ZESZYTY NAUKOWE
POLITECHNIKI ŚLĄSKIEJ

Bogdan SMOŁKA

NONLINEAR TECHNIQUES OF NOISE REDUCTION IN DIGITAL COLOR IMAGES

WYDAWNICTWO POLITECHNIKI ŚLĄSKIEJ

Gliwice 2004

POLITECHNIKA ŚLĄSKA
ZESZYTY NAUKOWE
Nr 1620

Bogdan SMOŁKA

Nonlinear Techniques of Noise Reduction in Digital Color Images

Gliwice 2004

Opiniodawcy

Prof. dr hab. inż. Jan PIECHA
Prof. dr hab. inż. Andrzej KASIŃSKI

Redakcja techniczna

Alicja NOWACKA

PL ISSN 0434-0760

Properly speaking, such work is never finished; one must declare it so when, according to time and circumstances, one has done one's best.

J. W. von Goethe

Contents

Dealing with impulsive noise remains one of the great challenges of modern engineering. It is hard to model, predict, and filter and yet it pervades the world, [146].

Preface

WITH breathtaking pace, computers are becoming more powerful and at the same time less expensive. Thus, the hardware needed for digital image processing is readily available. In this way, image processing is becoming a common tool to analyze *multidimensional scientific data* in all areas of natural science. For more and more scientists, processing of monochrome and especially *multichannel images* is the key to study complex scientific problems, they could not have dreamed to tackle only a few years ago.

Multichannel signal processing is of paramount importance in application areas such as biomedicine, computer and machine vision, robotics, entertainment and multimedia applications, industrial inspection, remote sensing and many others. In all these areas the users and system developers work with *multidimensional data sets*.

It is well known that humans and computer vision systems use *color information* to sense the environment and that the correct perception of *color* can help in different tasks of image understanding and pattern recognition. Unfortunately, *noise* and other impairments associated with the measurement or the transmission apparatus can significantly degrade the value of the *color information* carried by the digital images. This usually declines their perceptual fidelity and also decreases the performance of the task for which the image was created.

It comes therefore as no surprise that the most common signal processing task is the *noise filtering*. *Noise filtering* is an essential part of any image processing based system, whether the final information is used for human perception or for an automatic inspection and analysis.

The amount of research published in the last ten years indicates a *growing interest* in the area of color image processing and analysis. Furthermore, the surge of emerging applications such as web-based processing of color images and videos, image retrieval systems indexing large multimedia databases, enhancement of medical and biological images, digital archiving, cultural heritage preservation projects and the proliferation of smart devices such as video-enabled wireless phones, wearable computers and personal digital assistant tools, suggests that the demand for new, more powerful and cost effective multichannel filtering solutions will continue.

The correction of the *signal distorsions* is a digital process, by which disturbances introduced by the sensor system are rectified, with the goal being to obtain the image or generally the signal, which corresponds as closely as possible to the output of an *ideal imaging system*. Thus, correcting *signal artifacts*, in practice means adjusting the characteristics of the imaging system to meet specific demands of the human observer or the computer vision system.

Digital image processing is based on the conversion of a continuous image field into equivalent digital form. The synthesis of images from the signals arising from various sensor systems is accomplished by a digital process directed to transforming the signal into a form allowing visual or machine perception. The requirements for an ideal conversion system are usually expressed in terms of certain technical properties such as the resolution of the imaging systems, photometric accuracy, quantization levels, intensity of intrinsic noise and many others.

Improvement of the *quality* of images has always been one of the central tasks of *digital image processing*. In modern terms, improvements in sensitivity, resolution and *noise reduction* have equated higher quality with greater informational throughput. *Image noise* is an unwanted feature, which is either contained in the relevant light signal or is added by the imaging process and it compromises a precise evaluation of the light signal distribution, which should be measured.

The analysis of the image noise in digital image acquisition systems often focuses on *random noise sources*, such as those associated with quantum signal detection (shot noise) and *signal independent fluctuations* (dark current, readout noise). Other important source of image noise is the inhomogeneity of the responsiveness of the sensor elements and signal disturbances that introduce repeatable patterns into image data.

During image *formation*, *acquisition*, *storage* and *transmission* many additional types of *distorsions* limit the *quality* of digital images. Transmission errors, periodic or random motion of the camera system during exposure, electronic instability of the image signal, electromagnetic interferences from natural or man-made sources, sensor malfunctions, optic imperfections, electronics interference or aging of the storage material all disturb the *image quality*.

In the last years, the area of *vector valued* (*multichannel*, *multispectral*, *multicomponent*) signal processing has dramatically increased. The leading edge of development and interest is in the domain of remote sensing, but the *classical color imaging* still remains the preferred research domain.

Typically, a *color image* is represented in each pixel by a three component vector. The vector components quantify in general the amounts of pure red, green and blue that compose the local color. These *vector valued signals* cannot be reduced to a stack of separately processed scalar components, due to the inherent *correlation* between the channels.

In the literature, several *noise reduction techniques* have been proposed. They can be divided into *linear* and *nonlinear* techniques.

Linear processing techniques have been widely used in digital signal processing applications, since their mathematical simplicity and the availability of a unifying linear system theory make them relatively easy to analyze and implement. Unfortunately, most of the linear techniques tend to blur structural elements such as lines, edges and fine image details and therefore many *multichannel* image processing tasks cannot be efficiently accomplished by linear techniques.

Image signals are *nonlinear* in nature, due to the presence of structural information and are perceived through the *human visual system*, which has strong *nonlinear* characteristics. *Nonlinear methods* are able to preserve important multichannel structural elements and eliminate degradations occurring during *signal formation* or *transmission* through *nonlinear* channels and they proved to be efficient in the suppression of *impulsive*, *Gaussian* and *mixed* type of noise.

The most popular *nonlinear filters* are based on *order statistics*. However their common *drawback* is that they ignore the *temporal* or *spatial* information of the signal samples. Therefore many different techniques alleviating this problem have been proposed to date.

The algorithms developed by the author of this monograph are oriented towards the *improvement* of the efficiency of the standard filtering approaches and are especially focused on the *impulsive noise removal*. The special emphasis is placed on the suppression of *impulsive* and *mixed* noise, as the nonlinear techniques are *especially well suited* to this particular task.

THE purpose of this book is to present the *state of the art* in *nonlinear color image noise removal* techniques and also to *organize* and *integrate* the authors's *original contributions* to the dynamic development of this field, scattered in numerous refereed scientific publications. The book itself can be characterized as a *monograph* of author's own solutions put on the background of the state of the art.

The content of this *monograph* is structured into seven Chapters. The first part of this book is devoted to the overview of the problems of *noise reduction* in *color images*. Its purpose is to give some insight into the *fundamentals* of *color image processing* and basic *color image filtering* designs. This Chapter also covers the various color image *noise sources*, their *models* and *measures* of the *quality* of *image restoration*.

The second Chapter is devoted to the *adaptive* schemes of *noise reduction* in gray scale images, as many techniques primarily developed for monochrome images can be *reformulated* to work in the *multichannel domain*. In this Chapter *special emphasis* is put on the *weighted*

median filters and their *optimization*, as this *filter family*, extended to the vectorial case is considered in Chapter 3 and 7.

Chapter 3 presents an *overview* of the *noise reduction filters* used in *color imaging*. This Chapter provides the *state of the art* in *color image filtering* and serves as a basis for the next Chapters in which author's *original contributions* are presented. In this Chapter the modification of the *central weighted median filter* and the *rank conditioned vector median filter* are also briefly outlined.

The next Chapter covers the *anisotropic diffusion technique* in the scalar and *multichannel case*. The evaluation of the *efficiency* of this *nonlinear technique* shows that the approach, in which the central pixel of the filtering window is *excluded* from the processing, yields much better results and increases the robustness of the *anisotropic diffusion* method and its various modifications. This observation is utilized in the design of novel efficient techniques of impulsive *noise removal* presented in Chapter 5 and 6.

Chapter 5 is devoted to the *digital paths approach* to *color image filtering*, originally developed by the author for the gray scale imaging using the concept of *random* and *self-avoiding walks*. This *novel technique*, based on the exploration of the local neighborhood through *digital paths* and on the utilization of the *fuzzy concepts*, can be seen as a *powerful generalization* of the *anisotropic diffusion* approach introduced in Chapter 4. The performed simulations indicate that the new filter class *excels significantly* over the currently used *nonlinear multichannel techniques* especially in the case of mixed noise.

In the next Chapter the problem of *nonparametric impulsive noise reduction* in *multichannel images* is addressed. A new family of filters for *noise attenuation* elaborated by the author, based on *nonparametric probability density estimation* of sample data, is introduced and its relationship to commonly used filtering techniques is investigated. Extensive simulation experiments indicate that the presented family of filters *outperform* the standard techniques used to eliminate *impulsive noise* in *color images*.

The last Chapter deals with the *adaptive optimization* of the *weighted vector median filters* described in Chapters 2, 3 and also introduces the *new technique* based on the so called *sigma-filtering*, which can be seen as an *extension* of the *rank conditioned vector median* introduced in Chapter 3. This *novel adaptive filtering technique* is based on *robust order statistic* concepts and simplified statistical measures of vectors' dispersion. Simulation studies indicate that the presented filters are *computationally attractive*, yield good performance and are able to *preserve* fine details, while efficiently suppressing impulsive noise.

* * *

1

Noise Reduction in Color Images

Visual information processing is increasingly becoming widespread as multimedia becomes common in everyday life. With the expanding use of color images in multimedia applications and the proliferation of color capturing and display units, the interest in color imaging is rapidly growing.

Very often the quality of color images is decreased by different types of noise distortions. Noise can appear during the process of image acquisition, transmission and storage. Therefore its removal or reduction is one of the most important image processing tasks.

This chapter presents the fundamentals of color image processing, describes the various noise sources and its models, introduces the image quality measures and also describes briefly some basic filtering designs.

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 in computer vision, because in many practical applications the location of scene objects can be obtained only when color information is considered, [66, 117, 157, 229, 246, 250, 267, 419].

In many cases it is indispensable to remove the corrupted pixels to facilitate subsequent image processing operations such as *edge detection*, *image segmentation* and *pattern recognition*. To convey the desired information correctly, the noisy signal should be processed by a filtering

algorithm, that removes the noise component, but retains the image structure. Therefore, the goal of image filtering is the removal of unprofitable information that may corrupt any of the following image processing steps.

The computer vision systems can consist of a variety of mechanical, optical, electronic or chemical components, but all of them perform three basic operations: *image acquisition*, *signal processing* and *image formation*. However, these processes do not occur without serious problems. A number of undesired disturbances of the color image information result from the interference between the original signal and noise process. Noise affects the image quality level, decreasing not only its visual perception but also the performance of the task for which the image was acquired. Therefore *filtering*, the process of *signal transformation* into a more suitable form for a given task, is needed.

The noise removal process can be divided into *reconstruction* and *enhancement* filtering. In general, reconstruction filters utilize some knowledge about the type of image degradation, whereas image enhancement techniques attempt to improve (mostly subjectively measured) the quality of an image for human or machine interpretation. Both noise filtering and enhancement of the colors and structural information of the image are usually viewed as pre-processing stages in the image processing chain.

1.2 Fundamentals of Color Image Processing

The presence or absence of light is what causes the sensation of color. Light is a physical phenomenon, but color perception depends on the interaction of light with human visual system and is therefore a psycho-physiological experience, [117, 118, 234, 267, 367]. Since human high-intensity color vision is based on three types of photo-receptor cone cells, three numerical components are necessary and sufficient to define a color, if appropriate spectral weighting functions are used. The human cones respond to the short (*Blue*), medium (*Green*) and long (*Red*) wavelengths. Therefore, a color can be specified by a three-component vector, (Figs. 1.1, 1.6). The set of all colors forms a multidimensional space called *color space* or *color model*, [52, 86, 222, 267, 419].

Human perception of color is based on its *lightness*, *hue* and *saturation*, [66, 118, 248, 267, 367, 419]. Lightness is the perceptual response to luminance and distinguishes the gray levels. Hue is a color attribute associated with the dominant wavelength in a mixture of light waves and represents the dominant color as perceived by a human observer.

Saturation refers to the relative purity or the amount of white light mixed with a hue. Hue and saturation together describe the chrominance and the human perception of color is basically determined by *luminance* and *chrominance*, [118, 235, 271].

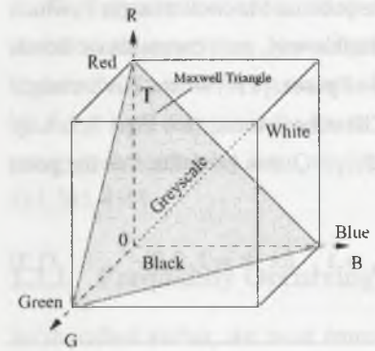


Fig. 1.1. RGB color space

To utilize color as a visual cue in multimedia, image processing, computer graphics and computer vision applications, an appropriate method for representing color signals is needed. The color image acquisition process consists in obtaining three monochromatic images representing the *Red* (R), *Green* (G) and *Blue* (B) components of the observed scene, [66] and the (R,G,B) triplet leads to an unambiguous formation of a color image, (Fig. 1.6).

Electronic devices digitize and represent color images using the three basic RGB color primaries. RGB based color models employ additive primaries and represent colors (defined as vectors in the RGB space, Fig. 1.1) as their combinations. Although the RGB sensor basis is distinct from the human experience of colors, it is widely used in the color image acquisition and processing.

Mathematically, a $K_1 \times K_2$ multichannel image (Fig. 1.6) is a mapping $\mathbb{Z}^l \rightarrow \mathbb{Z}^m$ representing a two-dimensional matrix of three-component samples (pixels), $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im}) \in \mathbb{Z}^l$, where l is the image domain dimension and m denotes the number of channels, (in the case of standard color images, parameters l and m are equal to 2 and 3, respectively). Components x_{ik}, for $k = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, Q$, $Q = K_1 \cdot K_2$, represent the color channel values quantified into the integer domain ranging from 0 to $(2^B - 1)$ levels, with B bits per color channel. The process of displaying an image creates a graphical representation (Fig. 1.6) of the image matrix, in which the pixel values are assigned particular colors. When x_{ik} is large, it indicates high amount of the k-th color primary in the vector $\mathbf{x}_i$. Green color channel $\{x_{i2}\}$ is the most similar to gray scale representation x of the color image $\mathbf{x}$, because x_{i2} has the largest coefficient in the transformation to gray scales, [117].

Using the introduced nomenclature, each color pixel $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})$ is considered as a 3-dimensional vector in the RGB cube. The magnitude $M_{\mathbf{x}_i}\colon \mathbb{Z}^2 \rightarrow \mathbb{R}^+$ of vectors, defined as the square root of the sum of squares of its components, constitutes a measure of their brightness. The direction $\mathbf{D}_{\mathbf{x}_i}\colon \mathbb{Z}^2 \rightarrow \mathbb{S}^2$ of color pixel vectors, where $\mathbb{S}^2$ is a unit ball in the $\mathbb{R}^3$ space, describes their chromaticity.

The magnitude $M_{\mathbf{x}_i}$ and direction $\mathbf{D}_{\mathbf{x}_i} = (D_{x_{i1}}, D_{x_{i2}}, D_{x_{i3}})$ corresponding to the sample $\mathbf{x}_i$, for $i = 1, 2, \ldots, Q$, are defined as, [386]

$$M_{\mathbf{x}_i} = \|\mathbf{x}_i\| = \sqrt{(x_{i1})^2 + (x_{i2})^2 + (x_{i3})^2}\,, \tag{1.1}$$

$$D_{\mathbf{x}_i} = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} = \frac{\mathbf{x}_i}{M_{\mathbf{x}_i}}\,, \quad D_{x_{ik}} = \frac{x_{ik}}{\|\mathbf{x}_i\|} = \frac{x_{ik}}{M_{\mathbf{x}_i}}\,, \quad \|D_{\mathbf{x}_i}\| = 1, \quad \text{for} \quad k = 1, 2, 3\,. \tag{1.2}$$

The color chromaticity $\mathbf{C}_{\mathbf{x}_i} : \mathbb{Z}^2 \rightarrow \mathbb{T}$ may be defined as the point on Maxwell triangle $\mathbb{T}$, which constitutes a parametrization of the chromaticity space. In this way, each chrominance line is entirely determined by its intersection point with the *Maxwell plane*, [117] where $\mathbb{T}$ is a triangle in $\mathbb{R}^3$, which intersects the RGB color primaries in the RGB cube corners, (see Figs. 1.1, 1.6). Thus, the color chromaticity of the vector $\mathbf{x}_i$, for $i = 1, 2, \ldots, Q$, can be defined as the point $\mathbf{C}_{\mathbf{x}_i} = (C_{x_{i1}}, C_{x_{i2}}, C_{x_{i3}})$ with coordinates given by

$$C_{x_{ik}} = \frac{x_{ik}}{x_{i1} + x_{i2} + x_{i3}}, \quad C_{x_{i1}} + C_{x_{i2}} + C_{x_{i3}} = 1\,, \quad \text{for } k = 1, 2, 3\,. \tag{1.3}$$

1.3 Color Image Noise and its Models

Because the acquisition or transmission of digital images through sensors or communication channels is often inferred by different kinds of noise, [20, 34, 224, 232, 246, 429] one of the aims of the pre-processing techniques is its efficient removal, [111, 226, 231].

Noise attenuation is of great importance in various applications, like the enhancement of biological images (for example cDNA microarrays, [90, 95, 348]), digital restoration of images of fine arts, [39, 169, 321, 323, 325, 327] old color movies enhancement, [153, 388] quality improvement of images acquired by different sensors, and many others, (Fig. 1.7).

In many practical applications, images are corrupted by noise originating from different sources. This *mixed noise* can be caused by faulty image sensors and errors due to image capture, transmission or storage and is usually modelled by additive Gaussian noise with superimposed impulsive noise, [20, 174, 246].

Noise in an image affects its perceptual quality, diminishing not only its visual fidelity, but also decreasing the performance of the task for which it was acquired, [412]. Noise introduces random distortions into sensor readings, making them different from the ideal values and thus introducing errors and undesirable errors in subsequent stages of the image processing based system.

Noise in the image sequence may result not only from sensor malfunctions, but also from electronic interference or flaws in the data transmission procedure. Faulty sensors, optic imperfections, electronics interferences, data transmission errors or aging of the storage material may introduce noise components to digital images, (Figs. 1.2, 1.7). Considering the signal-to-noise ratio over practical communication media, there can be also degradation in quality due to the low power of the received signal. Image quality degradation can also be caused by applied processing techniques, such as aperture correction, that amplifies both high frequency signals and noise or demosaicing procedures performed in CCD sensors.

The appearance of noise and its influence on the image quality, (Fig. 1.7) is related to its statistical characteristics. Noise signals can be either periodic in nature or random. They can be described in terms of the commonly used Gaussian noise model or they can be characterized by abrupt local changes in the image data, which occur in the form of short time duration, high energy spikes attaining large amplitudes and modelled by long-tailed distributions, [59, 141, 382, 383, 430].

1.3.1 Frequently Occurring Noise Classes

As described earlier, the most common noise sources are *sensors* and *transmission channel faults*, (Fig. 1.2). Both sources introduce different kinds of noise, which cause different visual effects.

Sensor Noise

Image sensors can be divided into two categories, namely *photochemical* and *photoelectronic sensors*, [136]. In photochemical sensors e.g. positive and negative photographic films, the appearance of noise can be attributed mainly to the silver grain in the active film surface. This so-called *film grain noise*, is modelled as a Poisson or Gaussian process. In addition to the film grain noise, photographic noise is due to dust that collects on the optics and on the films during the developing process.

Photoelectronic sensors have an advantage over film in that they can be used to drive an image digitizer directly. Two basic noise models are usually associated with image acquisition sensors:

- *thermal noise* usually modelled as additive, white, zero-mean Gaussian noise [246],
- *photoelectronic noise* [231], which is produced by the random fluctuation of the number of photons on the light sensitive surface of the sensor.

The noise characteristics are strongly dependent on the type of sensors used. The typical noise generated by the most commonly used CCD cameras include, [55, 77]:

- *Shot Noise* associated with the random arrival of photons at the detector,
- *Reset Noise* caused by the conversion from the charge domain to the voltage domain by means of a sense capacitor and source-follower amplifier,
- *Output Amplifier Noise* consisting of white noise, (thermal noise also called Johnson noise) and flicker noise (also called *1/f* noise),
- *Clocking Noise* caused by the number of clocks required to transfer the signal through a CCD and process its output,

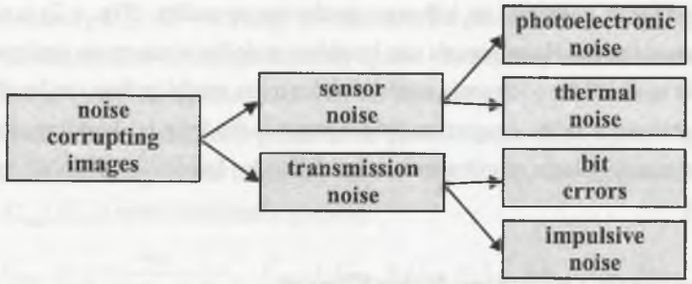


Fig. 1.2. Frequently occurring noise corrupting color images

- *Dark Current Noise* resulting from the imperfections or impurities in the depleted bulk silicon or at the silicon - silicon dioxide interface,
- *Surface Dark Current* caused by the generation centers at the sensor's surface, (these centers are surface states formed at the silicon - silicon dioxide interface),
- *Bulk Dark Current* attributed primarily to defects in the silicon, which generates dark current non-uniformity (each pixel generates a slightly different amount of dark current) and dark current shot noise (equal to the square root of the dark signal),
- *Photo Response Non-Uniformity* caused by the variation in light sensitivity of sensor's elements, which results in a faint checkerboard pattern in a flat-field image.

Taking into consideration that the general notion of noise describes the amount of random fluctuation in a given quantity, sensor noise should be considered as a 3-channel perturbation vector in the RGB color space affecting the spread of the actual RGB vectors, [375]. Such noisy samples can be characterized by a high angular distance to the neighboring samples. This can represent a strong artifact, to which the human visual system is very sensitive, [284].

Transmission Noise

The noise encountered in color images cannot always be described by the commonly assumed Gaussian model. Such noise is generated during the transmission and is frequently introduced through bit errors, [20, 49] i.e. random changes of bit values (from 1 to 0 or from 0 to 1) in an image digital representation.

Is it clear that the degree of such corruption depends on the frequency of the occurrence, as well as the bit level, that is affected by noise. It is unlikely that most of the noise in a color image will result in pure gray scale pixels. This would require the RGB color channels to be corrupted in such a way that the distorted pixels have the same color components, i.e. with full correlation among the color components. More likely is that the noisy vector will be composed

of independent channel values updated for each corrupted pixel or the noisy vectors of the RGB images will be only slightly correlated, [200, 246].

1.3.2 Noise Models

The noise modelling and evaluation of the efficiency of noise attenuation methods using the widely used test images allows the objective comparison of the noisy, restored and original images, [99].

Applying the nomenclature used in [19], in which $\mathbf{x}_i \in \mathbb{Z}^l$, $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ represents the observation (noisy) sample and $\mathbf{o}_i \in \mathbb{Z}^l$, $\mathbf{o}_i = (o_{i1}, o_{i2}, \ldots, o_{im})$ is the desired (noise free) sample, the noise corruption is modelled as

$$\mathbf{x}_i = \mathbf{o}_i + \mathbf{v}_i\,, \tag{1.4}$$

where $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{im})$ is the vector describing the noise process, (e.g. thermal noise mixed with bit errors) and $i = 1, 2, \ldots, Q$ characterizes the spatial position of the samples on the image domain. Note, that $\mathbf{v}_i$ can be described as a signal-dependent noise or as noise independent on the image signal.

Very often noise encountered in digital image processing applications has to be characterized in terms of random impulses. Thus, image filters should be able to suppress impulsive or generally heavy-tailed noise.

Impulsive Noise Models

In many practical situations, images are corrupted by noise caused either by faulty image sensors or due to transmission errors 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.

The impulsive noise is often generated by bit errors, especially during the scanning or transmission over noisy information channels. In the case of gray scale images, the model of such a corruption can be defined as, [20]

$$^*k_i^j = \begin{cases} k_i^j, & \text{with probability } 1-p, \\ 1-k_i^j, & \text{with probability } p, \end{cases} \tag{1.5}$$

where p is the bit change probability, k_i^j and $^*k_i^j$, for $j = 1, 2, \ldots, B$ are binary values $\{0, 1\}$ of B-bit original o_i sample and noisy sample x_i given by

$$o_i = k_i^1 2^{B-1} + k_i^2 2^{B-2} + \ldots + k_i^{B-1} 2^1 + k_i^B\,, \tag{1.6}$$

$$x_i = {}^*k_i^1 2^{B-1} + {}^*k_i^2 2^{B-2} + \ldots + {}^*k_i^{B-1} 2^1 + {}^*k_i^B\,. \tag{1.7}$$

In order to follow the corruption in the intensity domain, the above model can be simplified into the random valued impulsive noise given by, [20,231]

$$x_i = \begin{cases} o_i, & \text{with probability } 1-p, \\ v_i, & \text{with probability } p, \end{cases} \tag{1.8}$$

where x_i is the noisy image sample, o_i denotes the original image sample, i describes the sample location, v_i is the random value from the range $[0, 255]$ and p is the impulse probability.

In the case of color images, there exist some multichannel extensions of (1.8), [189, 232, 246]. In the first *salt & papper* noise model (NM1), the noisy signal is achieved as follows, [313,315]

$$\text{NM1}: \quad \mathbf{x}_i = \begin{cases} \{v_{i_1}, o_{i_2}, o_{i_3}\}, & \text{with probability } p, \\ \{o_{i_1}, v_{i_2}, o_{i_3}\}, & \text{with probability } p, \\ \{o_{i_1}, o_{i_2}, v_{i_3}\}, & \text{with probability } p, \end{cases} \tag{1.9}$$

where $\mathbf{x}_i$ represents the pixel in the corrupted image, $\mathbf{o}_i = \{o_{i_1}, o_{i_2}, o_{i_3}\}$ represents the original sample and $v_{i_1}, v_{i_2}, v_{i_3}$ are random, uncorrelated variables taking the value 0 or 255, with equal probability.

In the second model (NM2), [232,246,384,402,407] which reflects better the signal corruption and allows to simulate the correlation among noisy image channels, the sample distortion is given by

$$\text{NM2}: \quad \mathbf{x}_i = \begin{cases} \mathbf{o}_i, & \text{with probability } 1-p, \\ \{v_i, o_{i_2}, o_{i_3}\}, & \text{with probability } p_1\, p, \\ \{o_{i_1}, v_i, o_{i_3}\}, & \text{with probability } p_2\, p, \\ \{o_{i_1}, o_{i_2}, v_i\}, & \text{with probability } p_3\, p, \\ \{v_i, v_i, v_i\}, & \text{with probability } p_4\, p, \end{cases} \tag{1.10}$$

where p is the sample corruption probability and p_1, p_2, p_3 are corruption probabilities of each color channel, so that $\sum_1^4 p_k = 1$.[1] The impulses v_i can have either large 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. Similar corruption model can be applied using the HSV image representation, [155,402]

$$\text{NM3}: \quad \mathbf{x}_i = \begin{cases} \{H_i, S_i, V_i\}, & \text{with probability } (1-p), \\ \{v_{i_1}, S_i, V_i\}, & \text{with probability } p_1\, p, \\ \{H_i, v_{i_2}, V_i\}, & \text{with probability } p_2\, p, \\ \{H_i, S_i, v_{i_3}\}, & \text{with probability } p_3\, p, \\ \{v_{i_1}, v_{i_2}, v_{i_3}\}, & \text{with probability } p_4\, p, \end{cases} \tag{1.11}$$

[1] In this work, NM2 will generally denote the case with $p_k = 0.25, k = 1, \ldots, 4$, and $v_i = 0$ or 255.

where p denotes the degree of impulsive noise distortion, $\sum_1^4 p_k = 1$. In this noise model, v_{i_q}, $q = 1, 2, 3$ is a random variable in a small range, very close to the upper or lower bound of the pixel's HSV components.

