Pomierski and H.M. Gross. Biological neural architecture for chromatic adaptation resulting in constant color sensations. In *Proceedings of ICNN 1996, IEEE International Conference on Neural Networks*, pages 734–739, 1996.

[32] P.E. Trahanias, D. Karakos, and A.N. Venetsanopoulos. Directional processing of color images : theory and experimental results. *IEEE Trans. on Image Processing*, 5(6):868–880, 1996.

[33] M.I. Vardavoulia, I. Andreadis, and Ph. Tsalides. A new vector median filter for colour image processing. *Pattern Recognition Letters*, 22:675–689, 2001.

[34] T. Viero, K. Oistamo, and Y. Neuvo. Three-dimensional median-related filters for color image sequence filtering. *IEEE Trans. on Circuits and Systems for Video Technology*, 4(2):129–142, 1994.

[35] R. Wichman, K. Oistamo, Q. Liu, M. Grundstrom, and Y. Neuovo. Weighted vector median operation for filtering multispectral data. In *Proceedings of the Meeting Visual communications and image processing '92, Boston, MA*, pages 376–383, Nov 1992.

[36] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae.* John Willey, second edition, 1982.