

# **Nonlinear Techniques for Color Image Processing**

BOGDAN SMOLKA

Silesian University of Technology

Department of Automatic Control

Akademicka 16 Str., 44-101 Gliwice, Poland

Email: `bsmolka@ia.polsl.gliwice.pl`

KONSTANTINOS N. PLATANIOTIS

The Edward S. Rogers Sr. Department of

Electrical and Computer Engineering

University of Toronto, 10 King's College Road

Toronto ON, M5S 3G4, Canada

Email: `kostas@dsp.toronto.edu`

ANASTASIOS N. VENETSANOPOULOS

Faculty of Applied Science and Engineering

University of Toronto, 35 St. George Street

Toronto, ON, M5S 3G4, Canada

Email: `anv@dsp.toronto.edu`

*Invited Chapter to appear in "Nonlinear Signal and Image Processing: Theory, Methods, and Applications", CRC Press, Kenneth E. Barner and Gonzalo R. Arce, Editors.*

## **1.1 Introduction**

The perception of color is of paramount importance to humans since they routinely use color features to sense the environment, recognize objects and convey information. That is why, it is necessary to use color information for computer vision, because in many practical cases location of scene objects can be obtained only when color information is considered, [137].

Noise filtering is one of the most important tasks in many image analysis and computer vision applications. Its goal is the removal of unprofitable information that may corrupt any of the following image processing steps.

The reduction of noise in digital images without degradation of the underlying image structures has attracted much interest in the last years, [70, 73, 83, 69, 93, 138, 101]. Recently, increasing attention has been given to the nonlinear processing of vector valued signals. Many of the techniques used for color noise reduction are direct implementations of the methods used for gray-scale imaging. The independent processing of color image channels is however inappropriate and leads to strong artifacts. To overcome this problem, the standard techniques developed for monochrome images have to be extended in a way which exploits the correlation among the image channels.

The acquisition or transmission of digital images through sensors or communication channels is often inferred by mixed impulsive and Gaussian noise. In many applications it is indispensable to remove the corrupted pixels to facilitate subsequent image processing operations such as edge detection, image segmentation and pattern recognition.

Numerous filtering techniques have been proposed to date for color image processing. Nonlinear filters applied to color images are required to preserve edges and details and to remove different kinds of noise. Especially, edge information is very important for human perception. Therefore, its preservation and possibly enhancement, are very important subjective features of the performance of nonlinear image filters.

### **1.1.1 Noise in Color Images**

Noise introduces random variations into sensor readings, making them different from the real values, and thus introducing errors and undesirable side effects in subsequent stages of the image processing. Faulty sensors, optic imperfectness, electronics interference, data transmission errors or aging of the storage material 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 power of the received signal. Image quality degradation can be also a result of processing techniques, such as demosaicking or aperture correction, which introduce various noise-like artifacts.

The noise encountered in digital image processing applications cannot always be described by the com-

monly assumed Gaussian model. Very often it has to be characterized in terms of impulsive sequences, which occur in the form of short duration, high energy spikes attaining large amplitudes with probability higher than predicted by the Gaussian density model. Thus image filters should be robust to impulsive or generally heavy-tailed noise. In addition, when color images are processed, care must be taken to preserve image chromaticity, edges and fine image structures.

### Impulsive Noise Models

In many practical applications, images are corrupted by noise caused either by faulty image sensors or by 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.

Transmission noise, also known as *salt & pepper* noise in gray-scale imaging, is modelled by an impulsive distribution. However, one of the problems encountered in the research on noise effects on image quality is the lack of commonly accepted multivariate impulsive noise model.

A number of simplified models has been introduced to assist the performance evaluation of the different color image filters. The impulsive noise model considered in this chapter is as follows, [83, 130, 128]

$$\mathbf{F}_I = \begin{cases} (F_1, F_2, F_3) & \text{with probability } (1 - p) \\ (d, F_2, F_3) & \text{with probability } p_1 \cdot p \\ (F_1, d, F_3) & \text{with probability } p_2 \cdot p \\ (F_1, F_2, d) & \text{with probability } p_3 \cdot p \\ (d, d, d)^T & \text{with probability } p_4 \cdot p \end{cases}, \quad (1.1)$$

where  $\mathbf{F}_I$  denotes the noisy signal,  $\mathbf{F} = (F_1, F_2, F_3)$  is the noise-free color vector, and  $d$  is the impulse value,  $p_1 + p_2 + p_3 + p_4 = 1$ . Impulse  $d$  can have either positive or negative values and we assume that when an impulse is introduced, forcing the pixel value outside the  $[0, 255]$  range, clipping is applied to push the corrupted noise value into the integer range specified by the 8-bit arithmetic.

### Mixed Noise

In many practical situations, an image is often corrupted by both additive Gaussian noise due to 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

$$\mathbf{F}_M = \begin{cases} \mathbf{F} + \mathbf{F}_G & \text{with probability } (1 - p), \\ \mathbf{F}_I & \text{otherwise,} \end{cases} \quad (1.2)$$

where  $\mathbf{F}$  is the noise-free color signal, the additive noise  $\mathbf{F}_G$  is modelled as zero mean, *white* Gaussian noise and  $\mathbf{F}_I$  is the transmission noise modelled as multivariate impulsive noise, [83].

This chapter is organized as follows. In the second section a short introduction to the adaptive techniques of noise removal in gray-scale images is presented. In the next section the anisotropic diffusion approach is described and its relation to the adaptive smoothing presented in Section 2 is discussed. In Section 4 a brief survey of the noise attenuation techniques applied in color image processing is presented. Section 5 is devoted to the new technique of noise reduction based on the concept of digital paths. In the last section the effectiveness of the new filtering framework is evaluated, a comparison between the new filter class and some of the filters presented in Section 4 is provided and the relation of the new filter class to the anisotropic diffusion presented in Section 3 is shown.

## 1.2 Adaptive Noise Reduction Filtering

In this section we examine some adaptive techniques used for the reduction of noise in gray-scale images. Some of the presented concepts can be redefined, so that they can be used to suppress noise in the multidimensional case.

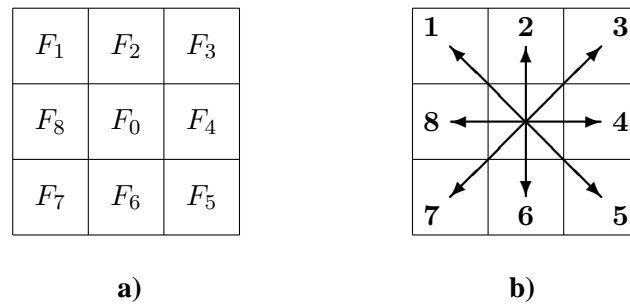


Figure 1.1: The filtering mask of size  $3 \times 3$  with the pixel  $F_0$  in the center a) and the directions between the central pixel and its neighbors b).

The most frequently used noise reduction transformations are the linear filters, which are based on the convolution of the image with the filter kernel of constant coefficients. This kind of filtering replaces the central pixel value  $F_0$  from the set of pixels  $F_0, F_1, \dots, F_n$ , (Fig. 1.1), belonging to the filter mask  $W$ , with the weighted average of the gray-scale values of the central pixel  $F_0$  and its  $n$  neighbors  $F_1, \dots, F_n$ , [38, 62]. The result of the convolution  $F_0^*$  of the kernel  $H$  with the pixels in  $W$  is

$$F_0^* = \frac{1}{Z} \sum_{k=0}^n H_k F_k, \quad Z = \sum_{k=0}^n H_k. \quad (1.3)$$

Linear filters are simple and fast, especially when they are separable, but their major drawback is that they cause blurring of the edges. This effect can be diminished choosing an appropriate adaptive nonlinear filter kernel, which performs the averaging in a selected neighborhood. The term *adaptive* means [41, 33], that the filter kernel coefficients change their values according to the image structure, which is to be smoothed.

Adaptive smoothing can be seen as a nonlinear process, in which noise is removed, while important image features are being preserved.

Different kinds of edge and structure preserving filter kernels have been proposed in the literature [47, 138, 38]. One of the simplest nonlinear schemes works with a filter kernel of the form  $H_k = 1 - |F_0 - F_k|$ ,

$$F_0^* = \frac{1}{Z} \sum_{k=0}^n [1 - |F_0 - F_k|] \cdot F_k, \quad Z = \sum_{k=0}^n [1 - |F_0 - F_k|], \quad F_k \in [0, 1]. \quad (1.4)$$

This filter takes with greater weighting coefficients those pixels of the neighborhood, whose intensity are close to the intensity of the central pixel  $F_0$ , and does not take into consideration the value of  $F_0$ , when defined as [96, 132, 52, 131, 61]

$$F_0^* = \frac{1}{Z} \sum_{k=1}^n [1 - |F_0 - F_k|] \cdot F_k, \quad Z = \sum_{k=1}^n [1 - |F_0 - F_k|], \quad (1.5)$$

which leads to a more robust filter performance. Similar structure has the gradient inverse weighted operator, which forms a weighted mean of the pixels belonging to a filter window. Again, the weighting coefficients depend on the difference of the gray-scale values between the central pixel and its neighbors, [132, 131],

$$F_0^* = \frac{1}{Z} \sum_{k=0}^n \frac{F_k}{\max\{\gamma, |F_0 - F_k|\}}, \quad Z = \sum_{k=0}^n \frac{1}{\max\{\gamma, |F_0 - F_k|\}}, \quad (\text{in [132] } \gamma = 0.5). \quad (1.6)$$

The Lee's local statistics filter [52, 51, 50], estimates the local mean and variance of the intensities of pixels belonging to a specified filter window  $W$  and assigns to the pixel  $F_0$  the value  $F_0^* = F_0 + (1 - \alpha)\hat{F}$ , where  $\hat{F}$  is the arithmetic mean of the image pixels belonging to the filter window and  $\alpha$  is estimated as  $\alpha = \max\{0, (\sigma_0^2 - \sigma^2)/\sigma_0^2\}$ , where  $\sigma_0^2$  is the local variance calculated for the samples in the filter window and  $\sigma^2$  is the variance calculated over the whole image. If  $\sigma_0 \gg \sigma$  then  $\alpha \approx 1$  and no changes are introduced. When  $\sigma_0 \ll \sigma$  then  $\alpha \approx 0$  and the central pixel is replaced with the local mean. In this way, the filter smooths with a local mean when the noise is not very intensive and leaves the pixel value unchanged when a strong signal activity is detected.

In [92, 91] a powerful adaptive smoothing technique related to the anisotropic diffusion, which will be discussed in the next section, was proposed. In this approach, the central pixel  $F_0$  is replaced by a weighted sum of all the pixel contained in the filtering mask

$$F_0^* = \frac{1}{Z} \sum_{k=0}^n w_k F_k, \quad \text{with} \quad w_k = \exp\left\{-\frac{|G_k|^2}{\beta^2}\right\}, \quad Z = \sum_{k=0}^n w_k, \quad (1.7)$$

where  $|G_k|$  is the magnitude of the gradient calculated in the local neighborhood of the pixel  $F_k$  and  $\beta$  is a smoothing parameter.

In [102] another efficient adaptive technique was proposed

$$F_0^* = \frac{1}{Z} \sum_{k=1}^N \exp\left\{-\frac{\rho_k^2}{\beta_1^2}\right\} \exp\left\{-\frac{|F_k - F_0|^2}{\beta_2^2}\right\} \cdot F_k, \quad (1.8)$$

where  $\rho_k$  denotes the topological distance between the central pixel  $F_0$  and the pixels  $F_k$ , ( $k = 1, 2, \dots, N$ ) of the filtering mask,  $\beta_1, \beta_2$  and  $N$  (number of neighbors of  $F_0$  in  $W$ ) are filter parameters. The concept of combining the topological distance between pixels with their intensity similarities has been further developed in the so called bilateral filtering [119, 27, 10], which can be seen as a generalization of the adaptive smoothing proposed in [67, 92, 91, 102, 112, 39].

Good results of noise reduction can usually be obtained by performing the  $\sigma$ -filtering [50, 54, 138]. This procedure computes a weighted average over the filter window, but only those pixels, whose gray values do not deviate too much from the value of the center pixel are permitted into the averaging process. This procedure computes a weighted mean over the filter window, but only those pixels whose values lie within  $\kappa \cdot \sigma$  of the central pixel value are taken into the average. This filter attempts to estimate a new pixel value with only those neighbors, whose values do not deviate too much from the value of  $F_0$

$$F_0^* = \frac{1}{Z} \sum_k H_k F_k, \quad \{k : |F_k - F_0| \leq \kappa \sigma\}, \quad (1.9)$$

where  $Z$  is the normalizing factor,  $\kappa$  is a parameter, (typically  $\kappa = 2$ ),  $\sigma$  is the standard deviation of all pixels belonging to  $W$  or the value of the standard deviation estimated from the whole image and  $H_k$  values are filter parameters.

Another adaptive scheme, called *k-nearest neighbor filter*, suggested in [30], replaces the gray level of the central pixel  $F_0$  by the average of its  $k$  neighbors whose intensities are closest to that of  $F_0$ , ( $k = 6$  and a window of size  $3 \times 3$  was recommended in [61]). The image noise can be also reduced by applying a filter, which substitutes the gray-scale value of the central pixel, by a gray tone from the neighborhood, which is closest to the average of all points in the filter window  $W$ , (*nearest neighbor filter*). In this way  $F_0^* = F_q$ , where  $q = \arg \{ \min \{ |F_k - \hat{F}| \} \}$ .

Another class of filters divides the filter masks into a set of regions, in which the variance of the pixel intensities is calculated. The aim of these filters is to find clusters of pixels which are similar to the central pixel of the filtering mask. Their output is defined as a mean value of the pixel values belonging to the sub-window in which the variance reaches the minimum. The Kuwahara filter [49, 120, 88], divides the  $5 \times 5$  filtering mask into four sub-windows as depicted in Fig. 1.2 a). In each of the sub-windows, the mean and the variance is calculated and the output of the filter is the mean value of the pixels from that sub-window, whose pixels have the smallest variance. This filtering scheme, based on searching for pixel clusters with similar intensities was further extended by introducing new regions in which the variance was measured [64, 63, 111], (Fig. 1.2 b, c) and [111], d).

This approach is in some way similar to the technique we propose in Section 1.5, in which the filters based on digital path are introduced. In the new approach, instead of looking for sub-windows with similar pixels, we investigate digital paths linking the central pixel with pixels belonging to the filter window.

Another class of adaptive algorithms is based on the rank transformations, defined using an ordering operator, which goal is the transformation of the set of pixels lying in a given filtering window  $W$  into a monotonically increasing sequence  $\{F_0, F_1, \dots, F_n\} \rightarrow \{F_{(0)}, F_{(1)}, \dots, F_{(n)}\}$ , with the property:  $F_{(k)} \leq F_{(k+1)}, k = 0, \dots, n - 1$ . In this way the rank operator is defined on the ordered values from the set  $\{F_{(0)}, \dots, F_{(n)}\}$  and has the form

$$F_0^* = \frac{1}{Z} \sum_{k=0}^n \varrho_{(k)} F_{(k)}, \quad Z = \sum_{k=0}^n \varrho_{(k)}, \quad (1.10)$$

where  $\varrho_k$  are nonzero weighting (ranking) coefficients. Taking appropriate ranking coefficients allows the definition of a variety of useful operators. The sequence

- $\{1, 1, \dots, 1\}$  corresponds to the moving average operator,
- $\{0, \dots, 0, \varrho_m = 1, 0, \dots, 0\}$ ,  $m = (1 + n)/2$ , generates the median, (for even number of neighbors  $n$ ),
- $\{0, \dots, 0, \varrho_{m-\alpha} = 1 = \dots = \varrho_m = \dots = \varrho_{m+\alpha} = 1, 0, \dots, 0\}$ ,  $0 \leq \alpha \leq m$  defines the  $\alpha$ -trimmed mean, which is a compromise between the median ( $\alpha = 0$ ) and the moving average ( $\alpha = m$ ),
- $\{\varrho_0 = 1, 0, \dots, 0, \varrho_n\}$  determines the so called mid-range filter.

The standard median exploits the rank-order information (order statistics) to eliminate impulsive noise. This filter substitutes the corrupted pixel with the middle-position element (median) of the ordered input samples. Since its introduction, it has been extensively studied and extended to the weighted median and its special case center weighted median filter.

The median filter is one of the most commonly used nonlinear filters. It has the ability of attenuating strong impulse noise, while preserving image edges. Its major drawback however, is that it wipes out structures, which are of the size of the filter window and this effect causes that the texture of a filtered image is strongly distorted. Another drawback of the standard median, is that it inevitably alters the details of the image not distorted by the noise process, since the standard median cannot distinguish between the corrupted and original pixels, and whether a pixel is corrupted or not, it is replaced by the local median within a filtering window. Therefore a trade-off between the suppression of noise and preservation of fine image details and edges has to be found. This can be accomplished in different ways, their goals is however always to diminish the filtering effect in image regions not affected by the noise process, [7, 6, 8, 11, 28, 2, 1, 48, 98, 4, 22].

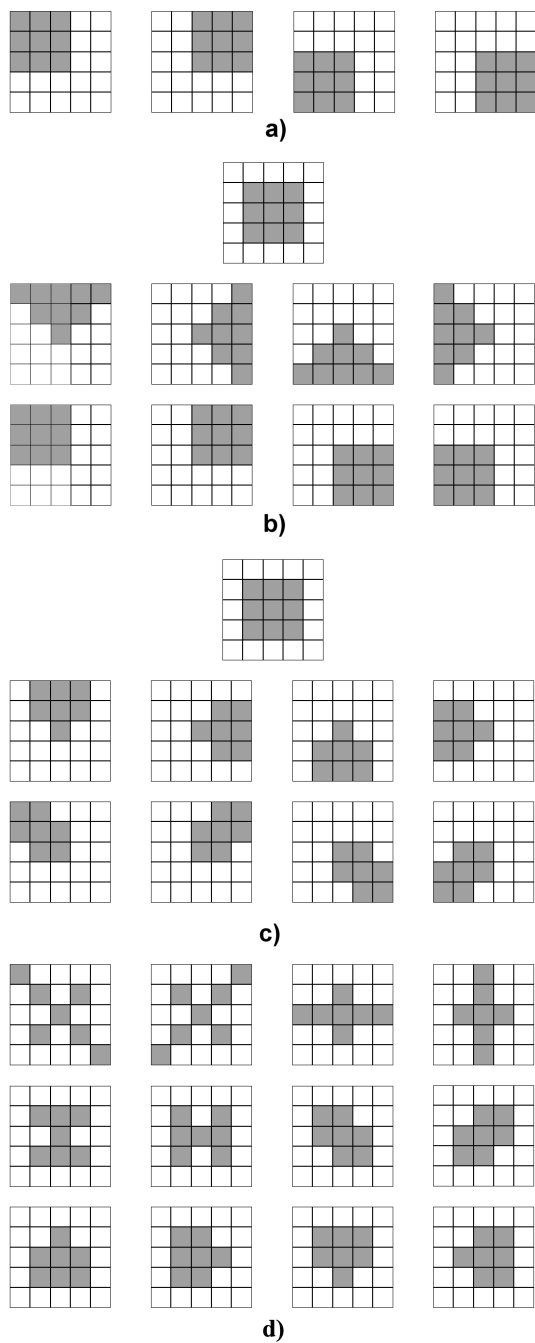
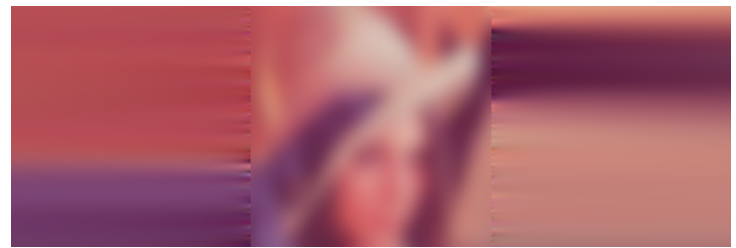
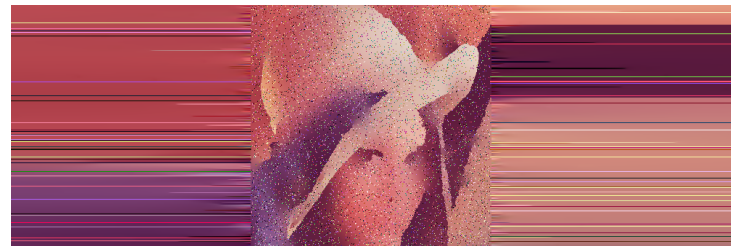


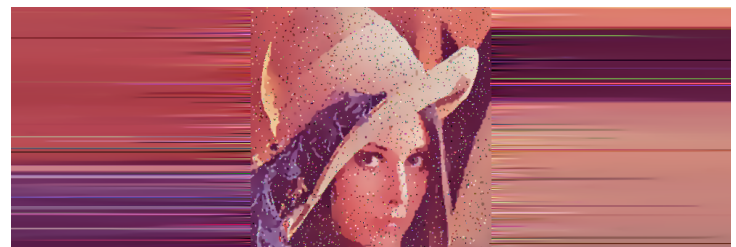
Figure 1.2: Different subwindow structures used in the filtering framework proposed in [49, 64] **a)**, [64, 63] **b), c)** and in [111], **d)**.



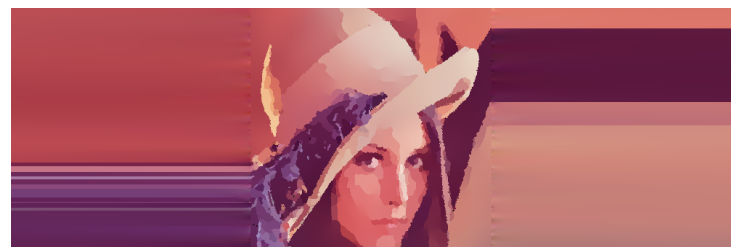
**a)**



**b)**



**c)**



**d)**

Figure 1.3: Illustrations of the development of the anisotropic diffusion process. The central part of the images shows the result obtained after 300 iterations. Left and right parts show the evolution of the column 25 and 325 of the  $350 \times 350$  color *LENA* image distorted by mixed impulsive and Gaussian noise, **a)** isotropic diffusion process (1.12), **b)** PMAD with  $c_1$ , (1.14), **c)** regularized AD of Catté [24, 25], **d)** new filter DPAF introduced in 1.5.



### 1.3 Anisotropic Diffusion

A powerful filtering technique, called anisotropic diffusion (AD), has been introduced by Perona and Malik, (P-M), [68, 67] in order to selectively enhance image contrast and reduce noise using a modified heat diffusion equation and the concepts of scale space, [136].