Very often the noise model NM2, (1.10) is being simplified setting $p_4 = 1$, [183, 189]

$$\text{NM4}: \quad \mathbf{x}_i = \begin{cases} \mathbf{v}_i, & \text{with probability } p, \\ \mathbf{o}_i, & \text{with probability } 1-p, \end{cases} \tag{1.12}$$

but in this model $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3})$ is now a noisy vector of random, uniformly distributed, uncorrelated integer values, (channel intensities) in the range $[0, 255]$.

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. Therefore, an image can be corrupted by *mixed noise* according to the following model, [385]

$$\text{NM5}: \quad \mathbf{x} = \begin{cases} \mathbf{o} + \mathbf{v}_G, & \text{with probability } (1-p), \\ \mathbf{v}_I, & \text{otherwise}, \end{cases} \tag{1.13}$$

where $\mathbf{o}$ is the noise-free color signal with the additive noise $\mathbf{v}_G$ modelled as zero mean, Gaussian noise and $\mathbf{v}_I$ is the transmission noise modelled as multichannel impulsive noise, [246].

Measurement of the Color Image Quality

It is clear that subjective and objective evaluation of the image quality can be applied in processing and non-processing context as well. In the filtering and enhancement applications, subjective evaluation of image quality can be summarized into three main points, (Tab. 1.1):

- Is the noise removed ?

  This is the basic requirement of the filter design. The human visual system is very sensitive to the presence of image distortions and noise introduced into the image inhibits the correct understanding of the image information.

- Is the structural content (edges, textures and fine details) of the image preserved ?

  One of the most important criteria in the subjective evaluation of filter performance is the edge preservation. Image edges, which may be defined as discontinuities or abrupt changes in color attributes, are important features, since they provide an indication of the shape of the objects in the image. Maintaining the sharpness of the edges is as important

as the removal of the image noise. The same holds true for fine image details. An image devoid of details looks plain and unpleasant. Therefore, it is important for the filter to distinguish the fine structures from the noise, so that they can be preserved during the filtering process.

- Are there some color artifacts in the image caused by faulty processing ?

  This requirement follows the classification of any imperfection such as blocking artifacts or new color pixels, that were not present in the original (noise-free) image. The human visual system is very sensitive to changes in color. Therefore, it is important to keep the chromaticity (hue and saturation) unchanged while removing noise. The natural appearance of the color features of the scene must be preserved, while artificial contrast, color drifts and other abberations that make the filtered image look unpleasant should be avoided.

From this point of view it is evident that the noise removal tasks in color images may be understood as a process of achieving the best balance between the above-mentioned criteria. According to the image processing fundamentals, which describe the filtering as a multi-criteria task, it is necessary to use at least two objective measures that correspond to the signal-detail preservation and also express the noise attenuation capability. Moreover, certain objective criteria for the measurement of the preservation of color information should be used as well, [24, 118, 195, 283].

The *Root Mean Squared Error* (RMSE), *Signal to Noise Ratio* (SNR), *Peak Signal to Noise Ratio* (PSNR), *Normalized Mean Squared Error* (NMSE) and the *Normalized Color Difference* (NCD) are used in this work for the analysis of the efficiency of the described filters, [72, 131, 135, 145, 246, 288, 401]. The objective quality measures are defined by the following formulas

$$MAE = \frac{\sum_{i=1}^{Q}\sum_{k=1}^{m}|x_{ik}-o_{ik}|}{Qm}, \quad MSE = \frac{\sum_{i=1}^{Q}\sum_{k=1}^{m}(x_{ik}-o_{ik})^2}{Qm}, \quad NMSE = \frac{\sum_{i=1}^{Q}\sum_{k=1}^{m}(x_{ik}-o_{ik})^2}{\sum_{i=1}^{Q}\sum_{k=1}^{m}(o_{ik})^2}, \tag{1.14}$$

$$SNR = 10\log_{10}\left[\frac{\sum_{i=1}^{Q}\sum_{k=1}^{m}(o_{ik})^2}{\sum_{i=1}^{Q}\sum_{k=1}^{Q}(x_{ik}-o_{ik})^2}\right], \quad PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right), \tag{1.15}$$

where Q is the number of image pixels, and x_{ik}, o_{ik} denote the k-th component of the noisy image pixel and its estimation at a pixel position i, respectively.

Since RGB is not a perceptually uniform space, [8, 73] in the sense that differences between colors in this color space do not correspond to color differences perceived by humans, the

Score	Overall evaluation of the distortion	Noise removal evaluation
1	very disruptive	poor
2	disruptive	fair
3	destructive, but not disruptive	good
4	perceivable, but not destructive	very good
5	imperceivable	excellent

Tab. 1.1. *Subjective image evaluation guidelines*

restoration errors are analyzed using the perceptually uniform color spaces such as CIE LAB, CIE LUV and color difference criteria such as ΔE_1, ΔE_2 and *Normalized Color Difference* criteria (NCD) are commonly used, [175, 203, 246, 283]

$$\Delta E_1 = \frac{1}{Q}\sum_{i=1}^{Q}\sqrt{\left(L^*_{\mathbf{o}_i}-L^*_{\mathbf{x}_i}\right)^2+\left(a^*_{\mathbf{o}_i}-a^*_{\mathbf{x}_i}\right)^2+\left(b^*_{\mathbf{o}_i}-b^*_{\mathbf{x}_i}\right)^2}, \tag{1.16}$$

$$\Delta E_2 = \frac{1}{Q}\sum_{i=1}^{Q}\sqrt{\left(L^*_{\mathbf{o}_i}-L^*_{\mathbf{x}_i}\right)^2+\left(u^*_{\mathbf{o}_i}-u^*_{\mathbf{x}_i}\right)^2+\left(v^*_{\mathbf{o}_i}-v^*_{\mathbf{x}_i}\right)^2}, \tag{1.17}$$

$$NCD_1 = \frac{Q\,\Delta E_1}{\sum_{i=1}^{Q}\sqrt{\left(L^*_{\mathbf{o}_i}\right)^2+\left(a^*_{\mathbf{o}_i}\right)^2+\left(b^*_{\mathbf{o}_i}\right)^2}}, \quad NCD_2 = \frac{Q\,\Delta E_2}{\sum_{i=1}^{Q}\sqrt{\left(L^*_{\mathbf{o}_i}\right)^2+\left(u^*_{\mathbf{o}_i}\right)^2+\left(v^*_{\mathbf{o}_i}\right)^2}}, \tag{1.18}$$

where L^* represents lightness values and (a^*, b^*), (u^*, v^*) chrominance values corresponding to original $\mathbf{o}_i$ and noisy (filtered) $\mathbf{x}_i$ samples expressed in CIE LAB and CIE LUV color spaces. [2]

It is worth noticing that the threshold value of ΔE_1 criteria established in [203, 283] at around 2.3 characterizes the limit of human sensitivity to color distortion. In terms of the difference between two colors, the human visual system is not capable to recognize a color difference smaller than this threshold value and therefore the designed filters should decrease the color difference between original and filtered samples just to this value or below.

In general, the criteria such as ΔE_1, ΔE_2 and respective NCD values, express well the measure of color difference or chromaticity preservation, however they do not measure the noise attenuation capability and signal-detail preservation of the noise filtering schemes. For that reason, it is necessary to combine ΔE_1, ΔE_2 based criteria with measures such as MSE, SNR, PSNR etc. computed in the RGB or other color spaces. It can be easily observed that one filter can produce low values of MSE, whereas its color chromaticity preservation capability (expressed through NCD) can be significantly worse, and vice versa.

Although quantitative measures, such as ΔE_1, ΔE_2 and NCD are close approximations of the perceptual error, they cannot exactly characterize the quite complex attributes of human perception. Therefore, an alternative subjective approach shown in Tab.1.1 is commonly used for the estimation of the perceptual image quality, [246].

[2] In this work the NCD defined on the CIE LUV color space will be used.

1.4 Color Image Filtering Designs

The filtering of image noise is an important part of any image processing system, whether the final image is utilized for manual interpretation or for automatic analysis and therefore a plethora of filtering techniques have been proposed in the literature, [230, 231, 234], (Fig. 1.4).

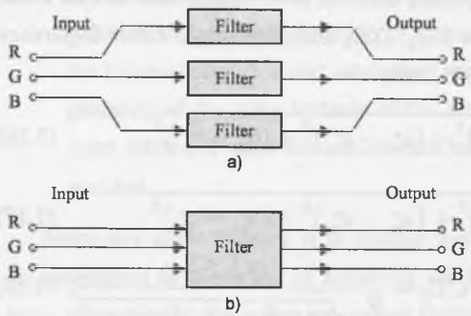


Fig. 1.3. Marginal (a) and vector processing (b)

It is clear that there are some significant aspects, which influence the design and selection of the appropriate filtering technique. A good filter for processing of color images should be designed mainly with respect to the *trichromatic nature of color image*, its *nonlinear characteristics* and *statistics of noise corruption*. According to the trichromatic nature of color, the color image processing techniques can be divided into two main classes:

- **Marginal (componentwise) methods**, [257, 430]

This framework operates on each color channel separately, (Fig. 1.3a). Since each processing step is usually accompanied with a certain inaccuracy, making the output values different from the desired ones, ignoring the correlation which exists between the RGB channels, the projection of separately processed color channels into the color image output usually results in perceivable color artifacts. Componentwise processing is appropriate in the case of highly decorrelated color spaces (e.g. YC_BC_R used in the digital television, YUV in the Pal/Secam television format, YIQ in the NTSC television format or opponent color spaces, [66, 200, 311].

- **Vector methods**, [243, 246]

In the vector processing of color images, the input samples are processed as a set of vectors. Since natural images are characterized by high correlation between their RGB components, this is an important feature that predetermines the success of vector processing.

An example of the distorsion caused by componentwise processing is shown in Fig. 1.5, where impulsive noise has been added to a signal component and then the channels were separately filtered by a median of length 5. The filtering removes the impulses on flat signal areas but causes the edge shift to the left, if there is an impulse in front of it. As a result of the edge shift, the output color sample will not be one of the inputs, [19, 220].

Besides the preservation of image colors, the filtering operators are required to preserve salient image features, such as edges and texture and of course to remove noise. The most common approach to the problem of noise reduction is the utilization of some kind of smoothing operation, which filters out random fluctuations due to noise. The rationale of this approach is the need to determine suitable values of image pixels, which are statistically close to the original, uncorrupted color image signal, [231].

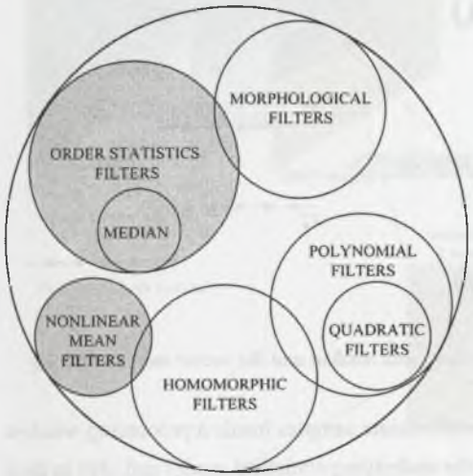


Fig. 1.4. Family of nonlinear filters

The smoothing approach is based on a special type of sliding (moving, running) window $W = \{\mathbf{x}_k \in \mathbb{Z}^l,\ k = 1, 2, \ldots, N\}$, which usually affects one image sample (mostly the sample $\mathbf{x}_1$ placed in the center of the window) at a time, changing its value by some function of a local neighborhood $\{\mathbf{x}_2, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ determined by W. Thus, the value of the estimated sample depends on the values of image samples in its neighborhood and the window operator slides over the image to process individually all the image pixels.

It should be emphasized that the window size N influences considerably the performance of the filters. If a window size is large, the filtering techniques operate on a large supporting area and in general they efficiently attenuate image noise. On the other hand, their detail preservation capability is low, which results in image blurring. It has been widely observed [139, 231], that for small image corruption a 3×3 square filter window provides the best accuracy of the local information estimation to achieve the trade-off between the noise smoothing and the image detail preservation.

Following the robust estimation and order statistic theory, [78, 233] the most popular multichannel filtering class operating on a window, sliding over the image domain, is based on sample ordering. Performing the scalar ordering operation on a gray scale image, the atypical image samples, are moved to the borders of the ordered set. Thus, the center of the ordered sequence known as a median, [36, 231] represents the sample, which has the largest probability to be noise-free. The direct application of the median filter (marginal filter, [257, 430]) to the RGB color channels leads however to strong color artifacts, (Fig. 1.5).

If the noise corrupting the image is of impulsive nature, [20, 246] e.g. bit errors and outliers, filtering approaches based on the order statistic theory are often employed, [20, 208, 231, 233].

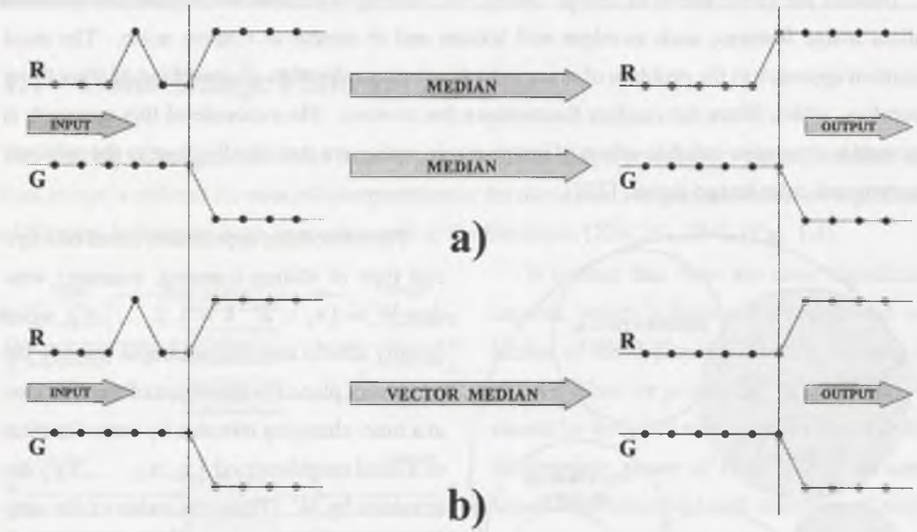


Fig. 1.5. Illustration of the difference between the marginal median and the vector median filtering

These nonlinear filters operate by ordering the multivariate samples inside a processing window and their popularity lies in the ability to match the underlying statistical model and also in their computational simplicity.

In the vector case, outliers are associated with the maximum extremes of the aggregated distances to other input samples in the sliding window. For this reason, the output of the vector filters based on ranking, is defined according to a specific ordering technique, [35,123,232,385] as the lowest ranked vector in a predefined sliding window. Since the lowest ranked vector is the sample of the input set, vector filters do not generate new color samples (color artifacts) and such behavior is desirable due to the correlation that exists between the RGB channels. The ordering scheme has been adopted by the most popular vector filters such as *Vector Median Filter*, [19] and *Vector Directional Filters*, [397].

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 impulsive and Gaussian noise. 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.

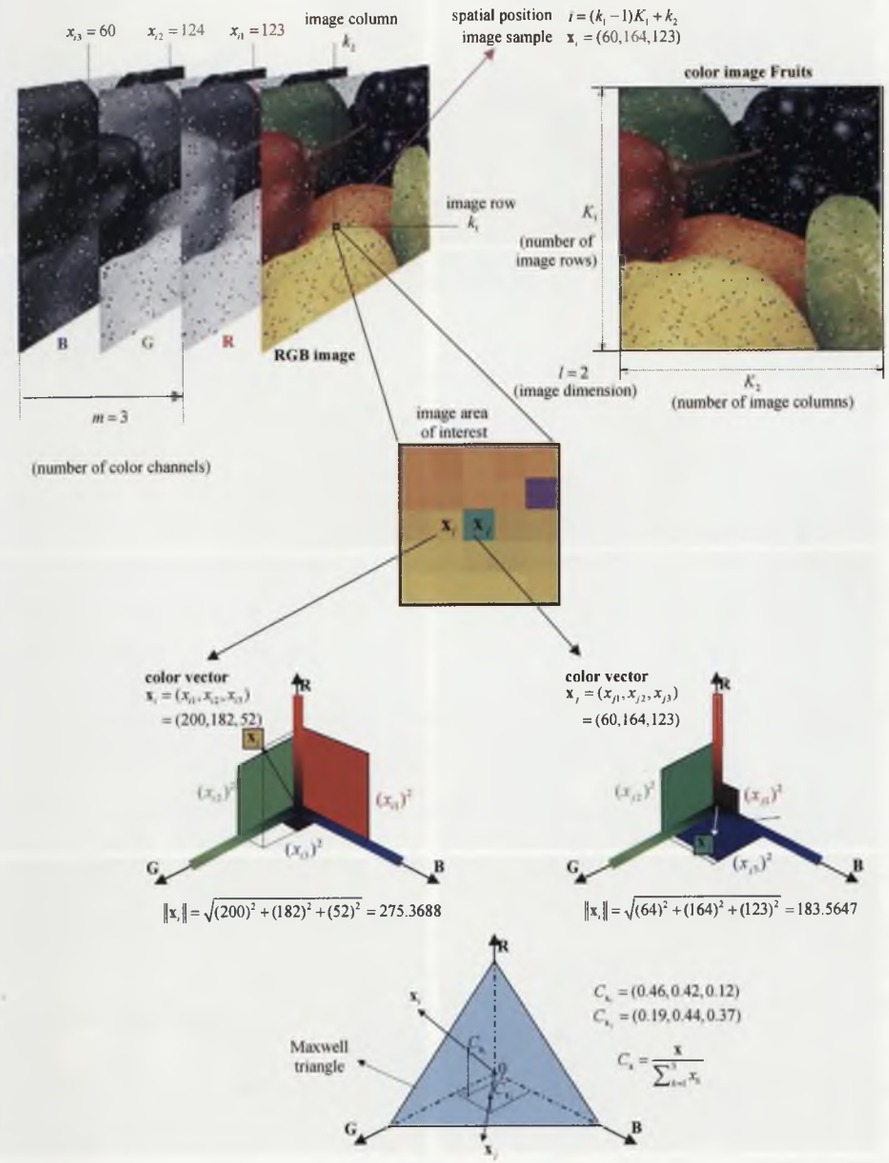


Fig. 1.6. Color image processing fundamentals

*Fig. 1.7. Examples of noise appearing in: **a)** scanned images of fine arts, **b)** digital photographs, **c)** digital images corrupted by transmission errors, (some of the image rows are distorted), **d)** DNA microarray images, (two-channel images consisting of Red and Green components only). Each of the examples consists of an image and a zoomed part to better visualize the noise distorsions*

2

Adaptive Noise Reduction Filtering

In this Chapter, various adaptive image filtering techniques, primarily developed for the noise reduction in gray scale images are examined. The presented concepts can be often extended to the multichannel case and directly applied as componentwise filtering techniques, capable of removing noise in color images. As will be shown in the third Chapter, very often the scalar techniques can be reformulated using the vector approach, to exploit the inherent correlation among image channels.

The second part of this Chapter is focused on the order statistics based filters and presents the optimization of the weighted median filters, which will be used in the last Chapter of this book.

2.1 Weighted Averaging Framework

NOISE suppression in digital images has been a topic of considerable interest in the past decades, due to its importance in numerous applications in various fields of computer vision. The most frequently used noise reducing transformations are the linear filters, which are based on the convolution of the image with a filter kernel of fixed coefficient. This kind of filtering replaces the central pixel x_1 from the pixel set $\{x_1, x_2, \ldots, x_N\}$, (Fig. 2.1) belonging to the filter mask W containing N image pixels, with the weighted mean of the samples inside W, [118,207,369]. The filter output y is given by the convolution of the filtering kernel determined by the weight coefficients $\{\psi_1, \psi_2, \ldots, \psi_N\}$ with the pixels in W

$$y = \frac{1}{S}\sum_{k=1}^{N} \psi_k x_k\,, \quad S = \sum_{k=1}^{N} \psi_k\,. \tag{2.1}$$

a)

x_2	x_3	x_4
x_5	x_1	x_6
x_7	x_8	x_9

b)

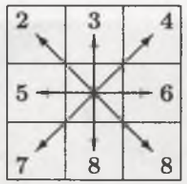


Fig. 2.1. The filtering mask of size 3×3, with the pixel x_1 in the center (a) and the directions between the central pixel and its neighbors (b)

Linear filters are relatively simple and fast, but their major drawback is that they cause blurring of the edges and suppress tiny details. This effect can be diminished incorporating a certain kind of nonlinearity and adaptiveness into the weight coefficients, performing the averaging in a selected neighborhood determined by the shape of the filter mask. The term *adaptive* means that the weights change according to the image structure, which is to be smoothed. In this way the *adaptive smoothing* can be seen as a nonlinear process capable of noise removal, while preserving important image features, [120].

Different kinds of edge and structure preserving adaptive filter kernels have been proposed in the rich literature on this subject, [43, 70, 109, 118, 150, 250, 421, 427]. One of the simplest nonlinear schemes works with a filter kernel weights expressed as [1]

$$\psi_k = 1 - |x_1 - x_k|, \text{ for } k = 1, 2, \ldots, N. \tag{2.2}$$

This filter takes with greater weighting coefficients those pixels of the neighborhood, whose intensity are close to the intensity of the central pixel x_1

$$y = \frac{1}{S}\sum_{k=1}^{N}(1 - |x_1 - x_k|)\, x_k\,, \quad S = N - \sum_{k=2}^{N}|x_1 - x_k|\,. \tag{2.3}$$

This improves dramatically the detail preserving capability of the *Averaging Filtering* (AF) scheme (2.1). Other similar designs, [164, 206, 275, 409, 410] do not take the central pixel into consideration, which leads to a much more robust filter performance.

The *gradient inverse weighted operator* employs a similar structure and forms a weighted mean of the pixels belonging to the filter window. Weighting coefficients depend on the difference of the gray scale values between the central pixel and its neighbors in W, [171, 409, 410]

$$y = \frac{1}{S}\sum_{k=1}^{N}\frac{x_k}{\max\{\gamma, |x_1 - x_k|\}}\,, \quad S = \sum_{k=1}^{N}\frac{1}{\max\{\gamma, |x_1 - x_k|\}}\,, \tag{2.4}$$

where γ is a parameter, which influences the degree of contribution of the central pixel to the weighted sum.

[1] We assume here that the pixel values are normalized to the range $[0, 1]$.

Local statistic filters constitute a class of linear minimum mean squared error estimators, based on the non-stationarity of the signal and the noise model, [161, 164–166]. These filters make use of the local mean and the variance of the input set W and define the filter output as

$$y = \hat{x} + \alpha\,(x_1 - \hat{x}) = \alpha x_1 + (1 - \alpha)\hat{x}\,, \tag{2.5}$$

where $\hat{x}$ is the arithmetic mean of the image pixels belonging to the filter window W and α can be defined as the normalized correlation between two images: $x_1 = o + v_1$, $x_2 = o + v_2$,

$$\alpha = \frac{\langle x_1 - \langle x_1\rangle\rangle\,\langle x_2 - \langle x_2\rangle\rangle}{\left(\langle |x_1 - \langle x_1\rangle|^2\rangle\right)^{\frac{1}{2}}\left(\langle |x_2 - \langle x_2\rangle|^2\rangle\right)^{\frac{1}{2}}}\,, \tag{2.6}$$

estimated through, [173, 231, 374]

$$\alpha = \frac{\sigma_x^2}{\sigma_n^2 + \sigma_x^2}\,, \quad \hat{x} = \frac{1}{N}\sum_{k=1}^{N}x_k\,, \quad \nu^2 = \frac{1}{N}\sum_{k=1}^{N}(x_k - \hat{x})^2\,, \tag{2.7}$$

$$\sigma_x^2 = \max\left\{0, \nu^2 - \sigma_n^2\right\}\,, \quad \alpha = \max\left\{0, 1 - \frac{\sigma_n^2}{\nu^2}\right\}\,, \tag{2.8}$$

where $<\cdot>$ denotes the expected value, ν is the local variance calculated from the samples in the filter window and σ_n^2 is the estimate of the variance of the noise process. If $\nu \gg \sigma_n$, then $\alpha \approx 1$ and practically no changes are introduced. When $v < \sigma_n$, then $\alpha = 0$ and the central pixel is replaced with the local mean. In this way, the filter smooths with the local mean, when the noise is not very intensive and leaves the pixel value unchanged when a strong signal activity is detected. The major drawback of this filter is that it fails to remove impulses and leaves noise in the vicinity of high gradient image features. Equation (2.5) can be rewritten as, [374]

$$y = \alpha x_1 + (1 - \alpha)\hat{x} = \frac{1 - \alpha}{N}\left(\psi_1 \cdot x_1 + x_2 + \ldots + x_N\right)\,, \quad \psi_1 = \frac{1 - \alpha + N\alpha}{1 - \alpha}\,, \tag{2.9}$$

and in this way the local statistic filter (2.5) is reduced to the *central weighted average*, with an adaptive weighting coefficient ψ_1.

In [268, 269] a powerful adaptive smoothing technique related closely to the *anisotropic diffusion*, which will be discussed in Chapter 4, was proposed. In this approach, the central pixel x_1 is replaced by a weighted sum of all pixel contained in the filtering mask W

$$y = \frac{1}{S}\sum_{k=1}^{N}\psi_k\, x_k\,, \quad \text{with} \quad S = \sum_{k=1}^{N}\psi_k, \quad \psi_k = \exp\left\{-\frac{g_k^2}{\beta^2}\right\}\,, \tag{2.10}$$

where g_k is the magnitude of the gradient calculated in the local neighborhood of the pixel x_k.

In [170] an adaptive smoothing technique based on the estimation of the directional gradients was proposed, (Fig. 2.2)

$$y = \left(1 - \sum_{k=2}^{N}\psi_k\right)x_1 + \sum_{k=2}^{N}\psi_k x_k\,, \tag{2.11}$$

where

$$\psi_k = f(g_k),\ g_k = |x_k - x_1|,\ k > 2, \qquad f(\chi) = \begin{cases} 1 - 2\,(\chi/\alpha)^2 & \text{if} \quad |\chi| \le \alpha/2, \\ 2\,(\chi/2 - 1)^2 & \text{if} \quad \alpha/2 \le |\chi| \le \alpha, \\ 0 & \text{if} \quad |\chi| \ge \alpha. \end{cases} \tag{2.12}$$

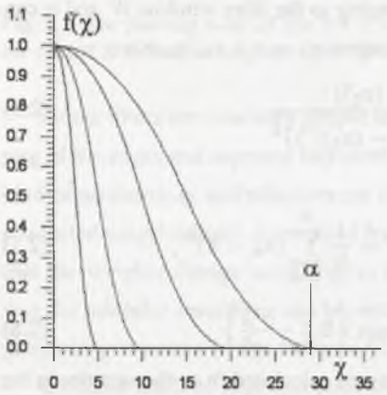


Fig. 2.2. Shape of the functions used in [170], (2.12)

Another efficient adaptive scheme, [290, 300, 301, 317, 318, 392] has been proposed as

$$y = \frac{1}{S}\sum_{k=2}^{N} x_k \exp\left\{-\frac{\rho_k^2}{\beta_1^2}\right\} \exp\left\{-\frac{|x_k - x_1|^2}{\beta_2^2}\right\}, \tag{2.13}$$

where ρ_k denotes the topological distance between the central pixel x_1 and other pixels in W and β_1, β_2 are filter parameters. The important feature of this algorithm is that it excludes the central pixel x_1 from the averaging.

Good results of noise reduction can usually be obtained by performing the *σ-filtering*, [25,166,172,421]. This procedure computes a weighted mean over the filter window, but only those pixels whose values lie within $\gamma \cdot \sigma$ of the central pixel intensity are taken into the average. In this way, this filter attempts to estimate a new pixel value with only those neighbors, whose values do not deviate too much from the value of x_1

$$y = \frac{1}{S}\sum_{k=1}^{N} \psi_k x_k, \quad \text{for } k : |x_k - x_1| \le \gamma\sigma, \tag{2.14}$$

where S is the normalizing factor, $\{\psi_1, \psi_2, \ldots, \psi_N\}$ are weighting coefficients, γ is a design parameter, (typically $\gamma = 2$) and σ is the standard deviation of pixels belonging to the local window W or the value of the standard deviation estimated from the whole image.

Similar idea lies behind the filtering scheme defined as, [250]

$$y = \begin{cases} \hat{x}, & \text{if} \quad |x_1 - \hat{x}| > \delta, \\ x_1, & \text{otherwise}. \end{cases} \tag{2.15}$$

If the difference between the central pixel and the local mean exceeds a specific threshold (design parameter) δ, this filter determines as the output, the mean value of the pixels in W, otherwise the central sample remains unchanged.