The main concept of *anisotropic diffusion* is based on the modification of the isotropic diffusion equation (1.12), with the aim to inhibit the smoothing across image edges. This modification is done by introducing a conductivity function that encourages intra-region smoothing over inter-region smoothing.

Since the introduction of the P-M method, a wide variety of techniques have been elaborated including multi-scale approaches, extensions to vector valued imaging [95, 37], multigrid methods [3], mathematical morphology inspired techniques and many others, [17, 60, 37, 121, 139, 34, 43, 44, 99].

Diffusion is a transport process that tends to level out concentration differences and in this way it leads to equalization of the spatial concentration differences. The elementary law of diffusion states that flux density  $\mathfrak{S}$  is directed against the gradient of concentration  $F$  in a given medium  $\mathfrak{S} = -c\nabla F$ , where  $c$  is the diffusion coefficient. If we use the continuity equation

$$\frac{\partial F}{\partial t} + \nabla \mathfrak{S} = 0, \quad \text{we obtain} \quad \frac{\partial F}{\partial t} = \nabla [c \nabla F]. \quad (1.11)$$

If  $F(x, y, t)$  denotes a real valued function representing the digital image, the equation of linear and isotropic diffusion is

$$\frac{\partial F(x, y, t)}{\partial t} = c \left[ \frac{\partial^2 F(x, y, t)}{\partial x^2} + \frac{\partial^2 F(x, y, t)}{\partial y^2} \right], \quad (1.12)$$

where  $x, y$  are the image coordinates,  $t$  denotes time,  $c$  is the conductivity coefficient.

Perona and Malik suggested that conductivity coefficient  $c$  should be dependent on the image structure and therefore they proposed the following partial derivative equation (PDE)

$$\frac{\partial F(x, y, t)}{\partial t} = \nabla [c(x, y, t) \nabla F(x, y, t)]. \quad (1.13)$$

The conductivity coefficient  $c(x, y, t)$  is a monotonically decreasing function of the image gradient magnitude and usually contains a free parameter  $K$ , which determines the amount of smoothing introduced by the nonlinear diffusion process. Different functions of  $c(x, y, t)$  have been suggested in the literature [18, 3, 89, 94, 5, 26, 90]. The most popular are those introduced in [67]

$$c_1 = \exp\left(-\frac{|\nabla F(x, y, t)|^2}{2K^2}\right), \quad c_2 = \left(1 + \frac{|\nabla F(x, y, t)|^2}{2K^2}\right)^{-1}. \quad (1.14)$$

The conductivity function  $c(x, y, t)$  is time and space-varying, it is chosen to be large in homogeneous regions to encourage smoothing and small at edges to preserve image structures.

The discrete version of Eq. (1.13) is

$$F_0^{t+1} = F_0^t + \lambda \sum_{k=0}^n c_k^t [F_k^t - F_0^t], \quad \text{for stability } \lambda \leq \lambda_0 = \frac{1}{n}, \quad (1.15)$$

where  $t$  denotes discrete time, (iteration number),  $c_k^t$  are the diffusion coefficients in  $n$  directions, (Fig. 1.1 b),  $F_0^t$  denotes the central pixel of the filtering window at time  $t$ ,  $F_k^t$  are its neighbors and  $\lambda_0$  is the largest value of  $\lambda$ , which guarantees the stability of the diffusion process.

It is quite easy to notice [10], that this equation is quite similar to the adaptive smoothing scheme proposed in [92, 91] and [87]. The Eq. (1.7) formulated in an iterative way

$$F_0^{t+1} = \frac{\sum_{k=0}^n w_k F_k^t}{\sum_{k=0}^n w_k}, \quad (1.16)$$

can be written as

$$F_0^{t+1} = F_0^t + \frac{\sum_{k=0}^n w_k F_k^t - F_0^t \sum_{k=0}^n w_k}{\sum_{k=0}^n w_k} = F_0^t + \frac{\sum_{k=0}^n w_k (F_k^t - F_0^t)}{\sum_{k=0}^n w_k} = F_0^t + \sum_{k=0}^n w_k^* (F_k^t - F_0^t), \quad (1.17)$$

where  $w_k^*$  are the normalized weighting coefficients. In this way, every adaptive smoothing scheme based on the averaging with weighting coefficients can be seen as a special realization of the general nonlinear diffusion scheme.

The equation of anisotropic diffusion, (1.15) can be written as

$$F_0^{t+1} = F_0^t \left[ 1 - \lambda \sum_{k=0}^n c_k^t \right] + \lambda \sum_{k=0}^n c_k^t F_k^t, \quad \lambda \leq \lambda_0 = \frac{1}{n}. \quad (1.18)$$

If we set  $[1 - \lambda \sum_{k=0}^n c_k^t] = 0$ , then we can switch off to some extent the influence of the central pixel  $F_0$  in the iteration process. This requires however that in each iteration step the  $\lambda$  values has to be a variable, dependent on time and image structure, equal to  $\lambda^t = [\sum_{k=0}^n c_k^t]^{-1}$ . The effect of diminishing the influence of the central pixel can be however achieved in a more natural way. Introducing the normalized conductivity coefficients  $C_k^t$

$$C_k^t = \frac{c_k^t}{\sum_{k=0}^n c_k^t}, \quad \sum_{k=0}^n C_k^t = 1, \quad (1.19)$$

Eq. (1.18) takes the form

$$F_0^{t+1} = F_0^t (1 - \lambda^*) + \lambda^* \sum_{k=0}^n C_k^t F_k^t, \quad \lambda^* = \lambda \sum_{k=0}^n c_k^t, \quad \lambda^* \in [0, 1], \quad (1.20)$$

which has the nice property, that for  $\lambda^* = 0$  no filtering takes place:  $F_0^{t+1} = F_0^t$  and for  $\lambda^* = 1$ , the central pixel is not taken into the weighted average and the anisotropic smoothing scheme reduces to a nonlinear, weighted average of the neighbors of  $F_0$

$$F_0^{t+1} = \sum_{k=1}^n C_k^t F_k^t. \quad (1.21)$$

In this way the central pixel is being replaced by a weighted average of its neighbors and the weights correspond to the similarity measure of the central pixel and its neighbors.

This scheme is very similar to the iterative approach proposed by Wang [132], (1.6), who recommended a gradient-inverse weighted noise smoothing algorithm

$$F_0^{t+1} = c_0 F_0^t + \sum_{k=0}^n c_k F_k^t \quad \text{with} \quad c_k = \frac{\max\{\gamma, |F_k - F_0|\}}{\sum_{k=0}^n \max\{\gamma, |F_k - F_0|\}}, \quad (1.22)$$

and is also quite similar to the approach of Lee [50] and to the algorithm of Smith [102], Eq. (1.8)

$$F_0^{t+1} = \frac{1}{Z} \sum_{k=1}^n c_k \cdot F_k^t, \quad c_k = \exp\left\{-\frac{\rho_k^2}{\beta_1^2}\right\} \exp\left\{-\frac{|F_k - F_0|^2}{\beta_2^2}\right\}, \quad k = 1, \dots, n. \quad (1.23)$$

which corresponds to the case of  $\lambda^* = 1$  in Eq. (1.20). The robustness of this scheme is achieved by rejecting the central pixel value of the filter mask when calculating the filter output. This scheme is especially efficient when the image is corrupted by heavy impulsive noise process.

Setting  $\lambda^* = 1$  in (1.20) is similar to taking the largest possible value of  $\lambda$  in (1.18),  $\lambda_0 = 1/n$  which ensures the stability of the anisotropic diffusion process, [89]. The good performance of an anisotropic diffusion scheme with  $\lambda^* = 1$  is confirmed by Fig. 1.4, which depicts the dependence of the efficiency of the P-M approach using the  $c_1$  conductivity function on the  $K$  and  $\lambda$  parameters for the gray scale *LENA* image distorted by Gaussian noise of different intensity. In this Figure, it is clearly visible that the best filter performance in terms of PSNR is achieved for  $\lambda$  close to  $\lambda_0 = 1/8$ , ( $3 \times 3$  mask), especially in the case of images distorted by Gaussian noise process of high  $\sigma$ . Such a setting of  $\lambda$  enables the diminishing of the influence of the central pixel, which ensures the suppression of the outliers injected by the noise process.

One of the major drawbacks of the anisotropic approach is that the optimal values of the parameters  $K$  and  $\lambda$  are unknown. Although  $K$  can be calculated using some a priori knowledge or can be estimated using some heuristic rules, the algorithm is very slow and needs many iterations to achieve the desired solution and also some stopping criterion is needed to finish the iteration process, before the image converges to the trivial solution, (the average value of the image pixels), [139, 133].

Another disadvantage of the Perona-Malik approach is that this algorithm is not able to cope with impulsive noise and as a result the noisy images goes through the diffusion process without perceptible improvement. The only way to force the diffusion to smooth out the impulsive noise is to increase the  $K$  value in (1.14), which results however in a higher blurring.

In order to improve the efficiency of the original scheme a regularized version was proposed, in which the conductance coefficient is a function of the gradient convolved with the Gaussian linear filter, [24, 25]

$$\frac{\partial F(x, y, t)}{\partial t} = \text{div} [\tilde{c}(x, y, t) \nabla F(x, y, t)], \quad (1.24)$$

where  $\tilde{c}(x, y, t) = f(|\nabla \mathcal{G}_\sigma * F(x, y, t)|)$ ,  $\mathcal{G}$  denotes the Gaussian kernel with standard deviation  $\sigma$ ,  $*$  denotes the convolution and  $f$  is a decreasing function. The advantage of this formulation is that it is mathematically well posed in contrary to the P-M scheme. However, the drawback of this approach is that the image discontinuities tend to be blurred and the whole scheme leads to a higher computational complexity of the anisotropic diffusion process.

Another solution to the impulsive noise problem is the introduction of robust conductivity functions. In [18] robust statistic norms were chosen to design the anisotropic diffusion process. However, these conductivity functions do not help increase the efficiency of the filtering in case of strong Gaussian or impulsive noise.

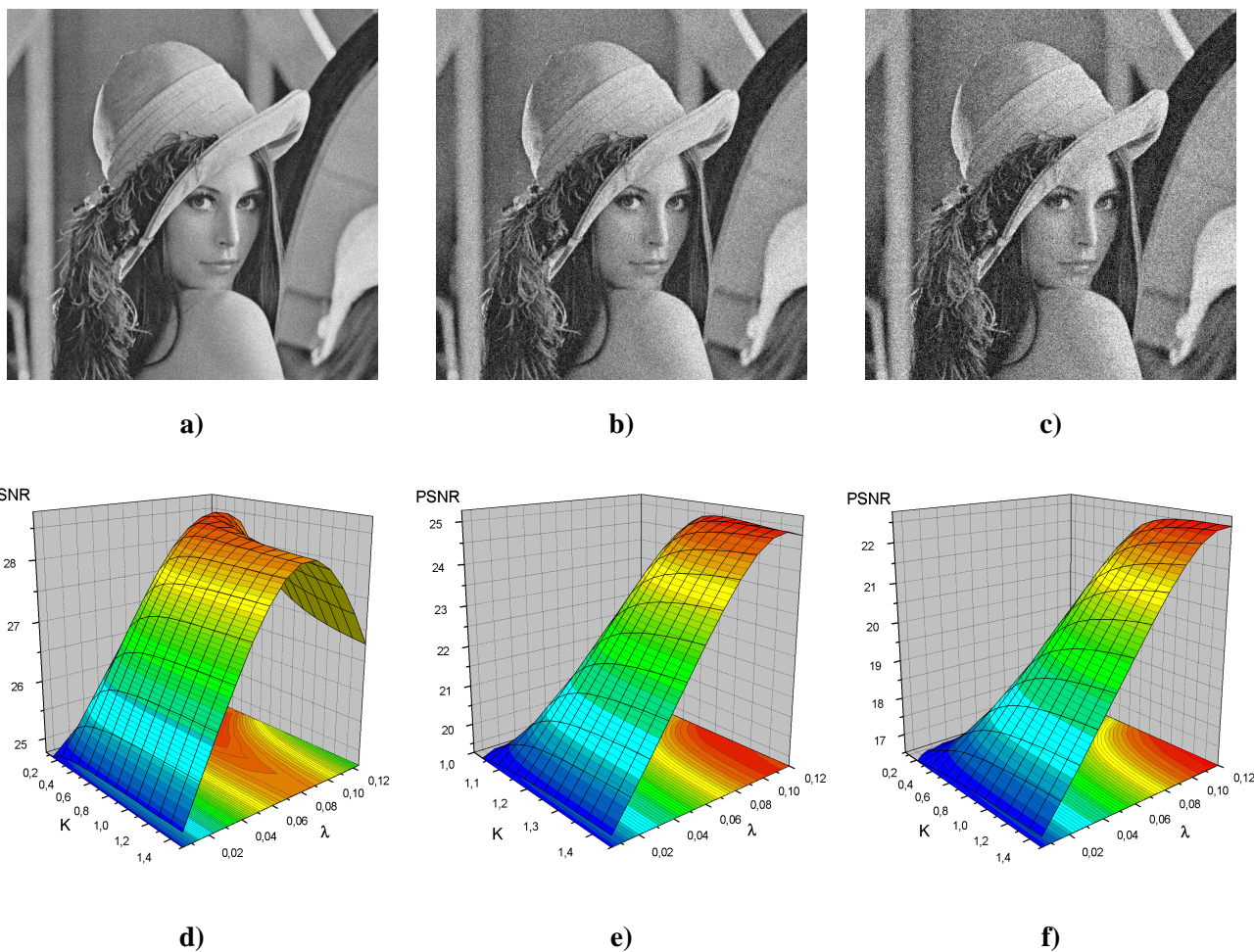


Figure 1.4: Dependence of the efficiency of the P-M scheme in terms of PSNR using the  $c_1$  conductivity function on the  $\lambda$  and  $K$  parameters, (1.14, 1.15). The test gray scale image *LENA* contaminated with Gaussian noise of: **a)**  $\sigma = 10$ , **b)**  $\sigma = 20$ , **c)**  $\sigma = 30$  are shown and below the respective plots of the noise reduction efficiency in terms of PSNR, after 3 iterations are presented, (**d-f**).

### 1.3.1 Anisotropic Diffusion Applied to Color Images

Let  $\mathbf{F}(x, y, t) = [F_r(x, y, t), F_g(x, y, t), F_b(x, y, t)]$  denote a color image pixel at position  $(x, y)$ , where  $F_r(x, y, t), F_g(x, y, t), F_b(x, y, t)$  are the red, green and blue channel respectively. The PDE equation (1.13) can be written for the multichannel case as

$$\frac{\partial \mathbf{F}(x, y, t)}{\partial t} = \nabla [c(x, y, t) \nabla \mathbf{F}(x, y, t)], \quad \mathbf{F}(x, y) = \begin{bmatrix} F_r(x, y) \\ F_g(x, y) \\ F_b(x, y) \end{bmatrix}, \quad \frac{\partial \mathbf{F}(x, y)}{\partial t} = \begin{bmatrix} \frac{\partial F_r(x, y)}{\partial t} \\ \frac{\partial F_g(x, y)}{\partial t} \\ \frac{\partial F_b(x, y)}{\partial t} \end{bmatrix}, \quad (1.25)$$

where  $c(x, y, t) = f(\|\mathbf{G}\|)$  is a conductivity function, which couples the three color image channels, [37, 134, 23, 53, 86]. The conductivity function is the same for all the image channels and is a function of the local gradient vector  $\mathbf{G}(x, y)$

$$\begin{bmatrix} \frac{\partial F_r(x, y, t)}{\partial t} \\ \frac{\partial F_g(x, y, t)}{\partial t} \\ \frac{\partial F_b(x, y, t)}{\partial t} \end{bmatrix} = \begin{bmatrix} \nabla [c(x, y, t) \nabla F_r(x, y, t)] \\ \nabla [c(x, y, t) \nabla F_g(x, y, t)] \\ \nabla [c(x, y, t) \nabla F_b(x, y, t)] \end{bmatrix}, \quad \mathbf{G}(x, y) = \begin{bmatrix} \frac{\partial \mathbf{F}(x, y)}{\partial x} \\ \frac{\partial \mathbf{F}(x, y)}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_r(x, y)}{\partial x}, \frac{\partial F_g(x, y)}{\partial x}, \frac{\partial F_b(x, y)}{\partial x} \\ \frac{\partial F_r(x, y)}{\partial y}, \frac{\partial F_g(x, y)}{\partial y}, \frac{\partial F_b(x, y)}{\partial y} \end{bmatrix}. \quad (1.26)$$

Estimating the local multichannel image gradient is one of the most important tasks, when designing an anisotropic diffusion scheme. Many of the approaches devised for color images are based on the *vector gradient norm* introduced by Di Zenzo [31]. Local variations of the color image  $\|d\mathbf{F}\|^2$  are expressed as

$$\|d\mathbf{F}\|^2 = \begin{bmatrix} dx \\ dy \end{bmatrix}^T \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}, \quad (1.27)$$

$$\text{where } \begin{cases} g_{11} = \left(\frac{\partial F_r(x, y)}{\partial x}\right)^2 + \left(\frac{\partial F_g(x, y)}{\partial x}\right)^2 + \left(\frac{\partial F_b(x, y)}{\partial x}\right)^2 \\ g_{22} = \left(\frac{\partial F_r(x, y)}{\partial y}\right)^2 + \left(\frac{\partial F_g(x, y)}{\partial y}\right)^2 + \left(\frac{\partial F_b(x, y)}{\partial y}\right)^2 \\ g_{12} = \left(\frac{\partial F_r(x, y)}{\partial x}\right) \left(\frac{\partial F_r(x, y)}{\partial y}\right) + \left(\frac{\partial F_g(x, y)}{\partial x}\right) \left(\frac{\partial F_g(x, y)}{\partial y}\right) + \left(\frac{\partial F_b(x, y)}{\partial x}\right) \left(\frac{\partial F_b(x, y)}{\partial y}\right) \end{cases}, \quad (1.28)$$

The eigenvalues of the matrix  $[g_{i,j}], i = 1, 2$

$$\lambda_+ = \frac{g_{11} + g_{22} + \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}, \quad \lambda_- = \frac{g_{11} + g_{22} - \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}, \quad (1.29)$$

are the extremum of  $\|d\mathbf{F}\|^2$  and the orthogonal eigenvectors determine the corresponding variation directions  $\eta$  and  $\xi$

$$\eta = \frac{1}{2} \arctan \frac{2g_{12}}{g_{11} - g_{22}}, \quad \xi = \eta + \frac{\pi}{2}. \quad (1.30)$$

Based on the eigenvalues, different gradient norms leading to various PDE schemes can be developed, [126, 127, 95, 94, 99, 19].

## 1.4 Noise Reduction Filters for Color Image Processing

Several nonlinear techniques for color image processing have been proposed over the years. Among them are linear processing methods, whose mathematical simplicity and the existence of a unifying theory make their design and implementation easy. However, not all filtering problems can be efficiently solved 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 color image processing techniques are introduced. Nonlinear techniques, to some extent, are able to suppress non-Gaussian noise and preserve important image elements, such as edges, corners and fine details, and eliminate degradations occurring during image formation and transmission through noisy channels.

### 1.4.1 Order-statistics Filters

One of the most popular families of nonlinear filters for impulsive noise removal are order-statistics filters, [129, 124, 73, 72, 75, 55, 65]. These filters utilize algebraic ordering of a windowed set of data to compute the output signal.

The early approaches to color image processing usually comprised extensions of the scalar filters to color images. Ordering of scalar data, such as the values of pixels in gray-scale images is well defined and it was extensively studied, [73]. However, the concept of input ordering, initially applied to 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 [12, 84, 65, 117]

- *marginal ordering* (M-ordering), where the multivariate samples are ordered along each dimension independently,
- *reduced or aggregated 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.

#### R-ordering filters

Let  $\mathbf{F}(\mathbf{x})$  be a multichannel image and let  $W$  be a window of finite size  $n + 1$ , (filter length). The noisy image vectors inside the filtering window  $W$  will be denoted as  $\mathbf{F}_j$ ,  $j = 0, 1, \dots, n$ . 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

$$R_i = \sum_{j=0}^n \rho(\mathbf{F}_i, \mathbf{F}_j), \quad (1.31)$$

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:  $R_{(0)} \leq R_{(1)} \leq \dots \leq R_{(\tau)} \leq \dots, \leq R_{(n)}$ , implies the same ordering of the corresponding vectors  $\mathbf{F}_i: \mathbf{F}_{(0)}; \mathbf{F}_{(1)}; \dots; \mathbf{F}_{(\tau)}; \dots; \mathbf{F}_{(n)}$ . Nonlinear ranked type multichannel filters define the vector  $\mathbf{F}_{(0)}$  as the output 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 [71, 40].

### Vector Median Filter (VMF)

The best known member of the family of the ranked type multichannel filters is the so called *Vector Median Filter*, (VMF) [9, 128, 13, 15, 36, 105, 107, 109, 130, 135]. 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 [9]. The output of the VMF is the pixel  $\mathbf{F}^* \in W$  for which the following condition is satisfied

$$\sum_{j=0}^n \rho(\mathbf{F}^*, \mathbf{F}_j) \leq \sum_{j=0}^n \rho(\mathbf{F}_i, \mathbf{F}_j), \quad i = 0, \dots, n. \quad (1.32)$$

It has been observed through experimentation that the Vector Median Filter (VMF) discards impulses and preserves edges and details in the image [9]. However, its performance in the suppression of additive white Gaussian noise, which is frequently encountered in image processing, is inferior to that of the Arithmetic Mean Filter (AMF). If a color image is corrupted by both additive Gaussian noise and impulsive noise, an effective filtering scheme should make an appropriate compromise between the Arithmetic Mean Filter and the Vector Median Filter.

### Extended Vector Median Filter (EVMF)

The VMF concept may be combined with linear filtering when the vector median is inadequate for 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 [9]. If the output of the *Arithmetic Mean Filter*, (AMF) is denoted as  $F_{AMF}$  then

$$\mathbf{F}^* = \begin{cases} \mathbf{F}_{AMF} & \text{if } \sum_{j=0}^n \|\mathbf{F}_{AMF} - \mathbf{F}_j\| < \sum_{j=0}^n \|\mathbf{F}_{VMF} - \mathbf{F}_j\| \\ \mathbf{F}_{VMF} & \text{otherwise} \end{cases}, \quad (1.33)$$

### $\alpha$ -trimmed Vector Median Filter (VMF $^\alpha$ )

In this filter, the  $1 + \alpha$  samples closest to the vector median are selected as inputs to an average type of filter, (see page 7). The output of the  $\alpha$ -trimmed VMF can be defined as follows [130, 84]

$$\mathbf{F}^* = \sum_{i=0}^{\alpha} \frac{1}{1 + \alpha} \mathbf{F}_{(i)}, \quad (1.34)$$

The trimming operation guarantees good performance in the presence of long tailed or impulsive noise and helps in the preservation of sharp edges. On the other hand, the averaging operation causes the filter to perform well in the presence of short tailed noise.

### Crossing Level Median Mean Filter (CLMMF)

On the basis of the vector ordering another efficient technique combining the idea of the VMF and the AMF can be proposed. Let  $w_i$  be a weight associated with  $i^{th}$  element of the ordered vectors  $\mathbf{F}_{(0)}; \mathbf{F}_{(1)}; \dots; \mathbf{F}_{(n)}$ , then the filter output is declared as  $\mathbf{F}_0^* = \sum_{i=0}^n w_{(i)} \cdot \mathbf{F}_{(i)}$ . One of the simplest possibilities of weight selection is

$$w_{(i)} = \begin{cases} 1 - \frac{n}{\sqrt{(n+1)(n+1+\gamma)}} & \text{for } i = 0 \\ \frac{1}{\sqrt{(n+1)(n+1+\gamma)}} & \text{for } i = 1, \dots, n, \end{cases} \quad (1.35)$$

where  $\gamma$  is the filter parameter. For  $\gamma \rightarrow \infty$  we obtain the standard vector median filter, and for  $\gamma = 0$  this filter reduces to the arithmetic mean (AMF).

### Weighted Vector Median Filter (WVMF)

In [135, 130, 4] 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  $\mathbf{F}^*$ , for which the following condition holds

$$\sum_{j=0}^n w_j \rho(\mathbf{F}^*, \mathbf{F}_j) \leq \sum_{j=0}^n w_j \rho(\mathbf{F}_i, \mathbf{F}_j), \quad i = 0, \dots, n. \quad (1.36)$$

### 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 using the scalar measure

$$A_i = \sum_{j=0}^n \alpha(\mathbf{F}_i, \mathbf{F}_j), \quad \text{with } \alpha(\mathbf{F}_i, \mathbf{F}_j) = \cos^{-1} \left( \frac{\mathbf{F}_i \cdot \mathbf{F}_j}{|\mathbf{F}_i| |\mathbf{F}_j|} \right). \quad (1.37)$$



As in the case of vector median filter, the ordering of the  $A_i$ 's implies the same ordering of the corresponding vectors  $\mathbf{F}_i$ . 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, 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, [122]. The GVDF generalizes BVDF in the sense that its output is a superset of the single BVDF output. The first vector in the ordered sequence constitutes the output of the Basic Vector Directional Filter, whereas the first  $\tau$  vectors constitute the output of the *Generalized Vector Directional Filter*, (GVDF)

$$BVDF\{\mathbf{F}_0, \mathbf{F}_1, \dots, \mathbf{F}_n\} = \mathbf{F}_0, \quad GVDF\{\mathbf{F}_0, \mathbf{F}_1, \dots, \mathbf{F}_n\} = \{\mathbf{F}_0, \mathbf{F}_1, \dots, \mathbf{F}_\tau\}, \quad 1 \leq \tau \leq n. \quad (1.38)$$

The output of GVDF is subsequently passed through an additional filter in order to produce a single output vector. In this step the designer can only consider the magnitudes of the vectors  $\mathbf{F}_0, \mathbf{F}_1, \dots, \mathbf{F}_\tau$  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, [57, 58].

### Directional Distance Filter (DDF)

To overcome the deficiencies of the directional filters, another method called *Directional - Distance Filter* (DDF) was proposed [42]. 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 accumulated distance inside the processing window is defined as

$$B_i = \left( \sum_{j=0}^n \alpha(\mathbf{F}_i, \mathbf{F}_j) \right)^\zeta \left( \sum_{j=0}^n \rho(\mathbf{F}_i, \mathbf{F}_j) \right)^{1-\zeta}, \quad (1.39)$$

where  $\alpha(\mathbf{F}_i, \mathbf{F}_j)$  is the directional (angular) distance defined in (1.37) and distance  $\rho(\mathbf{F}_i, \mathbf{F}_j)$  could be calculated using Minkowski  $L_p$  norm. The parameter  $\zeta$  regulates the influence of angle and distance components. As for any other ranked-order filter, an ordering of the  $B_i$ 's implies the same ordering of the corresponding vectors  $\mathbf{F}_i$ . Thus, DDF defines the  $\mathbf{F}_{(0)}$  vector as its output:  $\mathbf{F}_{DDF} = \mathbf{F}_0$ . For  $\zeta = 0$  we obtain the VMF and for  $\zeta = 1$  the BVDF. The DDF is defined for  $\zeta = 0.5$  and its usefulness stems from the fact that it combines both the criteria used in BVDF and VMF, [122, 56].

### Hybrid Directional Filter (HDF)

Another efficient rank-ordered operation called *Hybrid Directional Filter* HDF was proposed in [36]. This filter operates on the direction and 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

$$\mathbf{F}^* = \begin{cases} \mathbf{F}_{VMF} & \text{if } \mathbf{F}_{VMF} = \mathbf{F}_{BVDF} \\ \frac{\|\mathbf{F}_{VMF}\|}{\|\mathbf{F}_{BVDF}\|} \mathbf{F}_{BVDF} & \text{otherwise} \end{cases}, \quad (1.40)$$

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

### 1.4.2 Fuzzy Adaptive Filters

The performance of the different nonlinear filters based on order statistics depends heavily on the problem under consideration. The types of noise which are present in an image affect considerably the filter performance. To overcome difficulties associated with the uncertainty associated with the data, adaptive designs based on local statistics have been introduced [80, 79, 16, 32, 77, 78]. Such filters, utilize data-dependent coefficients to adapt to local image characteristics. The weights of the adaptive filters are determined by fuzzy transformations based on features from local data. The general form of the fuzzy adaptive filters is given as a nonlinear transformation of a weighted average of the input vectors inside the processing window

$$\mathbf{F}^* = f \left( \sum_{i=0}^n w_i^* \mathbf{F}_i \right) = f \left( \sum_{i=0}^n w_i \mathbf{F}_i / \sum_{i=0}^n w_i \right), \quad (1.41)$$

where  $f(\cdot)$  is a nonlinear function that operates over the weighted average of the input set. The relationship between the pixel under consideration and each pixel in the window should be reflected in the decision for the filters weights. In the adaptive design, the weights provide the degree to which an input vector contributes to the output of the filter. They are determined adaptively using fuzzy transformations of a distance criterion at each image position.

In this framework the weights are determined by fuzzy transformations based on features from local data. The fuzzy module extracts information without any a-priori knowledge about noise characteristics. The weighting coefficients are transformations of the distance between the vector under consideration, (center of the processing window  $W$ ) and all other vector samples inside the processing window  $W$ . This transformation can be considered to be a membership function with respect to a specific window component. The adaptive algorithm evaluates a membership function based on a given vector signal and then uses the membership values to calculate the filter output. Adaptive fuzzy algorithms utilize features extracted

from local data, here in the form of a sum of distances, as inputs to the fuzzy weights. In this case, the distance functions are not used to order input vectors. Instead they provide selected features in reduced space; features used as inputs for the fuzzy membership function.

Several candidate functions, such as triangular, trapezoidal, piecewise linear or Gaussian-like functions can be used as a membership function. If the distance criterion described by (1.37) is used as a distance measure, a sigmoidal membership function can be selected, [76, 83]

$$w_i = \beta (1 + \exp \{A_i\})^{-r}, \quad (1.42)$$

where  $A_i$  is a cumulative distance from (1.37), while  $\beta$  and  $r$  are parameters to be determined. The  $r$  value is used to adjust weighting effect of the membership function and  $\beta$  is a weight scale threshold. If the Minkowski  $L_p$  metric is used as the distance function, the fuzzy membership function with exponential form gives good results

$$w_i = \exp \left( -\frac{R_i^r}{\beta} \right), \quad (1.43)$$

where  $R_i$  is a cumulative distance associated with  $i^{\text{th}}$  vector in the processing window  $W$  using generalized Minkowski norm,  $r$  is a positive constant and  $\beta$  is a distance threshold.

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

### Fuzzy Weighted Average Filter

The first class of filters derived from the general nonlinear fuzzy algorithm is the so called Fuzzy Weighted Average Filters (FWAF). In this case, the output of the filter is a fuzzy weighted sum of the input set. The form of the filter is given as

$$\mathbf{F}_0^* = \frac{1}{Z} \sum_{i=0}^n w_i \mathbf{F}_i, \quad Z = \sum_{i=0}^n w_i. \quad (1.44)$$

This filter provides a vector-valued signal which is not included in the original set of inputs. The weighted average form of the filter provides a compromise between a nonlinear order statistics filter and an adaptive filter with data dependent coefficients. Depending on the form of the distance criterion and the corresponding fuzzy transformation, different fuzzy filters can be designed. If the distance criterion selected is the sum of vector angles, the Fuzzy Vector Directional Filter (FVDF) is obtained. If an  $L_1$  norm is used as the distance criterion, a fuzzy generalization of the Vector Median Filter (VMF) is constructed.

### Maximum Fuzzy Vector Directional Filters

Another possible choice of the nonlinear function  $f(\cdot)$  is the maximum selector. In this case, the output of the nonlinear function is the input vector that corresponds to the maximum fuzzy weight. Using the maximum selector concept, the output of the filter is a part of the original input set. The form of this filter is

$$\mathbf{F}_0^* = \mathbf{F}_i \quad \text{with} \quad i = \arg \max w_i, \quad i = 0, \dots, n. \quad (1.45)$$

In other words, as an output the input vector associated with the maximum fuzzy weight is selected. It must be emphasized that through the fuzzy membership function, the maximum fuzzy weight corresponds to the minimum distance. If the vector angle criterion is used to calculate distances, the fuzzy filter delivers the same output as the BVDF [76, 83]. If the  $L_1$  or  $L_2$  is adopted as distance criterion, the filter provides the same output as the VMF. Utilizing the appropriate distance function, different filters can be obtained. Thus, filters such as VMF or BVDF can be seen as special cases of this specific class of fuzzy filters.

### Fuzzy Ordered Vector Directional Filters

In many cases it is favorable not use all the inputs inside the operational window to produce the final output of the nonlinear filter. Instead, only a part of the vector-valued input signals can be used. The input vectors are ordered according to their respective fuzzy membership strengths. The form of the fuzzy ordered vector directional filter is given as

$$\mathbf{F}^* = \frac{1}{Z} \sum_{i=0}^{\tau} w_{(i)} \mathbf{F}_{(i)}, \quad Z = \sum_{i=0}^{\tau} w_{(i)}, \quad (1.46)$$

where  $w_{(i)}$  represents the  $i$ th ordered fuzzy membership function and  $w_{(\tau)} \leq w_{(\tau-1)} \leq \dots \leq w_{(0)}$ , with  $w_{(0)}$  being the fuzzy coefficient with the largest membership strength.

The above form of the filter constitutes a fuzzy generalization of the  $\alpha$ -trimmed filters, (1.34), [73]. Through the fuzzy transformation, the weights to be sorted are scalar values. In this way the nonlinear ordering process does not introduce any significant computational burden. Depending on the distance criterion and the associate fuzzy chosen by the designer, a number of different  $\alpha$ -trimmed filters can be obtained.

The fuzzy transformations of (1.42) and (1.43) are not the only way in which the adaptive weights of can be constructed. In addition to fuzzy membership functions, other design concepts can be utilized for the task. One of such designs is the *nearest neighbor rule* [82], in which the value of the weight  $w_i$  in (1.41) is calculated according to the following formula

$$w_i = \frac{D_{(n)} - D_{(i)}}{D_{(n)} - D_{(0)}}, \quad (1.47)$$

where  $D_{(n)}$  is the maximum distance in the filtering window, measured using an appropriate distance criterion, and  $D_{(0)}$  is the minimum distance, which is associated with the center-most vector inside the

window. As in the case of the fuzzy membership function, the value of the weight in (1.47) expresses the degree to which the vector  $\mathbf{F}_i$  is close to the center-most vector, and far away from the worst value, the outer rank.

In [82] an adaptive vector processing filter named Adaptive Nearest Neighbour Filter, (ANNF) was devised utilizing the general framework of (1.41). The weights in ANNF were calculated by using the formula of (1.47) with the angular distance as a measure of dissimilarity between the color vectors.

It is evident that the outcome of such an adaptive vector processing filter depends on the choice of the distance criterion selected as a measure of dissimilarity among vectors. As before, the  $L_p$  norm or the angular distance (sum of angles) between the color vectors can be used to remove vector signals with atypical directions. However, both these distance metrics utilize only part of the information carried by the color image vectors. As in the case of DDF, it is anticipated that an adaptive vector processing filter based on an ordering criterion, which utilizes both vector features, namely magnitude and direction, will provide a robust solution whenever the noise characteristics are unknown.

In [81] a distance measure for the noisy vectors was introduced

$$J_i = \sum_{j=0}^n [1 - S(\mathbf{F}_i, \mathbf{F}_j)], \quad \text{with} \quad S(\mathbf{F}_i, \mathbf{F}_j) = \left( \frac{\mathbf{F}_i \cdot \mathbf{F}_j}{\|\mathbf{F}_i\| \|\mathbf{F}_j\|} \right) \left( 1 - \frac{|\|\mathbf{F}_i\| - \|\mathbf{F}_j\||}{\max(\|\mathbf{F}_i\|, \|\mathbf{F}_j\|)} \right). \quad (1.48)$$

As can be seen, the similarity measure of (1.48) takes into consideration both the direction and the magnitude of the vector inputs. The first part of the measure  $S$  is equivalent to the angular distance (*vector angle criterion*) and the second part is related to the normalized difference in magnitude. Thus, if the two vectors under consideration have the same length, the second part of  $S(\mathbf{F}_i, \mathbf{F}_j)$  equals to one and only the directional information is used in (1.48). On the other hand, if the vectors under consideration have the same direction in the vector space (collinear vectors), the first part of  $S(\mathbf{F}_i, \mathbf{F}_j)$ , (directional information) equals to one and the similarity measure of (1.48) is based only on the magnitude of the difference part.

Utilizing this similarity measure, an adaptive vector processing filter based on the general framework of (1.41) and the weighting formula of (1.48) was devised in [81]. The so called Adaptive Nearest Neighbour Multichannel Filter (ANNMF) belongs to the adaptive vector processing filter family defined through (1.41). However, ANNMF combines the weighting formula of (1.47) with the new distance measure of (1.48) to evaluate its weights.

### 1.4.3 Nonparametric Adaptive Multichannel Filter

Consider the following model for the color image degradation process.

$$\mathbf{F}_j = \mathbf{X}_j + \mathbf{G}_j, \quad (1.49)$$

where  $\mathbf{X}_j$  is a *three-dimensional* uncorrupted image vector,  $\mathbf{F}_j$  is the corresponding noisy vector to be filtered and  $\mathbf{G}_j$  is an additive noise vector. In our analysis, it is assumed that the color image vectors are unknown and that the noise vectors are uncorrelated at the different image locations and signal independent.

Let us denote with  $\Phi(\mathbf{F})$  the minimum variance estimator of the color vector  $\mathbf{X}$ , given the noisy measurement vector  $\mathbf{F}$ . The expected square error of the filter, when the image vectors are corrupted by additive noise as in (1.49), can be written as

$$V = \int \int [\mathbf{X} - \Phi(\mathbf{F})][\mathbf{X} - \Phi(\mathbf{F})]^T f(\mathbf{X}|\mathbf{F}) f(\mathbf{F}) d\mathbf{X} d\mathbf{F}, \quad (1.50)$$

$$V = \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} [\mathbf{X} - \Phi(\mathbf{F})][\mathbf{X} - \Phi(\mathbf{F})]^T f(\mathbf{X}|\mathbf{F}) d\mathbf{X} \right] f(\mathbf{F}) d\mathbf{F}, \quad (1.51)$$

where  $z^T$  denotes the transpose of  $z$ . Since  $\Phi(\mathbf{F})$  does not enter into the outer integral and  $f(\mathbf{F})$  is always positive, it is sufficient for the optimal minimum variance estimator to minimize the expected value of the estimation cost (conditional Bayesian risk), given the observation  $\mathbf{F}$ . Thus, it is sufficient to minimize the quantity

$$V_{BR} = \int_{-\infty}^{\infty} [\mathbf{X} - \Phi(\mathbf{F})][\mathbf{X} - \Phi(\mathbf{F})]^T f(\mathbf{X}|\mathbf{F}) d\mathbf{X}. \quad (1.52)$$

The minimum variance estimator, which minimizes the above cost is then known to be

$$\Phi(\mathbf{F})_{MV} = \int_{-\infty}^{\infty} \mathbf{X} f(\mathbf{X}|\mathbf{F}) d\mathbf{X} = \int_{-\infty}^{\infty} \frac{\mathbf{X} f(\mathbf{X}, \mathbf{F})}{f(\mathbf{F})} d\mathbf{X}, \quad (1.53)$$

with

$$f(\mathbf{F}) = \int_{-\infty}^{\infty} f(\mathbf{X}, \mathbf{F}) f(\mathbf{X}) d\mathbf{X}. \quad (1.54)$$

If the densities in (1.52) are known and a training record of the sample pairs  $(\mathbf{X}, \mathbf{F})$  is available, the minimum variance estimator can be derived. Unfortunately, in a realistic image processing scenarios, no a-priori knowledge about the noise process or the image itself is available. Thus, a nonparametric estimator must be utilized to approximate the probability density functions (PDF) in (1.52).

Let us assume a window of finite length  $n$  centered around a noisy vector  $\mathbf{y}$ . Through this window, a set of multivariate noisy samples  $W = (\mathbf{F}_0, \mathbf{F}_1, \dots, \mathbf{F}_n)$  becomes available. Based on the samples from the filtering window  $W$ , an adaptive, data dependent multivariate kernel estimator can be devised to approximate the densities in (1.52). The form of the adaptive kernel estimator selected, is as follows

$$\hat{f}(\mathbf{X}, \mathbf{F}) = \frac{1}{N} \sum_{i=0}^n \frac{1}{h_i^L} K \left( \frac{\mathbf{F} - \mathbf{F}_i}{h_i} \right), \quad N = n + 1, \quad (1.55)$$

where  $\mathbf{F}_i$  is the  $i^{th}$  training vector, with  $i = 0, 1, \dots, n$ ,  $L = 3$  is the dimensionality of the measurement space and  $h_i$  is the data dependent smoothing parameter which regulates the shape of the kernel. The

variable kernel density estimator exhibits local smoothing, which depends both on the point at which the density is evaluated and also on the information on the local neighborhood in  $W$ .

The  $h_i$  can be any function of the sample size  $N = n + 1$ , [35]. The bandwidths  $h_i$  (smoothing factors) can be defined as a function of the aggregated distance between the local observation under consideration and all the other vectors inside the  $W$  window. Thus,

$$h_i = N^{-\frac{k}{L}} A_i = N^{-\frac{k}{L}} \sum_{k=0}^n \|\mathbf{F}_i - \mathbf{F}_k\|, \quad (1.56)$$

where  $k$  is a design parameter. The choice of the kernel function in (1.55) is not nearly as important as the bandwidth, (smoothing factor). For the applications, the multivariate extension of the exponential kernel  $K(z) = \exp(-|z|)$  or the Gaussian kernel  $K(z) = \exp(-|z^T z|/2)$  can be selected [35].

Given (1.52)-(1.55), the non-parametric estimator can be defined as

$$\Phi(\mathbf{F})_{NP} = \int_{-\infty}^{\infty} \frac{\mathbf{X} \hat{f}(\mathbf{X}, \mathbf{F})}{\hat{f}(\mathbf{F})} d\mathbf{X} = \sum_{i=0}^n \mathbf{X}_i \left( \frac{(N^{-1})h_i^{-L} K\left(\frac{\mathbf{F}-\mathbf{y}_i}{h_i}\right)}{\sum_{i=0}^n (N^{-1})h_i^{-L} K\left(\frac{\mathbf{F}-\mathbf{F}_i}{h_i}\right)} \right) \quad (1.57)$$

$$\Phi(\mathbf{F})_{NP} = \sum_{l=0}^n \mathbf{X}_i \left( \frac{h_i^{-L} K\left(\frac{\mathbf{F}-\mathbf{F}_i}{h_i}\right)}{\sum_{i=0}^n h_i^{-L} K\left(\frac{\mathbf{F}-\mathbf{F}_i}{h_i}\right)} \right) = \sum_{i=0}^n w_i^* \mathbf{X}_i \quad (1.58)$$

where  $w_i^*$  is a weighting function defined in the interval  $[0,1]$ .

To obtain the required estimate we must assume that, in the absence of noise, discrete sample vectors  $\mathbf{X}_i$  are available. This is not a severe restriction, since in many cases such samples may be obtained by a calibration procedure in a controlled environment, perhaps at a very high signal-to-noise ratio. In a real time image processing application however, that is not the case. Therefore, alternative suboptimal solutions are introduced. In a first approach, we substitute the vectors  $\mathbf{X}_i$  in (1.57) with their noisy measurements. The resulting Adaptive Nonparametric Multichannel Filter (ANMF) is solely based on the available noisy vectors and the form of the minimum variance estimator. Thus, the form of the ANMF is

$$\Phi_1(\mathbf{F})_{ANMF} = \sum_{i=0}^n \mathbf{F}_i \left( \frac{h_i^{-L} K\left(\frac{\mathbf{F}-\mathbf{F}_i}{h_i}\right)}{\sum_{i=0}^n h_i^{-L} K\left(\frac{\mathbf{F}-\mathbf{F}_i}{h_i}\right)} \right). \quad (1.59)$$

A different form of the adaptive nonparametric estimator can be obtained if a reference vector is used instead of the actual noisy measurement. The ideal reference vector is of course the actual value of the multidimensional signal in the specific location under consideration. However, since the  $\mathbf{X}_0$  vector is not available, a robust estimate, usually evaluated in a small subset of the input vector set, is utilized instead. Usually the vector median  $\mathbf{X}^{VM}$  is the preferable choice, since it smooths out impulsive noise and preserves

to some extent the edges. The median based Adaptive Nonparametric Multichannel Filter has then the following form

$$\Phi_2(\mathbf{F})_{ANMF} = \sum_{i=0}^n \mathbf{X}_i^{VM} \left( \frac{h_i^{-L} K \left( \frac{\mathbf{F} - \mathbf{F}_i}{h_i} \right)}{\sum_{i=0}^n h_i^{-L} K \left( \frac{\mathbf{F} - \mathbf{F}_i}{h_i} \right)} \right). \quad (1.60)$$

This filter can be viewed as a double-window, two stage estimator. First the original image is filtered by a multichannel median filter in a small processing window in order to reject possible outliers and then an adaptive nonlinear filter with data dependent coefficients defined in (1.57) is utilized to provide the final filtered output.