Maximum homogeneity neighbor filters, [163] divide 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 the mean value of the pixels belonging to the sub-window, in which the variance reaches the minimum, [27].

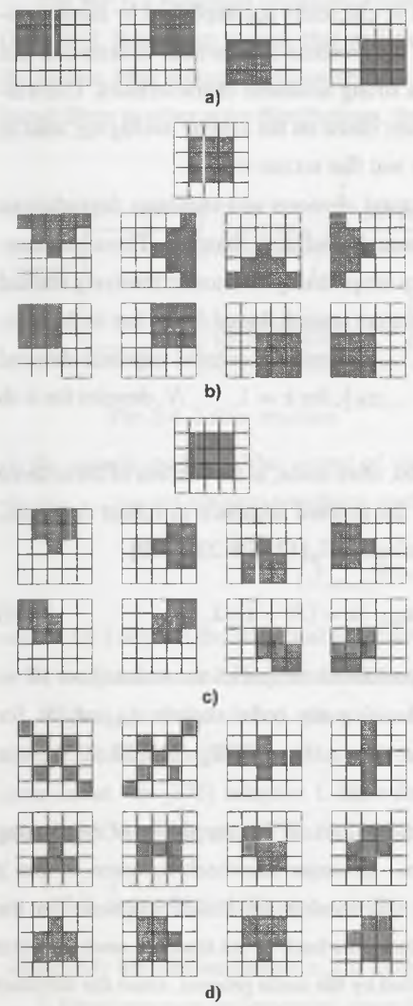


Fig. 2.3. Different sub-window structures used in the filtering frameworks proposed in [163, 213] a), [213, 214] b, c) and [368], d)

The filter introduced in [163,260,393] divides the 5×5 filtering mask into four sub-windows as depicted in Fig. 2.3a). In each of the subwindows, the mean and the variance is calculated and the output of the filter is the mean value of the pixels from that square, which has the smallest variance. To improve the efficiency of these filters, another window placed in the middle of the filtering can be added [393] and the partition of the filtering window into nine sub-windows can be performed, [213, 214], Fig. 2.3b), c). Similar filtering structure has been proposed in [368], (Fig. 2.3d). This approach is in some way similar to the technique which will be described in Chapter 5, in which the filters based on *digital paths* are introduced. Instead of looking for subwindows with similar pixels, this novel technique investigates digital paths linking the central pixel with pixels belonging to the filter window of size determined by the practitioner, [313, 322].

The *k-nearest neighbor filter* proposed in [81] replaces the gray level of the central pixel x_1 by the average of its τ neighbors whose intensities are closest to that of x_1, [25], ($\tau = 6$ and a window of size 3×3 was recommended in [206]).

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*).

2.2 Order Statistic Based Filters

It is evident that many image processing tasks cannot be efficiently accomplished by linear techniques. Image signals are nonlinear in nature due to the presence of structural information and are perceived by the human visual system, which has strong nonlinear characteristics. Unfortunately, most of the linear and also nonlinear techniques based on the *sample averaging*, tend to blur structural elements such as lines, edges, corners and fine texture details.

Nonlinear methods can preserve important structural elements and eliminate degradations occurring during signal formation or transmission through nonlinear channels. Therefore, nonlinear filters based on the robust *order statistics theory* are probably the most extensively studied class of image processing filters. Nonlinearity of the order statistic based filters lies in the sample ordering, which transforms the input set $\{x_1, x_2, ..., x_N\}$ into the ordered sequence denoted as $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(N)}$, where $x_{(k)} \in \{x_1, x_2, \ldots, x_N\}$, for $k = 1, \ldots, N$, denotes the k-th order statistic.

The ordering operation moves the atypical samples, often noise, to the borders of the ordered set and provides the middle positioned samples of the ordered sequence as robust estimates. Based on this property, the *Median Filter* (MF) is defined as [2], [12, 108, 231, 279]

$$y = MED\{x_1, x_2, ..., x_N\} = x_{(\mu)}, \; \mu = (N+1)/2\,, \tag{2.16}$$

where $x_{(\mu)}$ is the middle order statistic or middle positioned sample of the ordered set $\mathcal{W} = \{x_{(1)}, x_{(2)}, \ldots, x_{(N)}\}$, $x_{(k)} \leq x_{(j)}$, for $k < j$. Choosing any order statistic $x_{(k)}$ of $\mathcal{W}$, for $k = 1, 2, \ldots, N$, constitutes the output of the *Rank Order Filter* (ROF), [13, 20, 233]. It is evident that the MF is a special case of the ROF, for $k = \mu$.

The MF filter is the most commonly used selection filters. It has the ability of attenuating strong impulse noise, while preserving sharp edges. Its major drawback however, is that it wipes out the 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. As a result, after the application of the median filter, the image noise is removed, but details are lost and artifacts like streaks and blotches are produced. Additionally, the median filtered image is prone to edge jittering, when the noise ratio is high. Therefore a trade-off between the suppression of noise and

[2] We assume a filtering mask with odd number of samples N.

the preservation of fine image details and edges has to be achieved. 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, [34].

The MF is a *maximum likelihood estimate* (MLE) of location for the Laplacian distribution, [122, 231]. It has been proven, that statistical properties and the robust smoothing capability of the median filter makes it very suitable for impulsive noise filtering. To adapt the order statistic based filters to other noise distributions, the so called L filters have been introduced, (Fig. 2.4) .

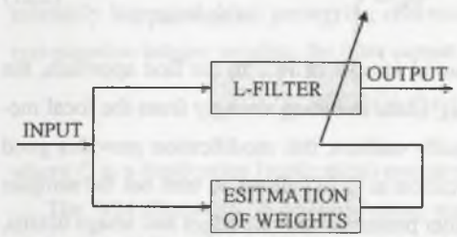


Fig. 2.4. L-filter structure

The L filters are estimators achieving the compromise between the nonlinear operation given by the sample ordering and linear operation given by the weighting of the sample data. The L filter output is achieved as a weighted sum of the ordered data. Thus, the corresponding mean squared error will always be less than or equal to that achieved with the sample mean or the sample median. The output of the L filters is defined as $y = \sum_{k=1}^{N} \psi_k x_{(k)}$, where $\psi_1, \psi_2, \ldots, \psi_N$ are nonzero weighting coefficients, which can be optimized by minimizing

$$E\left[\sum_{k=1}^{N} \left(\psi_k x_{(k)} - o_k\right)^2\right] = \boldsymbol{\psi}^T \mathbf{R} \boldsymbol{\psi}\,, \tag{2.17}$$

where $E[\cdot]$ denotes the statistical expectation, o signifies the desired signal, $\boldsymbol{\psi} = \{\psi_1, \ldots, \psi_N\}$ is the weight vector and $\mathbf{R}$ is the correlation matrix of ordered noise components $\{v_{(1)}, \ldots, v_{(N)}\}$.

A great advantage of the L filter class is that for a known noise distribution, it is possible to choose the filter weights in such a way, that it becomes the optimal filter in the mean squared error sense. In [375] adaptive L-filter structures exploiting temporal information have been introduced. In this approach the filter takes the form

$$y = \frac{1-\alpha}{N}\left(x_{(1)} + x_{(2)} + \ldots + \psi_\mu \cdot x_{(\mu)} + \ldots + x_{(N)}\right) = \alpha x_{(\mu)} + (1-\alpha)\hat{x}\,, \tag{2.18}$$

where only the median sample $x_{(\mu)}$ is assigned a weight ψ_μ.

L filters represent an important generalization of MF, ROF, and α-trimmed mean filters. The α-trimmed mean filter has been introduced as the compromise between the median filters and linear filters, since the AF suppresses additive Gaussian noise better than the MF and the MF has better impulsive noise characteristics. The output of the α-trimmed mean filter is given by

$$y = \frac{1}{N-2\alpha} \sum_{k=\alpha}^{N-\alpha+1} x_{(k)}\,. \tag{2.19}$$

In order to improve its signal-detail preserving characteristics and provide estimates closer to the MF output, two modifications of α-trimmed mean filters have been introduced

$$y = \sum_{k=1}^{N} \psi_k x_{(k)} \Big/ \sum_{k=1}^{N} \psi_k \,, \tag{2.20}$$

where weight coefficients $\psi_1, \ldots, \psi_N$ are chosen as

$$\psi_k = \begin{cases} 1, & \text{if } |x_k - x_{(\mu)}| \le \delta_1, \\ 0, & \text{otherwise,} \end{cases} \quad \text{or} \quad \psi_k = \begin{cases} 1, & \text{if } |x_k - x_1| \le \delta_2, \\ 0, & \text{otherwise,} \end{cases} \tag{2.21}$$

where $x_{(\mu)}$ is the local median and x_1 is the central sample of W. In the first approach, the amount of trimming depends on the parameter δ_1, (data deviating strongly from the local median are trimmed out). Since such data are usually outliers, this modification provides good noise attenuation properties. The second modification in (2.21) allows to trim out the samples deviating strongly from the central pixel. Such filter preserves well the edges and image details, however in some applications its noise attenuation capability can be inefficient, [308].

Wilcoxon filters are the most important filter family in the large class of R estimators. Since the output of Wilcoxon filter is defined as

$$y = MED\left\{[x_{(k)} + x_{(j)}]/2, \ \text{ for } 1 \le k \le j \le N\right\} , \tag{2.22}$$

these filters are effective in the removal of additive Gaussian noise. However, they do not preserve well edges because this filter structure is based on the averaging operation. A further disadvantage of the Wilcoxon filter is its high computational complexity. It can be decreased by introducing a parameter δ which leads to its simplified structure

$$y = MED\left\{[x_{(k)} + x_{(j)}]/2, , \text{ for } 1 \le k \le j \le N; \ (j-k) \le \delta\right\} . \tag{2.23}$$

By varying the range parameter δ from 1 to the window size N, the modified Wilcoxon filter can perform a wide range of smoothing operations from median to the standard Wilcoxon filter.

One of the main disadvantages of the L filters and the above mentioned order statistic based filter classes is that the ordering destroys the information about the local neighborhood structure. Therefore, their performance can be inefficient for larger window sizes. Also, the lower performance is particularly evident in the case of non-stationary signals. Therefore, a modification of L filters, which takes into account the information about the neighborhood has been developed, (L1 filters). Later, this idea has been extended to the nonlinear filters operating on permutation lattices, [14, 30–33, 125].

Combining the rank-temporal relations with the linear nature of the weighted averaging, makes that Ll filters are capable of removing additive Gaussian, impulsive and mixed noise in

digital images. However, these filters similarly as L filters, averaging based filters, Wilcoxon filters, α-trimmed mean filters, etc., produce new samples, which can increase local distortion of digital images. Therefore, *selection filtering* classes represent a better choice, especially for images corrupted by impulsive noise. These filters select the output as one of the samples belonging to W and their efficiency depends on the applied selection mechanism.

Weighted median (WM) filters constitute a class of the most natural selection filters. These filters have been developed as an extension of the median filter and are characterized by significantly improved detail preserving characteristics, [54, 264]. In the WM filtering, based on non-negative integer weights, the filter output is given by

$$y = MED\{\psi_1 \diamond x_1, \psi_2 \diamond x_2, \ldots, \psi_N \diamond x_N\} , \tag{2.24}$$

where $\diamond$ is a duplication (replication) operator defined as $w \diamond x_k = \overbrace{x_k, x_k, \ldots, x_k}^{w \text{ times}}$.

The WM filters can be designed using non-negative real weights. In such a case, the WM output $y \in W$ minimizes the expression $L(y) = \sum_{k=1}^{N} \psi_k |x_k - y|$. If $\psi_k \ge 0$, for $k = 1, \ldots, N$, and the function $L(y)$ is piecewise linear and convex, then y is the sample from the input set. In the case of positive real weights, the computation of WM filter output requires the ordering of the input samples and a successive summing up of the upper weights corresponding to ordered samples, until the sum exceeds half of the total sum of the weights. The WM filter output is the sample corresponding to the last added weight, [424].

To adapt the weight coefficients to varying signal and noise statistics, the WM adaptation algorithms, which originate from stack filter framework, [20, 21, 65, 105, 162, 171, 249] have been developed, [422–424]. The aim of the optimal WM filtering is to find a WM filter with the window size N, for which the error criteria such as mean absolute error MAE or mean squared error MSE between the filter output y and the desired output o is minimized.

Recently, the WM scheme has been extended, [15, 97] by assigning negative integer weights to the input samples. In this way, the WM filters can be also used as a sharpening filter class.

To avoid problems connected with searching for optimal parameters, a simple and powerful class of the *Lower-Upper-Middle* (LUM) smoothers has been introduced. The LUM smoothers are a subset of the LUM filters, [124] which can be designed to simultaneously perform smoothing and sharpening operations, [126, 179, 181]. The output of the LUM smoother is given by

$$y = MED\left\{x_{(k)}, x_1, x_{(N-k+1)}\right\} , \tag{2.25}$$

where $k = \{1, 2, \ldots, \mu\}$ denotes the smoothing parameter, x_1 is the central sample of the input set, $x_{(k)}$ is the lower and $x_{(N-k+1)}$ is the upper order statistic such that $x_{(k)} \le x_{(N-k+1)}$, (Fig. 2.5a). Other similar design assigns the median value to the central pixel if its rank in the ordered sequence is lower than k or higher than $N - k + 1$, (Fig. 2.5b).

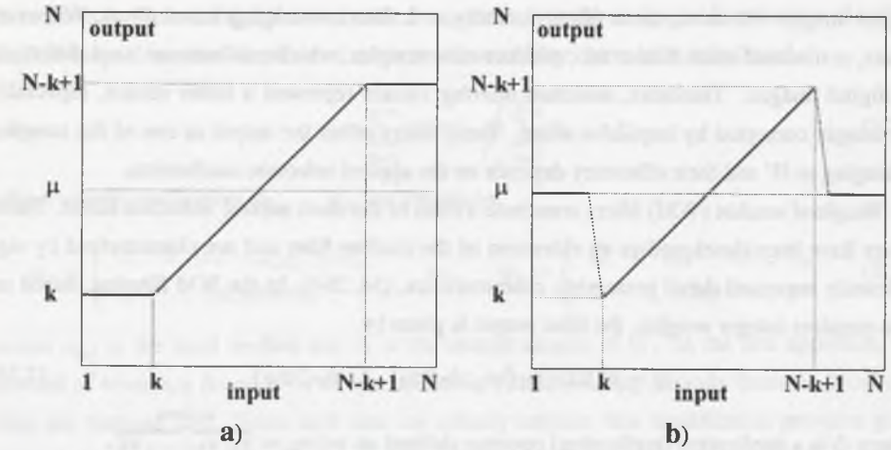


Fig. 2.5. Illustration of the construction of the LUM filter defined by (2.25) (a) and the rank conditioned filter, which assigns the median value μ to the samples with ranks lower than k or higher than $N-k+1$ (b), (adapted from [125])

The LUM smoother can be equivalently expressed as the *Central Weighted Median* (CWM) filter, [124, 151, 179] defined by

$$y_k = MED\{\psi_1 \diamond x_1, x_2, \ldots, x_n\} = MED\left\{\{x_1, x_2, ..., x_N\} \cup \left\{\overbrace{x_1, x_1, \ldots, x_1}^{\psi_1 - 1}\right\}\right\}, \quad (2.26)$$

where ψ_1 is the weight associated with the central sample x_1. The relationship between ψ_1 in (2.26) and k in (2.25) is given by: $\psi_1 = N - 2k + 2$.

It has been proven that definition (2.25) is more advantageous and useful than (2.26), especially in terms of the computational complexity, [124] and the filter analysis, [178]. On the other hand, the CWM definition (2.26) is widely used due to the popularity of the weighted median framework, [67, 68].

Using the more comprehensible form (2.25), the comparison of the lower x_k and the upper $x_{(N-k+1)}$ order statistic with the central sample x_1 from the filter window, forms the LUM smoothing operation. If x_1 lies in a range formed by these order statistics, it is not modified. However, if x_1 lies outside this range, it is replaced with a sample that lies closer to the median $x_{(\mu)}$. Varying the filter parameter k, the amount of smoothing performed by the LUM smoother can range from no smoothing, equivalent to the identity operation ($k = 1$) to the maximum amount of smoothing provided by the median filter, ($k = \mu$). The first case preserves the central sample x_1, whereas the last one often results in image blurring. Therefore, the intermediate values of k can provide a better trade-off between the smoothing and detail preserving LUM characteristics.

To employ an adaptive selection of k or ψ_1 in dependence on the local image statistics of the samples in W, a variety of selection mechanisms have been proposed to date, [68, 179, 185, 215]. However, their common drawback lies in their low design flexibility. Therefore, the switching median filters, [67, 96, 411, 429] based on the compromise between the identity operation and the robust MF, represent an interesting alternative, (Fig. 2.6).

The switching median filters can be viewed as an adaptive two-level LUM smoothers providing the maximum and the minimum amount of smoothing and the employed switching rule is often defined by

$$\begin{cases} \text{if } \varsigma \geq \delta & \text{then } x_1 \text{ is impulse,} \\ & \text{else } x_1 \text{ is noise} - \text{free,} \end{cases} \quad (2.27)$$

where δ is a threshold value and ς denotes a simple relationship, (usually absolute difference) between the central sample x_1 and the samples inside W. If $\varsigma \geq \delta$, then x_1 is considered as noisy and is being estimated by an appropriate filter. Otherwise, x_1 is declared to be noise-free and is being retained. This scheme confines the filter influence only to noisy samples and therefore significantly reduces the estimation error of the output image, [68, 151, 223].

Another important filtering class based on the order statistics is given by the permutation filters, [30, 32] and their extensions such as *Rank-Conditioned Rank-Selection* (RCRS) filters, [125, 171] and extended *permutation filters*, [31]. These filters utilize the whole potential of the permutation group theory and the information about the rank and spatial (temporal) position of the samples inside W. However, due to extreme computational complexity, their practical use is significantly limited, although a variety of methods for reduction of permutation group complexity have been introduced, [32].

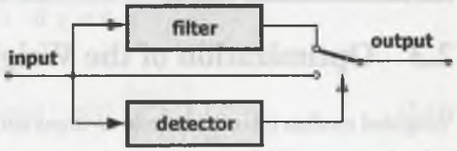


Fig. 2.6. Switching filtering concept

The majority of median filter modifications are implemented uniformly across the image, thus they modify also pixels that are undisturbed by noise. As a result, they still tend to remove details from the image or leave impulsive noise samples. To avoid excessive blurring of images during filtering process, the *Signal Dependent Rank-Ordered Mean* (SD-ROM) filter has been proposed, [3, 4, 209, 210]. In the SD-ROM approach, the filtering operation is conditioned on the differences between the input pixels and remaining rank-ordered pixels in the sliding window.

In this design, a vector containing neighbors x_k of x_1 from window W of size 3×3 is constructed. Assuming the ordering of neighbors of x_1: $x_{(2)} \leq x_{(3)} \leq, \ldots, \leq x_{(9)}$, a rank-

ordered mean $m_r = (x_{(5)} + x_{(6)})/2$ and rank ordered differences $[\varsigma_2, \varsigma_3, \varsigma_4, \varsigma_5]$ are defined as

$$\varsigma_k = \begin{cases} x_{(k)} - x_1 & \text{if } x_1 \le m_r, \\ x_1 - x_{(N-k)} & \text{if } x_1 > m_r, \end{cases} \quad k = 2, \ldots, 5. \tag{2.28}$$

The rank-ordered differences provide the information about the likelihood of corruption for the current pixel. The purpose of the impulse noise detector is to determine whether the current pixel is corrupted or noise-free. If a signal sample is detected as corrupted, it is replaced with an estimation of the true value, based on the order statistics of the remaining pixels in the processing window W, otherwise it is kept unchanged.

The SD-ROM filter output is defined as

$$y = \begin{cases} m_r, & \text{if } \varsigma_k > \delta_k, \quad \text{for} \quad k = 2, \ldots 5, \\ x_1, & \text{otherwise}, \end{cases} \tag{2.29}$$

where $\delta_2, \delta_3, \delta_4, \delta_5$ are threshold values, $(\delta_2 < \delta_3 < \delta_4 < \delta_5)$. In other words, if the algorithms detects x_1 as a noisy sample, and any of the four thresholds is exceeded, the central pixel is replaced by the rank-ordered mean m_r, otherwise, it is kept unchanged. It is worth noticing that this filtering procedure excludes the central pixel from the operation window.

2.3 Optimization of the Weighted Median Filters

Weighted median filters constitute an important nonlinear filtering class, [20,233]. Their robust smoothing capabilities in noisy environments and flexible design, [226] in conjunction with an optimization framework, [422,423] make this filtering class extremely attractive. Moreover, the WM filters are computationally efficient because their implementation may take advantage of binary operations, [21,22,416] and analysis, [182,426].

Let $W(i) = \{x_1(i), x_2(i), \ldots, x_N(i)\}$ be an input set of gray scale image samples determined by a filter window $W(i)$ of length N, where $i = 1, \ldots, Q$ denotes the position of the filtering window centered in $x(i)$. Let each input sample $x_k(i)$ from $W(i)$ be associated with a real valued weight ψ_k, for $k = 1, 2, \ldots, N$. The weighted median of the input set $W(i)$ is the sample $y(i) \in W(i)$ minimizing the expression $\sum_{k=1}^{N} \psi_k |y(i) - x_k(i)|$.

If each weight ψ_k is equal to 1, then the WM filter is equivalent to the MF, [231,233]. In order to choose an appropriate weight vector, so that the WM filter would be able to remove impulses and simultaneously preserve all desired image features, various optimization algorithms, [420,423] that originate from the stack filter design, [75,422] have been developed. The adaptive algorithms described here are based on linear and sigmoidal approximations of the sign function.

Given an input set $W(i)$ and a weight vector $\psi = \{\psi_1, \psi_2, \ldots, \psi_N\}$, the WM output is denoted as $y(i) = y(\psi, W(i))$. The estimation of the desired signal $o(i)$ is accompanied with the *estimation error* $e(i) = o(i) - y(i)$. The *cost function* defined under the *Mean Absolute Error* (MAE) or *Mean Squared Error* (MSE) is defined as

$$J_{MAE}(\psi, i) = E\left\{|o(i) - y(\psi, W(i))|\right\}, J_{MSE}(\psi, i) = E\left\{[o(i) - y(\psi, W(i))]^2\right\}. \tag{2.30}$$

With the constraint of non-negative weights, the optimization problem can be expressed as

$$\boxed{\text{minimize } J_{MAE}(\psi, i) \text{ or } J_{MSE}(\psi, i) \text{ subject to } \psi_k \ge 0, \text{ for } k = 1, 2, \ldots, N.} \tag{2.31}$$

Both cost functions are non-convex and under the assumption that the optimal weights are at one of the local minima, the conditions for optimality can be derived as

$$\frac{\partial J_{MAE}(\psi, i)}{\partial \psi_k} = \frac{\partial}{\partial \psi_k} E\left\{|o(i) - y(\psi, W(i))|\right\} = E\left\{\mathcal{S}\left\{o(i) - y(i)\right\} \frac{\partial y(i)}{\partial \psi_k}\right\}, \tag{2.32}$$

where $\mathcal{S}$ denotes the sign function

$$\mathcal{S}\{\chi\} = \begin{cases} 1 & \text{if } \chi > 0, \\ 0 & \text{if } \chi = 0, \\ -1 & \text{if } \chi < 0, \end{cases} \tag{2.33}$$

and then

$$\frac{\partial J_{MSE}(\psi, i)}{\partial \psi_k} = 2E\left\{[o(i) - y(i)] \frac{\partial y(i)}{\partial \psi_k}\right\}. \tag{2.34}$$

Assuming the MAE criterion, the necessary condition for the filter optimality is, [189]

$$\mathcal{S}\left\{o(i) - y(i)\right\} \frac{\partial y(i)}{\partial \psi_k} = 0, \ \psi_k \ge 0 \ \ k = 1, 2, \ldots, N. \tag{2.35}$$

With respect to this analysis, adaptive WM algorithms based on *linear*, [422] and *sigmoidal approximation*, [423] of the sign function $\mathcal{S}$ have been developed. Using the *least mean squared* (LMS) method and the constraint of non-negative weighting coefficients, the adaptation step related to J_{MSE} is given by

$$\psi_k(i+1) = \left[\psi_k(i) + 2\epsilon \frac{\partial J(\psi, i)}{\partial \psi_k}\right]^+, \quad \text{where} \quad \{\chi\}^+ = \begin{cases} 0, & \text{if } \chi < 0, \\ \chi, & \text{otherwise}, \end{cases} \tag{2.36}$$

is a projection function, which sets the negative values to zero. Replacing the statistical expectation in (2.32) with the instantaneous estimates, results in the following adaptation formula

$$\psi_k(i+1) = \left\{\psi_k(i) + 2\epsilon \frac{\partial y(\psi, i)}{\partial \psi_k} \mathcal{S}\{o(i) - y(i)\}\right\}^+. \tag{2.37}$$

Combining the principle of minimal LMS errors with the simultaneous principle of orthogonality, the adaptation formula (2.37) is redefined as

$$\psi_k(i+1) = \{\psi_k(i) + 2\epsilon[o(i) - y(i)][x_k(i) - o(i)]\}^+ . \tag{2.38}$$

Let us consider the sliding filtering window centered in position i, moving over an image domain. During the processing, the weight coefficients are adjusted by adding the contribution of the samples multiplied by a certain regulation factor ϵ. If the adaptive WM algorithm based on the sigmoidal approximation of the sign function is considered, an adjustment of the filter weights can be expressed as, [423]

$$\psi_k(i+1) = \left[\psi_k(i) + 2\epsilon[o(i) - y(i)]\bar{\mathcal{S}}\{x_k(i) - y(i)\}\right]^+ , \tag{2.39}$$

where $o(i)$ is the desired sample, $y(i)$ is the WM output in the i-th iteration, ϵ is the iteration constant and $\bar{\mathcal{S}}\{\cdot\}$ is the sign function approximated by the sigmoidal function (2.40)

$$\bar{\mathcal{S}}\{\chi\} = \frac{2}{1+e^{-\chi}} - 1, \tag{2.40}$$

Let us assume for a moment that $\{\cdot\}^+$ is an identity function, whose argument remains unchanged. If $x_k(i) \gg y(i)$ and ϵ is positive, then the adaptation formula (2.39) is given by

$$\psi_k(i+1) = \psi_k(i) + 2\epsilon[o(i) - y(i)] , \tag{2.41}$$

i.e. the importance of the sample occupying the k-th position in a supporting window $W(i)$ increases if $o(i)$ is greater than the actual WM output $y(i)$ and decreases if $o(i)$ is less than $y(i)$. In general, the initial weight vector $\psi(1)$ can be set to arbitrary positive values, but most advantageous is to start the weight adaptation with equal weights corresponding to the median. Regarding the optimal value of ϵ, it has been shown in [423] that the algorithm converges to sub-optimal solution for sufficiently small positive value of $\epsilon \approx 10^{-5}$.

In the case of adaptive WM filtering with the linear approximation, [422] the weight coefficients are updated as

$$\psi_k(i+1) = \Big\{\psi_k(i) + 2\epsilon\Big[x_{(N)}(i) - x_{(1)}(i) + \\ -2|o(i) - x_k(i)| - \sum_{j=1}^{N} \psi_j(i)[x_{(N)}(i) - x_{(1)}(i) - 2|x_k(i) - x_j(i)|]\Big]\Big\}^+ , \tag{2.42}$$

where $k, j = 1, 2, \ldots, N$, $x_{(N)}(i)$ and $x_{(1)}(i)$ represent the maximum and minimum of the input set $\{x_1(i), x_2(i), \ldots, x_N(i)\}$ respectively and ϵ is the positive adaptation step-size, [422].

3

Overview of Noise Reduction Filters for Color Imaging

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 and attractive. However, many filtering problems cannot be efficiently solved with linear techniques, as they cannot cope with nonlinearities of the image formation and fail to preserve edges and image details. To this end, nonlinear image processing techniques intended for color image filtering are presented. Nonlinear techniques are able to suppress mixed noise, preserve salient image features and eliminate degradations occurring during image acquisition and transmission through noisy channels.

3.1 Order Statistic in Color Image Filtering

ONE of the most popular families of nonlinear filters for noise removal are order statistic filters, [155,231,233,246,396,403]. These filters utilize algebraic ordering of a windowed set of data to compute the output signal using the theory of robust statistics.

The early approaches to color image processing usually comprise direct extensions of the scalar filters to color images, (Fig. 1.3). Ordering of scalar data, such as samples of gray scale images, is well defined and it was extensively studied, [231]. However, the concept of input ordering, initially applied to scalar quantities is not easily extendable to multichannel data, since there is no universal way to define ordering in vector spaces and therefore a number of

different ways to order multivariate data has been proposed. These techniques are generally classified into, [35, 156, 243]:

- *marginal ordering* (M-ordering), where the multivariate samples are ordered in each dimension independently,
- *reduced* or *aggregated ordering* (R-ordering), where each multivariate observation is reduced to a scalar value according to a chosen 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.

Let the mapping $\mathbb{Z}^l \rightarrow \mathbb{Z}^m$ represents a multichannel image, where l is an image dimension and m denotes the number of color channels. Let $W = \{\mathbf{x}_k \in \mathbb{Z}^l; k = 1, 2, \ldots, N\}$ represents a filter window of a finite length N, where $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ is a set of noisy samples and the central sample $\mathbf{x}_1$ determines the position of the filter window. Note that x_{kq}, for $q = 1, 2, \ldots, m$ is the q-th element of the input sample $\mathbf{x}_k = (x_{k1}, x_{k2}, \ldots, x_{km})$.

In the case of color image filtering, the most popular filtering approaches are based on *vector ordering* scheme defined through the ordering of *aggregated distance functions* or *dissimilarity measures*. Such an ordering should:

- be useful from a robust estimation perspective, allowing the extension of the operations of scalar order statistic filters to the multivariate domain,
- preserve the notion of varying levels of extremes that is present in the scalar ordering,
- take into consideration the type of the multivariate data being used.

Therefore, since the RGB color space is used throughout this work, the ordering scheme should give equal importance to the three primary color channels and should consider all the information contained in each of the three channels.

Based on these three principles, the ordering scheme that will be utilized here, is a variation of the reduced ordering scheme, [246, 258, 385] that assigns a dissimilarity measure to the set of samples in W. In this way the aggregated measure of a distance of sample $\mathbf{x}_k$, for $k = 1, 2, \ldots, N$, to all other samples in the filtering window W

$$R_k = \sum_{j=1}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j), \tag{3.1}$$

is used for ranking purposes. Note that $\rho(\cdot)$ denotes the chosen distance or dissimilarity function. The scalar quantities $R_1, R_2, \ldots, R_N$ are then ranked in the order of their value and the associated vectors are correspondingly ordered as follows, [145, 232, 246, 385, 405]

$$R_{(1)} \leq R_{(2)} \leq \ldots \leq R_{(\tau)} \leq \ldots \leq R_{(N)} \quad \Rightarrow \quad \mathbf{x}_{(1)} \prec \mathbf{x}_{(2)} \prec \ldots \prec \mathbf{x}_{\tau} \prec \ldots \prec \mathbf{x}_{(N)}. \tag{3.2}$$

Note that $R_{(k)} \in \{R_1, R_1, \ldots, R_N\}$ and $\mathbf{x}_{(k)} \in \{\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_N\}$, for $k = 1, 2, \ldots, N$. The proposed ordering scheme focuses on the relationships between the multivariate samples, since it computes dissimilarity or distance between all pairs of data points belonging to W. The output of the ranking procedure depends on the type of data used for the determination of the aggregated distance R in (3.1) and on the function $\rho(\mathbf{x}_k, \mathbf{x}_j)$ selected to evaluate the dissimilarity or distance between the vectors $\mathbf{x}_k$ and $\mathbf{x}_j$.

According to the used dissimilarity measure, it is possible to differentiate the techniques operating on the *vector distance domain*, [19, 29, 87–89, 182, 379, 407], *angular domain*, [189, 192, 236, 239, 395, 397] or their combinations, [106, 138, 183, 194, 195, 246].

3.2 Family of Vector Median Filters

Let us assume that each input multichannel sample $\mathbf{x}_k$ is associated with the distance measure

$$R_k = \sum_{j=1}^{N} \|\mathbf{x}_k - \mathbf{x}_j\|_\gamma, \qquad \text{for } k = 1, 2, \ldots, N, \tag{3.3}$$

where $\|\mathbf{x}_k - \mathbf{x}_j\|_\gamma$ quantifies the distance among two m-channel samples $\mathbf{x}_k = (x_{k1}, \ldots, x_{km})$ and $\mathbf{x}_j = (x_{j1}, \ldots, x_{jm})$ using the Minkowski metric

$$\|\mathbf{x}_k - \mathbf{x}_j\|_\gamma = \left(\sum_{q=1}^{m} |x_{kq} - x_{jq}|^\gamma \right)^{\frac{1}{\gamma}}, \tag{3.4}$$

where x_{kq} is the q-th element of $\mathbf{x}_k$ and γ characterizes the used norm. Note that the Minkowski metric includes the city-block distance ($\gamma = 1$), Euclidean distance ($\gamma = 2$) and the chess-board distance ($\gamma = \infty$) as special cases.

The sample $\mathbf{x}_{(1)} \in W$ associated with the minimal aggregated distance $R_{(1)} \in \{R_1, \ldots, R_N\}$ constitutes the output of the *Vector Median Filter* (VMF), which minimizes the distance to other samples inside the sliding filtering window W, [19].

Nonlinear ranked type multichannel filters generally define the vector $\mathbf{x}_{(1)}$ 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 (3.2), [123, 191, 232, 405].

The definition of the vector median is a direct extension of the ordinary scalar median definition with the appropriate norm utilized to order vectors according to their relative magnitude differences, [19, 40, 79, 85, 289, 391, 430]. The output of the VMF is the pixel $\mathbf{x}_{(1)} \in W$ for

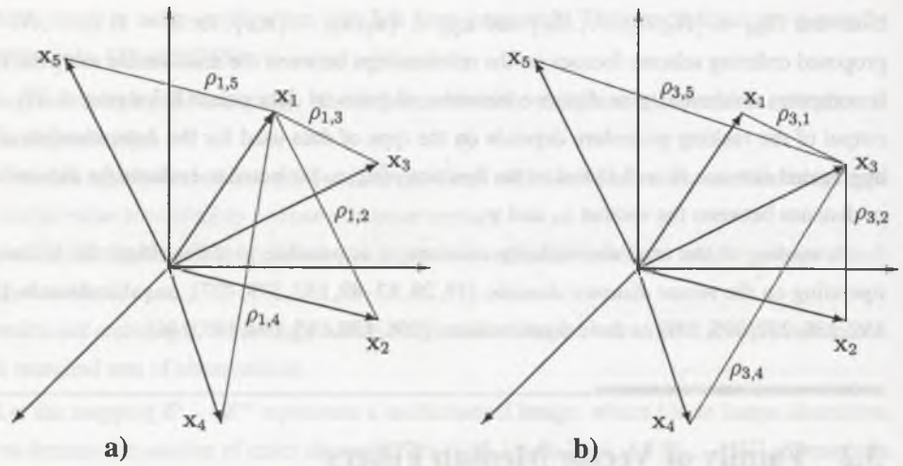


Fig. 3.1. Construction of the cumulative distance: $R_1 = \rho(1,2) + \rho(1,3) + \rho(1,4) + \rho(1,5)$, *(a) and similarly the distance* R_3 *associated with* $\mathbf{x}_3$ *equals* $R_3 = \rho(3,1) + \rho(3,2) + \rho(3,4) + \rho(3,5)$, *(b)*

which the following condition is satisfied

$$\sum_{j=1}^{N} \rho(\mathbf{x}_{(1)}, \mathbf{x}_j) < \sum_{j=1}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j), \quad k = 1, \ldots, N. \tag{3.5}$$

In this way the VMF consists of computing and comparing the values of R_k in (3.3) and the output is the vector $\mathbf{x}_k$ for which R_k minimizes the function R in (3.3). In other words, if for some k the value $R_k = \sum_{j=1}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j)$, is smaller than $R_1 = \sum_{j=1}^{N} \rho(\mathbf{x}_1, \mathbf{x}_j)$, and minimizes the function R, then the original pixel $\mathbf{x}_1$ in the filter window W is being replaced by $\mathbf{x}_k$ which satisfies the condition (3.5), which means that $k = \arg\min R$. The construction of the VMF is illustrated in Fig. 3.1, where the Euclidean distance is used, however different norms can be applied for noise suppression using the VMF concept, [37, 38, 64, 391].

Extended Vector Median Filter

The VMF concept may be combined with the linear filtering for the case where the 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 proposed in [18, 19, 121, 220]. If the output of the *Arithmetic Mean Filter*, (AMF) is denoted as $\mathbf{x}_{AMF}$ then

$$\mathbf{x}_{EVMF} = \begin{cases} \mathbf{x}_{AMF}, & \text{if } \sum_{j=1}^{N} \|\mathbf{x}_{AMF} - \mathbf{x}_j\| < \sum_{j=1}^{N} \|\mathbf{x}_{VMF} - \mathbf{x}_j\|, \\ \mathbf{x}_{VMF}, & \text{otherwise.} \end{cases} \tag{3.6}$$

The output of the extended VMF is the same as that of VMF or AMF, whichever gives a smaller value of the sum of distances. In smooth areas the EVMF outputs the average value of the samples in W, whereas at strong signal transitions its output is the VMF.

α-trimmed Vector Median Filter

It has been observed through experimentation, that the VMF discards well impulses and preserves to some extent image edges, [19]. However, its performance in the suppression of additive Gaussian noise, which is frequently encountered in image processing, is inferior to that of the linear AMF. If a color image is corrupted by both additive Gaussian and impulsive noise, an effective filtering scheme should make an appropriate compromise between the AMF and VMF. The so called α*-trimmed Vector Median Filter* (αVMF) exemplifies this trade-off. In this filter, the α samples closest to the vector median output are selected as inputs to an averaging type of filter. The output of the α-trimmed VMF, which is a modification of the α-trimmed mean (2.19), can be defined as, [243, 407]

$$\mathbf{x}_{\alpha VMF} = \frac{1}{\alpha} \sum_{k=1}^{\alpha} \mathbf{x}_{(k)}, \tag{3.7}$$

with ordering defined in (3.3). The trimming operation guarantees good performance in the presence of impulsive noise, whereas the averaging operation causes that the filter performs well in the presence of short-tailed noise.

The class of filters based on order statistics is very rich. In addition to the filters discussed above, it includes other filters such as the max/min vector filters or the L-vector estimators. The L-vector filter family is an important generalization of the Vector Median Filter, [217] and is closely related to the large class of robust scalar estimators called L-estimators discussed in Chapter 2. These robust filters are modelled by means of weighting coefficients, which can be chosen optimally according to the input noise intensity and its statistical characteristics, [218].

Crossing Level Median-Mean Filter

On the basis of the vector ordering and L-estimator concepts, an efficient technique called Crossing Level Median-Mean Filter (CLMMF) combining the idea of the VMF and AMF can be proposed. Let ψ_k be a weight associated with the k-th element of the ordered set of vectors $\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(N)}$, then the filter output is declared as $\mathbf{y} = \sum_{k=1}^{N} \psi_k \mathbf{x}_{(k)}$. One of the efficient weight selection scheme is

$$\psi_k = \begin{cases} 1 - \frac{N-1}{\sqrt{N(N+\gamma)}}, & \text{for } k = 1, \\ \frac{1}{\sqrt{N(N+\gamma)}}, & \text{for } k = 2, \ldots, N, \end{cases} \tag{3.8}$$

where γ is a parameter, (for $\gamma \to \infty$ we obtain VMF, and for $\gamma = 0$ the filter reduces to AMF).

Weighted Vector Median Filter

In [7, 151, 198, 407, 415] the VMF concept has been generalized and the so-called *Weighted Vector Median Filter* (WVMF) has been proposed. Using the *weighted vector median* approach, the filter output is the vector $\mathbf{x}_{(1)}$ belonging to W, for which the following condition holds

$$\sum_{j=1}^{N} \psi_j \, \rho(\mathbf{x}_{(1)}, \mathbf{x}_j) < \sum_{j=1}^{N} \psi_j \, \rho(\mathbf{x}_k, \mathbf{x}_j), \quad k = 1, \ldots, N. \tag{3.9}$$

If $\psi_1 > 1$ and $\psi_k = 1$ for $k = 2, \ldots, N$, ($\psi = \{\psi_1, 1, 1 \ldots, 1\}$), then the simplified *Central Weighted VMF* (CWVMF) is obtained, [34, 68, 151, 199, 373]. In this way searching for the vector $\mathbf{x}_k$ satisfying (3.9) is equivalent to finding the smallest value of accumulated distances R_k, and in this way the vector $\mathbf{x}_k$ assigned to

$$R_k = \sum_{j=1}^{N} \psi_j \rho(\mathbf{x}_k, \mathbf{x}_j), \quad k = 1, \ldots, N, \tag{3.10}$$

with

$$R_1 = \sum_{j=1}^{N} \psi_1 \rho(\mathbf{x}_1, \mathbf{x}_j) = \sum_{j=2}^{N} \psi_1 \rho(\mathbf{x}_1, \mathbf{x}_j), \quad R_{k>1} = \sum_{j=1}^{N} \psi_j \rho(\mathbf{x}_k, \mathbf{x}_j) = \psi_1 \rho(\mathbf{x}_k, \mathbf{x}_1) + \sum_{j=2}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j), \tag{3.11}$$

can be found.

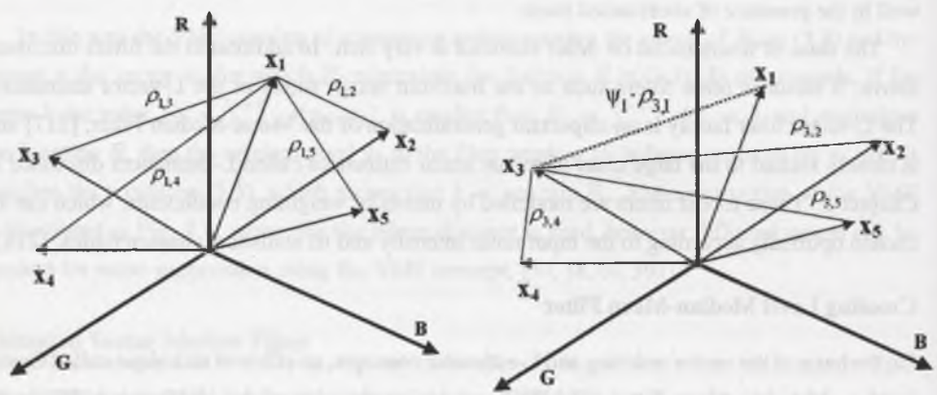


Fig. 3.2. Illustration of the CWVMF construction, in which the distance to the central pixel $\mathbf{x}_1$ is multiplied by the weighting factor ψ_1

If R_1 is larger than $R_{k>1}$, then the central pixel $\mathbf{x}_1$ is being replaced by one of its neighbors $\mathbf{x}_k$. The condition for the central pixel replacement is then $R_1 > R_{k>1}$, which yields

$$\rho(\mathbf{x}_k, \mathbf{x}_1) < \left(\sum_{j=2}^{N} [\rho(\mathbf{x}_1, \mathbf{x}_j) - \rho(\mathbf{x}_k, \mathbf{x}_j)] \right) \Big/ \psi_1, \quad k = 2, \ldots, N. \tag{3.12}$$

The difference between the VMF and CWVMF is that the distance between the central pixel $\mathbf{x}_1$ and its neighbors is multiplied by the weighting coefficient ψ_1, which privileges the central pixel $\mathbf{x}_1$, as shown in Fig. 3.2.

Modified Central Weighted Vector Median Filter

An efficient modification of the CWVMF called *Modified CWVMF* (MCWVMF) was proposed in [292,342,344]. The construction of this filter is to some extent similar to the WVMF proposed in [7, 151, 180, 193, 407, 415].

METHOD	REF.	NMSE $[10^{-3}]$	RMSE	SNR [dB]	PSNR [dB]	NCD $[10^{-4}]$
AMF	[231]	79.317	12.627	21.006	26.105	82.745
VMF	[19]	18.766	6.142	27.266	32.365	40.467
CWVMF	[407]	**12.105**	**4.933**	**29.170**	**34.269**	**19.019**
BVDF	[395]	24.587	7.030	26.093	31.192	41.151
GVDF	[395]	19.474	6.257	27.105	32.204	41.773
DDF	[138]	18.872	6.159	27.242	32.340	40.237
HDF	[106]	18.610	6.116	27.303	32.401	41.275
AHDF	[106]	18.310	6.067	27.373	32.472	41.166
FVDF	[240]	22.251	6.688	26.527	31.625	44.686
ANNF	[237]	26.800	7.340	25.719	30.817	48.009
MCWVMF	[292]	**8.950**	**4.034**	**30.918**	**36.017**	**10.753**

Tab. 3.1. Comparison of the efficiency of the MCWVMF with the VMF, CWVMF and other techniques, using the LENA standard image contaminated by 4% impulsive noise, (NM2, $p = 0.04$), [292, 344]

Let the aggregated distance R_k^* associated with the pixel $\mathbf{x}_k$ be defined in a slightly different way as in (3.10)

$$R_k^* = \sum_{j=1}^{N} \psi_k^* \rho(\mathbf{x}_k, \mathbf{x}_j) = \psi_k^* \sum_{j=1}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j) = \psi_k^* R_k, \quad k = 1, \ldots, N. \tag{3.13}$$

with $\psi^* = \{\psi_1^*, 1, 1 \ldots, 1\}$, $\psi_1^* \in [0, 1]$.

Then we obtain

$$R_1^* = \psi_1^* R_1, \quad \text{and} \quad R_{k>1}^* = R_k, \qquad \text{with} \qquad R_k = \sum_{j=1}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j), \tag{3.14}$$

and then the condition for the replacement of $\mathbf{x}_1$ is simply

$$\psi_k^* R_1 > R_k, \quad k = 2, \ldots, N. \tag{3.15}$$

For $\psi_1^* = 0$ no changes are introduced to the image, and for $\psi_1^* = 1$ the standard VMF is obtained. If $\psi_1^* \in (0, 1)$, then the modified CWVMF (MCWVMF) has the ability of noise

removal, while preserving fine image details, (lines, edges, corners, texture) and it outperforms the standard central weighted vector median scheme as shown in Tab. 3.1 and in Figs. 3.3, 3.4, [292, 342].

It is easy to notice that the new filter is faster than the CWVMF, as the only weighting is applied to the sum of distances R_1^*. As a result, the new filter needs only one additional multiplication compared with the VMF. The CWVMF needs 7 additional multiplications to perform the weighting of the distances between $\mathbf{x}_1$ and all its neighbors. As a result the new filtering scheme is faster than CWVMF and is also more efficient.

For the efficiency comparisons, the color test image LENA has been contaminated by impulsive noise ranging from 1% to 10%, (NM2). The comparison with standard filtering techniques (Tab. 3.1, Figs. 3.3, 3.4) shows that the new filter outperforms significantly the VMF and also the standard Central Weighted Vector Median Filter.

Thresholded Vector Median Filter

The VMF gives acceptable results for impulsive noise removal, but it has a severe shortcoming. Namely it changes more image pixels than it is necessary, and thus causes excessive over-smoothing. One of the possibilities to reduce this effect is to introduce a threshold value which reduces the amount of changes introduced to the filtered image. This concept, similar to (2.15) and (2.21) is used in the *Thresholded Vector Median Filter* (TVMF) which is defined as, [210]

$$\mathbf{x}_{TVMF} = \begin{cases} \mathbf{x}_{(1)}, & \text{if } \|\mathbf{x}_{VMF} - \mathbf{x}_1\| > \delta, \\ \mathbf{x}_1, & \text{otherwise}, \end{cases} \tag{3.16}$$

where $\mathbf{x}_{TVMF}$ is the output of the TVMF filter, $\mathbf{x}_{(1)}$ is the output of the VMF, $\mathbf{x}_1$ denotes the original image pixel, $\|\cdot\|$ denotes the vector norm and δ is a threshold parameter.

Rank Conditioned Vector Median Filter

Another modification of the vector median filter, so called *Rank Conditioned Vector Median Filter* (RCVMF), which aims to alleviate the excessive smoothing of the VMF is based on the ordering of the accumulated distances, which implies an ordering of the vector samples

$$\begin{matrix} R_1, & R_2, & \dots & , R_\tau, & \dots & , R_N \\ \mathbf{x}_1, & \mathbf{x}_2, & \dots & , \mathbf{x}_\tau, & \dots & , \mathbf{x}_N \end{matrix} \longrightarrow \begin{matrix} R_{(1)}, & R_{(2)}, & \dots & , R_{(\tau)}, & \dots & , R_{(N)} \\ \mathbf{x}_{(1)}, & \mathbf{x}_{(2)}, & \dots & , \mathbf{x}_{(\tau)}, & \dots & , \mathbf{x}_{(N)} \end{matrix} . \tag{3.17}$$

In order to decrease the number of samples replaced by the VMF, which are not distorted by the corruption process, the following switching scheme can be applied

$$\mathbf{y} = \begin{cases} \mathbf{x}_1, & \text{if } R_1 \le R_{(\tau)}, \\ \mathbf{x}_{(1)}, & \text{if } R_1 > R_{(\tau)}, \end{cases} \quad \mathbf{y} = \begin{cases} \mathbf{x}_1, & \text{if } R_1 \in \{R_{(1)}, R_{(2)}, \dots, R_{(\tau)}\}, \\ \mathbf{x}_{(1)}, & \text{if } R_1 \in \{R_{(\tau+1)}, R_{(\tau+2)}, \dots, R_{(N)}\}, \end{cases} \tag{3.18}$$

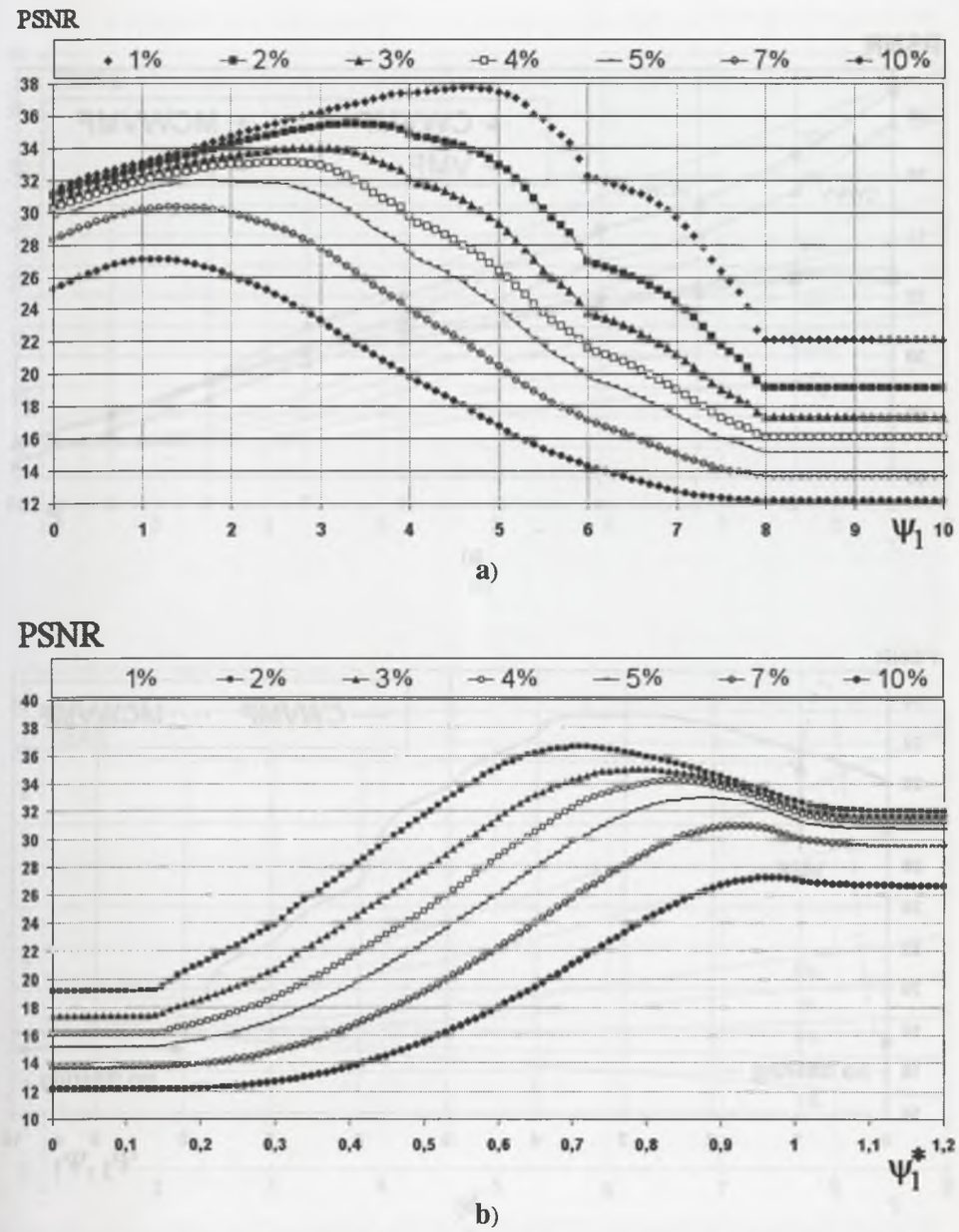
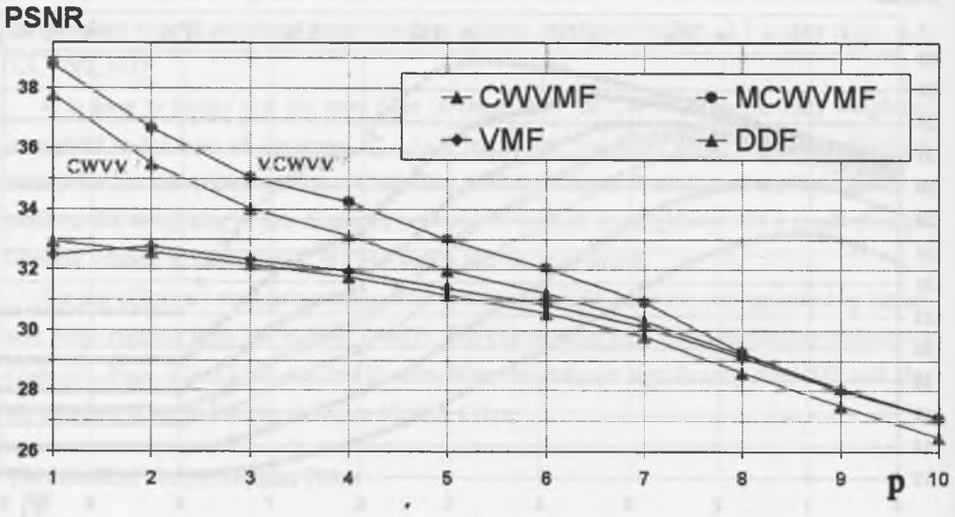
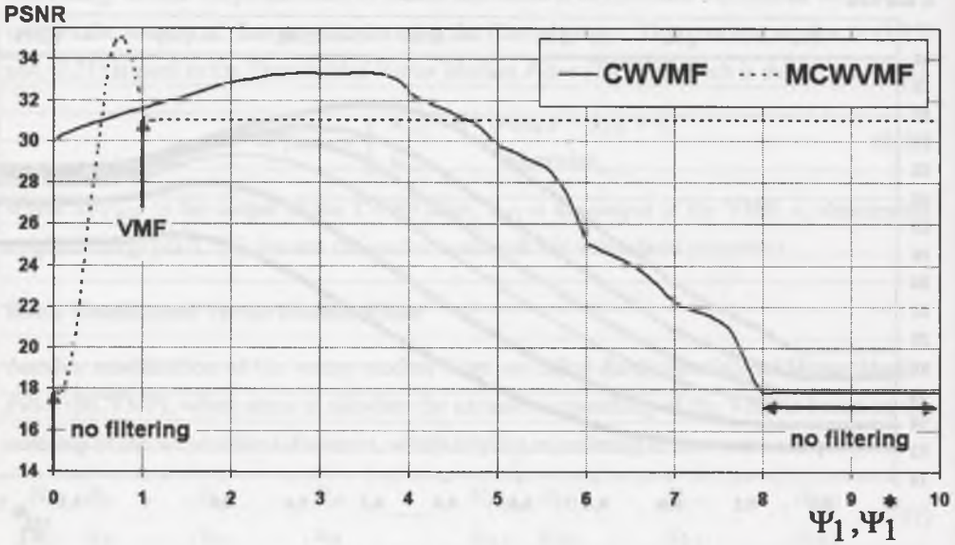