## 1.5 Digital Paths Approach to Color Image Filtering

In this section a novel approach to color image filtering is proposed. Instead of using a fixed window, the new method exploits connections between image pixels using the concept of digital paths. According to the proposed methodology, image pixels are grouped together, forming paths that reveal the underlying structural dynamics of the image, (see Figs. 1.5, 1.6). Depending on the design principles and the computational constraints, the new filter framework allows the paths to be considered on the entire image or to be restricted to a predefined search area, [108, 104]. The new approach focuses on the latter case.

To facilitate comparisons with existing ranked type operations and to illustrate the computational efficiency of the proposed framework, the path searching area is allowed to match the window  $W$  used by the ranked type filters. However, instead of the indiscriminately use of the window pixels, an approach advocated by the majority of existing multichannel filters, the proposed here framework allows for the formation of a number of digital path models, which in turn are used to determine the coefficients of a weighted average type of filtering operation.

The new filter class based on digital paths and connection cost can be seen as a powerful generalization of the multichannel anisotropic diffusion presented in Section 1.3 and an extension of the fuzzy adaptive filters described in 1.4.2. The filters discussed there are shown in this Section to be a special case of the new filtering scheme, when a digital path is degenerated to a step of length 1.

The path connection costs evaluated over all possible digital paths, are used to derive fuzzy membership functions that quantify the similarity between vectorial inputs. The proposed filtering structure is then using the function outputs to appropriately weight input contributions in order to determine the filtering result. The proposed filtering schemes parallelize the familiar structure of the adaptive multichannel filter introduced in [74] and they can successfully eliminate Gaussian, impulsive as well as mixed-type noise. However, thanks to the introduction of the digital paths in its supporting element, the new filters not only preserve edges and fine image details, but can also act as an image sharpening operators.



### 1.5.1 Connection Cost Defined Over Digital Paths

In order to perform operations based on the distances we first need to precisely define the notion of a topological distance. The concept of a topological distance between image points is of extreme importance in many applications based on the distance transformation, which is one of the fundamental operations of mathematical morphology, [20, 21, 100, 85].

Let  $\mathcal{B}$  be a nonempty set. We can measure distances between points in  $\mathcal{B}$ , which amounts to defining a real valued function on the Cartesian product  $\mathcal{B} \times \mathcal{B}$  of  $\mathcal{B}$  with itself. Let the function  $\rho : \mathcal{B} \times \mathcal{B} \rightarrow \mathbb{R}$  be called a distance if it is positive definite:  $\rho(x, y) \geq 0$ , with  $\rho(x, y) = 0$ , when  $x = y$  and symmetric:  $\rho(x, y) = \rho(y, x)$ , for all  $x, y \in \mathcal{B} \times \mathcal{B}$ . A distance is called a metric if additionally it satisfies the triangle inequality [46]:  $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$ , for all  $x, y, z \in \mathcal{B} \times \mathcal{B}$ .

In digital image processing three basic distance functions are usually applied. If  $p = (p_1, p_2)$  and  $q = (q_1, q_2)$  denote two image points ( $p, q \in Z^2$ ) then we define the *City-Block Distance*:  $\rho_4(p, q) = |p_1 - q_1| + |p_2 - q_2|$ , *Chessboard Distance*:  $\rho_8(p, q) = \max\{|p_1 - q_1|, |p_2 - q_2|\}$  and *Euclidean Distance*:  $\rho_E(p, q) = [(p_1 - q_1)^2 + (p_2 - q_2)^2]^{\frac{1}{2}}$ . Using the city-block and chessboard distances we are able to define the two basic types of neighborhoods, 4-neighborhood  $\mathcal{N}_4(x) = \{y : \rho_4(x, y) = 1\}$  and 8-neighborhood  $\mathcal{N}_8(x) = \{y : \rho_8(x, y) = 1\}$ .

Let  $\omega \in \{4, 8\}$ . Two points  $p, q \in Z^2$  are said to be in  $\mathcal{N}_\omega$ -neighborhood relation, (denoted as  $\sim$ ), or to be  $\mathcal{N}_\omega$ -adjacent if  $q \in \mathcal{N}_\omega(p)$  or equivalently  $p \in \mathcal{N}_\omega(q)$ . This  $\mathcal{N}_\omega$ -adjacency relation defines a graph structure on the image domain, called  $\mathcal{N}_\omega$ -adjacency graph. On the graph, a finite  $\mathcal{N}_\omega$ -path can be defined as a sequence of points  $(p_0, p_1, \dots, p_\eta)$  such that for  $i \in \{1, 2, \dots, \eta\}$  the point  $p_{i-1}$  is  $\mathcal{N}_\omega$  adjacent to  $p_i$ . A path is called simple if  $i \neq j$  implies that  $p_i \neq p_j$ . This is a very important property of a path, as it means that a path does not intersect itself or in other words it is self-avoiding, [59, 113].

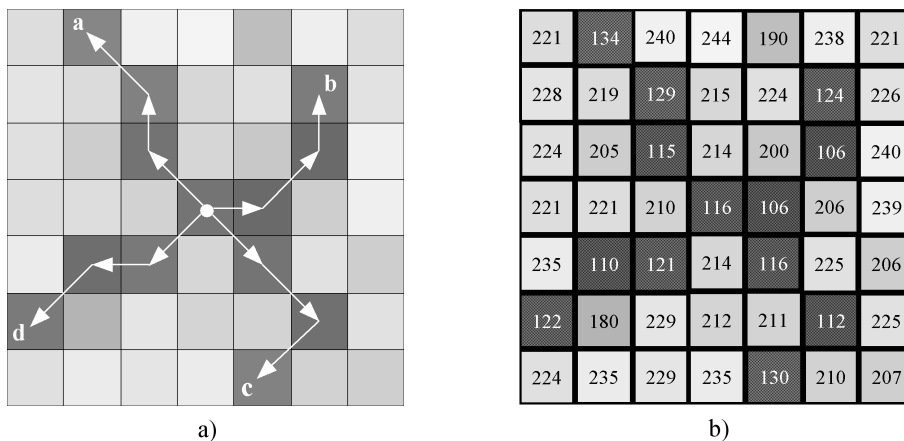


Figure 1.5: Illustration of the concept of digital paths and connection cost. The pixels a, b, c, d are connected with the central pixel along paths whose connection costs are minimal.

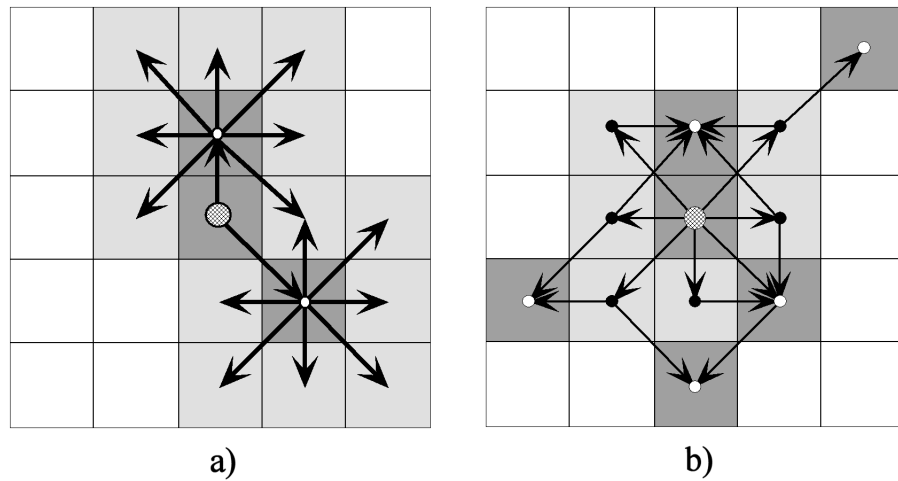


Figure 1.6: In the DPAF and DPAL filters, the weights are assigned to the pixels surrounding the central pixel and are determined in different ways. In the DPAF approach **(a)**, the weights in (1.74) are calculated exploring all digital paths starting from the central pixel and crossing its nearest neighbors, then a weighted average of the nearest neighbors of the central pixel is calculated, (1.75). In the DPAL approach, the weights are obtained by exploring all digital paths leading from the central pixel to the pixels contained in the filtering window **(b)** and then a weighted average of all pixels from that window is calculated, (1.81).

Using the distances between neighboring points, which are called prime distances [114], we are able to define a distance between any two image points by following all admissible paths linking those points and then taking the minimum of the total length over all possible paths, which is the sum of the prime distances between the nodes of the paths. In this way, the distance between two image points is the length of the path for which the sum of the prime distances between the path nodes is minimal. For the city-block distance the admissible paths consist of horizontal and vertical moves only, whereas for the chessboard distance also the diagonal moves are allowed. The prime distances for the two kinds of neighborhood are declared in this work to be equal to 1.

Let us now introduce the definition of a geodesic distance. Let us assume, that  $\mathbb{R}^2$  is the Euclidean space,  $S$  is a planar subset of  $\mathbb{R}^2$  and  $x, y$  are points belonging to set  $S$ . A path from  $x$  to  $y$  is a continuous mapping  $\Pi: [a, b] \rightarrow S$ , such that  $\Pi(a) = x$  and  $\Pi(b) = y$ . The point  $x$  is considered as the starting point, while  $y$  is the ending point on the path  $\Pi$ , [21].

An increasing polygonal line  $P$  on the path  $\Pi$  is any polygonal line such that  $P = \{\Pi(\lambda_i)\}_{i=0}^{\eta}$ ,  $a = \lambda_0 < \dots < \lambda_{\eta} = b$ . The length of the polygonal line  $P$  is considered to be the total sum of its constitutive line segments  $L(P) = \sum_{i=1}^{\eta} \rho(\Pi(\lambda_{i-1}), \Pi(\lambda_i))$ , where  $\rho(x, y)$  is the distance between the points  $x$  and  $y$ , when a specific metric is adopted. A path  $\Pi$  from  $x$  to  $y$  is called rectifiable, if and only if  $L(P)$ , where  $P$

is an increasing polygonal line, is bounded. Its upper bound is called the length of the path  $\Pi$ .

The geodesic distance  $\rho^S(x, y)$  between points  $x$  and  $y$  is the lower bound of the length of all paths leading from  $x$  to  $y$  which are totally included in  $S$ . If such paths do not exist, then the value of the geodesic distance is set to  $\infty$ . In general  $\rho^S(x, y) \geq \rho(x, y)$ . However, if the set  $S$  is convex, meaning that there are no points on the line between  $x$  and  $y$  that are not members of  $S$ , the geodesic distance verifies  $\rho^S(x, y) = \rho(x, y)$ .

The notion of a path can be extended to a lattice, which is a set of discrete points on the plane, in our case the spatial locations of the image pixels. Let a digital lattice  $\mathcal{H} = (\mathbf{F}, \mathcal{N})$  be defined by  $\mathbf{F}$ , which is the set of all points of the plane, (pixels of a color image) and a neighborhood relation  $\mathcal{N}$  between the lattice points [97].

A digital path  $P = \{p_i\}_{i=0}^\eta$  defined on the lattice  $\mathcal{H}$  is a sequence of neighboring points  $(p_{i-1}, p_i) \in \mathcal{N}$ . The length  $L(P)$  of the digital path  $P \{p_i\}_{i=0}^\eta$  is simply  $\sum_{i=1}^\eta \rho^{\mathcal{H}}(p_{i-1}, p_i)$ , where  $\rho^{\mathcal{H}}$  denotes the distance between two neighboring points of the lattice  $\mathcal{H}$  and the geodesic distance between  $p_0$  and  $p_\eta$  is the minimal length of  $L(P)$ .

Constraining the paths to be totally included in a predefined set  $W$  yields the digital geodesic distance  $\rho^W$ . In this work  $\mathcal{N}_\omega$ -neighborhood system ( $\omega = 4$  or  $\omega = 8$ ) is considered, with a topological distance of 1 assigned to any neighboring points and the set  $W$  is the supporting window of appropriate size. All paths considered in this chapter are included in the filtering window  $W$ , (Fig. 1.7).

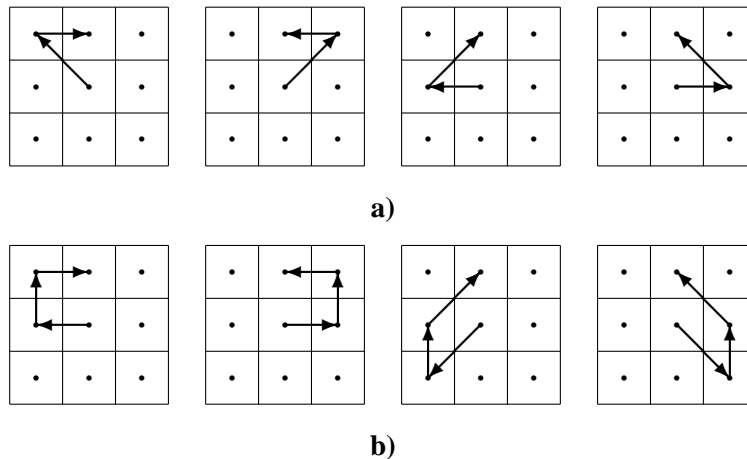


Figure 1.7: Digital paths of **a)** length 2 and **b)** length 3, connecting two neighboring points within a predefined window  $W$  of size  $3 \times 3$ , when the 8-neighborhood system is applied.

Let us now adopt the following notation, which will help us define the distance functions defined over geodesic paths. The starting point of a path will be denoted as  $p_0 = (x_0, y_0)$ . Its neighbors will be denoted as  $p_1 = (x_{u_1}, y_{v_1})$ , which means that the neighbors are the second points of all digital paths originating at  $p_0$ . Then the third point of a digital path starting at  $p_0$  will be  $p_2 = (x_{u_2}, y_{v_2})$  and so on, till the path

reaches in  $\eta$  steps the ending point  $p_\eta = (x_{u_\eta}, y_{v_\eta})$ . In this way the sequences  $x_{u_1}, \dots, x_{u_\eta}$  and  $y_{v_1}, \dots, y_{v_\eta}$  uniquely define the digital path starting at  $x_0, y_0$  and ending at  $x_{u_\eta}, y_{v_\eta}$ . The set of all possible digital paths contained in  $W$  joining two points  $x, y \in W$  will be denoted as  $\Psi^W(x, y)$ .

Two pixels  $x$  and  $y$  will be called connected, (hereafter denoted as  $x \Leftrightarrow y$ ), if there exists a digital path  $P^W(x, y)$  contained in the set  $W$  starting from  $x$  and ending at  $y$ . If two pixels  $p_0$  and  $p_\eta$  are connected by a digital path  $P^{W,\eta} \{p_0, p_1, \dots, p_\eta\}$  of length  $\eta$  then let  $\Lambda^{W,\eta} \{p_0, p_1, \dots, p_\eta\}$  be a measure of the connection cost defined over the digital path linking the starting point  $p_0$  and ending point  $p_\eta$ , ( $f$  is a nonnegative scalar function of  $\eta$  vector variables)

$$\Lambda^{W,\eta} \{p_0, \dots, p_\eta\} = f \{ \mathbf{F}(p_0), \dots, \mathbf{F}(p_\eta) \} = f \{ \mathbf{F}(x_0, y_0), \mathbf{F}(x_{u_1}, y_{v_1}), \dots, \mathbf{F}(x_{u_\eta}, y_{v_\eta}) \}. \quad (1.61)$$

The connection cost over the digital path  $\Lambda^{W,\eta}$  can be seen as a measure of dissimilarity between color image pixels at points  $p_0, p_1, \dots, p_\eta$  forming a specific path linking  $p_0$  and  $p_\eta$ , [118, 29, 85]. If a path joining two distinct points  $x, y$ , such that  $\mathbf{F}(x) = \mathbf{F}(y)$  consists of the pixels of the same channel values, then the connection cost should be zero, otherwise  $\Lambda^{W,\eta} > 0$ .

Let us now define a generalized connection cost function, based on the Distance Transform on the Curved Space (DTCOS), [85, 118] introduced by Toivanen for the gray scale images. For two given points  $p_i = (x_{u_i}, y_{v_i})$  and  $p_{i-1} = (x_{u_{i-1}}, y_{v_{i-1}})$ ,  $i = 1, 2, \dots, \eta$ , which are in neighborhood relation, let the generalized distance between the two points be called connection cost defined on the hybrid spatial-color space discussed in [45, 110]:  $\Lambda^{W,1} \{p_{i-1}, p_i\} = \|\mathbf{F}(p_i) - \mathbf{F}(p_{i-1})\| + \xi \cdot \rho^W(p_i, p_{i-1})$ , where  $\xi$  establishes a proper weighting in the hybrid spatial-color space. The connection cost of a whole digital path  $p_0, p_1, \dots, p_\eta$  will be then

$$\Lambda^{W,\eta} \{p_0, p_1, \dots, p_\eta\} = \sum_{i=1}^{\eta} [\|\mathbf{F}(p_i) - \mathbf{F}(p_{i-1})\| + \xi \cdot \rho^W(p_i, p_{i-1})]. \quad (1.62)$$

As we will work with small filtering window, we will focus on the color space only, by setting  $\xi = 0$ .

Similarly to the gray-scale case, we will call the minimal connection cost  $\Gamma^{W,\eta}(x, y)$  of a path of length  $\eta$  linking two points  $x, y \in W$ , the  $\eta$ -geodesic between  $x$  and  $y$ :  $\Gamma^{W,\eta}(x, y) = \min \{ \Lambda(\gamma), \gamma \in \Psi^{W,\eta} \}$ .

In this way the  $\eta$ -geodesic is defined as the path of length  $\eta$ , which gives the minimal connection cost between two points linked by a digital path. If we take the minimum of the connection costs generated by all possible paths joining two points  $x$  and  $y \in W$ , then we obtain the generalized multichannel geodesic distance between these points:  $\Gamma^W(x, y) = \min_{\eta} \{ \Gamma^{W,\eta}(x, y) \} = \min \{ \Lambda(\pi), \pi \in P^{W,\eta}(x, y), \eta \in \mathbb{N} \}$ .  $\Gamma^W(x, y)$  defines the multidimensional distance transform, which is a generalization of DTCOS, [118].

In general, two distinct pixel's locations on the image lattice can be connected by many paths. Moreover, the number of possible geodesic paths of certain length  $\eta$  connecting two distinct points depends on their locations, length of the path and the neighborhood system used, (Fig. 1.7).

### 1.5.2 General Filter Framework

In this work, fuzzy filtering structure proposed in [80, 76, 79] will be used. The general form of the fuzzy adaptive filters presented here is defined as a weighted average of input vectors inside  $W$

$$\mathbf{F}_0^* = \sum_{i=0}^n w_i^* \mathbf{F}_i = \frac{\sum_{i=0}^n w_i \mathbf{F}_i}{\sum_{i=0}^n w_i}. \quad (1.63)$$

The relationship between the pixel under consideration  $\mathbf{F}_0$  and each pixel in the window should be reflected in the decision how to define the filter weights. In our case, the weights will be determined using the similarity functions calculated over digital paths included in the processing window  $W$ .

On the basis of the connection cost function concept, it is possible to define different classes of similarity functions. Choosing a specific form of a similarity function yields different filters of specific properties, which can be applied for a wide range of low-level vision tasks.

### 1.5.3 Digital Paths Approach Filter Class

Let us now define a similarity function, analogous to a membership function used in fuzzy systems, between two pixels connected through all possible digital paths leading from  $x$  to  $y$

$$w^{W,\eta}(x,y) = \sum_{m=1}^{\omega} f\{\Lambda_m^{W,\eta}(x,y)\}, \quad (1.64)$$

where  $\omega$  is the number of all paths connecting  $x$  and  $y$ ,  $\Lambda_m^{W,\eta}(x,y)$  is a dissimilarity value along a specific path  $m$  from the set of all  $\omega$  possible paths leading from  $x$  to  $y$  and  $f(\cdot)$  is a smooth function of  $\Lambda_m^{W,\eta}$ . By definition  $w^{W,\eta}(x,y)$  returns a value evaluated over all possible routes linking the starting point  $x$  with the endpoint  $y$ . The smooth function  $f : (0; \infty) \rightarrow \mathbb{R}$  should satisfy following conditions:  $f$  is a decreasing in  $(0; \infty)$ ,  $f$  is convex in  $(0; \infty)$ ,  $f(0) = 1$ ,  $f(\chi) \rightarrow 0$ , when  $\chi \rightarrow \infty$ . Several functions satisfying the above conditions have been proposed in the literature [83, 103, 74, 53, 106]. However, for the impulsive noise removal good results are obtained using the exponential form of the function  $f$ , [14]. Therefore,

$$w^{W,\eta}(x,y) = \sum_{m=1}^{\omega} \exp[-\beta \cdot \Lambda_m^{W,\eta}(x,y)], \quad (1.65)$$

where  $\beta$  is the filter design parameter.

For  $\eta = 1$  and a square ( $3 \times 3$ ) window  $W$ , the similarity function  $w$  is defined according to (1.62) as  $w^{W,1}(x,y) = \exp\{-\beta\|\mathbf{F}(x) - \mathbf{F}(y)\|\}$ , and then if  $\mathbf{F}(x) = \mathbf{F}(y)$ ,  $\Lambda^{W,n}(x,y) = 0$ ,  $w(x,y) = 1$ , and for  $\|\mathbf{F}(x) - \mathbf{F}(y)\| \rightarrow \infty$  then  $w \rightarrow 0$ , [76].

Figure 1.8 illustrates the calculation of the similarity function between two points connected by two geodesic paths of length  $\eta = 2$ . In this case, the cost functions related to paths  $P_1$  and  $P_2$  are

$$\Lambda_1^{W,2}(x,y) = d_1^1 + d_1^2, \quad \Lambda_2^{W,2}(x,y) = d_2^1 + d_2^2, \quad (1.66)$$

where  $d_1^1$  and  $d_1^2$  are connection costs between neighboring points on the path  $P_1$  defined according to (1.62), while  $d_2^1, d_2^2$  are connection costs defined on path  $P_2$ . The total similarity value can be expressed as

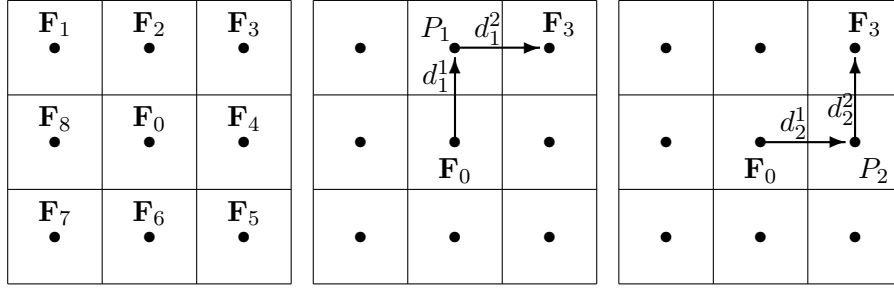


Figure 1.8: Digital paths of length  $n = 2$  connecting points  $\mathbf{F}_0$  and  $\mathbf{F}_3$ .

$$w^{W,2} = \exp\left(-\beta \cdot \Lambda_1^{W,2}\right) + \exp\left(-\beta \cdot \Lambda_2^{W,2}\right). \quad (1.67)$$

A normalized form of the similarity function is defined as

$$w^*(x, y) = \frac{w^{W,n}(x, y)}{\sum_{z \leftrightarrow x} w^{W,n}(x, z)}, \quad (1.68)$$

where  $y \leftrightarrow x$  denotes all points  $y$  connected by digital paths with  $x$  contained in  $W$ .

Assuming that the pixel  $\mathbf{F}_x$  is the pixel under consideration, with  $\mathbf{F}_y$  representing the pixel included in the supporting element  $W$ , which is connected to  $\mathbf{F}_x$  via a digital path, the filter output  $\mathbf{F}_x^*$  is given as

$$\mathbf{F}_x^* = \sum_{y \leftrightarrow x} w^*(x, y) \cdot \mathbf{F}_y, \quad w^*(x, y) = \frac{w^{W,\eta}(x, y)}{\sum_{z \leftrightarrow x} w^{W,\eta}(x, z)}. \quad (1.69)$$

As can be easily noticed,  $\mathbf{F}_x^*$  is the weighted average of all points  $\mathbf{F}_y^*$  connected by digital paths with the pixel  $\mathbf{F}_x^*$ . The pixel  $\mathbf{F}_y$  is the ending point of a path leading from  $x$  and therefore this filter structure is called DPA Last (DPAL), as  $y$  is the last point on the path, (see Fig. 1.6 b).

Another possible filtering scheme takes into account the similarity between the starting point  $x = p_0$  and point  $y = p_1$  crossed by a digital path connecting pixel  $p_0$  and its neighbor  $p_1$  with all possible points  $p_\eta \in W$  which can be reached in  $\eta$  steps from  $p_0$ , (Digital Path Approach First, DPAF).

The aim of taking into account the points  $p_2, \dots, p_\eta$  when calculating the filter output is to explore not only the direct neighborhood of  $p_0$  but also to use the information on the local image structure. This can be done by acquiring the information on the local image features investigating the connection costs of digital paths originating at  $p_0$ , passing  $p_1$  and then visiting successive points, till the path reaches length  $\eta$ . In this case the new similarity function takes the form

$$w^{W,\eta}(x, y) = w^{W,\eta}(p_0, p_1) = \sum_{\{p_2^*, p_3^*, \dots, p_\eta^*\}} f\left(\Lambda^{W,\eta}\{p_0, p_1, p_2^*, p_3^*, \dots, p_\eta^*\}\right), \quad (1.70)$$

where  $\{p_0, p_1, p_2^*, \dots, p_\eta^*\}$  denotes all paths originating at  $x = p_0$  crossing  $y = p_1$  end ending at  $p_\eta^*$ , which are totally included in  $W$ ,  $f(\cdot)$  is a smooth function of  $\Lambda^{W,\eta}$ .

If the exponential function is used, then the similarity function takes the form

$$w^{W,\eta}(x, y) = w^{W,\eta}(p_0, p_1) = \sum_{\{p_2^*, p_3^*, \dots, p_\eta^*\}} \exp[-\beta \cdot \Lambda^{W,\eta} \{p_0, p_1, p_2^*, \dots, p_\eta^*\}], \quad (1.71)$$

where  $\beta$  is the smoothing parameter. A normalized form of the similarity function can be defined as follows

$$w^*(x, y) = w^*(p_0, p_1) = \frac{\sum_{\{p_2^*, p_3^*, \dots, p_\eta^*\}} \exp[-\beta \cdot \Lambda^{W,\eta} \{p_0, p_1, p_2^*, \dots, p_\eta^*\}]}{\sum_{\{p_1^*, p_2^*, \dots, p_\eta^*\}} \exp[-\beta \cdot \Lambda^{W,\eta} \{p_0, p_1^*, p_2^*, \dots, p_\eta^*\}]}, \quad (1.72)$$

where  $\{p_0, p_1, p_2^*, \dots, p_\eta^*\}$  denotes a path joining  $x = p_0$  and  $p_\eta$ , crossing  $y = p_1$ , whereas  $\{p_0, p_1^*, p_2^*, \dots, p_\eta^*\}$  do not necessarily cross  $y = p_1$  when joining  $p_0$  and  $p_\eta$ .

Assuming that the pixel  $\mathbf{F}_x$  at the position  $x = p_0$  is the pixel under consideration, with  $\mathbf{F}_y$  representing the pixel at  $y = p_1$ , the filter output  $\mathbf{F}_x^*$  is given as

$$\mathbf{F}_x^* = \mathbf{F}_{p_0}^* = \sum_{y \leftrightarrow x} w^*(x, y) \cdot \mathbf{F}_y = \sum_{y \sim x} w^*(x, y) \cdot \mathbf{F}_y = \sum_{p_1^* \sim p_0} w^*(p_0, p_1^*) \cdot \mathbf{F}_{p_1^*}, \quad (1.73)$$

and combining this with (1.72) gives

$$\mathbf{F}_x^* = \mathbf{F}_{p_0}^* = \sum_{p_1^* \sim p_0} \frac{\sum_{\{p_2^*, p_3^*, \dots, p_\eta^*\}} \exp[-\beta \cdot \Lambda^{W,\eta} \{p_0, p_1^*, p_2^*, \dots, p_\eta^*\}]}{\sum_{\{p_1^*, p_2^*, \dots, p_\eta^*\}} \exp[-\beta \cdot \Lambda^{W,\eta} \{p_0, p_1^*, p_2^*, \dots, p_\eta^*\}]} \cdot \mathbf{F}_{p_1^*} = \sum_{p_1^* \sim p_0} w^*(p_0, p_1^*) \cdot \mathbf{F}_{p_1^*}. \quad (1.74)$$

Using the notation from Sections 1.3, 1.2 and Eq. (1.20) we can formulate Eq. (1.74) as

$$\mathbf{F}_0^* = \sum_{k=1}^n w_k^* \mathbf{F}_k, \quad (1.75)$$

where  $w_k^*$ , the normalized weighting coefficients, play the role of the generalized conductivity coefficients from Section 1.3 and  $\mathbf{F}_k$  are the neighbors of  $\mathbf{F}_0$ , which is the central pixel in the filter mask  $W$ .

The general form of the anisotropic diffusion scheme based on the digital paths, can be written as

$$\mathbf{F}_0^* = (1 - \lambda^*) \mathbf{F}_0^* + \lambda^* \sum_{k=1}^n w_k^* \mathbf{F}_k, \quad (1.76)$$

or using the iterative notation, as

$$\mathbf{F}_0^{t+1} = (1 - \lambda^*) \mathbf{F}_0^t + \lambda^* \sum_{k=1}^n w_k^* \mathbf{F}_k^t. \quad (1.77)$$

Using the relation  $\lambda^* = \lambda \sum_{k=0}^n c_k$ , (1.20) it is possible to obtain the classical form of the anisotropic diffusion scheme defined by Eq. (1.15).

Figure 1.9 shows the dependence of PSNR on the  $\lambda^*$  and  $K$  values for color *LENA* image contaminated by impulsive and mixed noise for the classic multichannel anisotropic diffusion scheme (AD) and the new DPAF filter defined by Eq. (1.75). Especially interesting is the behavior of the plots as a function of  $\lambda^*$ . As can be seen, for images contaminated by a noise process of high intensity, the maximum of PSNR is obtained for  $\lambda^*$  very close to 1, which means that it is favorable to omit the central pixel while calculating the weighted average in Eq. (1.20) and also in (1.75). This was already noticed in the scheme of Smith, [102], (Eq. 1.8), who did not take the central pixel into the averaging process, which is equivalent to setting  $\lambda^* = 1$ . That is why we set  $\lambda^* = 1$  in Eq. (1.75) to define the new DPAF filter, (1.74), (1.75).

The superiority of this approach over the classic scheme is clearly seen in Fig. 1.9, where especially for highly corrupted images, the difference in terms of PSNR is quite significant, (see also Tab. 1.4 and 1.5).

In a similar way the DPAL filter can be defined as

$$\mathbf{F}_x^* = \mathbf{F}_{p_0}^* = \frac{\sum_{\{p_1^*, p_2^*, p_3^*, \dots, p_\eta^*\}} \exp[-\beta \cdot \Lambda^{W, \eta} \{p_0, p_1^*, p_2^*, \dots, p_\eta^*\}] \cdot \mathbf{F}_{p_\eta^*}}{\sum_{\{p_1^*, p_2^*, \dots, p_\eta^*\}} \exp[-\beta \cdot \Lambda^{W, \eta} \{p_0, p_1^*, p_2^*, \dots, p_\eta^*\}]} = \sum_{p_\eta^*} w^*(p_0, p_\eta^*) \cdot \mathbf{F}_{p_\eta^*}, \quad (1.78)$$

which can be written as

$$\mathbf{F}_0^* = \sum_{k=1}^N w_k^* \mathbf{F}_k, \quad (1.79)$$

where  $N$  denotes the number of pixels surrounding  $\mathbf{F}_0$  in the filtering window. Analogously to (1.76), we can introduce the general form of DPAL defined by (1.78)

$$\mathbf{F}_0^* = (1 - \lambda^*) \mathbf{F}_0^* + \lambda^* \sum_{k=1}^N w_k^* \mathbf{F}_k, \quad (1.80)$$

and its iterative version

$$\mathbf{F}_0^{t+1} = (1 - \lambda^*) \mathbf{F}_0^t + \lambda^* \sum_{k=1}^N w_k^* \mathbf{F}_k^t, \quad (1.81)$$

where  $w_k^*$  are the normalized weighting coefficients from (1.78).

The concept of the DPAF and DPAL filters is presented in Fig. 1.6. The weights assigned to the pixels surrounding the central pixel  $\mathbf{F}_0$  are determined in different ways. In the DPAF approach, the weights in (1.74) are calculated exploring all digital paths starting from the central pixel and crossing its neighbors, (Fig. 1.6 a) and then a weighted average of the nearest neighbors of the central pixel is calculated, (1.75).

In the DPAL approach, the weights are obtained by exploring all digital paths leading from the central pixel to any of the pixel in the filtering window, (Fig. 1.6 b) and then a weighted average of all pixels contained in that window is calculated, (1.81).

Although, both schemes work on supporting windows of the same size, determined by the number of steps  $\eta$  and the kind of neighborhood relation  $\sim$ , the DPAL has more powerful smoothing properties, as it



involves all the  $N$  pixels from the filtering window into the averaging process, whereas the DPAF determines the weighted output using only its nearest neighbors. The efficiency of the new class of filters DPAF and DPAL is evaluated and compared with some of the standard filtering techniques in Section 1.6.

The computational complexity of the DPA filters depends on the path length  $\eta$  and the number of paths, which can be constructed in the supporting window  $W$  of size  $(k \times k)$ . It is not hard to see that for large  $k$ , which may be required in certain applications, the computational complexity of the filters makes them inapplicable. To decrease the computational burden, another filter structure is introduced. In the Fast Digital Paths Approach (FDPA), the size of the supporting window  $W$  is set to  $(3 \times 3)$  independently of the digital paths length  $\eta$ .

It is possible to construct both the fast DPAF and fast DPAL filters, however their properties are quite similar and therefore only the filtering approach based on DPAL, (denoted as FDPA) will be investigated. Using the FDPA formulation a number of interesting properties of the proposed filtering structure can be observed. For example, let us assume that parameter  $\beta$  used in (1.65), is very small,  $\beta \rightarrow 0$ . Then the weights in (1.69) reduce to  $w^*(x, y) = \omega(x, y) / \Omega$ , where  $\omega(x, y)$  is the number of digital paths of length  $\eta$  connecting points  $x$  and  $y$ , and  $\Omega$  denotes the number of all possible digital paths starting from  $x$ , which are totally included in  $W$ .

The examination of the convolution masks obtained in this way, reveals their similarity to the masks obtained through Gaussian kernels. Therefore, the FDPA can be viewed as a non-linear generalization of the Gaussian kernel based schemes, which are widely used in many image processing tasks.

The parameter  $\beta$  in (1.65, 1.71) regulates the smoothness of the similarity function. Since the filtering structure of (1.63) is a regression estimator, which enables a smooth interpolation among the observed, noise-corrupted image vectors, the parameter  $\beta$  provides the required balance between smoothing and detail preservation. Therefore, it is not surprising that the best results are obtained when the smoothing operators  $\mathbf{F}^*$  in (1.69) and (1.73) are applied in an iterative way. Starting with a low value of  $\beta$  enables the smoothing of the image noise components. At each iteration step the parameter  $\beta$  can be increased, following a scheme similar to that used in simulated annealing applications. In particular,  $\beta$  can be increased exponentially:  $\beta(\kappa) = \beta(\kappa - 1) \cdot \alpha$ ,  $\kappa \in \mathbb{N}$ , where  $\kappa$  is the iteration number and  $\alpha$  is a design parameter. The increasing of the  $\beta$  parameter causes that after a few iterations no further changes are introduced to the image, as for high  $\beta$  the filter output is that pixel, which lies on the geodesic digital path in the color space. The influence of  $\alpha$  on the performance of the DPAL and FDPA filters is shown in Fig. 1.11. The value of  $\alpha$  is not critical for the efficiency of the new filter class, and taking  $\alpha$  from the interval  $[1, 2]$  guarantees fast filter convergence and good filtering results.

### 1.5.4 Computational Complexity and Fast Filter Design

Apart from the numerical behavior of any proposed algorithm, its computational complexity is a realistic measure of its practicality and usefulness, since it determines the required computing power and processing, (execution) time. A general framework to evaluate the computational requirements of image filtering algorithms based on fixed processing window is given in [13] and [84]. The requirement of this approach is that the filter window  $W$  is symmetric ( $k \times k$ ) and contains  $k^2$  vector samples of dimension  $L$ . In most image processing applications a value  $k = 3$  is considered, while for color RGB images  $L = 3$ .

The computational complexity of a specific filter is given in terms of the total execution time needed for a complete filtering cycle. The total time is calculated as:  $TIME = \sum w_{OPER} \cdot OPER$ , where  $OPER$  is the number of particular operations required for a complete cycle, and  $w_{OPER}$  is the relative weight of this operation. In the following analysis the following operations are used: *ADDS* (additions), *MULTS* (multiplications), *DIVS* (divisions), *SQRTS* (square roots), *COMPS* (comparisons), *ARCCOS* (arc cosines) and *EXPS* (exponents). Mostly  $w_{ADDS}$  is assumed to be 1, while other  $w_{OPER}$  values depend on the computing platform. The determination of the weights of different operations is beyond the scope of this work.

Since the structure of the new filters is not based on fixed window, the methodology presented in [13] and [84] cannot be directly applied to evaluate the new filter class complexity. The computational burden of the proposed filters depends mostly on the number of possible digital paths, which in turn depends on the path length. For a given path of length  $\eta$ , the number of simple paths  $\Omega$  can be easily computed. Table 1.1 depicts the number of possible paths corresponding to the DPA and FDPA filters, [115, 116, 106, 104].

Table 1.1: Number of possible simple digital paths  $\Omega$  in dependence on path length  $\eta$ .

$n$	1	2	3	4
DPA	8	56	368	2336
FDPA	8	24	56	69

The complexity of the DPA and FDPA filters can be determined as follows, [115, 116]

1. Filtering of 1 pixel requires computation of all weights  $w^*$  (see point 2),  $L(\Omega - 1)$  additions and  $L \cdot \Omega$  multiplications.
2. Computation of all weights  $w^*$  requires computation of all similarity functions  $w^{W,\eta}$  (see point 3),  $\Omega$  divisions and  $(\Omega - 1)$  additions.
3. Computation of all similarity functions  $w^{W,\eta}$  requires  $\Omega$  computations of distance  $\Lambda_m^{W,\eta}$  (see point 4),  $(\Omega - 1)$  additions,  $\Omega$  multiplications and  $\Omega$  computations of an exponent.
4. Computation of one distance  $\Lambda_m^{W,\eta}$  along path  $m$  requires  $n$  computations of Euclidean distance (if the  $L_2$  metric is used) and  $(\eta - 1)$  additions.
5. Computation of one particular Euclidean distance requires  $L$  multiplications,  $2L$  additions and 1 square root.

Thus the total number of operations needed to implement the filters is

$$(2\eta L\Omega + \Omega p + L\Omega - L - 2) \cdot ADDS + (\Omega + L\Omega + 2\eta) \cdot MULTS + \Omega \cdot DIVS + \Omega\eta \cdot SQRTS + \Omega \cdot EXPS. \quad (1.82)$$

Using the framework of [13] and assuming that the size of the processing window is  $(k \times k)$ , the computational complexity for the VMF, BVDF and DDF can be evaluated, (Tab.1.2).

It should be emphasized at this point that the computational complexity analysis of the new filter was based on straightforward application of the described algorithms without any consideration of a particular implementation. However, it is possible to significantly reduce the computational complexity of the proposed filters. To illustrate this, the FDPA filter is considered. The analysis of the filtering structure reveals that the  $L_2$  distance should be evaluated  $\eta$  times for each path of length  $\eta$ . If the total number of paths in the supporting window is  $\Omega$ , the number of  $L_2$  norm evaluations is  $(\Omega \cdot \eta)$ . However, most of these calculations are unnecessary, since values already computed for other paths can be used. For example in a  $(3 \times 3)$  window there are only 20 possible distances to be calculated. These values can be computed and stored in order to be used to determine the path related weights for a neighboring pixel. Furthermore, other techniques used to improve the performance of the VMF presented in [13] can be applied in the DPA or FDPA filter design.

Table 1.2 summarizes the total number of operation for different filter, with  $DPA_\eta$  denoting the basic DPA filter of length  $\eta$ ,  $FDPA_\eta$  denoting straightforward application of FDPA algorithms and  $FDPA_\eta^*$  the optimized version of FDPA. As can be seen the fast implementation of the proposed filter is computationally more attractive than the VMF and it significantly outperforms filters based on angular distances.

Table 1.2: Number of elementary operations for a complete processing cycle.

FILTER	ADDS	MULTS	DIVS	SQRTS	EXPS	COMPS	ARCCOS	TOTAL
DPA <sub>2</sub>	947	228	56	112	56	—	—	1399
DPA <sub>3</sub>	8827	1478	368	1104	368	—	—	12145
FDPA <sub>2</sub>	403	100	24	48	24	—	—	599
FDPA <sub>3</sub>	1139	230	56	168	56	—	—	1649
FDPA <sub>2</sub> *	169	22	24	9	24	—	—	<b>248</b>
FDPA <sub>3</sub> *	721	24	56	9	56	—	—	866
VMF <sub>3×3</sub>	186	63	—	21	—	8	—	<b>278</b>
VMF <sub>5×5</sub>	855	330	—	110	—	24	—	1319
BVDF <sub>3×3</sub>	375	210	21	21	—	8	21	656
BVDF <sub>5×5</sub>	1970	1100	110	110	—	24	110	3424
DDF <sub>3×3</sub>	540	282	21	42	—	8	21	914
DDF <sub>5×5</sub>	2785	1455	110	220	—	24	110	4704

Table 1.3: Filters taken for comparison with the proposed noise reduction techniques.

Notation	METHOD	REF.
AMF	Arithmetic Mean Filter	[83]
VMF	Vector Median Filter	[9]
BVDF	Basic Vector Directional Filter	[125, 123]
GVDF	Generalized Vector Directional Filter	[122]
DDF	Directional-Distance Filter	[42]
HDF	Hybrid Directional Filter	[36]
AHDF	Adaptive Hybrid Directional Filter	[36]
FVDF	Fuzzy Vector Directional Filter	[76]
ANNF	Adaptive Nearest Neighbor Filter	[81, 82]
ANP-E	Adaptive Non Parametric (Exponential) Filter	[80, 83]
ANP-G	Adaptive Non Parametric (Gaussian) Filter	[80, 83]
ANP-D	Adaptive Non Parametric (Directional) Filter	[80, 83]
VBAMMF	Vector Bayesian Adaptive Median/Mean Filter	[80, 83]
AD	Perona-Malik Anisotropic Diffusion Filter with $c_1$	[68, 67]
GD-PDE	Geometric Diffusion PDE	[126]

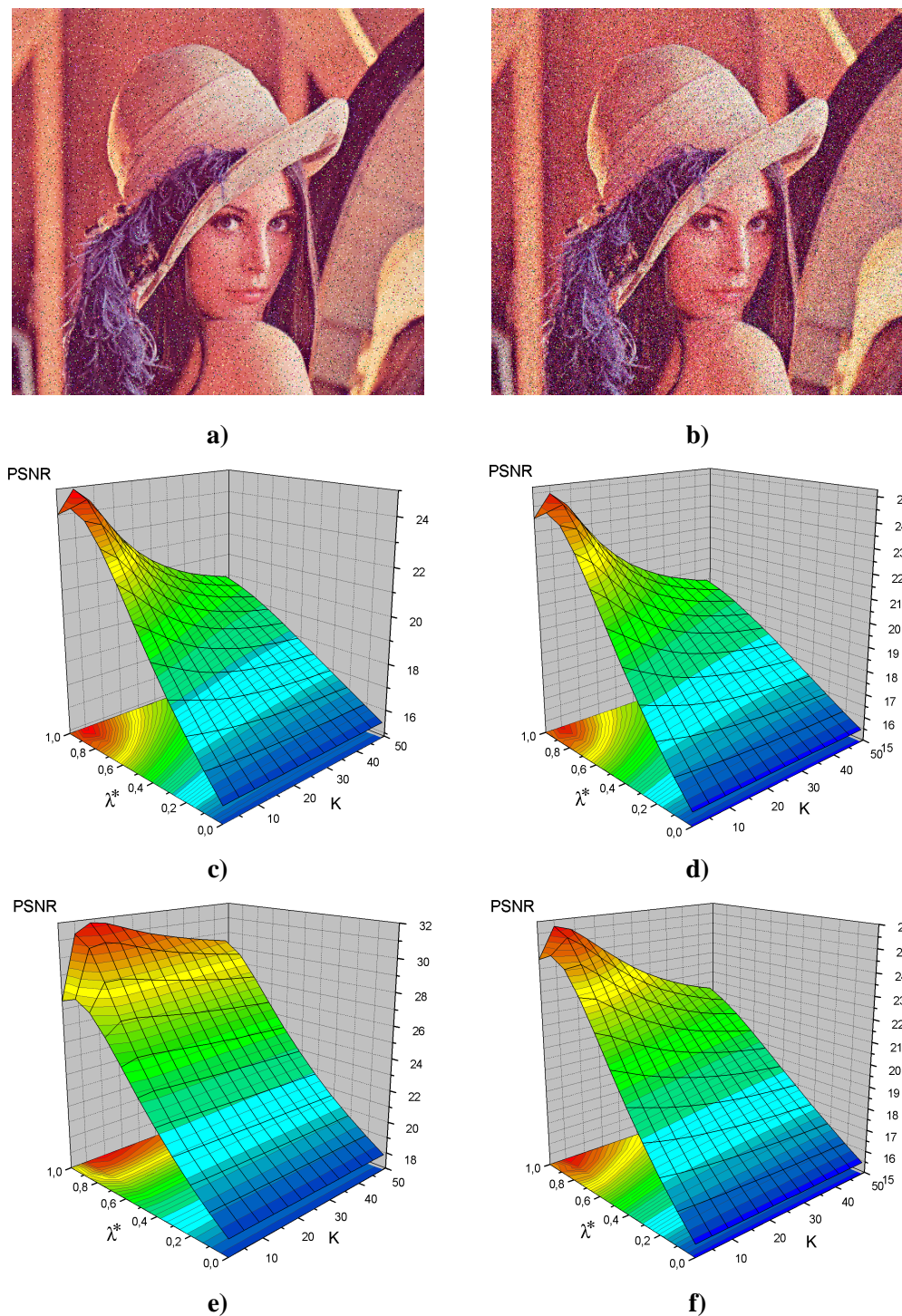


Figure 1.9: Dependence of the efficiency of the P-M anisotropic diffusion filter and the DPAF on the  $\lambda^*$  parameter: **a)** color image LENA contaminated with impulsive noise, ( $p = 0.12, p_1 = p_2 = p_3 = 0.3$ ), **b)** test image corrupted by mixed noise, ( $\sigma = 30, p = 0.12, p_1 = p_2 = p_3 = 0.3$ ), **c)** and **d)** results obtained with the P-M anisotropic diffusion filter, **e)** and **f)** results obtained with the DPAF, ( $\eta = 2$ ). As expected the maximum of PSNR is achieved for  $\lambda^*$  close to 1.