Fig. 3.3. Dependence of the PSNR on the ψ_1 value for the CWVMF defined by (3.9) and (3.12) (a) and ψ_1^* in the modified scheme defined by (3.13) and (3.15) (b) for the LENA color image corrupted by impulsive noise, (NM2, $p = 1\% - 10\%$). The difference between the modified central weighted scheme and the classical approach is presented in Fig. 3.4

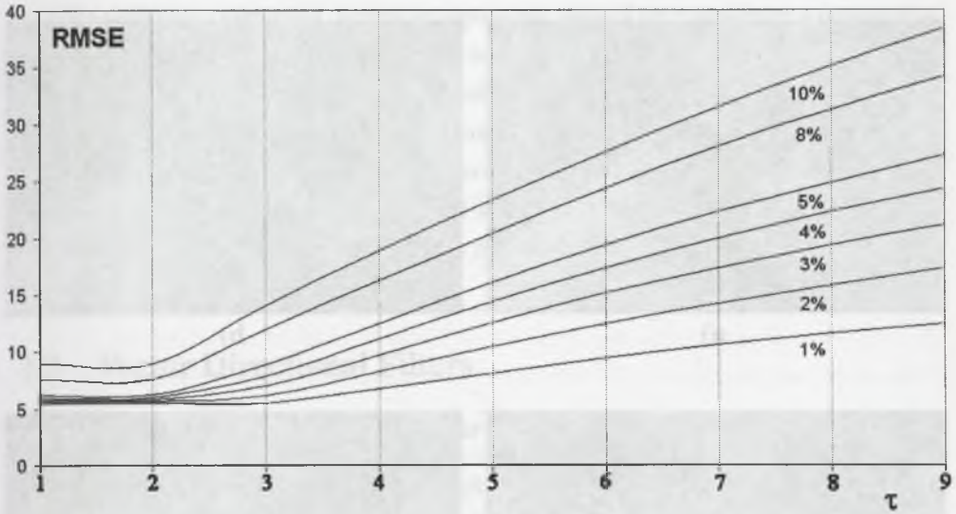


a)

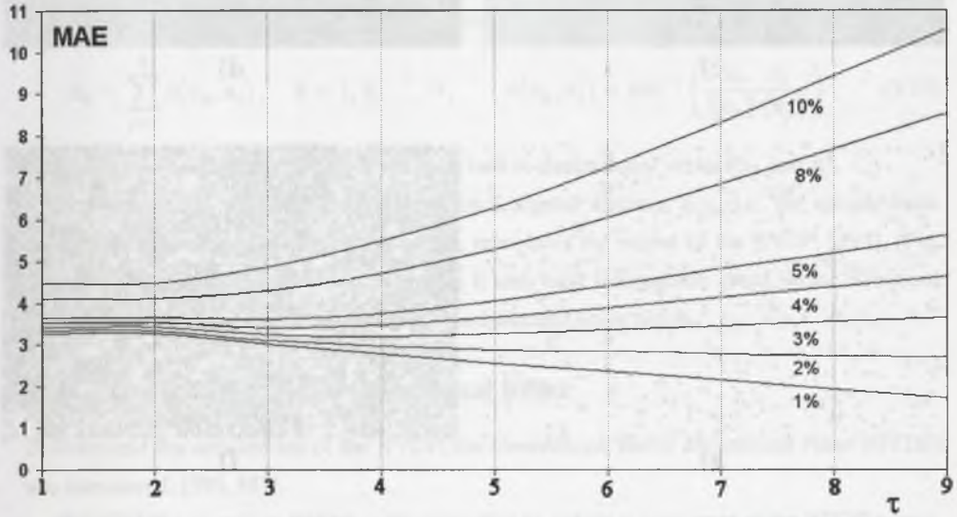


b)

Fig. 3.4. Efficiency of the modified central weighted vector median filter (MCWVMF) in comparison with the CWVMF, VMF and DDF for the LENA image contaminated by impulsive noise ($p = 1\% - 10\%$, NM2), (a) and below the plot of PSNR versus weighting coefficients ψ_1 and ψ_1^* for the noise intensity $p = 4\%$, (NM2), (b)



a)



b)

Fig. 3.5. Dependence of RMSE (a) and MAE (b) on the τ parameter for the Rank Conditioned Vector Median Filter (RCVMF), (LENA, NM2, $p = 1\% - 10\%$), (VMF is obtained for $\tau = 1$)

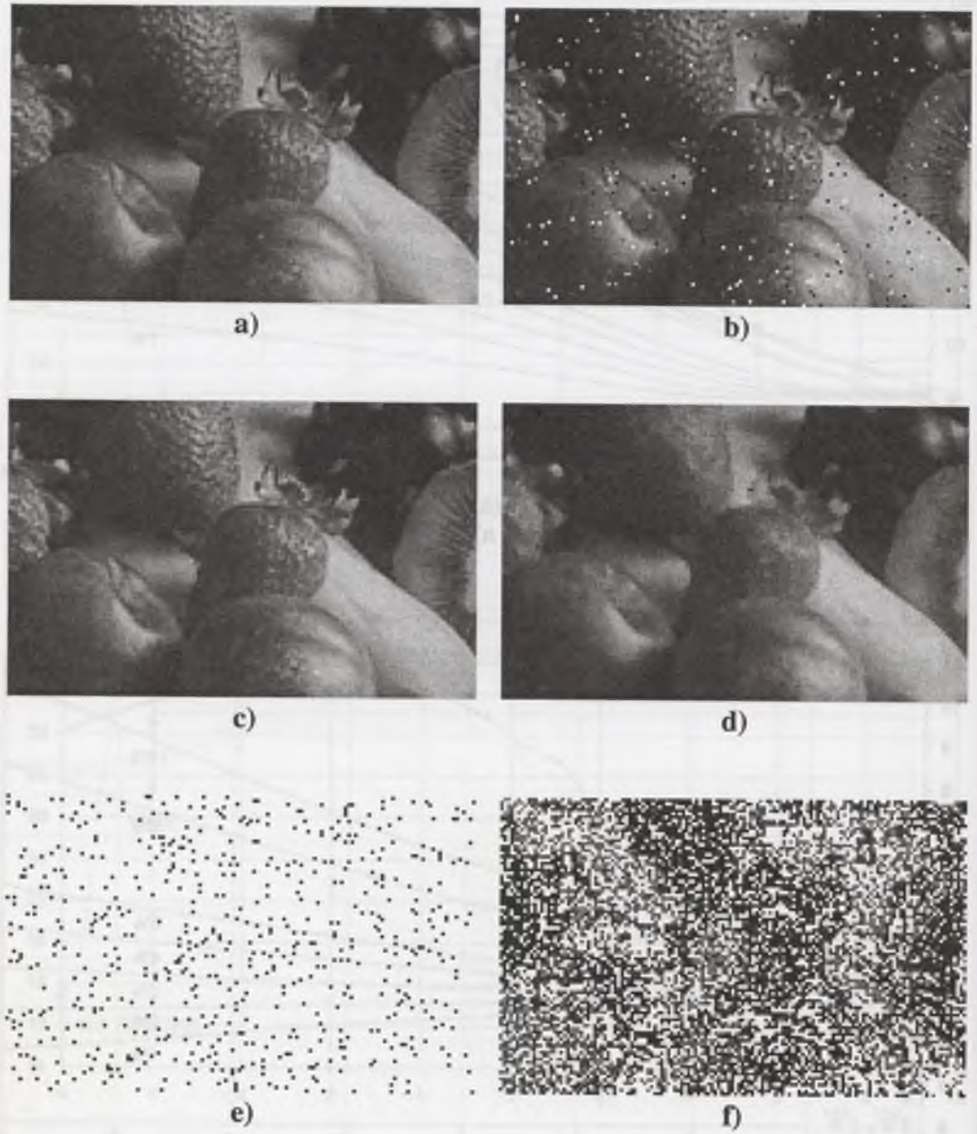


Fig. 3.6. Illustration of the efficiency of the MCWVMF in comparison with the VMF: a) part of the test image FRUITS, b) test image corrupted by impulsive noise, ($p = 0.02$, NM2), c) MCWVMF output, d) VMF output, e) and f) depict the difference between the original image (a) and (c), (d) respectively

where τ is a filter design parameter. For $\tau = 1$ this filter is identical with VMF, whereas for $\tau = N$ no filtering is performed. Setting an appropriate value of the τ parameter, a compromise between the VMF and identity operation can be obtained. However, as can be derived from Fig. 3.5 the optimal τ value depends on the intensity of noise corruption. For small impulsive noise intensity, $\tau = 3$ is a good choice, guaranteing a good trade-off between the cancellation of impulses (Fig. 3.5a) and image detail preservation, (Fig. 3.5b). However, as the scheme is not adaptive to noise intensity, its performance is not satisfactory and therefore in Chapter 7 an adaptive modification of this algorithm will be presented.

3.3 Vector Directional Filters

3.3.1 Basic Vector Directional Filter

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

In the directional processing of color images, [216, 239, 244, 395, 397] each input vector $\mathbf{x}_k$ is associated with the aggregated *angular measure*

$$A_k = \sum_{j=1}^{N} a(\mathbf{x}_k, \mathbf{x}_j), \quad k = 1, 2, \ldots, N, \qquad a(\mathbf{x}_k, \mathbf{x}_j) = \cos^{-1}\left(\frac{\mathbf{x}_k \cdot \mathbf{x}_j}{\|\mathbf{x}_k\| \, \|\mathbf{x}_j\|}\right), \tag{3.19}$$

where $a(\mathbf{x}_k, \mathbf{x}_j)$ represents the angle between two m-dimensional vectors $\mathbf{x}_k$ and $\mathbf{x}_j$.

The sample $\mathbf{x}_{(1)}$ associated with the minimal angular distance $A_{(1)}$, i.e. the sample minimizing the sum of angles with other vectors, represents the output of the BVDF, [395], (Fig. 3.7). A drawback of the BVDF is that since it uses only information about vector directions (chromaticity information), it cannot remove achromatic noisy pixels.

3.3.2 Generalized Vector Directional Filter

To overcome the deficiencies of the BVDF, the *Generalized Vector Directional Filter* (GVDF) was introduced, [395, 397].

The GVDF generalizes BVDF in the sense that its output is a superset of the BVDF output. The first vector in the ordered sequence using the angular distance constitutes the output of the BVDF, whereas the first τ vectors constitute the output of the GVDF

$$BVDF\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} = \mathbf{x}_{(1)}, \quad GVDF\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} = \{\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(\tau)}\}, \; 1 \le \tau \le N. \tag{3.20}$$

The output of the GVDF is subsequently passed through an additional filter in order to produce a single vector output. In this step the designer may only consider the magnitudes of the vectors $\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(\tau)}$ since they have approximately the same direction in the vector space. As a result, the GVDF separates the processing of color vectors into directional and then into magnitude processing as 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.

3.3.3 Directional Distance Filter

To improve the eficiency of the directional filters, another method called *Directional - Distance Filter* (DDF) was proposed, [137, 138]. The DDF is a combination of VMF and BVDF and is derived by simultaneous minimization of their defining functions. Specifically, in the case of the DDF, the distance inside the processing window is defined as

$$D_k = \left(\sum_{j=1}^{N} a\left(\mathbf{x}_k, \mathbf{x}_j\right)\right)^{\kappa} \left(\sum_{j=1}^{N} \rho\left(\mathbf{x}_k, \mathbf{x}_j\right)\right)^{1-\kappa}, \tag{3.21}$$

where $a\left(\mathbf{x}_k, \mathbf{x}_j\right)$ is the directional (angular) distance defined in (3.19) and the distance $\rho\left(\mathbf{x}_k, \mathbf{x}_j\right)$ can be calculated using the L_γ norm. The parameter κ regulates the influence of the angle and distance components. As for any other ranked-order filter, an ordering of the D_k values $D_{(1)} \leq D_{(2)} \leq \ldots \leq D_{(N)}$, implies the same ordering of the corresponding vectors $\mathbf{x}_k$: $\mathbf{x}_{(1)} \prec \mathbf{x}_{(2)} \prec \ldots \prec \mathbf{x}_{(N)}$, thus DDF defines the $\mathbf{x}_{(1)}$ vector as its output. For $\kappa = 0$ we obtain the VMF and for $\kappa = 1$ the BVDF. The DDF is defined for $\kappa = 0.5$ and its usefulness stems from the fact that it combines both the criteria used in BVDF and VMF, [397].

3.3.4 Hybrid Directional Filter

Another efficient rank-ordered operation called *Hybrid Directional Filter* (HDF) was proposed in [106]. This filter operates on the direction and the magnitude of 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, produces an output according to the rule

$$\mathbf{x}_{HDF} = \begin{cases} \mathbf{x}_{VMF}, & \text{if } \mathbf{x}_{VMF} = \mathbf{x}_{BVDF}, \\ \left(\frac{\|\mathbf{x}_{VMF}\|}{\|\mathbf{x}_{BVDF}\|}\right) \mathbf{x}_{BVDF}, & \text{otherwise}, \end{cases} \tag{3.22}$$

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

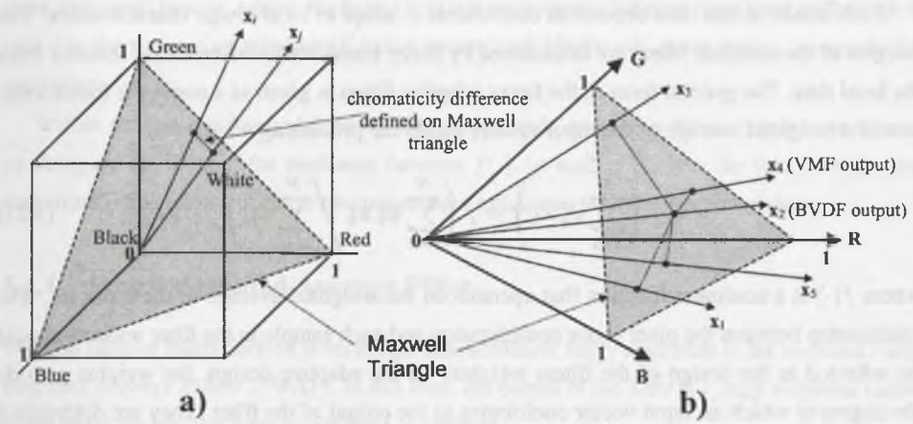


Fig. 3.7. The principle of the directional processing of color images: a) chromaticity difference between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, b) angular minimization property of the BVDF scheme

More complex hybrid filter, which involves the utilization of the *Arithmetic Mean Filter* (AMF), has also been proposed. The structure of this so-called *Adaptive Hybrid Directional Filter* (AHDF) is defined as

$$\mathbf{x}_{AHDF} = \begin{cases} \mathbf{x}_{VMF}, & \text{if } \mathbf{x}_{VMF} = \mathbf{x}_{BVDF}, \\ \mathbf{x}_1^*, & \text{if } \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{x}_1^*\| < \sum_{k=1}^{N} \|\mathbf{x}_k - \mathbf{x}_2^*\|, \\ \mathbf{x}_2^*, & \text{otherwise}, \end{cases} \tag{3.23}$$

where

$$\mathbf{x}_1^* = \left(\frac{\|\mathbf{x}_{VMF}\|}{\|\mathbf{x}_{BVDF}\|}\right) \mathbf{x}_{BVDF}, \quad \mathbf{x}_2^* = \left(\frac{\|\mathbf{x}_{AMF}\|}{\|\mathbf{x}_{BVDF}\|}\right) \mathbf{x}_{BVDF}, \tag{3.24}$$

and $\mathbf{x}_{AMF}$ denotes the output of the arithmetic mean filter operating inside the same processing window. Both hybrid filters are computationally demanding, since they require the evaluation of the VMF and BVDF outputs, [143, 144].

3.4 Fuzzy Adaptive Filters

The performance of different nonlinear filters based on order statistics depends heavily on the problem under consideration, as the type of noise which is present in an image affects the filter's performance. To overcome difficulties associated with the uncertainty associated with the data, adaptive designs based on local statistic have been introduced, [44, 94, 236, 239, 240, 242].

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 the 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 W

$$\mathbf{y} = f\left(\sum_{k=1}^{N} \psi_k^* \mathbf{x}_k\right) = f\left(\sum_{k=1}^{N} \psi_k \mathbf{x}_k \Big/ \sum_{k=1}^{N} \psi_k\right), \tag{3.25}$$

where $f(\cdot)$ is a nonlinear function that operates on the weighted average of the input set. The relationship between the pixel under consideration and each sample in the filter window should be reflected in the design of 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 sample position, [101, 102, 144, 406].

In this framework the weights are determined by fuzzy transformations based on features from the local filtering window. 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 and all other vector samples inside 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 way, the distance functions are not used to order input vectors. Instead, they provide selected features in a reduced space; features used as inputs for the fuzzy membership function.

Several candidate functions, such as triangular, trapezoidal, piecewise linear and Gaussian-like functions can be used as a membership function. If the distance criterion described by (3.19) is used as a distance measure, a sigmoidal membership function can be selected, [240, 246]

$$\psi_k = \gamma_1 \left[1 + \exp\left(A_k\right)\right]^{-\gamma_2}, \tag{3.26}$$

where A_k is the cumulative distance from (3.19), while γ_1 and γ_2 are parameters to be determined. The γ_2 value is used to adjust the weighting effect of the membership function and γ_1 is a scale threshold. If the Minkowski L_γ metric is used as the distance function, the fuzzy membership function with exponential form

$$\psi_k = \exp\left(-R_k^{\gamma_1}/\gamma_2\right), \tag{3.27}$$

gives also good results, where R_k from (3.3) is a cumulative distance associated with the k-th vector in the processing window W using generalized Minkowski norm and γ_1, γ_2 are design parameters.

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 determined. The choice of these two parameters influences the filter characteristics.

3.4.1 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 output* of the input set and the form of the filter is given as

$$\mathbf{y} = \frac{1}{S}\sum_{k=1}^{N} \psi_k \mathbf{x}_{(k)}, \qquad S = \sum_{k=1}^{N} \psi_k. \tag{3.28}$$

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 statistic 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 selected criterion is the sum of vector angles, the *Fuzzy Vector Directional Filter* (FVDF) is obtained, [240]. If the L_γ norm is used as the distance criterion, a fuzzy generalization of the Vector Median Filter is constructed.

3.4.2 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 $\mathbf{y} = \mathbf{x}_k$ with $k = \arg\max \psi_j$, $j = 1, \ldots, N$. 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, [240, 246]. If the L_γ is adopted as a distance criterion, the filter provides the same output as the VMF. In this way, utilizing an 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.

3.4.3 Fuzzy Ordered Vector Directional Filters

It is not necessary for the designer to use all the inputs inside the operational window to produce the final output of the nonlinear filter. If desired, only a subset of the vector-valued input signals can be used. The input vectors are then ordered according to their respective fuzzy membership strengths. The form of the *Fuzzy Ordered Vector Directional Filter* (FOVDF) is given as

$$\mathbf{y} = \frac{1}{S}\sum_{k=1}^{\tau} \psi_{(k)}\, \mathbf{x}_{(k)}\,, \qquad S = \sum_{k=1}^{\tau} \psi_{(k)}\,, \tag{3.29}$$

where $\psi_{(k)}$ represents the k-th ordered fuzzy membership function, and $\psi_{(\tau)} \leq \psi_{(\tau-1)} \leq \ldots, \leq \psi_{(1)}$, with $\psi_{(1)}$ being the fuzzy coefficient with the largest membership strength.

The above form of the algorithm constitutes a fuzzy generalization of the α-trimmed filters, (3.7), [231]. 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 fuzzy membership function chosen by the designer, a number of different α-trimmed filters can be obtained.

The fuzzy transformations (3.26) and (3.27) are not the only way in which the adaptive weights 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*, [237] in which the value of the weight ψ_k in (3.25) is determined according to the following formula

$$\psi_k = \frac{D_{(N)} - D_{(k)}}{D_{(N)} - D_{(1)}}, \tag{3.30}$$

where $D_{(N)}$ is the maximum distance in the filtering window, measured using an appropriate distance criterion, and $D_{(1)}$ is the minimum distance, which is associated with the center-most vector inside the window W. As in the case of the fuzzy membership function, the value of the weight in (3.30) expresses the degree to which the vector $\mathbf{x}_k$ is close to the center-most vector, and far away from the worst value, the outer rank.

In [237, 238] an adaptive vector processing filter named *Adaptive Nearest Neighbor Filter* (ANNF) was devised utilizing the general framework of (3.25). The weights in ANNF are calculated using the formula of (3.30), 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_γ norm or the angular distance between the vectors can be used to remove samples with atypical directions. However, both these distance metrics utilize only a part of the information carried by the image vectors. As in the case of DDF, it is anticipated that an adaptive vector processing filter, based on the ordering criterion, which utilizes both vector features, namely *magnitude* and *direction*, will provide a robust solution whenever the noise characteristics are unknown, [74].

In [11, 238] a novel distance measure was introduced

$$J_k = \sum_{j=1}^{N} [1 - \psi(\mathbf{x}_k, \mathbf{x}_j)]\,, \quad \psi(\mathbf{x}_k, \mathbf{x}_j) = \left(\frac{\mathbf{x}_k \cdot \mathbf{x}_j}{\|\mathbf{x}_k\|\|\mathbf{x}_j\|}\right)\left(1 - \frac{|\,\|\mathbf{x}_k\| - \|\mathbf{x}_j\|\,|}{\max(\|\mathbf{x}_k\|, \|\mathbf{x}_j\|)}\right), \tag{3.31}$$

which takes into consideration both the direction and the magnitude of the vector inputs. The first part of the measure ψ 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 $\psi(\mathbf{x}_k, \mathbf{x}_j)$ equals one and only the directional information is used in (3.31). On the other hand, if the vectors under consideration have the same direction in the vector space, the first part of $\psi(\mathbf{x}_k, \mathbf{x}_j)$, (*directional information*) equals one and the similarity measure 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 (3.25) and the weighting formula of (3.31) was proposed in [238]. The so-called *Adaptive Nearest Neighbor Multichannel Filter* (ANNMF) belongs to the adaptive vector processing filter family defined through (3.25). However, ANNMF combines the weighting formula of (3.30) with the new distance measure of (3.31) to evaluate its weights.

3.5 Nonparametric Adaptive Multichannel Filters

Based on the samples from the filtering window, an adaptive multivariate kernel density estimator [1] can be devised to approximate the samples probability density function $\Psi(\mathbf{x})$

$$\Psi(\mathbf{x}) = \frac{1}{N}\sum_{k=1}^{N} \frac{1}{h_k^m} \mathcal{K}\left\{\frac{\|\mathbf{x} - \mathbf{o}_k\|}{h_k}\right\}, \tag{3.32}$$

where m denotes the dimensionality of the measurement space and h_k is the data dependent smoothing parameter, which regulates the shape of the kernel $\mathcal{K}$, [245, 246].

The variable *kernel density estimator* exhibits local smoothing, which depends both on the point at which the density is evaluated and on the information about samples in W. The choice of the kernel function is not nearly as important as the smoothing factor h. Very often the multivariate extension of the exponential kernel or the Gaussian kernel is selected.

[1] Detailed description of the nonparametric estimation is given in Chapter 6.

The non-parametric estimator can be defined as

$$\mathbf{y} = \sum_{k=1}^{N} \mathbf{o}_k \frac{h_k^{-m}\mathcal{K}\{\|\mathbf{x}_1 - \mathbf{o}_k\|/h_k\}}{\sum_{j=1}^{N} h_j^{-m}\mathcal{K}\{\|\mathbf{x}_1 - \mathbf{o}_j\|/h_j\}} = \sum_{k=1}^{N} \psi_k \mathbf{o}_k, \tag{3.33}$$

where ψ_k are weighting coefficients and $\mathbf{x}_1$ is the central sample or its estimate obtained through an appropriate noise filtering technique.

To obtain the required estimate, we must assume that in the absence of noise, discrete sample vectors $\mathbf{o}_k$ are available. This is not a very severe restriction, since in some cases such samples may be obtained by a calibration procedure in a controlled environment, perhaps at a very high signal-to-noise ratio conditions. In many image processing applications however, that is not the case. Therefore, alternative suboptimal solutions are needed. In a first approach, we can substitute the vectors $\mathbf{o}_k$ in (3.33) with their noisy measurements.

The resulting *Adaptive Nonparametric Multichannel Filter* (ANMF) is solely based on the available noisy vectors. Thus, the form of the ANMF is

$$\mathbf{y}_1 = \sum_{k=1}^{N} \mathbf{x}_k \frac{h_k^{-m}\mathcal{K}\{\|\mathbf{x}_1 - \mathbf{x}_k\|/h_k\}}{\sum_{j=1}^{N} h_j^{-m}\mathcal{K}\{\|\mathbf{x}_1 - \mathbf{x}_j\|/h_j\}}. \tag{3.34}$$

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 $\mathbf{o}_1$ 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 $\mathbf{x}^*$ 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

$$\mathbf{y}_2 = \sum_{k=1}^{N} \mathbf{x}_k^* \frac{h_k^{-m}\mathcal{K}\{\|\mathbf{x}_1^* - \mathbf{x}_k\|/h_k\}}{\sum_{j=1}^{N} h_j^{-m}\mathcal{K}\{\|\mathbf{x}_1^* - \mathbf{x}_j\|/h_j\}}. \tag{3.35}$$

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

4

Application of Anisotropic Diffusion to Image Enhancement

Recently, growing attention has been given to the nonlinear processing of vector valued noisy image signals through the anisotropic diffusion technique. Anisotropic diffusion is a relatively new method derived from the scale space theory, which allows to reduce the image noise without blurring the frontiers between image regions of different color or brightness. In this Chapter some basic concepts of anisotropic diffusion are presented, its efficiency is evaluated and it is shown how this technique can be modified, so that it can better cope with the removal of impulsive noise in multichannel images.

This Chapter also presents an implementation of the anisotropic diffusion based on the forward and backward diffusion concept, which allows to reduce the Gaussian noise, enhance edges and better preserve important image structures.

4.1 Anisotropic Diffusion Framework

VERY powerful filtering technique, called *anisotropic diffusion* (AD), has been introduced by Perona and Malik, (PM), [227, 228] in order to selectively enhance image contrast and reduce noise, using a modified *heat diffusion equation* and the concepts of *scale space*, [263,273,376,418,418]. The main concept of *anisotropic diffusion* is based on the modification of the *isotropic diffusion equation*, so that the smoothing across image edges can be inhibited. This modification is done by introducing a conductivity function that encourages intra-region over inter-region smoothing.

Since the introduction of the PM method, a variety of techniques have been elaborated including multi-scale approaches [142, 277], extensions to vector valued imaging [112, 272], multigrid methods [5], mathematical morphology inspired techniques and many others, [45, 98, 112, 147, 201, 221, 270, 282, 394, 425] and applied to the processing of 2D and also 3D images, [83, 133, 160, 387].

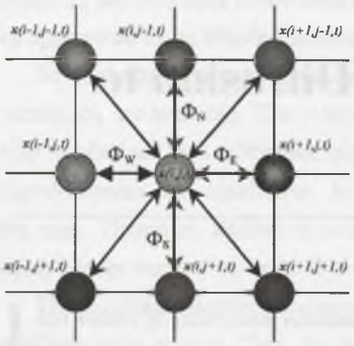


Fig. 4.1. Visualization of the anisotropic diffusion scheme, (i,j) denotes the discrete image coordinates

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

$$\frac{\partial x}{\partial t} + \nabla\zeta = 0\,, \text{ we obtain } \frac{\partial x}{\partial t} = \nabla\left(c\,\nabla x\right). \tag{4.1}$$

If $x(\xi,\eta,t)$ denotes a real-valued function representing the gray scale image, the equation of linear and isotropic diffusion is

$$\frac{\partial x(\xi,\eta,t)}{\partial t} = c\left[\frac{\partial^2 x(\xi,\eta,t)}{\partial\xi^2} + \frac{\partial^2 x(\xi,\eta,t)}{\partial\eta^2}\right], \tag{4.2}$$

where ξ,η are the continuous coordinates, t denotes time and c is the c o n s t a n t conductivity (diffusivity) coefficient.