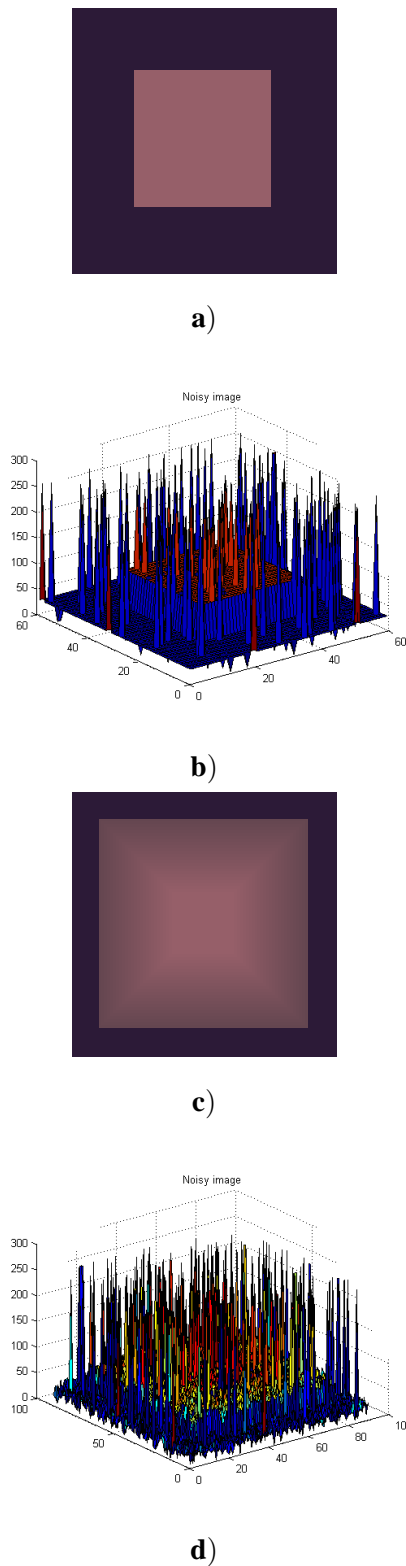


Figure 1.10: **a)** Test image *SQUARE*, **b)** *SQUARE* image corrupted by impulsive noise, (green channel), **c)** test image *PYRAMID*, **d)** *PYRAMID* image corrupted by mixed impulsive and Gaussian noise, (green channel).

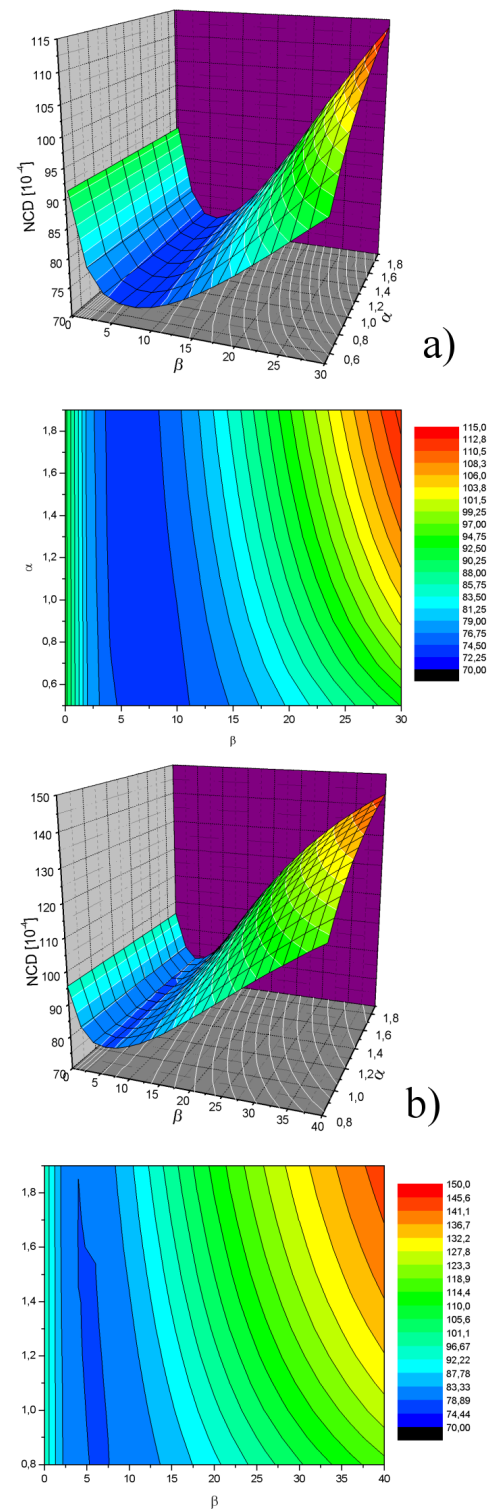


Figure 1.11: Efficiency of the **a)** DPAL and **b)** FDPA in terms of NCD and their dependence on  $\alpha$  and  $\beta$  for *LENA* image corrupted by impulsive ( $p = 0.12$ ,  $p_1 = p_2 = p_3 = 0.3$ ) and Gaussian noise ( $\sigma = 30$ ), ( $n = 2$ , third iteration).

## 1.6 Efficiency of the New Filter Class

In this section the performance of the new filter class is evaluated comparing the results with some of the noise reduction techniques listed in Tab. 1.3 using synthetic and natural color images corrupted by Gaussian and mixed Gaussian and impulsive noise.

### 1.6.1 Simulations Performed on Artificial Images

The use of nonlinear filters in color image processing is motivated primarily by the good performance of the filters near edges and other sharp signal transitions. Edges are basic images features which carry valuable information, useful in image analysis and object classification. Therefore, any nonlinear noise reduction operator is required to preserve edges and smooth out noise without altering sharp signal transitions.

In this section some examples of the efficiency of the new filter class are presented in order to illustrate its excellent noise reduction properties. To quantitatively evaluate the behavior of the proposed algorithms, two color synthetic images were prepared. To examine the performance of the new filters in case of an artificial step edge, a three-channel image called *SQUARE* of size  $(60 \times 60)$  containing a square of size  $(30 \times 30)$  was generated, (Fig. 1.10 a). Further, for the evaluation of the filter performance in case of a ramp edge, a synthetic test image called *PYRAMID* was constructed. The three-channel image of size  $(90 \times 90)$  contains a top-cut pyramid, which is used to simulate a "ramp-edge" scenario, (Fig. 1.10 c).

The test image *SQUARE* was corrupted by multivariate impulsive noise following the model given by Eq. (1.1.1) in Section 1.1 with the degree of contamination  $p = 0.1$  and  $p_1 = p_2 = p_3 = 0.25$ , (Fig. 1.10 b). The test image *PYRAMID* was corrupted by mixed impulsive noise with  $p = 0.1$  and  $p_1 = p_2 = p_3 = 0.25$  and  $\sigma = 20$ , (Fig. 1.10 d).

The new techniques based on the Digital Paths Approach (DPAF, DPAL), and the Fast Digital Paths Approach (FDPA) algorithms were compared in terms of objective quality criteria with the Vector Median Filter (VMF), with the Arithmetic Mean Filter (AMF), with the classic Perona-Malik anisotropic diffusion (AD) and other filtering techniques listed in Tab. 1.3.

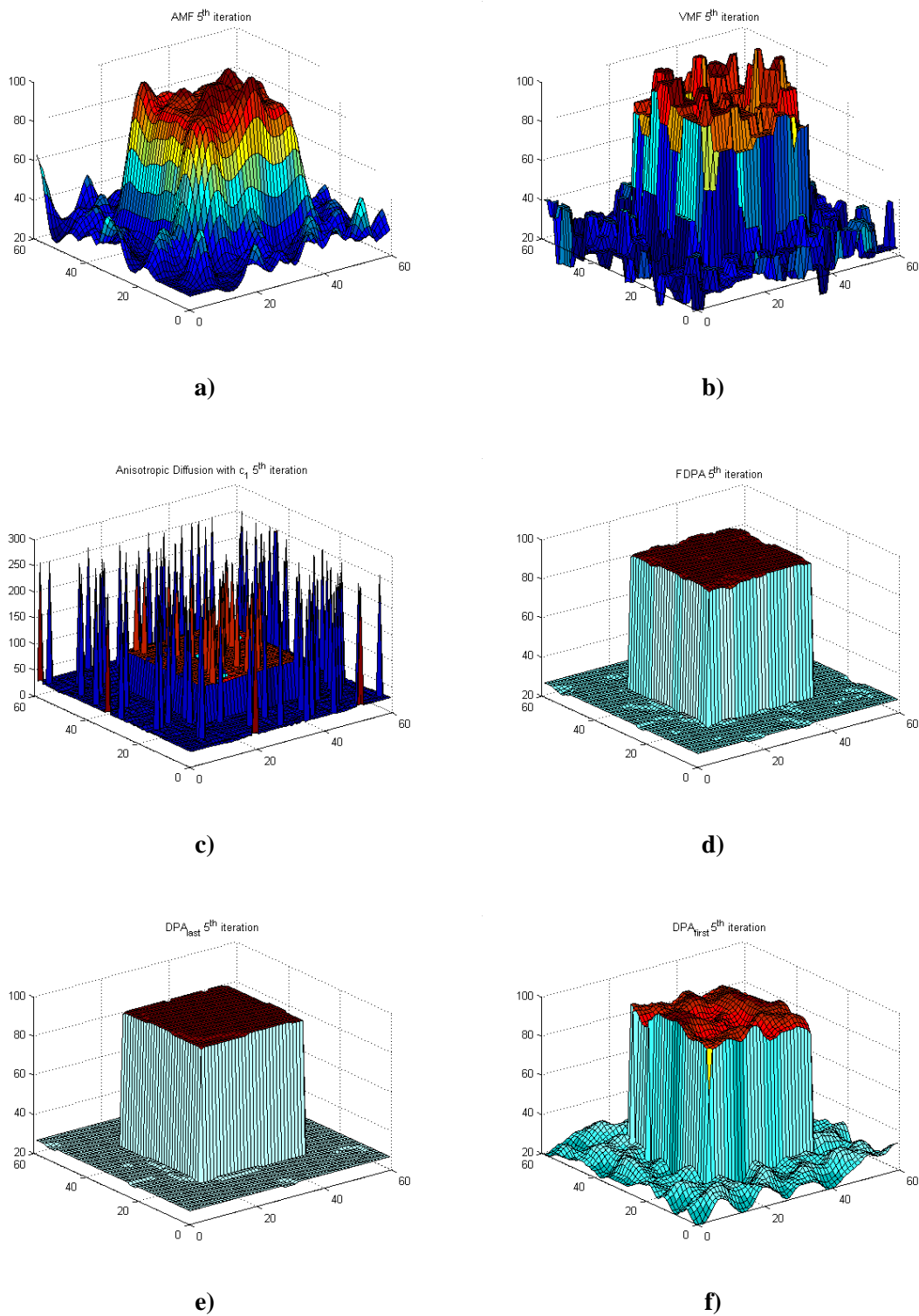


Figure 1.12: Three-dimensional representation of the results of noise attenuation in the green channel of the *SQUARE* image corrupted by impulsive noise, using the standard and new techniques: **a)** AMF, **b)** VMF, **c)** AD, **d)** FDPA, **e)** DPAL and **e)** DPAF, (five iterations,  $\eta = 2$ ).



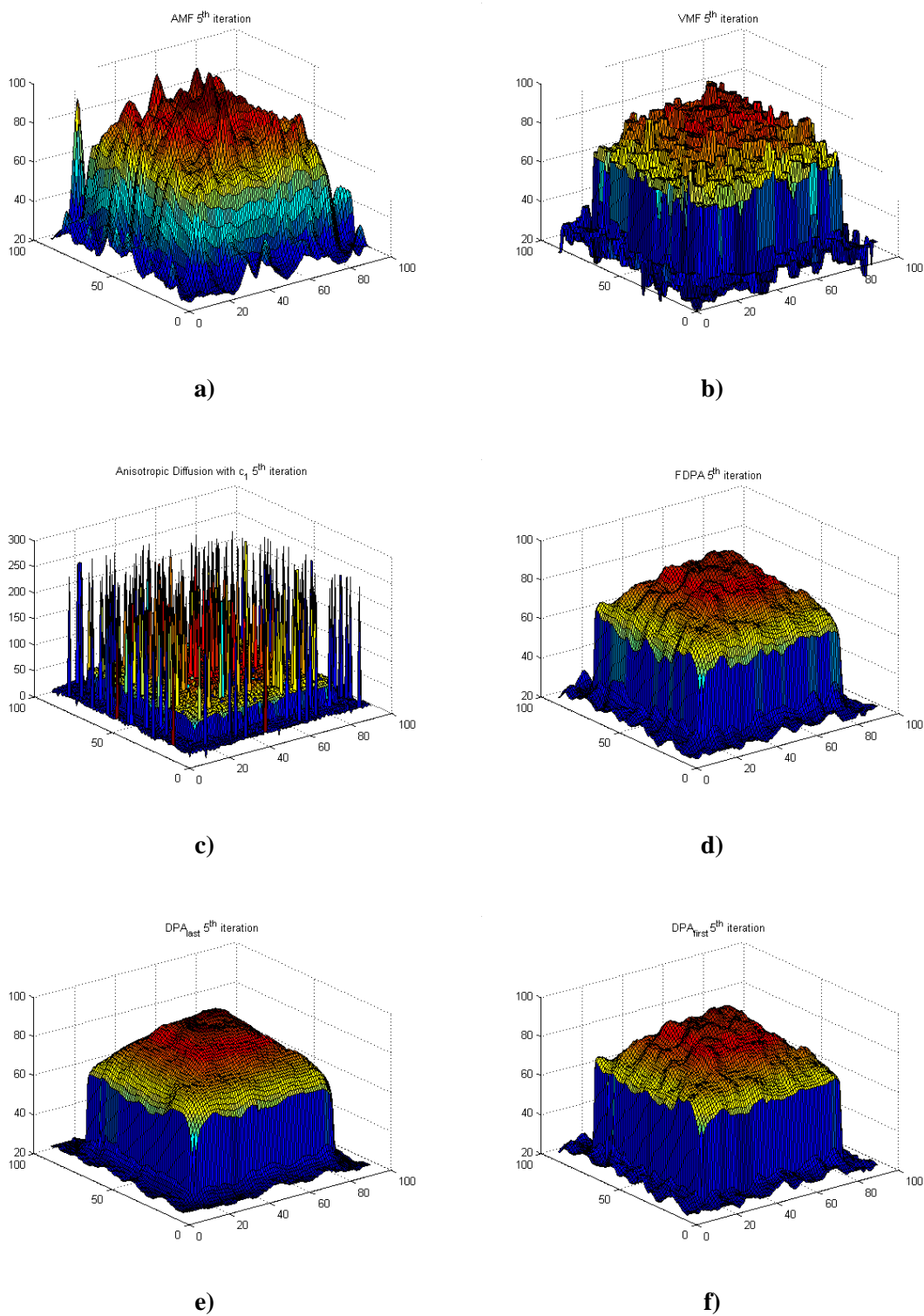


Figure 1.13: Three-dimensional representation of the results of noise attenuation in the the green channel of the *PYRAMID* image corrupted by mixed Gaussian and impulsive noise using the standard and new techniques: **a)** AMF, **b)** VMF, **c)** AD, **d)** FDPA, **e)** DPAL and **e)** DPAF, (five iterations,  $\eta = 2$ ).

Table 1.4: Comparison of the efficiency of the new algorithms with different techniques, (Tab. 1.3) using the *LENA* standard color image corrupted by Gaussian noise of  $\sigma = 30$ .

FILTER	NMSE [ $10^{-3}$ ]	RMSE	SNR [dB]	PSNR [dB]	NCD [ $10^{-4}$ ]
NONE	420.55	29.075	13.762	18.860	250.090
AMF	66.452	11.558	21.775	26.873	95.347
VMF	87.314	13.248	20.589	25.688	117.170
BVDF	279.54	23.705	15.536	20.634	117.400
GVDF	76.713	12.418	21.151	26.250	84.876
DDF	100.50	14.213	19.979	25.077	108.960
HDF	66.584	11.569	21.766	26.865	92.769
AHDF	60.166	10.997	22.206	27.305	91.369
FVDF	57.466	10.748	22.406	27.504	77.111
ANNF	63.341	11.284	21.983	27.082	82.587
ANP-E	60.396	11.018	22.190	27.288	76.896
ANP-G	60.443	11.023	22.187	27.285	76.890
ANP-D	58.389	10.834	22.337	27.435	78.486
AD	41.434	9.126	23.826	28.925	69.482
GD-PDE	34.530	8.296	24.618	29.753	72.100
DPAF	42.873	9.244	23.678	28.813	82.814
DPAL	43.005	9.258	23.665	28.800	77.932
FDPA	44.913	9.462	23.476	28.611	84.918

Table 1.5: Comparison of the new algorithms with the techniques from (Tab. 1.3) using the *LENA* color image corrupted by mixed Gaussian and impulsive noise, ( $\sigma = 30, p = 0.12, p_1 = p_2 = p_3 = 0.25$ ).

FILTER	NMSE [ $10^{-3}$ ]	RMSE [dB]	SNR	PSNR [dB]	NCD [ $10^{-4}$ ]
NONE	905.93	42.674	10.429	15.528	305.55
AMF	97.444	13.996	20.112	25.211	95.80
VMF	96.464	13.925	20.156	25.255	121.79
BVDF	336.46	26.006	14.731	19.829	123.93
GVDF	91.118	13.534	20.404	25.503	89.277
DDF	110.62	14.912	19.561	24.660	113.39
HDF	74.487	12.236	21.279	26.378	97.596
AHDF	68.563	11.740	21.639	26.738	96.327
FVDF	108.76	14.786	19.635	24.734	111.22
ANNF	75.652	12.332	21.212	26.310	86.836
ANP-E	90.509	13.488	20.433	25.532	97.621
ANP-G	90.523	13.489	20.432	25.531	97.603
ANP-D	74.203	12.213	21.296	26.394	85.026
AD	339.55	26.125	14.691	19.790	113.65
GD-PDE	59.371	10.924	22.264	27.363	77.510
DPAF	50.804	10.106	22.941	28.040	76.076
DPAL	49.999	10.025	23.010	28.109	72.851
FDPA	53.573	10.377	22.711	27.809	78.666

In the DPAF, DPAL and FDPA filters, the paths of length  $\eta = 2$  with design parameters set at  $\beta = 20$  and  $\alpha = 1.2$  were used. The AMF and VMF operated on a filtering window of size  $(3 \times 3)$ . Anisotropic diffusion filter used in the experiments denoted as AD is a vector implementation of the Perona-Malik anisotropic diffusion, which utilizes the conductivity function  $c_1$  (1.14), [67, 37]. For the AD filter the parameters which gave the best results in terms of PSNR were used.

It should be pointed out that the parameters used for the FDPA, DPAF and DPAL filters were not optimal and in majority of cases better results can be obtained for images corrupted by a specific noise process. However in practical situations the optimal values of the design filter parameters are generally unknown and therefore the experimental values of these parameters were used.

In case of images corrupted with Gaussian noise the AMF as expected gave better results than the VMF, especially in the flat homogeneous regions, but it blurred heavily the image edges. Classical P-M anisotropic diffusion gives good results for images corrupted with Gaussian noise of low intensity, but it requires many iterations to smooth the image till its performance can be comparable with the new filter class in terms of objective quality criteria. In case of images distorted by Gaussian noise process with high  $\sigma$ , the PM approach is not able to suppress the spikes, which leads to a poor overall performance.

The experimentations with images corrupted by mixed Gaussian and impulsive noise revealed as expected that the AMF filter introduces extensive smoothing into the image and impulses are still visible as blurred 'bumps'. Anisotropic diffusion with parameters used in the experiments does not blur the image edges but it leaves impulses almost unchanged, (of course when we increase the threshold parameter  $K$  in (1.14) we can smooth the noise out but then the AD will also destroy the image edges). The VMF efficiently reduces the noise component but tends to blur the edges and produces color blotches in flat image regions. The results obtained using the DPAF, DPAL and FDPA filters confirm their excellent properties in case of images corrupted by both impulsive and Gaussian noise.

The new filtering structure gives excellent results both in flat regions and also at the edges, (see Figs. 1.12, 1.13 and also 1.16). The results obtained with anisotropic diffusion and with filters proposed in this work are quite similar in case of images corrupted by low intensity Gaussian noise. Both the schemes provide efficient smoothing in homogeneous image regions and achieve excellent edge preservation. However, the new filters achieve its goal much faster and work efficiently even when the intensity of the Gaussian noise is high, (Fig. 1.15). For images corrupted with mixed Gaussian and impulsive noise neither the VMF nor AMF provide acceptable results. While anisotropic diffusion filter smoothes out only the Gaussian noise component and AMF introduces blurring, the DPAF, DPAL and FDPA filters performance is excellent. The new filters remove outliers introduced by impulsive noise, and smooth flat noisy regions leaving the edges of the objects almost unchanged. The simulations performed on the synthetic images revealed that:

- The VMF performs poorly in the presence of Gaussian noise.

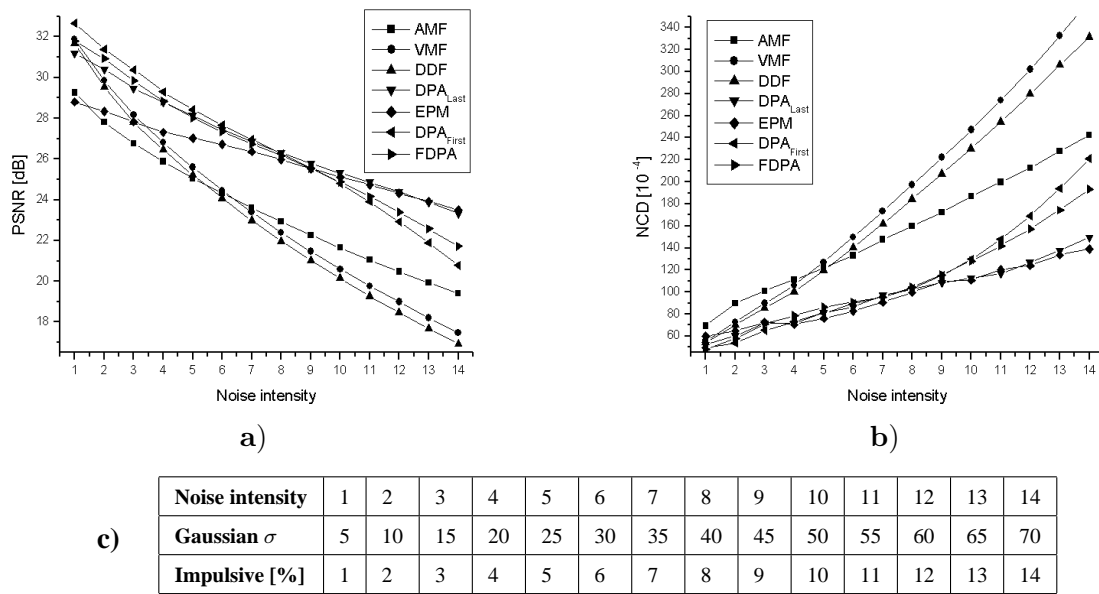


Figure 1.14: Comparison of the efficiency of the standard filters efficiency with the new filter class in terms of **a)** PSNR and **b)** NCD for different amounts of noise, (mixed Gaussian and impulsive noise intensities,  $p = 0.01 - 0.12, p_1 = p_2 = p_3 = 0.3$ ), **c)**. EPM denotes a path in which with every step the distance between the current point and the origin is increasing, (*Escaping Particle Model*).

- The AMF works well in homogeneous regions with additive Gaussian noise.
- Classical Perona-Malik anisotropic diffusion (AD) scheme performs well in images corrupted by low intensity Gaussian noise, but fails in the presence of impulsive noise.
- The proposed filtering class is able to suppress Gaussian as well as mixed Gaussian and impulsive noise in homogeneous regions and also near edges. The obtained results confirm the much better performance of the new filters when compared to the AMF, VMF and P-M AD scheme.

### 1.6.2 Filter Performance for Natural Color Images

The noise attenuation properties of different filters were examined using the color test image *LENA*, which has been contaminated by Gaussian and mixed Gaussian and impulsive noise in order to compare the new filters with the filtering techniques listed in Tab. 1.3. The test images were contaminated by additive Gaussian noise of  $\sigma = 30$  and also by mixed impulsive ( $p = 0.12, p_1 = p_2 = p_3 = 0.3$ ) and Gaussian noise of  $\sigma = 30$ . As the results for *LENA* and *PEPPERS* are consistent, only the results obtained with *LENA* image will be reported.

The *Root Mean Squared Error*, (RMSE), *Signal to Noise Ratio*, (SNR), *Peak Signal to Noise Ratio*, (PSNR), *Normalized Mean Square Error*, (NMSE) and the *Normalized Color Difference*, (NCD) [83] were

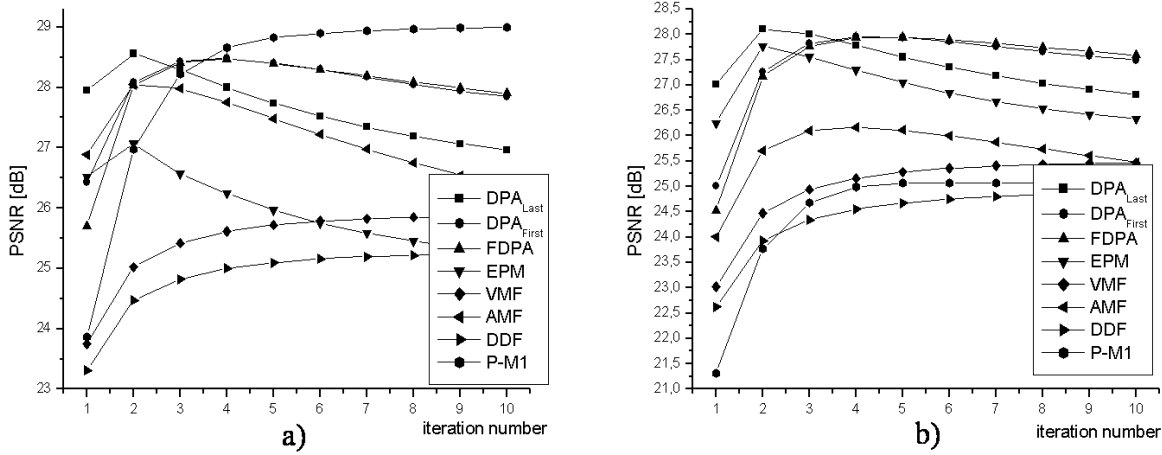


Figure 1.15: Plots of PSNR in subsequent iterations for various filters applied to color LENA image contaminated with Gaussian,  $\sigma = 30$ ) a) and mixed impulsive and Gaussian noise,  $\sigma = 30$ ,  $p = 0.12$ ,  $p_1 = p_2 = p_3 = 0.3$ ) b).

used for the analysis. The objective quality measures are defined by the following formulas

$$RMSE = \sqrt{\frac{1}{NML} \sum_{i=1}^N \sum_{j=1}^M \sum_{l=1}^L (F^l(i, j) - \hat{F}^l(i, j))^2}, \quad NMSE = \frac{\sum_{i=1}^N \sum_{j=1}^M \sum_{l=1}^L (F^l(i, j) - \hat{F}^l(i, j))^2}{\sum_{i=1}^N \sum_{j=1}^M \sum_{l=1}^L F^l(i, j)^2}, \quad (1.83)$$

$$SNR = 10 \log \left[ \frac{\sum_{i=1}^N \sum_{j=1}^M \sum_{l=1}^L F^l(i, j)^2}{\sum_{i=1}^N \sum_{j=1}^M \sum_{l=1}^L (F^l(i, j) - \hat{F}^l(i, j))^2} \right], \quad PSNR = 20 \log \left( \frac{255}{RMSE} \right), \quad (1.84)$$

where  $M, N$  are the image dimensions, and  $F^l(i, j)$  and  $\hat{F}^l(i, j)$  denote the  $l^{\text{th}}$  component of the original image vector and its estimation at pixel position  $(i, j)$ , respectively. The NCD perceptual measure is evaluated over the uniform  $L^*u^*v^*$  color space. The difference measure NCD is defined as

$$NCD = \frac{\sum_{i=1}^N \sum_{j=1}^M \Delta E}{\sum_{i=1}^N \sum_{j=1}^M E^*}, \quad \Delta E = [(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2]^{\frac{1}{2}}, \quad E^* = [(L^*)^2 + (u^*)^2 + (v^*)^2]^{\frac{1}{2}}, \quad (1.85)$$

where  $\Delta E$  is the perceptual color error and  $E^*$  is the *norm* or *magnitude* of the uncorrupted original color image pixel in the  $L^*u^*v^*$  space.

Results obtained using the new filtering techniques are compared with the filtering algorithms from Tab. 1.3 in Tab. 1.4 and Tab. 1.5. For the denoising of both contaminated *LENA* images with the new filtering techniques, predefined parameter values were used: path length  $\eta = 2$ ,  $\beta = 13$ ,  $\alpha = 1.2$ . For all evaluated filters 10 iterations were performed and the best result in terms of PSNR is presented in Tabs. 1.4, 1.5.

Figure 1.11 depicts the efficiency of the proposed algorithms, (DPAL and FDPA) in terms of NCD quality measure, as a function of the design parameters  $\alpha$  and  $\beta$ . It can be easily noticed that both algorithms yield comparable results with a flat minimum of NCD, which ensures their robustness to optimal parameter settings. The parameter  $\alpha$  ensures quick convergence of the proposed filters to a stable state and as can be seen in Fig. 1.11 good results can be obtained for any  $\alpha$  in the range  $[1, 2]$ .

Figure 1.17 presents the efficiency of the DPAL filter applied to a scanned road map. The new filtering technique was able to remove the raster structure, while image details such as roads, names etc. were preserved and even enhanced. The VMF gives much worse results, raster texture is still visible and image details are blurred.

Tables 1.4 and 1.5 indicate that the new filters yields especially good results in case of images corrupted by mixed Gaussian and impulsive noise. In addition to excellent noise attenuation properties, the new filters restore the noisy images so that they have well preserved, and even enhanced edges and corners, which make them interesting for many different computer vision applications, (Fig. 1.16).

The best results for the Gaussian and mixed noise attenuation, for the majority of existing filters were obtained after many iterations, while for filters based on the digital paths concept the best results were achieved in the second or third iteration, (see Fig. 1.15).

The comparison of the new filters efficiency with some of the standard filters is presented in Fig. 1.14, where for different filters, the PSNR and NCD dependence on the amount of mixed impulsive and Gaussian noise is shown. As the intensity of the noise increases, the quantitative results obtained using the new filters become significantly better than those obtained by the standard filters, (AMF, VMF, DDF).

The simulations revealed that in the case of both Gaussian and mixed Gaussian and impulsive noise very good results were obtained using the method GP-PDE, presented in [126, 127], which is based on the gradient norm described in Section 1.3.1. The visual comparison between the FDPA and the algorithm GP-PDE [126, 127] is shown in Fig. 1.18.

In conclusion, from the results listed in the Tables and shown in the Figures, it can be easily seen that the new filters, especially the FDPA filter, provide consistently good results. The DPAF, DPAL and FDPA filters can be seen as universal filters able to attenuate different types of noise, while preserving image edges and corners. Simulation results show that the new class of filters yield favorable noise reduction results for various kinds of color images in comparison with the standard adaptive smoothing algorithms.

*The contribution of Rachid Deriche and David Tschumperle who evaluated the GP-PDE algorithm, [126, 127] on a set of noisy images used in this work is gratefully acknowledged.*

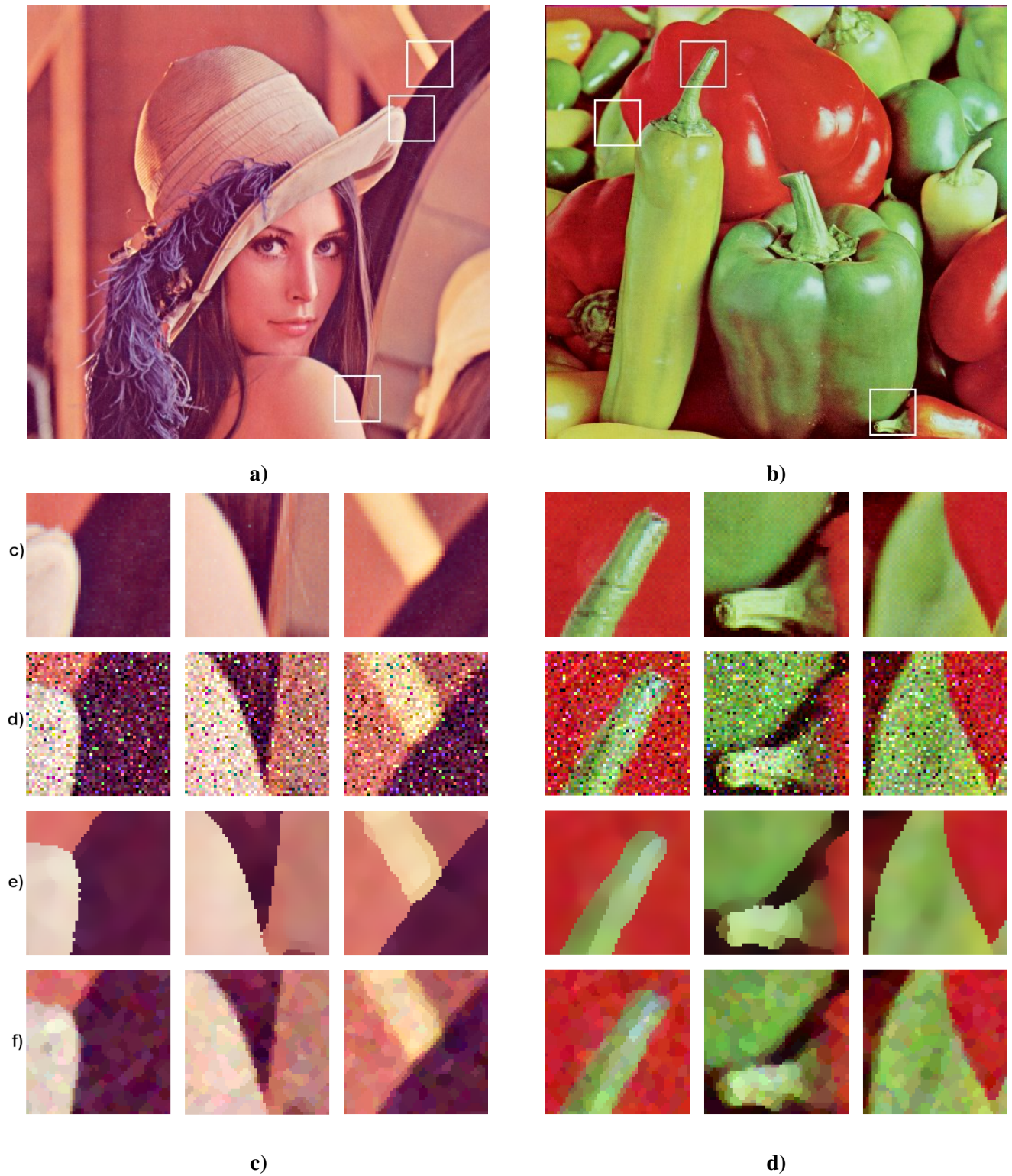


Figure 1.16: Color test images *LENA* **a)** and *PEPPERS* **b)** with depicted regions of interest **c)**. The chosen image regions were contaminated by mixed impulsive ( $p = 0.12$ ,  $p_1 = p_2 = p_3 = 0.3$ ) and Gaussian noise of  $\sigma = 30$ , **d)** and then restored with the DPAF method, **e)** and VMF, **f)**.

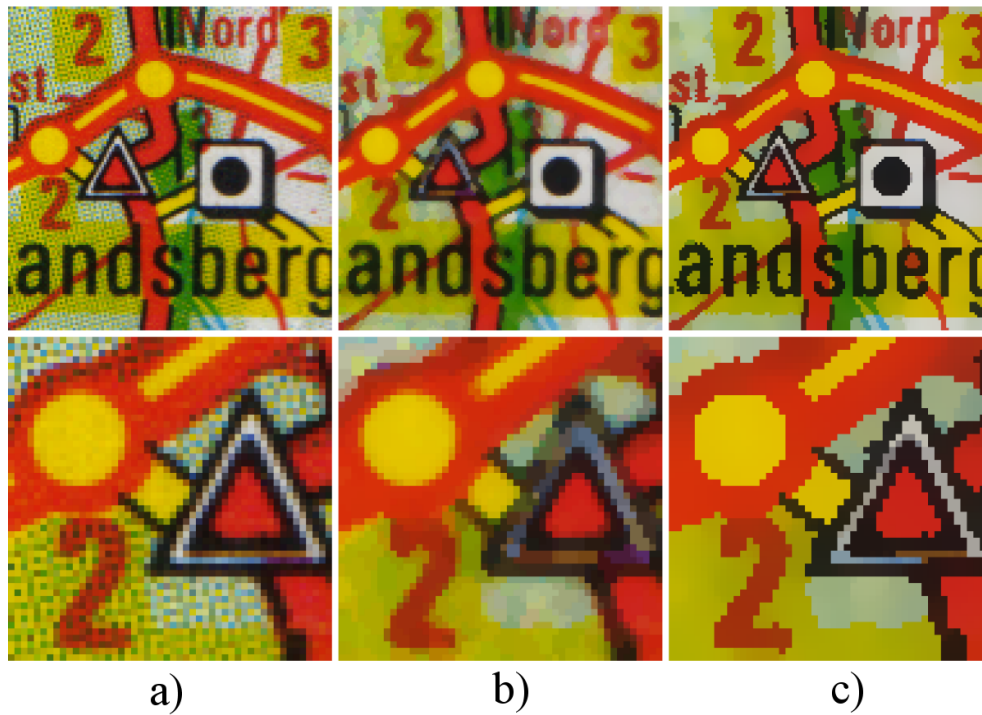


Figure 1.17: Comparison of efficiency of the vector median with the DPAF: **a)** test image, (part of a scanned map), **b)** VMF, ( $3 \times 3$  mask), **c)** DPAF, ( $\beta = 20$ ,  $\alpha = 1.25$ ,  $\eta = 2$ , 3 iterations).

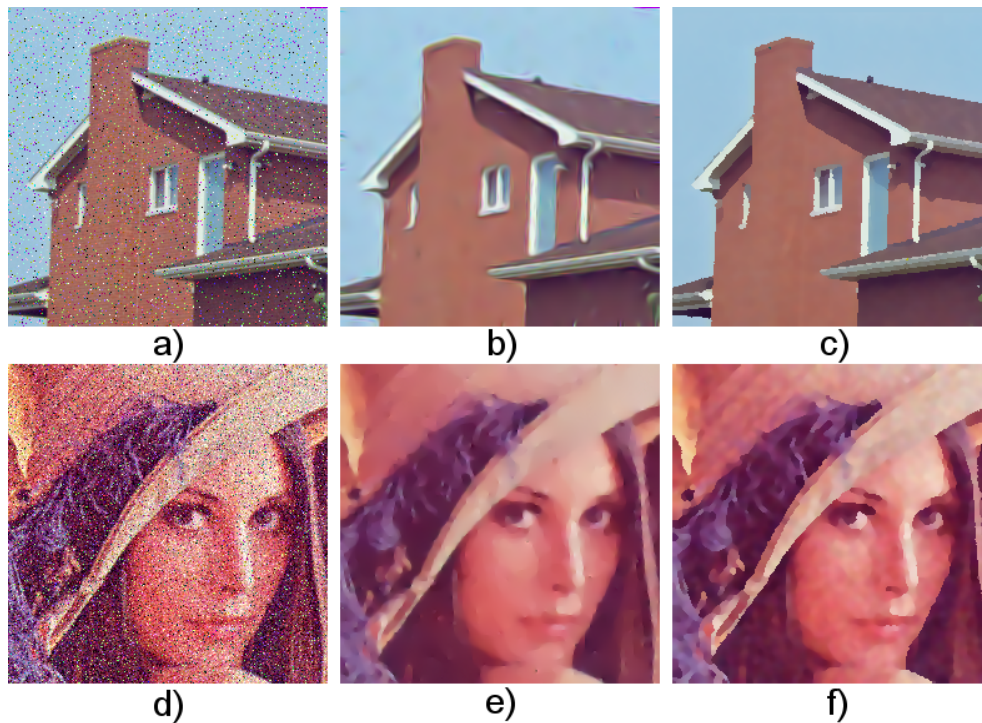


Figure 1.18: Comparison of the method proposed in [126, 127] with the new approach (DPAF): **a)** test image *HOUSE* contaminated with impulsive noise ( $p = 0.1$ ), **b)** GD-PDE [126, 127], **c)** DPAF, **d)** test image *LENA* contaminated with mixed impulsive and Gaussian noise, **e)** GD-PDE, **f)** DPAF.



# Bibliography

- [1] E. Abreu, "Signal-dependent rank-ordered mean (SD-ROM) filter" in S.K. Mitra, G.L. Sicuranza (Edts.) "Nonlinear Image Processing", Academic Press, 111-134, 2000.
- [2] E. Abreu, S.K. Mitra, "A signal-dependent rank-ordered mean (SD-ROM) filter -a new approach for removal of impulses from highly corrupted images", Proc. of International Conf. on Accoustic, Speech, and Signal Processing, vol. 4, pp. 2371-2374, 1995.
- [3] S.T. Acton, "Multigrid anisotropic diffusion", IEEE Transactions on Image Processing, vol 7, no.3, 280-291, March 1998.
- [4] L. Alparone, M. Barni, F. Bartolini, V. Cappellini, "Adaptively weighted vector-median filters for motion-fields smoothing", Proc. IEEE Internat. Conf. Acoustics, Speech and Signal Processing, pp. 2267-2270, May 7-10, 1996.
- [5] L. Alvarez, F. Guichard, P.L. Lions, J.M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion", SIAM J. Numer. Anal., vol 29, no. 3, pp. 845-866, 1992.
- [6] G. R. Arce, N. C. Gallagher, Jr., T. A. Nodes, "Median filters: theory and applications," in Advances in Computer Vision and Image Processing, (T. S. Huang, ed.), vol. 2, JAI Press, 1986.
- [7] G. R. Arce, "A generalized weighted median filter structure admitting negative weights", IEEE Transactions on Signal Processing, vol. 46, December 1998.
- [8] G. R. Arce, J. Paredes, "Image enhancement with weighted medians", in "Nonlinear Image Processing", (S. Mitra and G. Sicuranza, Eds.), pp. 2767, Academic Press, 2000.
- [9] J. Astola, P. Haavisto, Y. Neuovo, "Vector median filters", IEEE Proc. vol. 78, pp. 678-689, 1990.
- [10] D. Barash, "A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation", IEEE Transactions on PAMI, vol. 24, no. 6, pp. 844-847, June 2002.
- [11] K. E. Barner, R. C. Hardie, "Spatialrank order selection filters" in "Nonlinear Image Processing", (S. K. Mitra and G. Sicuranza, Eds.), San Diego, Academic Press, 2001.
- [12] V. Barnett, "The ordering multivariate data", Journal of Royal Statistical Society, A 139, part 3, pp. 318-355, 1976.
- [13] M. Barni, V. Cappellini, "On the computational complexity of multivariate median filters", Signal Processing, vol.71, pp 45-54, 1998.
- [14] M. Basu, M. Su, Image smoothing with exponential functions, International Journal of Pattern Recognition and Artificial Intelligence, vol. 15, no. 4, pp. 735-752, 2001.
- [15] M. Barni, V. Cappellini, A. Mecocci, "Fast vector median filter based on Euclidean norm approximation", IEEE Signal Processing Letters, vol. 1, no. 6, pp. 92-94, 1994.
- [16] J.C. Bezdek, S.K. Pal (Eds.), "Fuzzy Models for Pattern Recognition", IEEE Press, 1992.
- [17] S. Biswas, N.R. Pal, S.K. Pal, "Smoothing of digital images using the concept of diffusion process", Pattern Recognition, vol. 29, no. 3, pp. 497-510, 1996.
- [18] M.J. Black, G. Sapiro, D. H. Marimont, D. Heeger, "Robust anisotropic diffusion", IEEE Trans. on Image Processing, vol. 7, no. 3, pp. 421-432, March 1998.
- [19] P. Blpmgren, T.F. Chan, "Color tv: Total variation methods for restoration of vector-valued images", in "Special Issue on Partial Differential Equations and Geometry Driven Diffusion in Image Processing and Analysis", IEEE Trans. Image Processing, vol. 7, no. 3, pp. 304-309, March 1998.