Perona and Malik suggested that the conductivity coefficient c should be d e p e n d e n t on the image structure and therefore they proposed the following *partial derivative equation*, (PDE)

$$\frac{\partial x(\xi,\eta,t)}{\partial t} = \nabla\left[c(\xi,\eta)\nabla x(\xi,\eta,t)\right], \tag{4.3}$$

which can be expressed as a minimization of the energy $\mathcal{E}$ on the image domain Ω, [62, 211]

$$\mathcal{E}(x) = \int_\Omega \Gamma_R\left(|\nabla x|\right)\,\mathrm{d}\Omega\,, \tag{4.4}$$

where $\Gamma_R\left(|\nabla x|\right)$ is a regularization function that penalizes high gradients, while preserving edges: $\Gamma'_R\left(|\nabla x|\right) = c\left(|\nabla x|\right)|\nabla x|$, which leads to

$$\frac{\partial x}{\partial t} = \operatorname{div}\left(\frac{\Gamma'_R\left(|\nabla x|\right)\nabla x}{|\nabla x|}\right) = \operatorname{div}\left(c\left(|\nabla x|\right)\nabla x\right) = \operatorname{div}\Phi(x)\,, \tag{4.5}$$

where $\Phi(x) = c\left(|\nabla x|\right)\nabla x$ is the flux function.[1]

[1]For the sake of simplicity we will use $\Phi(x) = -\zeta(x)$.

The conductivity coefficient $c(\xi,\eta)$ is a monotonically decreasing function of the image gradient magnitude and usually contains a free parameter β, which determines the amount of smoothing introduced by the nonlinear diffusion process. Different functions of $c(\xi,\eta)$ have been suggested in the rich literature, [5, 9, 46, 63, 262, 274]. The most popular are those introduced in [228], (Figs. 4.2, 4.11),

$$c_1 = \exp\left(-\frac{|\nabla x(\xi,\eta)|^2}{2\beta^2}\right), \quad c_2 = \left(1 + \frac{|\nabla x(\xi,\eta)|^2}{\beta^2}\right)^{-1}. \tag{4.6}$$

The conductivity function $c(\xi,\eta)$ is space-varying and it is chosen to be large in a relatively homogeneous regions to encourage smoothing, and small in regions with high gradients to preserve image edges, (see Fig. 4.3).

In one-dimensional case, the gradient and divergence expressions in (4.3) reduce to derivatives, [112, 159, 172, 253]

$$\frac{\partial}{\partial t}x(\xi,t) = \frac{\partial}{\partial\xi}\left[c(\xi,t)\,\frac{\partial}{\partial\xi}x(\xi,t)\right]. \tag{4.7}$$

Substituting discrete approximations of the derivatives, we obtain

$$\frac{\partial}{\partial t}x(\xi,t) \approx \frac{\partial}{\partial\xi}\left\{c(\xi,t)\frac{1}{\Delta\xi}\left[x\left(\xi+\frac{\Delta\xi}{2},t\right) - x\left(\xi-\frac{\Delta\xi}{2},t\right)\right]\right\} \approx \tag{4.8}$$

$$\frac{1}{(\Delta\xi)^2}\left[c\left(\xi+\frac{\Delta\xi}{2},t\right)\left(x\left(\xi+\Delta\xi,t\right)-x\left(\xi,t\right)\right) - c\left(\xi-\frac{\Delta\xi}{2},t\right)\left[x\left(\xi,t\right)-x\left(\xi-\Delta\xi,t\right)\right]\right].$$

The conductivity values $c(\xi+\frac{\Delta\xi}{2},t)$ and $c(\xi-\frac{\Delta\xi}{2},t)$ can be determined as functions of the discrete gradient approximations

$$c(\xi,t) \approx f\left(\frac{1}{\Delta\xi}\left|x\left(\xi+\frac{\Delta\xi}{2},t\right) - x\left(\xi-\frac{\Delta\xi}{2},t\right)\right|\right), \tag{4.9}$$

$$c\left(\xi+\frac{\Delta\xi}{2},t\right) \approx f\left(\frac{1}{\Delta\xi}\left|x(\xi+\Delta\xi,t) - x(\xi,t)\right|\right), \tag{4.10}$$

$$c\left(\xi-\frac{\Delta\xi}{2},t\right) \approx f\left(\frac{1}{\Delta\xi}\left|x(\xi,t) - x(\xi-\Delta\xi,t)\right|\right). \tag{4.11}$$

Introducing the notation

$$c_R = \frac{1}{\Delta\xi^2}c\left(\xi+\frac{\Delta\xi}{2},t\right),\ c_L = \frac{1}{\Delta\xi^2}c\left(\xi-\frac{\Delta\xi}{2},t\right), \tag{4.12}$$

$\nabla_R x(\xi,t) = x(\xi+\Delta\xi,t) - x(\xi,t)$, $\nabla_L x(\xi,t) = x(\xi-\Delta\xi,t) - x(\xi,t)$, we get

$$\frac{\partial}{\partial t}x(\xi,t) = c_L\,\nabla_L x(\xi,t) + c_R\,\nabla_R x(\xi,t) = \Phi_L + \Phi_R\big|_{\Delta\xi=1}, \tag{4.13}$$

$$\boxed{x(\xi, t+\Delta t) \approx x(\xi, t) + \Delta t \frac{\partial}{\partial t} x(\xi, t) = x(\xi, t) + \Delta t \left(\Phi_L + \Phi_R\right)\big|_{\Delta\xi=1}.} \tag{4.14}$$

The 1-D discrete formulation can be extended to the 2-D case

$$\frac{\partial}{\partial t} x(\xi,\eta,t) = \frac{\partial}{\partial \xi}\left[c(\xi,\eta,t)\frac{\partial}{\partial \xi}x(\xi,\eta,t)\right] + \frac{\partial}{\partial \eta}\left[c(\xi,\eta,t)\frac{\partial}{\partial \eta}x(\xi,\eta,t)\right] \approx \tag{4.15}$$

$$\approx \frac{1}{(\Delta\xi)^2}\left[\begin{array}{l} c\left(\xi+\frac{\Delta\xi}{2},\eta,t\right)\left(x(\xi+\Delta\xi,\eta,t) - x(\xi,\eta,t)\right) + \\ -c\left(\xi-\frac{\Delta\xi}{2},\eta,t\right)\left(x(\xi,\eta,t) - x(\xi-\Delta\xi,\eta,t)\right) \end{array}\right] +$$

$$+\frac{1}{(\Delta\eta)^2}\left[\begin{array}{l} c\left(\xi,\eta+\frac{\Delta\eta}{2},t\right)\left(x(\xi,\eta+\Delta\eta,t) - x(\xi,\eta,t)\right) + \\ -c\left(\xi,\eta-\frac{\Delta\eta}{2},t\right)\left(x(\xi,\eta,t) - x(\xi,\eta-\Delta\eta,t)\right) \end{array}\right] = \tag{4.16}$$

$$= c_N(\xi,\eta,t)\,\nabla_N x(\xi,\eta,t) + c_S(\xi,\eta,t)\,\nabla_S x(\xi,\eta,t) + c_W(\xi,\eta,t)\,\nabla_W x(\xi,\eta,t) +$$

$$+ c_E(\xi,\eta,t)\,\nabla_E x(\xi,\eta,t) = \Phi_N + \Phi_S + \Phi_W + \Phi_E\big|_{\Delta\xi=1,\Delta\eta=1}, \tag{4.17}$$

$$c_N = c(\xi,\eta+\Delta/2,t)/\Delta\eta^2, \quad c_S = c(\xi,\eta-\Delta/2,t)/\Delta\eta^2, \tag{4.18}$$

$$c_E = c(\xi+\Delta/2,\eta,t)/\Delta\xi^2, \quad c_W = c(\xi-\Delta/2,\eta,t)/\Delta\xi^2, \tag{4.19}$$

$$\nabla_N x(\xi,\eta,t) = x(\xi,\eta+\Delta\eta,t) - x(\xi,\eta,t), \quad \nabla_S x(\xi,\eta,t) = x(\xi,\eta-\Delta\eta,t) - x(\xi,\eta,t), \tag{4.20}$$

$$\nabla_E x(\xi,\eta,t) = x(\xi+\Delta\xi,\eta,t) - x(\xi,\eta,t), \quad \nabla_W x(\xi,\eta,t) = x(\xi-\Delta\xi,\eta,t) - x(\xi,\eta,t), \tag{4.21}$$

and finally

$$\boxed{x(\xi,\eta,t+\Delta t) \approx x(\xi,\eta,t) + \Delta t\,\left(\Phi_N + \Phi_S + \Phi_W + \Phi_E\right)\big|_{\Delta\xi=1,\Delta\eta=1}.} \tag{4.22}$$

The filtering process consists of updating each pixel in the image by an amount equal to the flow contributed by its nearest neighbors. The parameter Δt should be equal or less than $1/2$ for the 2-neighborhood (4.14) and less than $1/8$ for the 8-neighborhood case (4.22) to ensure the stability of the iterative process. The 2D anisotropic diffusion for the 8-neighborhood is illustrated in Fig. 4.1, where the intensity of the central pixel is modified by the flow contributions from its eight neighboring points.

The discrete, iterative version of (4.3) can be written as

$$x_1^{t+1} = x_1^t + \lambda \sum_{k=2}^{N} \Phi_k = x_1^t + \lambda \sum_{k=2}^{N} c_k^t \left[x_k^t - x_1^t\right], \quad \text{for stability} \quad \lambda \le \lambda_0 = \frac{1}{N-1}, \tag{4.23}$$

where t denotes discrete time, (iteration number), c_k^t, $k = 2, \ldots, N$ are the diffusion coefficients in $N-1$ directions, (Fig. 2.1b), x_1^t denotes the central pixel of the filtering window, x_k^t are its neighbors and λ_0 is the largest value of λ (Δt in (4.14) and (4.22)), which guarantees the stability of the diffusion process.

It is quite easy to notice, [28] that this equation is s i m i l a r to the adaptive smoothing scheme proposed in [268, 269] and [255, 256]. Equation (2.10) formulated in an iterative way

$$x_1^{t+1} = \sum_{k=1}^{N} \psi_k\, x_k^t \Big/ \sum_{k=1}^{N} \psi_k\,, \tag{4.24}$$

can be written as

$$x_1^{t+1} = x_1^t + \frac{\sum_{k=1}^{N} \psi_k x_k^t - x_1^t \sum_{k=1}^{N} \psi_k}{\sum_{k=1}^{N} \psi_k} = x_1^t + \frac{\sum_{k=1}^{N} \psi_k (x_k^t - x_1^t)}{\sum_{k=1}^{N} \psi_k} = x_1^t + \sum_{k=2}^{N} \psi_k^* \left(x_k^t - x_1^t\right), \tag{4.25}$$

where ψ_k^* are the normalized weighting coefficients. In this way, e v e r y 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 (4.23) can be rewritten as

$$x_1^{t+1} = x_1^t \left[1 - \lambda \sum_{k=2}^{N} c_k^t\right] + \lambda \sum_{k=2}^{N} c_k^t\, x_k^t, \quad \lambda \le \lambda_0 = \frac{1}{N-1}. \tag{4.26}$$

If we set $[1 - \lambda \sum_{k=2}^{N} c_k^t] = 0$, then we can switch off to some extent the influence of the central pixel x_1^t in the iteration process. This requires however, that in each iteration step the λ has to be a variable, dependent on the image structure, equal to $\lambda^t = \left[\sum_{k=2}^{N} c_k^t\right]^{-1}$. The effect of diminishing the influence of the central pixel can be however achieved in a more natural way. Introducing the normalized conductivity coefficients $C_k^t = c_k^t / \sum_{k=2}^{N} c_k^t$, with $\sum_{k=2}^{N} C_k^t = 1$, the Eq. (4.26) takes the form

$$x_1^{t+1} = x_1^t\,(1 - \lambda^*) + \lambda^* \sum_{k=2}^{N} C_k^t\, x_k^t, \quad \lambda^* = \lambda \sum_{k=2}^{N} c_k^t, \quad \lambda^* \in [0,1], \tag{4.27}$$

which has the nice property, that for $\lambda^* = 0$ no filtering is performed: $x_1^{t+1} = x_1^t$ and for $\lambda^* = 1$, the central pixel is n o t t a k e n into the weighted average and the anisotropic smoothing scheme reduces to a weighted average of the neighbors of the central pixel x_1

$$x_1^{t+1} = \sum_{k=2}^{N} C_k^t\, x_k^t. \tag{4.28}$$

In this way the central pixel is being replaced by a weighted average of its neighbors and the weights correspond to the similarity measures of the central pixel and its neighbors. This scheme is very similar to the iterative approach proposed in [409, 410], where a gradient-inverse weighted noise smoothing algorithm was presented

$$x_1^{t+1} = c_1 x_1^t + \sum_{k=2}^{N} c_k\, x_k^t, \quad c_k = \frac{\max\{\gamma, |x_k - x_1)|\}}{S}, \quad S = \sum_{k=1}^{N} \max\{\gamma, |x_k - x_1|)\}, \tag{4.29}$$

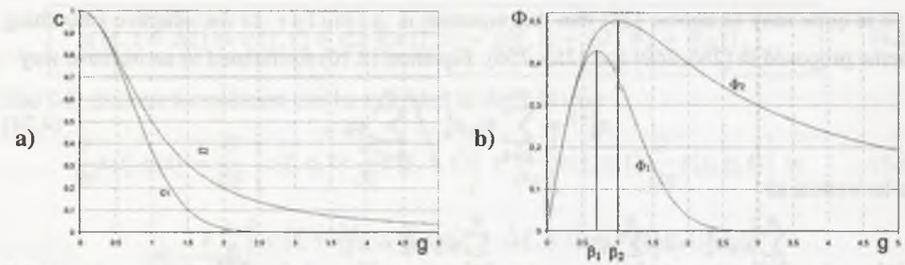


Fig. 4.2. Conductivity functions c_1, c_2 (4.6) (a) and respective plots of the flux functions (b)

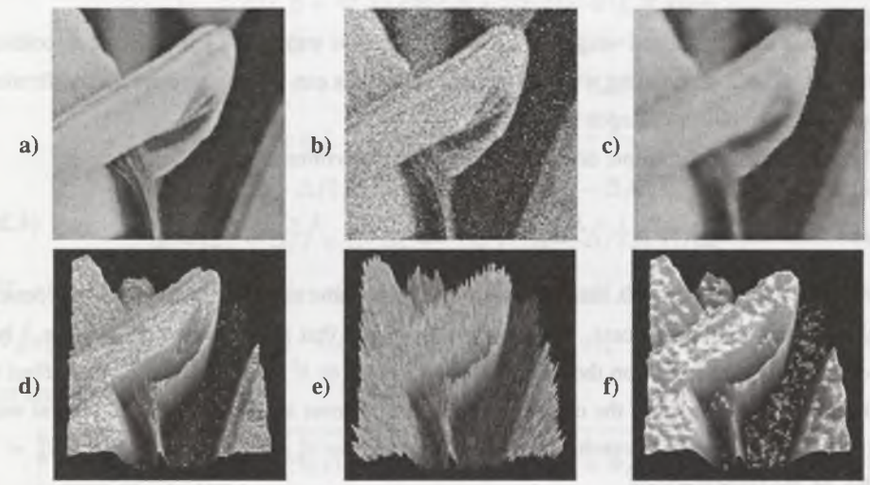


Fig. 4.3. Illustration of the PM scheme: a) part of LENA image, b) image contaminated by Gaussian noise of $\sigma = 30$, c) image restored with PM technique. Below respective 3-D visualizations, (d) - (f)

where γ influences the contribution of the central pixel in the averaging, (small γ value leads to (4.28)). Scheme (4.28) is also quite similar to the approach proposed in [166] and to the algorithm presented in [290]

$$x_1^{t+1} = \frac{1}{S}\sum_{k=2}^{N} c_k\, x_k^t\,, \quad c_k = \exp\left\{-\frac{\rho_k^2}{\beta_1^2}\right\}\exp\left\{-\frac{|x_k - x_1|^2}{\beta_2^2}\right\}, \quad k = 2,\ldots,N\,, \tag{4.30}$$

which corresponds to the case of $\lambda^* = 1$ in (4.27). The robustness of this scheme is achieved by r e j e c t i n g the central pixel value of the filter mask, when calculating the filter output. This technique is especially efficient when the image is corrupted by heavy impulsive noise process, as will be shown in Chapter 5.

Setting $\lambda^* = 1$ in (4.27) produces similar effect as taking the largest possible value of λ in (4.26), $\lambda_0 = 1/(N-1)$ which ensures the stability of the anisotropic diffusion process, [262].

The good performance of the anisotropic diffusion scheme with $\lambda^* = 1$ is confirmed by Fig. 4.5, which depicts the dependence of the efficiency of the PM approach using the c_1 conductivity function on the β and λ 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 λ close to $\lambda_0 = 1/8$, (3×3 filter mask), especially in the case of images distorted by Gaussian noise process of high σ. Such a setting of λ enables the diminishing of the influence of the central pixel x_1, which ensures the suppression of the outliers injected by the noise process.

One of the drawbacks of the anisotropic diffusion approach is that the optimal values of the parameters β and λ are unknown. Although β can be calculated using some a priori knowledge or can be estimated using some heuristic rules, [57, 168] the algorithm is relatively slow and requires 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, (average value of the image pixels), (Fig. 4.4), [276, 413, 425]. 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 image goes through the diffusion process without perceptible improvement, (see Fig. 4.14b).

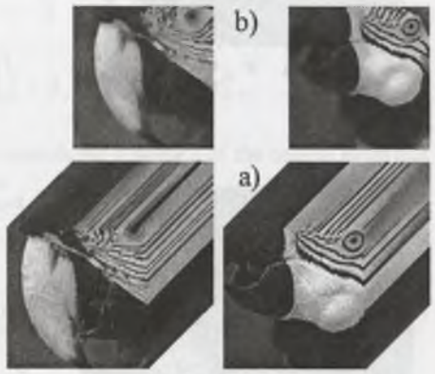


Fig. 4.4. Visualization of the the development of the anisotropic diffusion process. Parts of the test image PARROTS have been processed using the standard PM multichannel anisotropic diffusion scheme. The development of a selected image row and column through time (iterations) is presented, (a). It can be observed that weak edges are fused and only strong edges can be preserved. The final result obtained after 300 iterations is shown above, (b)

In order to improve the efficiency of the original PM scheme a regularized version was proposed, in which the conductance coefficient is a function of the gradient convolved with the Gaussian linear filter, [60, 61]

$$\frac{\partial x(\xi,\eta,t)}{\partial t} = \operatorname{div}\left[\tilde{c}(\xi,\eta,t)\nabla x(\xi,\eta,t)\right], \tag{4.31}$$

where $\tilde{c}(\xi,\eta,t) = f(|\nabla\, \mathcal{G}_\sigma \star x(\xi,\eta,t)|)$, $\mathcal{G}_\sigma$ denotes the Gaussian kernel with standard deviation σ, $\star$ 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 PM 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, (see Fig. 4.14c).

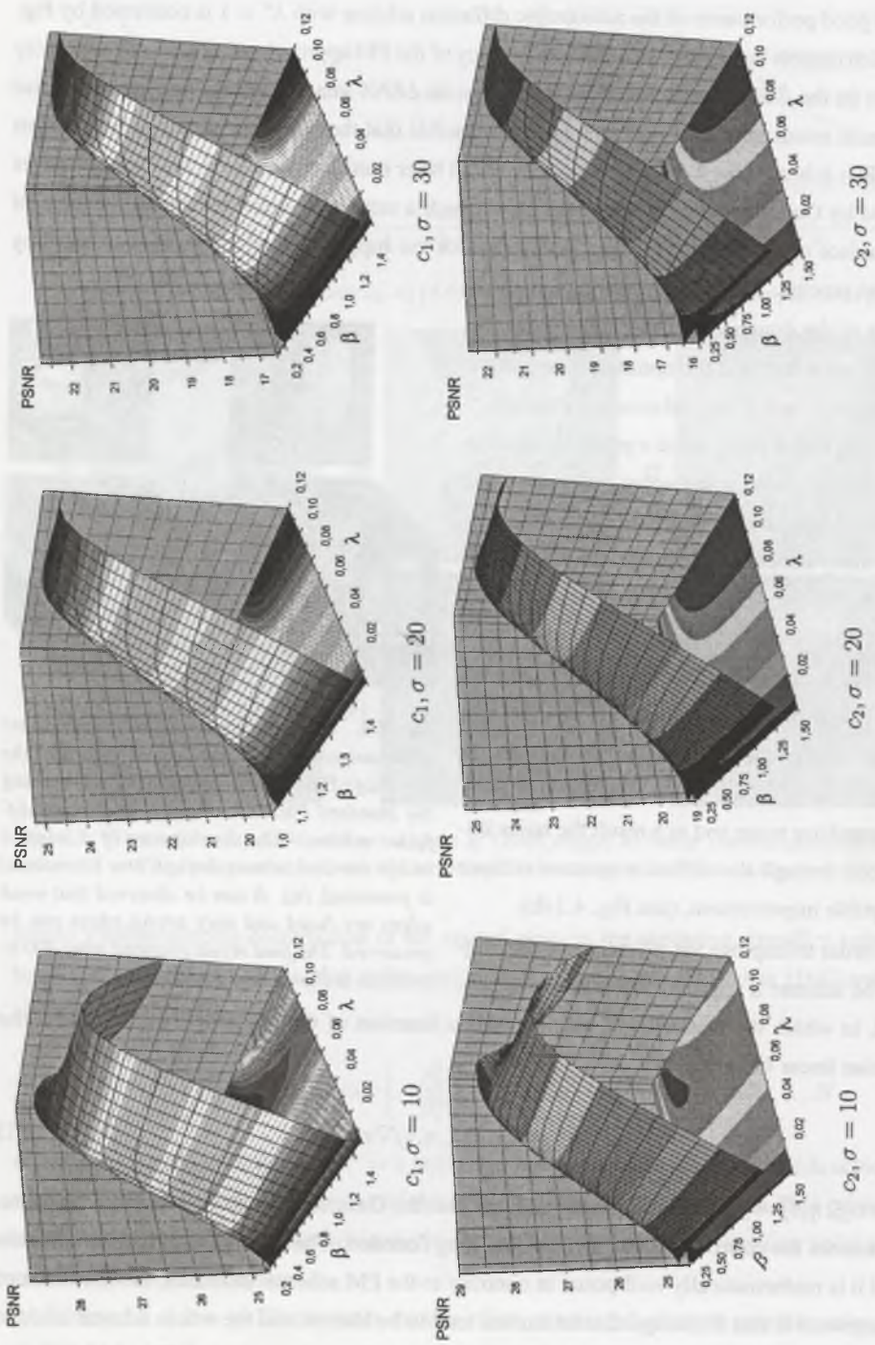


Fig. 4.5. Dependence of the efficiency of the PM scheme with the c_1 and c_2 conductivity function on the λ and β parameters, (4.6), (4.23) using the test gray scale image LENA contaminated with Gaussian noise of $\sigma = 10$, $\sigma = 20$ and $\sigma = 30$ in terms of PSNR for the third iteration

In [219] the so called *biased anisotropic diffusion* has been proposed. This scheme differs from the PM approach (4.23) in an additional term expressing the deviation between the initial image x^0 and the filtered image x

$$\frac{\partial x}{\partial t} = \nabla\left(c(x)\,\nabla x\right) + \alpha\left[x^0 - x\right], \tag{4.32}$$

where α ia a parameter. The discrete, iterative scheme is then given by

$$x_1^{t+1} = x_1^t + \lambda\left\{\left[\sum_{k=2}^{N} c_k^t\left(x_k^t - x_1^t\right)\right] + \alpha\left(x_1^0 - x_1^t\right)\right\}, \tag{4.33}$$

$$x_1^{t+1} = x_1^t\left(1 - \lambda\sum_{k=2}^{N} c_k^t - \lambda\alpha\right) + \lambda\sum_{k=1}^{N} c_k^t x_k^t + \lambda\alpha x_1^0, \tag{4.34}$$

Setting $[1 - \lambda\sum_{k=2}^{N} c_k^t - \lambda\alpha] = 0$ we are able to diminish the influence of the central pixel and obtain the time dependent variable $\lambda^* = \left(\alpha + \sum_{k=2}^{N} c_k^t\right)^{-1}$. Thus the iterative scheme, robust to impulsive noise is given by

$$x_1^{t+1} = \lambda^*\left(\sum_{k=2}^{N} c_k^t x_k^t + \alpha x_1^0\right). \tag{4.35}$$

The major advantage of this approach is that due to the bias term $(x_1^0 - x_1^t)$, the biased anisotropic diffusion scheme converges to a steady solution, which preserves image edges, [112]. In the case of very noisy images contaminated with mixed noise, the initial image x^0 can be replaced by its appropriate estimate (mean, median), which allows to significantly improve the filtering performance. Figure 4.6 depicts the dependence of PSNR on the iteration number for $\alpha = 0$ equivalent to (4.28) and $\alpha = 0.2$ in (4.35) for the *LENA* image contaminated by mixed noise NM5, ($p = 0.04$, NM2, $\sigma = 30$).

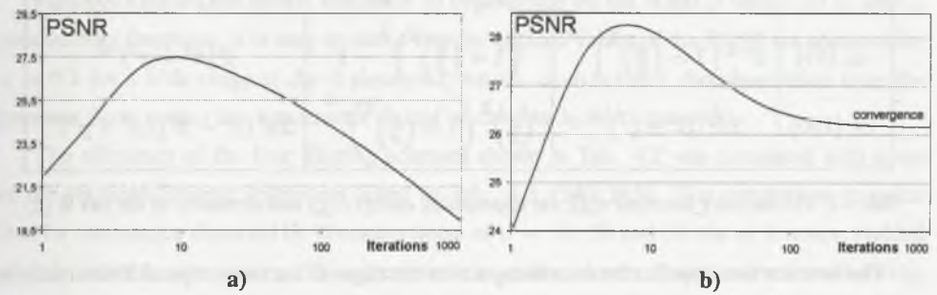


Fig. 4.6. Dependence of PSNR on the iteration number for a) $\alpha = 0$, (4.28) and b) $\alpha = 0.2$, (4.35), (LENA image contaminated by mixed noise, $p = 0.04$, NM2, $\sigma = 30$, c_1, $\beta = 20$)

4.2 Efficiency of the Anisotropic Diffusion Schemes

The properties of the image processing techniques, based on the anisotropic diffusion are determined by the *conductivity function* in the PDE equation, which defines the nonlinear diffusion process. Changing the shape of the conductivity function, we can tune the anisotropic diffusion filter to the image noise intensity and its statistical properties, in order to achieve optimal results of the image smoothing, [46, 47]. In this Section the behavior of the classical functions introduced by Perona and Malik together with the Tukey's biweight and Huber's estimator, [204] used in [46] is analyzed and the different filtering schemes are compared with the standard approaches used for the reduction of Gaussian noise in digital images.

One of the attempts to alleviate the problems connected with the inability of the classical anisotropic diffusion approach to suppress strong noise is the introduction of the so called *robust conductivity functions*. In [46] 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. This is caused by the strong influence of the central pixel in the filtering window on the development of the anisotropic diffusion scheme.

	$c(g)$	$\mathcal{E}(g)$	$\Phi'(g)$
c_1, [228]	$\exp\left(-\frac{g^2}{2\beta^2}\right)$	$\beta^2\left(1-\exp\left(-\frac{g^2}{2\beta^2}\right)\right)$	$\left[1-\frac{g^2}{\beta^2}\right]\exp\left(-\frac{g^2}{2\beta^2}\right)$
c_2, [228]	$\left(1+\left(\frac{g}{\beta}\right)^2\right)^{-1}$	$\frac{\beta^2}{2}\log\left(1+\left(\frac{g}{\beta}\right)^2\right)$	$\left[1-\frac{g^2}{\beta^2}\right]\left(1+\left(\frac{g}{\beta}\right)^2\right)^{-2}$
c_5, [119]	$\tanh\left(\frac{g}{\beta}\right)/(g\beta)$	$\log\cosh\left(\frac{g}{\beta}\right)$	$\beta^{-2}\left(\cosh\left(\frac{g}{\beta}\right)\right)^{-2}$
c_6, [63]	$\beta^{-2}\left(1+\left(\frac{g}{\beta}\right)^2\right)^{-\frac{1}{2}}$	$\left(1+\left(\frac{g}{\beta}\right)^2\right)^{\frac{1}{2}}-1$	$\beta\left(\beta^2+g^2\right)^{-\frac{3}{2}}$
c_7, [110]	$2\beta^2\left(\beta^2+g^2\right)^{-2}$	$\left(\frac{g}{\beta}\right)^2\left(1+\left(\frac{g}{\beta}\right)^2\right)^{-2}$	$2\beta^2\left(g^3-\beta^2\right)\left(\beta^2+g^2\right)^{-2}$

Tab. 4.1. Conductivity functions $c(g)$, the appropriate energy $\mathcal{E}(g)$ and derivative of the flux $\Phi'(g)$

The function that impedes the smoothing across the edges in the anisotropic diffusion scheme, is the diffusion coefficient. The conduction function $c(g)$ is space varying, (depending on the gradient magnitude g at a determined position) and is chosen to be large in homogeneous regions to encourage image smoothing and small at edges to preserve them, (Figs. 4.2, 4.3).