- [20] G. Borgefors, "Distance transformations in arbitrary dimensions", *Computer Vision, Graphics and Image Processing*, vol. 27, pp. 321-345, 1984.
- [21] G. Borgefors, "Distances transformations in digital images", *Computer Vision, Graphics and Image Processing*, vol 34, pp. 344-371, 1986.
- [22] D.R.K. Brownrigg, "The weighted median filter", *Commun. of Assoc. Computers*, pp. 807-818, 1984.
- [23] R.A. Carmona, S. Zhong, "Adaptive smoothing respecting feature directions", *IEEE Transactions on Image Processing*, Special Issue PDE's and Geometry Driven Diffusion in Image Analysis, vol. 7, no. 3, pp. 353-358, 1998.
- [24] F. Catte, F. Dibos, G. Koepfler, "Image selective smoothing and edge detection by nonlinear diffusion", *SIAM J. Numer. Anal.*, vol. 29, no. 1, pp. 182-193, February 1992.
- [25] F. Catte, P.-L. Lions, J.M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion-II", *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 845-866, 1992.
- [26] P. Charbonnier, G. Aubert, M. Blanc-Ferraud, and M. Barlaud, "Two-deterministic half-quadratic regularization algorithms for computed imaging", *Proc of IEEE International Conf. on Image Proc. ICIP*, pp. 1681-172, Austin (Texas), November 1994.
- [27] K. Chen, "A feature-preserving adaptive smoothing method for early vision", Technical Report, National Laboratory of Machine Perception and Center for Information Science, Peking University, Beijing, China.
- [28] T. Chen, H.R. Wu, "Adaptive impulse detection using center-weighted median filters", *IEEE Signal Processing Letters*, vol. 8, no. 1, pp. 1-3, January 2001.
- [29] O. Cuisenaire, "Distance transformations: fast algorithms and applications to medical image processing", PhD Thesis, Universite Catholique de Louvain, October 1999.
- [30] L.S. Davis, A. Rosenfeld, "Noise cleaning by iterated local averaging", *IEEE Trans. Syst. Man. Cybern., SMC*, vol. 8, pp. 705-710, 1978.
- [31] S. DiZenzo, "A note on the gradient of multi-image", *Computer Vision Graphics Image Processing*, vol. 33, pp. 116-125, 1986.
- [32] S.A. Durrani, "The distance-weighted k-nearest neighbor rule", *IEEE Trans. on Systems, Man and Cybernetics*, vol. 15, pp. 630-636, 1977.
- [33] K. Erler, E. Jernigan, "Adaptive image restoration using recursive image filters", *IEEE Transactions on Signal Processing*, vol. 42, no. 7, pp. 1877-1881, July 1994.
- [34] B. Fischl, E.L. Schwartz, "Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no 4, pp. 342-352, April 1997.
- [35] K. Fukunaga "Introduction to Statistical Pattern Recognition", Academic Press, Second Edition, London, UK, 1990
- [36] M. Gabbouj, F.A. Cheickh, "Vector median - vector directional hybrid filter for colour image restoration", *Proceedings of EUSIPCO*, pp. 879-881, 1996.
- [37] G. Gerig, R. Kikinis, O. Kuebler, F. Jolesz, "Nonlinear anisotropic filtering of MRI data", *IEEE Transaction on Medical Imaging*, vol. 11, no. 2, pp. 221-232, June 1992.
- [38] R.C. Gonzalez, R.E. Woods, "Digital Image Processing", Reading MA, Addison-Wesley, 1992.
- [39] S. Guillon, P. Baylou, M. Najim, N. Keskes, "Adaptive non-linear filters for 2D and 3D image enhancement", *Signal Processing*, vol. 67, pp. 237-254, 1998.
- [40] R.C. Hardie, G.R. Arce, "Ranking in  $R^p$  and its use in multivariate image estimation", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, no. 2, pp. 197-209, 1991.
- [41] S. Haykin, "Adaptive filter theory", Prentice Hall, Englewood Cliffs, New Jersey, 1991.
- [42] D. Karakos, P.E. Trahanias, "Generalized multichannel image filtering structures", *IEEE Trans. on Image Processing*, vol. 6, no. 7, pp. 1038-1045, 1997.
- [43] B.B. Kimia, K. Siddiqi, "Geometric heat equation and nonlinear diffusion of shapes and images", *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 305-322, November 1996.

- [44] B.B. Kimia, K. Siddiqi, "Geometric heat equation and non-linear diffusion of shapes and images", in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 113-120, 1994.
- [45] R. Kimmel, R. Malladi, N. Sochen, "Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images", *International J. Computer Vision*, vol. 39, no. 2, pp. 111, 2000.
- [46] C.O. Kislman, "Regularity properties of distance transformations in image analysis", *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 390-398, November 1996.
- [47] R. Klette, P. Zamperoni, "Handbuch der Operatoren für die Bildverarbeitung", Vieweg Verlag, Braunschweig, Wiesbaden, 1992.
- [48] S.J. Ko, Y.H. Lee, "Center weighted median filters and their applications to image enhancement", *IEEE Trans. on Circuits Syst.*, vol. 38, pp. 984-993, Sept. 1991.
- [49] M. Kuwahara, K. Hachimura, S. Eiho, M. Kinoshita, "Digital processing of biomedical images", Plenum Press, New York, pp. 187-203, 1976.
- [50] J.S. Lee, "Digital image smoothing and the sigma filter", *Computer Vision Graphics Image Process*, vol 24, pp. 255-269, 1983.
- [51] J.S. Lee, "Refined filtering of image noise using local statistics", *Computer Graphics Image Process.*, vol. 15, pp. 380-389, 1981.
- [52] J.S. Lee, "Digital image enhancement and noise filtering by use of local statistics", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 2, pp. 165-168, 1980.
- [53] R.S. Lin, Y.C. Hsueh, "Multichannel filtering by gradient information", *Signal Processing*, vol. 80, pp. 279-293, 2000.
- [54] G. Lohmann, "Volumetric Image Analysis", John Wiley and Teubner, 1988.
- [55] R. Lukac, "Color image filtering by vector directional order-statistics", *Pattern Recognition and Image Analysis*, vol. 12, no. 3, pp. 279-285, 2002.
- [56] R. Lukac, "Optimised directional distance filter", *Machine Graphics and Visions: Special Issue on Colour Image Processing and Applications*, vol. 11, no. 2/3, 311-326, 2002.
- [57] R. Lukac, "Introducing of the weight concept to vector directional filters", *Journal of Electrical Engineering*, vol. 52, no. 3-4, pp. 98-100, 2001.
- [58] R. Lukac, "Adaptive impulse noise filtering by using center-weighted directional information", *Proceedings of the CGIV 2002, France*, pp. 86-89, 2002.
- [59] N. Madras, G. Slade, "The Self-Avoiding Walk", Boston, Birkhauser, 1993.
- [60] J. Maeda, T. Iizawa, T. Ishizaka, C. Ishikawa, Y. Suzuki, "Segmentation of natural images using anisotropic diffusion and linking of boundary edges", *Pattern Recognition*, vol. 31, no. 12, pp. 1993-1999, December 1998.
- [61] G.A. Mastin, "Adaptive filters for digital image noise smoothing: an evaluation", *Computer Vision Graphics and Image Processing*, vol. 31, pp. 103-121, 1985.
- [62] M.J. McDonnell, "Box-filtering techniques", *Computer Graphics and Image Processing*, vol. 17, pp. 65-70, 1981.
- [63] M. Nagao, T. Matsuyama, "A structural analysis of complex aerial photographs", Plenum Press, New York, 1980.
- [64] M. Nagao, T. Matsuyama, "Edge preserving smoothing", *Computer Graphics and Image Processing*, vol. 9, pp. 394-407, 1979.
- [65] N. Nikolaidis, I. Pitas, "Multichannel  $L$  filters based on reduced ordering", *IEEE Transactions on Circuits and Systems for Video Technology*, vol.6, no. 5, pp. 470-482, October 1996
- [66] S. Osher, L.I. Rudin, "Feature-oriented image enhancement using shock filters", *SIAM Journal on Numerical Analysis*, vol. 27, no. 4, pp. 919-940, August 1990.
- [67] P. Perona, J. Malik, "Scale space and edge detection using anisotropic diffusion", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 629-639, 1990.
- [68] P. Perona, J. Malik, "Scale space and edge detection using anisotropic diffusion", in *Proceedings of IEEE Workshop on Computer Vision*, pp. 16-22, Miami, 1987.
- [69] I. Pitas, "Digital Image Processing Algorithms and Applications", Wiley-Interscience, 2000

- [70] I. Pitas, "Digital Image Processing Algorithms", Prentice Hall, 1993.
- [71] I. Pitas, P. Tsakalides, "Multivariate ordering in color image processing", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, no. 3, pp. 247-256, 1991.
- [72] I. Pitas, A.N. Venetsanopoulos, "Order statistics in digital image processing, *Proceedings of IEEE*, vol. 80, no. 12, pp. 1893-1923, 1992.
- [73] I. Pitas, A.N. Venetsanopoulos, "Nonlinear Digital Filters: Principles and Applications", Kluwer Academic Publishers, Boston, MA, 1990.
- [74] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, "Adaptive fuzzy systems for multichannel signal processing", *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1601-1622, 1999.
- [75] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, "Multichannel filters for image processing", *Signal Processing: Image Communications*, vol. 9, no. 2, pp. 143-158, 1997.
- [76] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, "Fuzzy adaptive filters for multichannel image processing", *Signal Processing Journal*, vol. 55, no. 1, pp. 93-106, 1996.
- [77] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, "Adaptive multichannel filters for color image processing", *Proceedings of the Visual Communications Image Processing Conference, SPIE vol. 2727, Part III*, pp. 1270-1279, 1996.
- [78] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, "Fuzzy adaptive filters for multichannel image processing", *Signal Processing Journal*, vol. 55, no. 1, pp. 93-106, 1996.
- [79] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, "Colour image processing using fuzzy vector directional filters", *Proceedings of the IEEE Workshop on Nonlinear Signal and Image Processing, Greece*, pp. 535-538, 1995.
- [80] K.N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, A.N. Venetsanopoulos, "Color image processing using adaptive multichannel filters", *IEEE Trans. on Image Processing*, vol. 6, no. 7, pp. 933-950, 1997.
- [81] K.N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, A.N. Venetsanopoulos, "An adaptive nearest neighbor multichannel filter", *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 699-703, December 1996.
- [82] K.N. Plataniotis, D. Androutsos, V. Sri, A.N. Venetsanopoulos, "A nearest neighbour multichannel filter", *Electronic Letters*, vol. 31, no. 22, pp. 1910-1911, 1995.
- [83] K.N. Plataniotis, A.N. Venetsanopoulos, "Color Image Processing and Applications", Springer Verlag, August 2000.
- [84] K.N. Plataniotis, A.N. Venetsanopoulos, "Vector filtering" in S.J. Sangwine, R.E.N. Horne(Eds.), "The Colour Image Processing Handbook", Chapman & Hall, Cambridge, Great Britain, 1998, pp. 188-209.
- [85] F. Preteux, N. Merlet, "New concepts in mathematical morphology: the topographical distance function", in P. D. Gader, E. R. Dougherty - Editors", *Proceedings of SPIE*, 1568, 66-77, 1991.
- [86] P. Pujas, M.J. Aldon, "Estimation of the colour image gradient with perceptual attributes", in *Proceedings of the 9th International Conference on Image Analysis and Processing*, pp. 103-110, Springer Verlag, September 1997.
- [87] G. Ramponi, C. Moloney, "Smoothing speckled images using an adaptive rational operator", *IEEE Signal Processing Letters*, vol. 4, no. 3, pp. 68-71, 1997.
- [88] D. de Ridder, R.P.W. Duin, P.W. Verbeek, L.J. van Vliet, "The applicability of neural networks to non-linear image processing", *Pattern Analysis and Applications*, vol. 2, pp. 111-128, 1999.
- [89] B. Romeny ter Haar, "Geometry-Driven Diffusion in Computer Vision", Kluwer, Boston, MA, 1994.
- [90] L.I. Rudin, S. Osher, E. Fatemi, "Nonlinear total variation based noise removal algorithms", *Physica D*, vol. 60, pp. 259-268, 1992.
- [91] P. Saint-Marc, J.S. Chen, G. Medioni, "Adaptive smoothing: a general tool for early vision", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 514, June 1991.
- [92] P. Saint-Marc, J.S. Chen, G. Medioni, "Adaptive smoothing: a general tool for early vision", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 618-624, San Diego, 1989.
- [93] S.J. Sangwine, R.E.N. Horne(Eds.), "The Colour Image Processing Handbook", Chapman & Hall, Cambridge, Great Britain, 1998.

- [94] G. Sapiro, "Geometric partial differential equations and image analysis", Cambridge University Press, Cambridge, MA, 2001.
- [95] G. Sapiro, D.L. Ringach, "Anisotropic diffusion of multivalued images with applications to color filtering", *IEEE Transactions on Image Processing*, vol. 5, no. 11, pp. 1582-1586, November 1996.
- [96] A. Scher, V. Dias, F.R. Rosenfeld, "Some new image smoothing techniques", *IEEE Trans. on SMC*, vol. 10, pp. 153-158, March 1980.
- [97] M. Schmitt, "Lecture notes on geodesy and morphological measurements", *Proceedings of the Summer School on Morphological Image and Signal Processing*, Zakopane, Poland, pp. 36-91, 1995.
- [98] M. Schulze, J.A. Pearce, "Some properties of the two-dimensional pseudomedian filter", *Proceedings of SPIE*, vol. 1451, pp. 48-57, 1991.
- [99] J. Shah, "A common framework for curve evolution, segmentation and anisotropic diffusion", in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 136-142, June 1996.
- [100] F.Y. Shih, J.J. Liu, "Size-invariant four-scan Euclidean distance transformation", *Pattern Recognition*, vol. 31, no. 11, pp. 1761-1766, 1998.
- [101] S. K. Mitra, G. Sicuranza, (Editors) *Nonlinear Image Processing (Communications, Networking and Multimedia)*, Academic Press, September, 2000
- [102] S.M. Smith, J.M. Brady, "SUSAN - a new approach to low level image processing", *Int. Journal of Computer Vision*, vol. 23, no. 1, pp. 45-78, 1997.
- [103] B. Smolka, A. Chydzinski, K. Wojciechowski, K. Plataniotis, A.N. Venetsanopoulos, "On the reduction of impulsive noise in multichannel image processing", *Optical Engineering*, vol. 40, no. 6, pp. 902-908, 2001.
- [104] B. Smolka, "On the new robust algorithm of noise reduction in color images", *Computers & Graphics*, vol. 11, no 2/3, 311-326, 2003.
- [105] B. Smolka, M.K. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, "Fast modified vector median filter" in W. Skarbek (Ed.), *Computer Analysis of Images and Patterns, LNCS 2124*, Springer-Verlag Berlin, pp. 570-580, 2001.
- [106] M.K. Szczepanski, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, "Enhancement of the DNA Microarray Chip Images", *Proceedings of Digital Signal Processing DSP 2002*, Santorini, Greece 2002.
- [107] B. Smolka, M.K. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos "On the modified weighted vector median filter", *Proceedings of Digital Signal Processing DSP2002*, Santorini, Greece, 2002.
- [108] B. Smolka, K. Wojciechowski, "Random walk approach to image enhancement", *Signal Processing*, vol. 81, no. 3, pp. 465-482, 2001.
- [109] B. Smolka, A. Chydzinski, K. Wojciechowski, K. Plataniotis, A. Venetsanopoulos, "Self-adaptive algorithm of impulsive noise reduction in color images", *Pattern Recognition*, vol. 35, pp. 1771-1784, 2002.
- [110] N. Sochen, R. Kimmel, R. Malladi, "A geometrical framework for low level vision", *IEEE Trans. Image Processing*, vol. 7, no. 3, p. 310, 1998.
- [111] A. Somboonkaew, S. Chitwong, F. Cheevasuvit, K. Dejhan, S. Mitatha, "Segmentation on the edge preserving smoothing image", *Proceedings of 20th Asian Conference on Remote Sensing*, November 22-25, 1999, Hong Kong, China.
- [112] M. Spann, A. Nieminen, "Adaptive Gaussian weighted filtering for image segmentation" *Pattern Recognition Letters*, vol. 8, pp. 251-255, 1988.
- [113] F. Spitzer, "Principles of random walk", D. van Nostrand Company, Princeton, New Jersey, 1975.
- [114] V. Starovoitov, "Towards a distance transform generalization", *Proceedings of the 9th Scandinavian Conference on Image Analysis*, (G. Borgfors Editor), Uppsala, pp. 499-506, 1995.
- [115] M. Szczepanski, B. Smolka, K. N. Plataniotis, A. N. Venetsanopoulos, "On the distance function approach to color image enhancement", *Discrete Applied Mathematics*, in press.
- [116] M. Szczepanski, B. Smolka, K. N. Plataniotis, A. N. Venetsanopoulos, "On the geodesic paths approach to color image filtering", *Signal Processing*, in press.

- [117] K. Tang, J. Astola, Y. Neuovo, "Nonlinear multivariate image filtering techniques", *IEEE Trans. on Image Processing*, vol. 4, no. 6, pp. 788-797, June 1995.
- [118] P.J. Toivanen, "New geodesic distance transforms for gray scale images", *Pattern Recognition Letters*, vol. 17, pp. 437-450, 1996.
- [119] C. Tomasi, R. Manduchi, "Bilateral filtering for gray and color image", *Proc. of the IEEE International Conference on Computer Vision*, Bombay, India, 1998.
- [120] F. Tomita, S. Tsuji, "Extraction of multiple regions by smoothing in selected neighbourhoods", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, pp. 107-109, 1977.
- [121] F. Torkamani-Azar, K.E. Tait, "Image recovery using the anisotropic diffusion equation", *IEEE Transactions on Image Processing*, vol. 5, no. 11, pp. 1573-1578, November 1996.
- [122] P.E. Trahanias, D. Karakos, A.N. Venetsanopoulos, "Directional processing of color images: theory and experimental results", *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 868-880, 1996.
- [123] P.E. Trahanias, D.G. Karakos, A.N. Venetsanopoulos, "Directional processing of color images: theory and experimental results", *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 868-880, 1996.
- [124] P.E. Trahanias, I. Pitas, A.N. Venetsanopoulos, "Color image processing", in "Advances in 2D and 3D Digital Processing (Techniques and Applications)", C.T. Leondes Ed., Academic Press, 1994.
- [125] P.E. Trahanias, A.N. Venetsanopoulos, "Vector directional filters: a new class of multichannel image processing filters", *IEEE Trans. on Image Processing*, vol. 2, no. 4, pp. 528-534, 1993.
- [126] D. Tschumperle, R. Deriche, "Constrained and unconstrained PDE's for vector image restoration", *Proceedings of the SCIA Conference*, 2001
- [127] D. Tschumperle, R. Deriche, "Diffusion PDE's on vector-valued images: local approach and geometric viewpoint", *IEEE Signal Processing Magazine*, Special Issue, September 2002.
- [128] M.I. Vardavoulia, I. Andreadis, Ph. Tsalides, "A new vector median filter for colour image processing", *Pattern Recognition Letters*, vol. 22, pp. 675-689, 2001.
- [129] A.N. Venetsanopoulos, K.N. Plataniotis, "Multichannel image processing", *Proceedings of the IEEE Workshop on Nonlinear Signal and Image Processing*, pp. 2-6, 1995.
- [130] T. Viero, K. Oistamo, Y. Neuvo, "Three-dimensional median-related filters for color image sequence filtering", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 2, 129-142, 1994.
- [131] D. Wang, Q. Wang, "A weighted averaging method for image smoothing", *Proceedings of the 8th. ICPR*, 981-983, Paris, 1988.
- [132] D. Wang, A.H. Vagnucci, C.C. Li, "Gradient inverse weighted smoothing scheme and the evaluation of its performance", *Computer Graphics and Image Processing*, vol. 15, pp. 167-181, 1981.
- [133] J. Weickert J, "Anisotropic diffusion in image processing", G.G. Teubner, Stuttgart, 1998.
- [134] R. Whitaker, G. Gerig, "Vector-valued diffusion" in B. Romeny ter Haar, "Geometry-Driven Diffusion in Computer Vision", Kluwer, Boston, MA, pp. 93-134, 1994.
- [135] R. Wichman, K. Oistamo, Q. Liu, M. Grundstrom, Y. Neuovo, "Weighted vector median operation for filtering multispectral data", *Proceedings of the Conference 'Visual Communications and Image Processing'*, pp. 376-383, 1992.
- [136] A. Witkin, "Scale-space filtering", *Proc. of Int. Joint Conf. Artif. Intell.*, pp. 1019-1021, 1983.
- [137] G. Wyszecki, W.S. Stiles, "Color science: concepts and methods", quantitative data and formulae, Second Edition, John Willey, New York, 1982.
- [138] L. Yaroslavsky, E. Murray, "Fundamentals of digital optics", Birkhäuser, Boston, 1996.
- [139] Y.L. You, W. Xu, A. Tannenbaum, M. Kaveh, "Behavioral analysis of anisotropic diffusion in image processing", vol. 5, no. 11, pp. 1539-1553, 1996.