Perona and Malik originally suggested two choices of $c(g)$

$$c_1(g) = \exp\left\{-(g/\beta)^2/2\right\}, \qquad c_2(g) = \left(1+(g/\beta)^2\right)^{-1}. \tag{4.36}$$

The constant β can be made adaptive, using the gradient estimator described in [57], where a histogram of the absolute values of the gradients in the image is computed and β is usually set to obtain 90% of its integral.

Apart from the PM functions, different types of conductivity functions have been proposed over the years in the literature, [9, 46, 63, 262], (Tab. 4.1). In [46] robust statistic norms were chosen to define the conduction functions. The authors proposed there the so called *Tukey's biweight* function - $c_3(g)$ and the *Huber's min/max* function - $c_4(g)$

$$c_3(g) = \begin{cases} \frac{1}{2}\left[1-(g/\beta)^2\right]^2, & g \le \beta, \\ 0, & \text{otherwise}, \end{cases}, \quad c_4(g) = \begin{cases} 1/\beta, & |g| \le \beta, \\ 1/g, & \text{otherwise}. \end{cases} \tag{4.37}$$

In order to compare the efficiency of the standard anisotropic diffusion filtering schemes based on different conductivity functions, the LENA standard gray scale image was contaminated with zero-mean, additive Gaussian noise of $\sigma = 10$, 20 and 30 respectively. For each combination of λ and β the diffusion process was iterated until the maximum PSNR value was achieved. From all the combinations of the filters' parameters the optimal values of λ and β in terms of PSNR for each conductivity function was found and this value was treated as an indicator of the filter's performance, (Tab. 4.2), [340, 345].

σ	c_1	c_2	c_3	c_4
10	28.34	**28.59**	28.51	28.08
20	25.14	**25.49**	25.35	25.05
30	23.01	**23.19**	23.04	22.96

Tab. 4.2. Optimal efficiency of the anisotropic diffusion filters in terms of PSNR. For the evaluation purposes the gray scale LENA image distorted by zero-mean additive Gaussian noise of ($\sigma = 10$, 20, 30) using conductivity functions c_1, c_2, c_3 and c_4 was used

Figure 4.5 shows the filters' efficiency in dependence on the λ and β values for c_1 and c_2 conductivity functions. It is easy to notice that the optimal values of the PSNR are obtained for $\lambda \approx 0.1$ for a wide range of the β parameter, which confirms the observation from the previous Section, that the λ parameter should be close to λ_0, (λ^* close to 1).

The efficiency of the four filtering schemes shown in Tab. 4.2 was compared with some of the standard filtering techniques listed in Tab. 4.3, [340, 345]. The simulations revealed that for the images distorted by Gaussian noise of $\sigma = 10$, 20 and 30, the c_2 function yielded slightly better results than c_1. The experiments have also shown, that the robust conductivity functions c_3 and c_4 were not superior to the functions c_1 and c_2 originally proposed by Perona and Malik, which indicates that the shape of the conductivity function is not as important as could be expected.

The efficiency of the anisotropic diffusion decreases with the intensity of the Gaussian noise and for $\sigma = 30$ this filtering scheme is significantly worse than the simple α-trimmed mean, which by the way performs better than the 3×3 median. This inability of the anisotropic diffusion filters to suppress strong Gaussian noise can be derived from the fact, that strong impulses introduced by the noise process are perceived by the filters as edges and are not eliminated, which leads to a poor overall filter performance. This however can be alleviated by diminishing the importance of the central pixel, as has been shown in the previous Section.

FILTER	1	2	3	4	5	6	7	8	9
$\sigma = 10$	27.37	27.53	26.18	22.32	25.06	23.07	25.66	23.30	**28.59**
$\sigma = 20$	**26.31**	26.22	25.35	22.94	23.73	22.82	24.09	22.09	25.49
$\sigma = 30$	**25.03**	24.73	24.27	21.93	22.21	22.48	22.38	22.45	23.19

Tab. 4.3. PSNR results obtained with some of the standard filters using the same distorted LENA images as evaluated in Tab. 4.2: (1) α-trimmed mean with 2 excluded pixels, (2) α-trimmed mean with 4 excluded pixels, (3) moving average (3x3), (4) moving average (5x5), (5) median filter (3x3, 2 iterations), (6) median filter (5x5, 2 iterations), (7) median filter (3x3), (8) median filter (5x5), (9) AD with c_2

The extensive simulations revealed, that the robust conductivity functions do not improve the filter performance, which indicates that the shape of the conductivity function is not crucial to the filter efficiency and more effective scheme is needed in case of highly corrupted images. Such a solution will be presented and evaluated in Chapter 5.

4.3 Anisotropic Diffusion Applied to Color Images

The extension of the anisotropic diffusion framework to the multichannel case is not a very difficult task. Let $\mathbf{x}(\xi,\eta,t) = [x_r(\xi,\eta,t), x_g(\xi,\eta,t), x_b(\xi,\eta,t)]$ denotes a color image pixel at position (ξ,η), where $x_r(\xi,\eta,t)$, $x_g(\xi,\eta,t)$, $x_b(\xi,\eta,t)$ are the *red*, *green* and *blue* channels respectively. The PDE Eq. (4.3) can be written for the multichannel case as

$$\frac{\partial \mathbf{x}(\xi,\eta,t)}{\partial t} = \nabla\left[c_{rgb}(\xi,\eta,t)\nabla\mathbf{x}(\xi,\eta,t)\right], \mathbf{x}(\xi,\eta) = \begin{bmatrix} x_r(\xi,\eta) \\ x_g(\xi,\eta) \\ x_b(\xi,\eta) \end{bmatrix}, \frac{\partial \mathbf{x}(\xi,\eta)}{\partial t} = \begin{bmatrix} \frac{\partial x_r(\xi,\eta)}{\partial t} \\ \frac{\partial x_g(\xi,\eta)}{\partial t} \\ \frac{\partial x_b(\xi,\eta)}{\partial t} \end{bmatrix}, \quad (4.38)$$

where $c_{rgb}(\xi,\eta,t) = f(\|\mathbf{G}\|)$ is the conductivity function, the same for each image channel, dependent on the magnitude of the local gradient $\|\mathbf{G}\|$, which couples the three color image channels, [112, 170, 277, 386, 414]

$$\begin{bmatrix} \frac{\partial x_r(\xi,\eta,t)}{\partial t} \\ \frac{\partial x_g(\xi,\eta,t)}{\partial t} \\ \frac{\partial x_b(\xi,\eta,t)}{\partial t} \end{bmatrix} = \begin{bmatrix} \nabla\left[c_{rgb}(\xi,\eta,t)\nabla x_r(\xi,\eta,t)\right] \\ \nabla\left[c_{rgb}(\xi,\eta,t)\nabla x_g(\xi,\eta,t)\right] \\ \nabla\left[c_{rgb}(\xi,\eta,t)\nabla x_b(\xi,\eta,t)\right] \end{bmatrix}, \mathbf{G}(\xi,\eta) = \begin{bmatrix} \frac{\partial x_r(\xi,\eta)}{\partial \xi}, & \frac{\partial x_g(\xi,\eta)}{\partial \xi}, & \frac{\partial x_b(\xi,\eta)}{\partial \xi} \\ \frac{\partial x_r(\xi,\eta)}{\partial \eta}, & \frac{\partial x_g(\xi,\eta)}{\partial \eta}, & \frac{\partial x_b(\xi,\eta)}{\partial \eta} \end{bmatrix}. \quad (4.39)$$

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*, [58, 100, 254, 428]. Local variations of the color image $\|d\mathbf{x}\|^2$ are expressed as

$$\|d\mathbf{x}\|^2 = \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}^T \begin{bmatrix} g_{11}, & g_{12} \\ g_{21}, & g_{22} \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix}, \quad (4.40)$$

$$\begin{aligned} g_{11} &= \left(\frac{\partial x_r(\xi,\eta)}{\partial \xi}\right)^2 + \left(\frac{\partial x_g(\xi,\eta)}{\partial \xi}\right)^2 + \left(\frac{\partial x_b(\xi,\eta)}{\partial \xi}\right)^2 \\ g_{22} &= \left(\frac{\partial x_r(\xi,\eta)}{\partial \eta}\right)^2 + \left(\frac{\partial x_g(\xi,\eta)}{\partial \eta}\right)^2 + \left(\frac{\partial x_b(\xi,\eta)}{\partial \eta}\right)^2 \\ g_{12} &= \left(\frac{\partial x_r(\xi,\eta)}{\partial \xi}\right)\left(\frac{\partial x_r(\xi,\eta)}{\partial \eta}\right) + \left(\frac{\partial x_g(\xi,\eta)}{\partial \xi}\right)\left(\frac{\partial x_g(\xi,\eta)}{\partial \eta}\right) + \left(\frac{\partial x_b(\xi,\eta)}{\partial \xi}\right)\left(\frac{\partial x_b(\xi,\eta)}{\partial \eta}\right). \end{aligned} \quad (4.41)$$

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

$$\lambda_+ = \frac{g_{11} + g_{22} + \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}, \lambda_- = \frac{g_{11} + g_{22} - \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}, \quad (4.42)$$

are the extremum of $\|d\mathbf{x}\|^2$ and the orthogonal eigenvectors determine the corresponding variation directions. Based on the eigenvalues, different gradient norms leading to various PDE schemes can be developed, [48, 272, 282, 398, 400].

Figure 4.4 shows the development of the multichannel anisotropic diffusion process. As can be observed, this technique efficiently suppresses texture and low intensity Gaussian noise, but preserves strong edges. However, the process has to be stopped, as the image after many iterations is being heavily blurred and converges to the homogeneous image of the same color.

The anisotropic diffusion scheme has been generalized using the concepts of digital paths and fuzzy adaptive filters, [200, 295, 306, 313] [2]. Instead of using a fixed window, this method exploits connections between image pixels using the concept of *fuzzy connectedness*. According to the proposed methodology, image pixels are grouped together, forming paths that reveal the underlying structural dynamics of the color image.

[2] The generalization of anisotropic diffusion will be described in detail in Chapter 5.

4.4 Forward and Backward Anisotropic Diffusion

The conductance coefficients in the PM process are chosen to be a decreasing function of the signal gradient. This operation selectively smoothes regions that do not contain high gradients. In the *Forward-and-Backward diffusion* (FB), a different approach is taken. Its goal is to emphasize the extrema, if they indeed represent singularities and do not come as a result of noise. As we want to emphasize large gradients, we would like to move "mass" from the lower part of a "slope" upwards. This process can be viewed as moving back in time along the scale space, or reversing the diffusion process, [247]. Mathematically, we can change the sign of the conductance coefficient to negative

$$\frac{\partial}{\partial t}\mathbf{x}(\xi,\eta,t) = \nabla\left[-c(\xi,\eta,t)\,\nabla\,\mathbf{x}(\xi,\eta,t)\right], \quad c(\xi,\eta,t) > 0\,. \tag{4.43}$$

However, we cannot simply use an inverse linear diffusion process, because it is highly unstable. Three major problems associated with the linear backward diffusion process are: *explosive instability*, *noise amplification* and *oscillations*.

One way to avoid instability explosion is to diminish the value of the inverse diffusion coefficient at high gradients. In this way, when the singularity exceeds a certain gradient threshold, it does not continue to affect the process any longer. The diffusion process can be also terminated after a limited number of iterations. In order not to amplify noise, the inverse diffusion force at low gradients should be eliminated and the oscillations should be suppressed at the moment they are introduced.

The result of this analysis is that two forces of diffusion working simultaneously on the signal are needed - one backward force needed for *edge sharpening* and the other forward one, used for *stabilizing oscillations* and *reducing noise*. These two forces can actually be combined into one coupled *forward-and-backward diffusion force* with a conductance coefficient possessing both positive and negative values. In [113–115] a rather *ad hoc* conductivity function that controls the FB diffusion process has been proposed, (Fig. 4.7a)

$$c_{FB}(g) = \begin{cases} 1-(g/k_f)^{\gamma_1}\,, & 0 \le g \le k_f, \\ \alpha^*\left[\left((g-k_b)/w\right)^{2\gamma_2}-1\right], & k_b - w \le g \le k_b + w, \\ 0\,, & \text{otherwise}\,, \end{cases} \tag{4.44}$$

where g is an *edge indicator*, (gradient magnitude or the value of the gradient convolved with the Gaussian smoothing operator), $k_f, k_b, w, \gamma_1 > \gamma_2$ are design parameters and $\alpha^* = k_f/(2k_b)$, $(k_f \le k_b)$ controls the ratio between the forward and backward diffusion.

Later another form of the diffusivity function was proposed, [116], (Fig. 4.7b)

$$c_{FB}(g) = \frac{1}{1+(g/k_f)^{\gamma_1}} - \frac{\alpha^*}{1+((g-k_b)/w)^{2\gamma_2}}\,. \tag{4.45}$$

In this work a more natural approach based on the widely used *unsharp masking* technique is proposed, [291, 307, 338–340, 352]. Let us define the *unsharp masking* operation, [132, 167]

$$\mathbf{y} = \mathbf{x} \star I + \delta\,(\mathbf{x} \star I - \mathbf{x} \star \mathcal{G}) = \mathbf{x} \star [I(1+\delta) - \delta\mathcal{G}]\,, \tag{4.46}$$

where I is the identity operator, $\mathcal{G}$ is the Gaussian operator, $\star$ denotes the componentwise convolution of the multichannel signals and δ is a parameter which influences the sharpening effect.

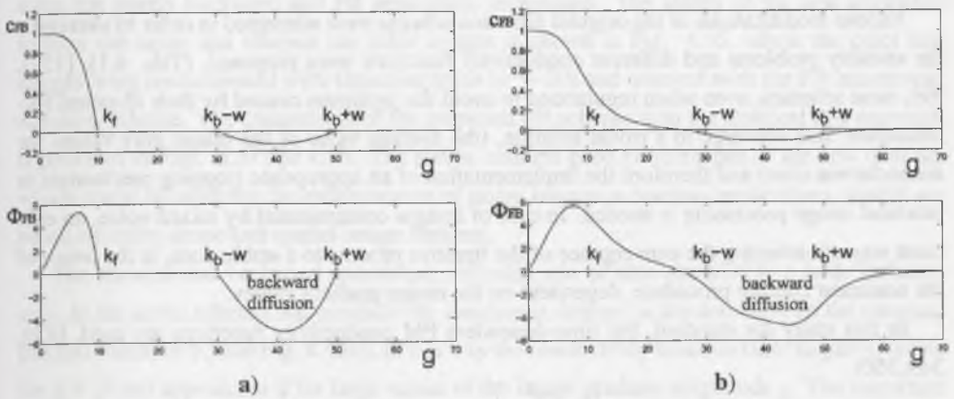


Fig. 4.7. Plots of the forward and backward conductivity and flux functions proposed in [113–115] (a) and in [116] (b)

The unsharp masking defined by (4.46) is equivalent to the anisotropic diffusion with a conductivity function defined as

$$\mathcal{C}(\delta, g, \beta) = 1 + \delta - \delta \exp\left\{-\frac{g^2}{2\beta^2}\right\}, \tag{4.47}$$

which satisfies $\mathcal{C}(g=0) = 1$. However for $\delta > 0$, $\mathcal{C} > 1$ which causes that the anisotropic diffusion based on the direct extension of the unsharp masking method would lead to a highly unstable scheme, which would very quickly collapse when used in an iterative way. Therefore a more stable solution is needed, which can be provided by the scheme defined by

$$\mathbf{y} = \mathbf{x} \star \mathcal{G}_1 + \delta(\mathbf{x} \star \mathcal{G}_1 - \mathbf{x} \star \mathcal{G}_2)\,, \tag{4.48}$$

where $\mathcal{G}_1$ and $\mathcal{G}_2$ are two Gaussian operators. The aim of $\mathcal{G}_1$ is to suppress the image noise and $\mathcal{G}_2$ is needed to perform the unsharp masking operation. Equation 4.48 can be rewritten as

$$\mathbf{y} = \mathbf{x} \star [\mathcal{G}_1 + \delta(\mathcal{G}_1 - \mathcal{G}_2)] = \mathbf{x} \star [(1+\delta)\mathcal{G}_1 - \delta\mathcal{G}_2)]\,. \tag{4.49}$$

The anisotropic diffusion scheme based on (4.49) is then parameterized by δ and β_1, β_2

$$\mathcal{C}(\delta, g, \beta) = (1+\delta)\exp\left\{-g^2/\left(2\beta_1^2\right)\right\} - \delta\exp\left\{-g^2/\left(2\beta_2^2\right)\right\}, \tag{4.50}$$

where $\beta_1 < \beta_2$. Defined in this way conductivity coefficient has the required properties: $\mathcal{C}(g = 0) = 1$, $\mathcal{C}(g > 0) < 1$ and $\lim_{g\to\infty}\mathcal{C}(g) = 0$. Setting the δ parameter to 1 we obtain two conduction coefficients directly based on the PM approach, [291,307,338]

$$c_{1_{FB}} = 2\exp\left\{-\frac{1}{2}\left(\frac{g}{\beta_1}\right)^2\right\} - \exp\left\{-\frac{1}{2}\left(\frac{g}{\beta_2}\right)^2\right\}, \; c_{2_{FB}} = \frac{2}{1+\left(\frac{g}{\beta_1}\right)^2} - \frac{1}{1+\left(\frac{g}{\beta_2}\right)^2}. \tag{4.51}$$

Various modifications of the original diffusion scheme were attempted in order to overcome the stability problems and different conductivity functions were proposed, (Tab. 4.1), [159]. Yet, most schemes, even when regularized to avoid the problems caused by their ill-posed formulations, still converge to a trivial solution, (the average value of the image gray values for monochrome case) and therefore the implementation of an appropriate stopping mechanism in practical image processing is needed. In case of images contaminated by mixed noise, an efficient way of enforcing the convergence of the iterative process to a stable state, is the usage of the nonlinear *cooling* procedure, dependent on the image gradient values.

In this study the standard, but time-dependent PM conductivity functions are used, [339, 349,350]

$$c_1(g,t) = \exp\left[-\frac{g^2}{2\beta(t)^2}\right], \quad c_2(g,t) = \frac{1}{1+\left(\frac{g}{\beta(t)}\right)^2}, \tag{4.52}$$

to obtain new *forward and backward* conductivity functions, derived from the unsharp masking technique

$$c_{1_{FB}} = 2\exp\left[-\frac{g^2}{2\beta_1(t)^2}\right] - \exp\left[-\frac{g^2}{2\beta_2(t)^2}\right], c_{2_{FB}} = \frac{2}{1+\left(\frac{g}{\beta_1(t)}\right)^2} - \frac{1}{1+\left(\frac{g}{\beta_2(t)}\right)^2}. \tag{4.53}$$

where $g = \|\nabla \mathbf{x}(\xi,\eta,t)\|$ is the vector norm, $\beta_j(n+1) = \beta_j(n)\cdot\alpha$, $\alpha \in (0,1]$, $\beta_j(1)$ is the starting parameter, $j = 1,2$, $\beta_1(n) < \beta_2(n)$, where n is the iteration number.

This scheme depends only on two (in case of forward or backward diffusion) or three (in case of FB diffusion) parameters: initial values of the two starting β_j parameters and the cooling rate α. Setting α to 1 means, that there is no cooling in the system. As α decreases, the cooling is faster, less noise is being filtered but edges are better preserved.

Figure 4.8 illustrates the dependence of the PM diffusion coefficients $c_1(g,t)$ and $c_2(g,t)$ on the iteration step n. The plots of the forward and backward diffusion coefficients $c_{1FB}(g,t)$ and $c_{2FB}(g,t)$ are presented in Fig. 4.9. In the FB diffusion process, smoothing is performed

when the conductivity function is positive and sharpening takes place for negative conduction coefficient values. If the cooling coefficient α is lower than 1, then the gradient threshold $\beta(t)$ decreases with time, allowing lower and lower gradients to take part in the smoothing process. As the iteration step advances, only smoother regions are being filtered, whereas large gradients can get enhanced due to local backward diffusion. The scheme converges to a steady state for $\beta \to 0$, which means that no diffusion at all is taking place.

The experiments revealed that better results of noise suppression using the FB scheme were achieved using the conductivity function c_2 from the original PM approach. The efficiency of the proposed technique is presented in Fig. 4.15, where two color images are enhanced using the purely backward and FB anisotropic techniques. The ability of the new algorithm to filter out noise and sharpen the color images is shown in Fig. 4.16, where the color test images were contaminated with Gaussian noise ($\sigma = 30$) and restored with the FB anisotropic diffusion scheme. The comparison of the proposed FB scheme with the classical PM approach is provided in Figs. 6.27 and 4.18. The results confirm good performance of the new method, which could be used for the enhancement of noisy images in various applications, which are based on color, shape and spatial image features.

The forward and backward anisotropic diffusion can be also obtained in a quite different way. In the novel scheme, the conductivity function is defined as the derivative of the classical PM flux function Φ, (see Fig. 4.13b). In this way the conductivity function takes negative values for $g > \beta$ and approaches 0 for large values of the image gradient magnitude g. The important feature of the FB scheme is that the parameter β is now time-dependent and is decreasing with time (iteration), (Fig. 4.13). This causes that the maximum and minimum of the flux function are approaching zero in successive iterations, which g u a r a n t e e s that the diffusion process converges quickly to a non-trivial solution.

Fig. 4.11 shows the plots of the conductivity functions used in the PM approach c_1, c_2, [228] and c_5 [119], c_6 [63] defined in Tab. 4.1. It can be seen that the c_5 and c_6 conductivities lead to convex energy functions, which is a condition for the regularization of the diffusion process, [82, 159]. As can be observed in Fig. 4.11 the PM conductivity functions c_1 and c_2 yield non-convex energy functions, which is the source of the stability problems of the PM approach, [82,276,413]. Figure 4.12 depicts the conductivity functions, the fluxes and energies for the FB scheme defined by (4.53). Note that the shapes of the functions defined by (4.53) are quite similar to those shown in Fig. 4.13, which confirms the similarity of the two proposed approaches. As can be seen the energy functions in Figs. 4.12 and 4.13 are like in the PM scheme non-convex, (see Fig. 4.11) and the shapes of the conductivity functions are determined by the term $[1 - g^2/\beta^2]$, (Tab. 4.1).

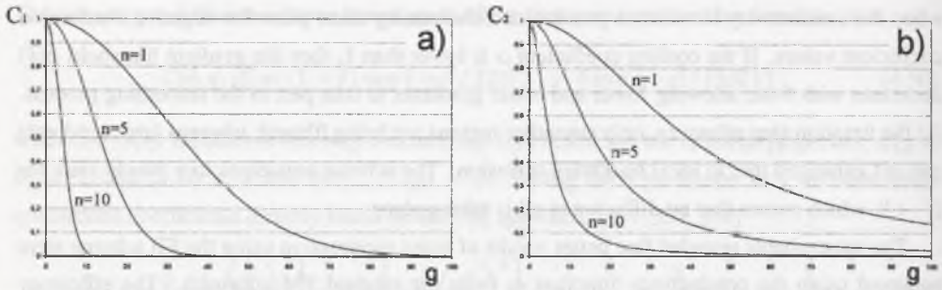


Fig. 4.8. Dependence of the conductivity functions on the iteration step and the image gradient g for the 'cooled' c_1 and c_2 conductivity functions, (forward diffusion, $\beta_1 = 40$, $\alpha = 0.8$)

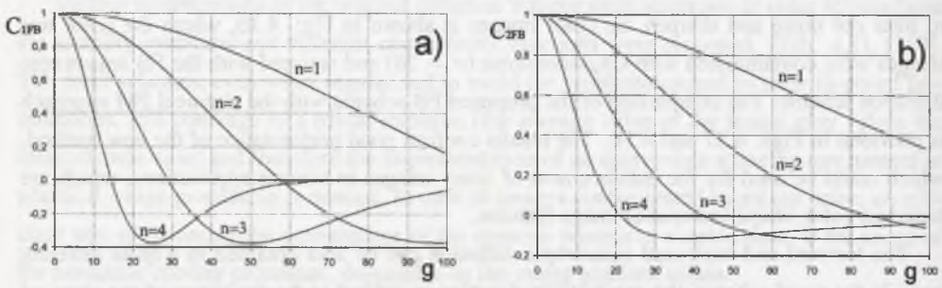


Fig. 4.9. Dependence of the forward and backward conductivity functions on the iteration number n and the image gradient g for the c_{1FB} and c_{2FB} conductivity functions defined by (4.51) and (4.53) for $\beta_1(1) = 40$, $\beta_2(1) = 80$ and $\alpha = 0.5$. Note, that because of low α already in the second iteration, the conductivity functions attain negative values

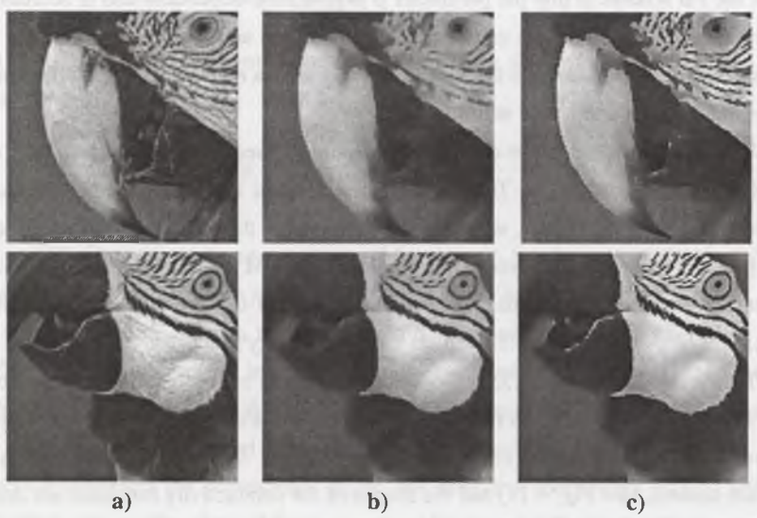


*Fig. 4.10. Comparison of the standard PM anisotropic diffusion scheme with the proposed forward and backward diffusion (FB), **a)** parts of the test image PARROTS, **b)** output of the PM technique after 50 iterations, **c)** result of the filtering using the proposed FB design after 6 iterations*

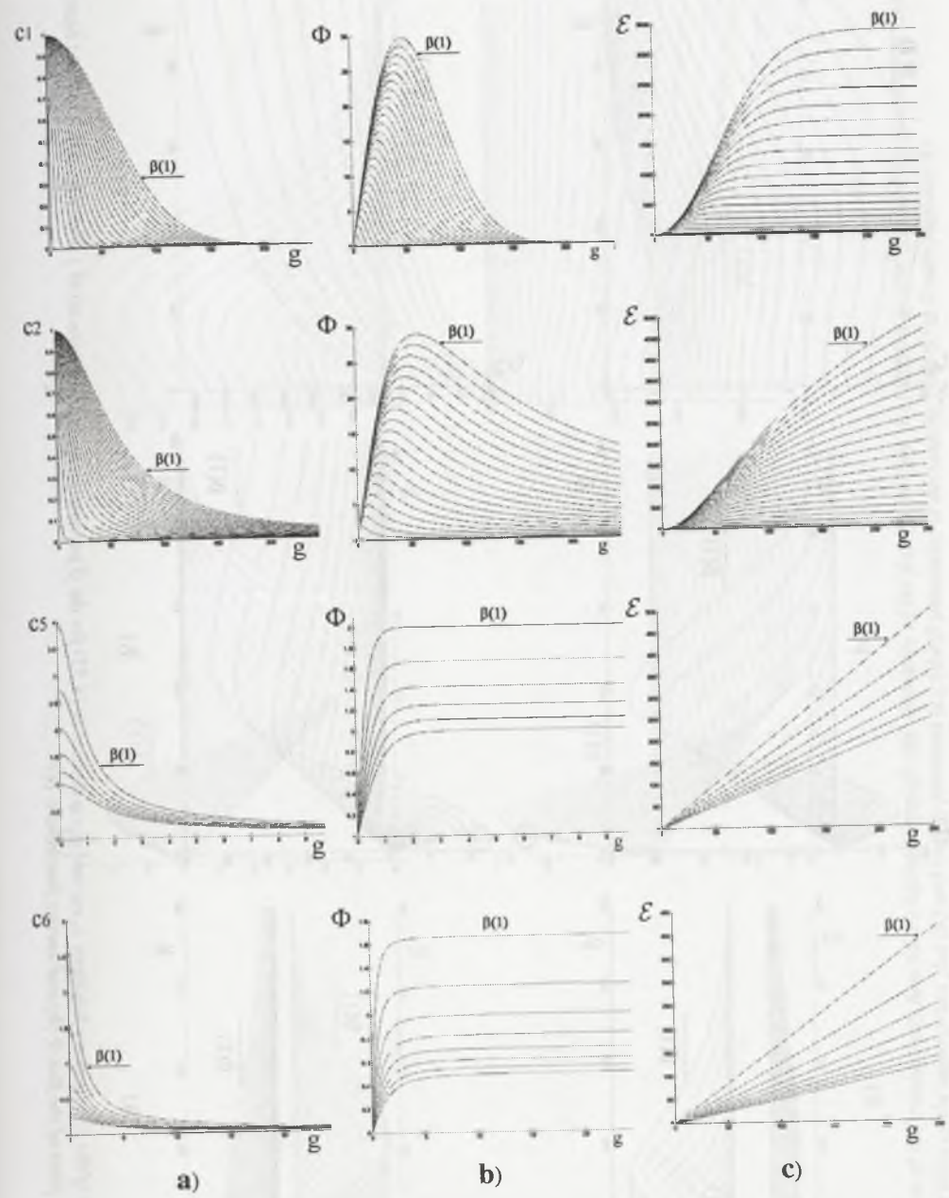


*Fig. 4.11. Plots of the conductivity functions used in the P-M approach c_1, c_2, [228] and c_5 [119], c_6 [63] defined in Tab. 4.1 for the β parameter decreasing from 60 to 1 (**a**), beside the appropriate plots of the flux functions Φ (**b**) and the energy functions $\mathcal{E}$ (**c**)*

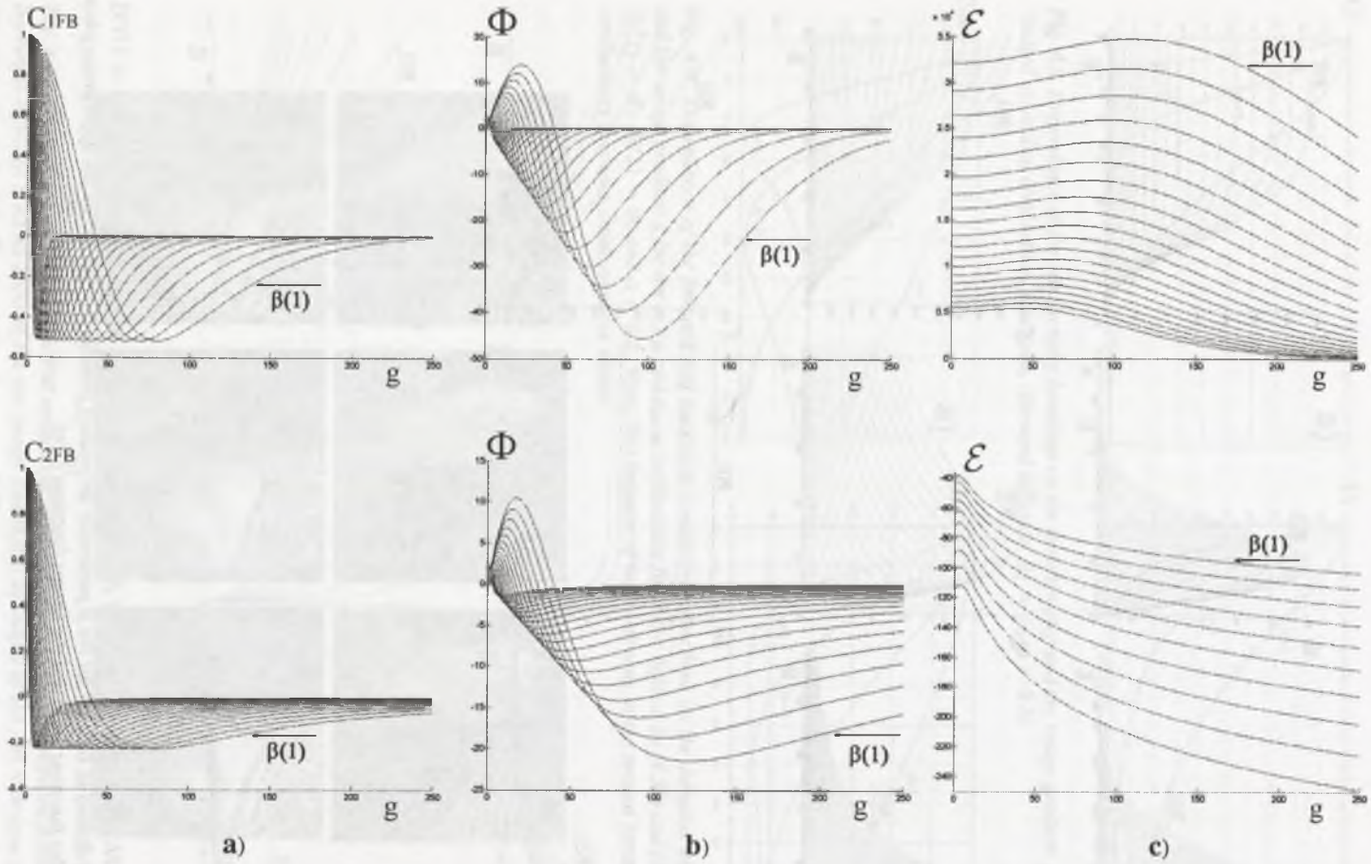


*Fig. 4.12. Plots of the conductivity functions c_{1FB} and c_{2FB} defined in (4.53) for the β parameter decreasing from 60 to 1 with step-size 3 (**a**) and beside the appropriate plots of the flux Φ (**b**) and energy functions $\mathcal{E}$ (c)*

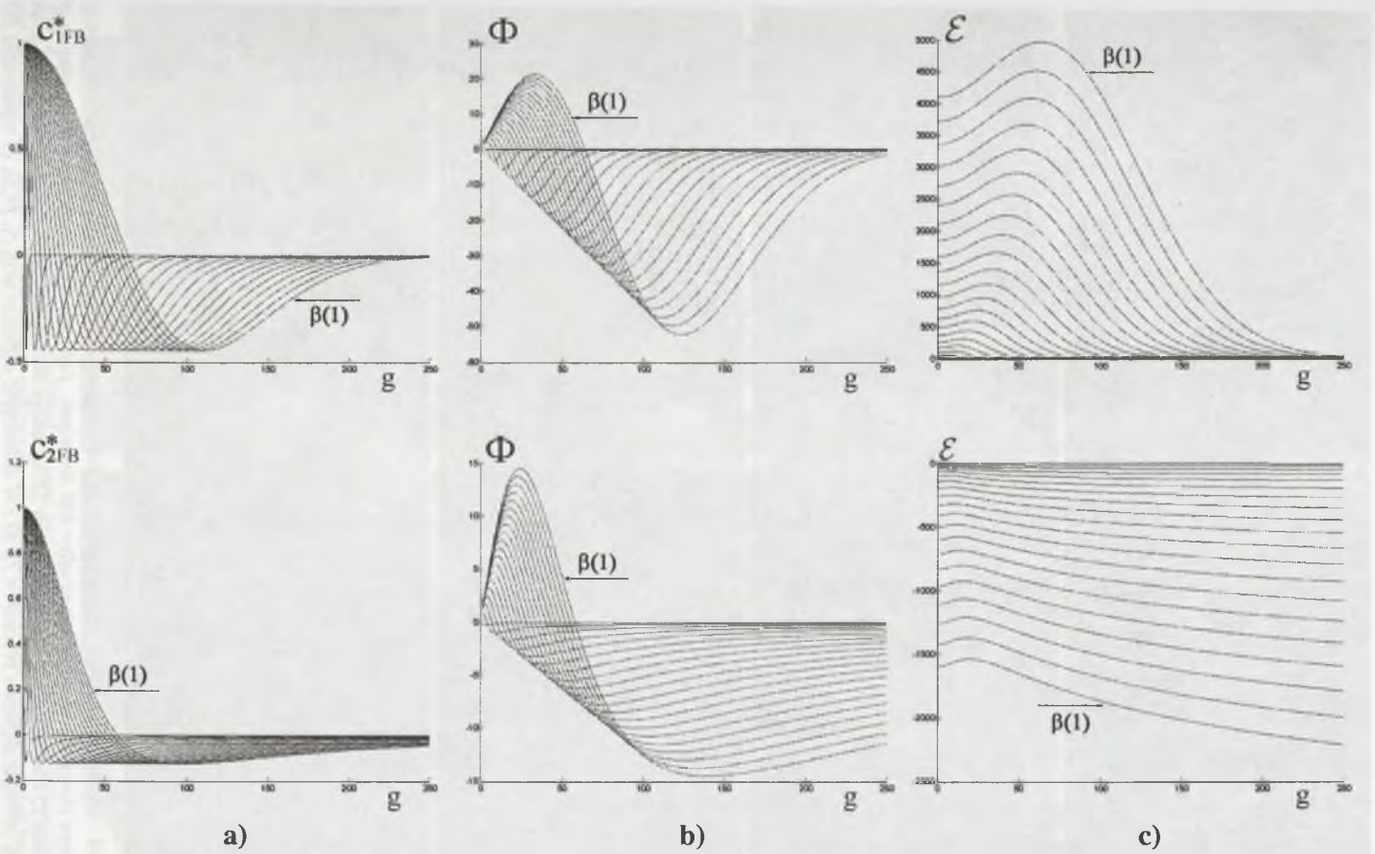


*Fig. 4.13. Plots of the conductivity functions obtained through the derivative of the flux (Φ' in Tab. 4.1) defined by PM conductivity functions c_1 and c_2 (**a**), beside the plots of the appropriate flux (**b**) and energy functions (**c**), (the β parameter is decreasing from 60 to 1 with step-size 3)*

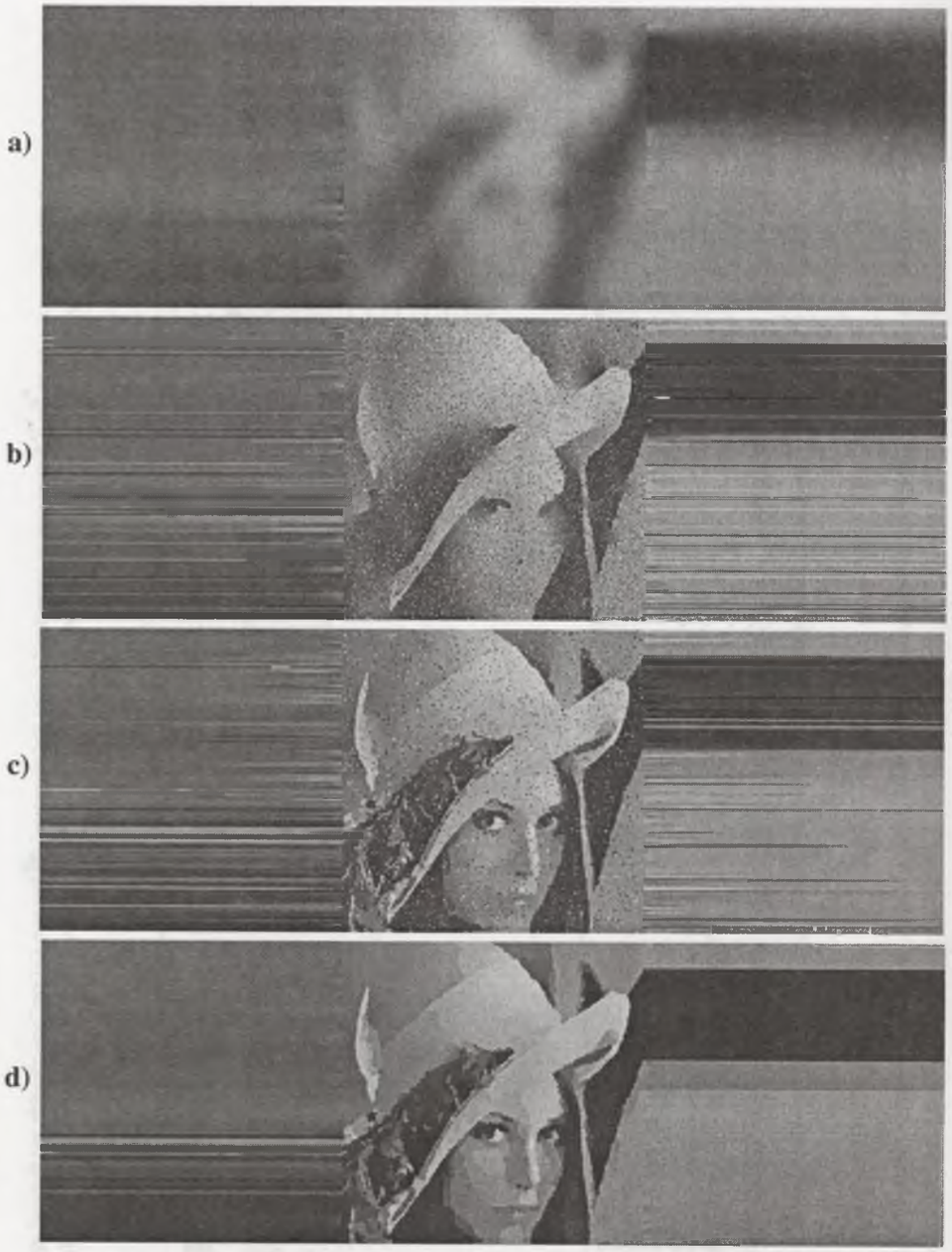


Fig. 4.14. Illustrations of the the development of the anisotropic diffusion process. The central part of the images shows the result obtained after 300 iterations. Left and right parts show the evolution of the column 25 and 325 of the 350×350 *color LENA image distorted by mixed impulsive and Gaussian noise,* ***a)*** *isotropic diffusion process (4.2),* ***b)*** *PM anisotropic diffusion with* c_1*, (4.6),* ***c)*** *regularized AD of Catté, [60, 61],* ***d)*** *new filter DPAF introduced in Chapter 5*

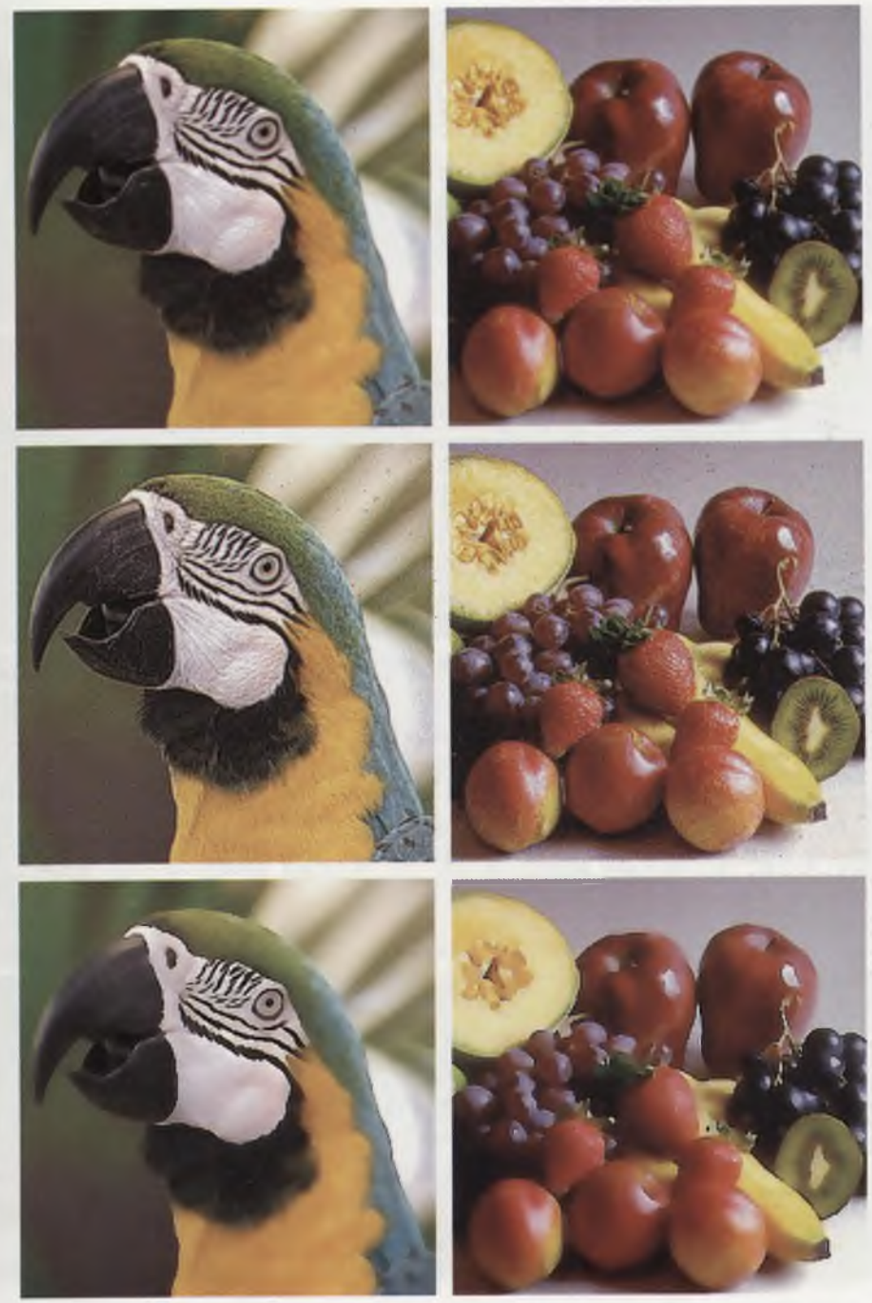


Fig. 4.15. Illustration of the proposed combined forward and backward anisotropic diffusion scheme. At the top: color test images, below images enhanced with the pure backward diffusion and at the bottom images enhanced with the FB diffusion scheme

Fig. 4.16. Effectiveness of the coupled FB diffusion scheme. Left column: color test images, in the center: images contaminated with additive Gaussian noise ($\sigma = 30$), to the right: images enhanced with the proposed FB anisotropic diffusion scheme

*Fig. 4.17. Illustration of the efficiency of the FB anisotropic diffusion scheme, **a)** color test images, **b)** images enhanced with the FB scheme. Figs. **c)** and **d)** depict the results obtained with the PM approach, using the conductivity functions c_1 and c_2, respectively after 50 iterations*

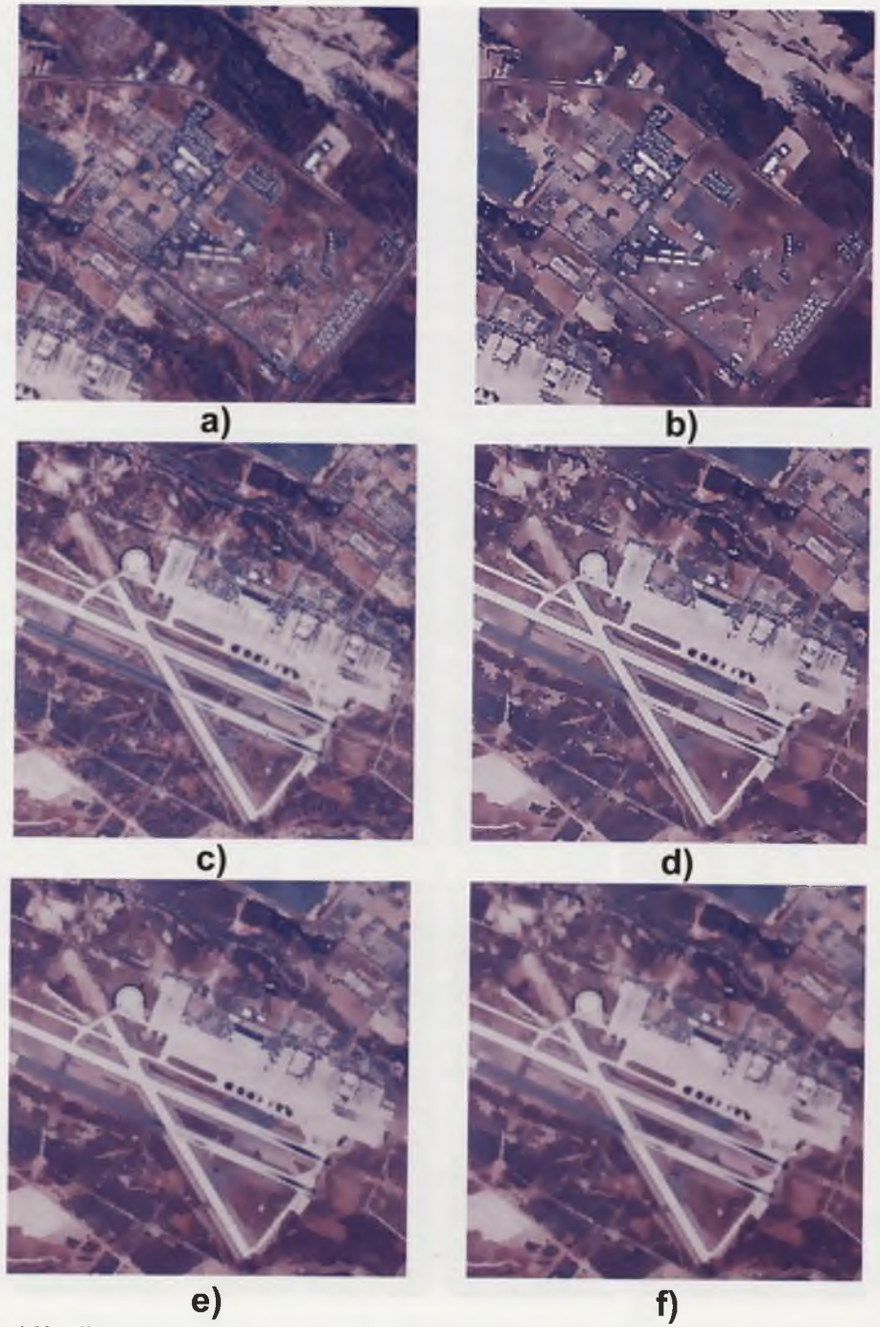


*Fig. 4.18. Illustration of the efficiency of the new filtering scheme: **a)** and **c)** test images and beside images enhanced with the proposed method (**b**), (**d**). Below the evolution of the test image **c)** using the standard PM scheme with the c_2 conductivity function after 20 and 50 iterations (**e**), (**f**)*

5

Digital Paths Approach to Color Image Filtering

In this Chapter a novel method of noise reduction in color images is presented. The class of filters presented here, utilizes fuzzy membership functions defined over vectorial inputs connected by digital paths. Instead of using a fixed window, the new method exploits connections between image pixels using the concept of digital paths. According to the described technique, image pixels are grouped together, forming paths that reveal the underlying structural dynamics of the image.

The efficiency of the new filters is evaluated under a variety of performance criteria and compared with the standard filters. It is shown that, compared to existing techniques, the filters presented here are better able to suppress impulsive, Gaussian and mixed noise. Furthermore, the computational analysis provided in this Chapter shows, that some members of the new filter family are computationally less demanding than the standard, widely used vector median filter.

5.1 Connection Cost over Digital Paths

DEPENDING on the design principles and the computational constraints, the new filter framework allows the digital paths exploring the image to be considered on the entire image domain, [298,299,302,319] or to be restricted to a predefined search area, [295,302,306, 313–316]. 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 enables 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, [313,326,380].

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 Chapter 4 and an extension of the *fuzzy adaptive filters* described in Section 3.4. 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 single step.

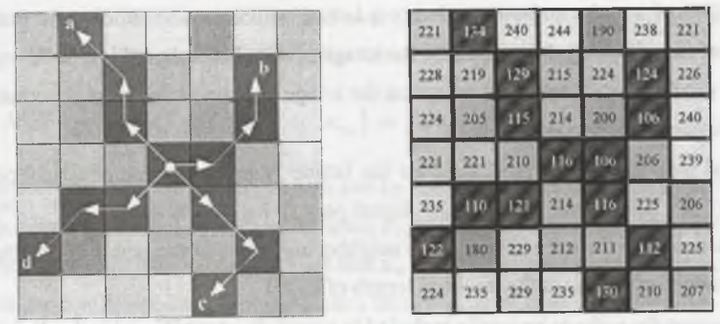
The path connection costs evaluated over all possible digital paths, are used here to derive *fuzzy membership functions* that quantify the similarity between vectorial inputs. The proposed filtering structure is then using the function outputs to appropriately weight input contributions in order to determine the filtering result. The proposed filtering schemes parallelize the familiar structure of the adaptive multichannel filters 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.

In order to perform operations based on the distances, we first need to precisely define the notion of the *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, [50,51,154,172,251,285].

Let $\mathcal{D}$ be any nonempty set. We can measure distances between points in $\mathcal{D}$, defining a real valued function on the Cartesian product $\mathcal{D}\times\mathcal{D}$ of $\mathcal{D}$. Let the function $\rho : \mathcal{D}\times\mathcal{D} \rightarrow \mathbb{R}$ be called a distance if it satisfies: $\rho(u,v) \geq 0$, with $\rho(u,v) = 0$ when $u = v$ and $\rho(u,v) = \rho(v,u)$, for all $u, v \in \mathcal{D}\times\mathcal{D}$. A distance is called a metric if additionally it satisfies the triangle inequality: $\rho(u,w) \leq \rho(u,v) + \rho(v,w)$, for all $u, v, w \in \mathcal{D}\times\mathcal{D}$, [149,259].

In digital image processing three basic distance functions are usually applied. If $u = (u_1, u_2)$ and $v = (v_1, v_2)$ denote two image points ($u, v \in \mathbb{Z}^2$), then we define the *city-block distance*: $\rho_4(u,v) = |u_1 - v_1| + |u_2 - v_2|$, *chessboard distance*: $\rho_8(u,v) = \max\{|u_1 - v_1|, |u_2 - v_2|\}$ and *Euclidean distance*: $\rho_E(u,v) = [(u_1 - v_1)^2 + (u_2 - v_2)^2]^{\frac{1}{2}}$. Using the city-block and chessboard distances we are able to define the two basic types of neighborhoods: *4-neighborhood* $\mathcal{N}_4(u) = \{v : \rho_4(u,v) = 1\}$ and *8-neighborhood* $\mathcal{N}_8(u) = \{v : \rho_8(u,v) = 1\}$.

Let $\iota \in \{4, 8\}$, then two points $u, v \in \mathbb{Z}^2$ are said to be in $\mathcal{N}_\iota$-neighborhood relation, (denoted as $\leftrightarrow$) or to be $\mathcal{N}_\iota$-adjacent if $v \in \mathcal{N}_\iota(u)$ or equivalently $u \in \mathcal{N}_\iota(v)$. This $\mathcal{N}_\iota$-adjacency relation defines a *graph structure* on the image domain, called $\mathcal{N}_\iota$-adjacency graph. On the graph, a finite $\mathcal{N}_\iota$-path can be defined as a sequence of points $(q_0, q_1, \ldots, q_\eta)$ such that for $k \in \{1, 2, \ldots, \eta\}$ the point q_{k-1} is $\mathcal{N}_\iota$ adjacent to q_k. A path is called simple if $k \neq j$ implies that $q_k \neq q_j$. 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*, [71,202,320,322,370].



a) b)

Fig. 5.1. 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

Using the distances between neighboring points, which are called *prime distances*, [371] 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 routes, 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 assumed in this work to be equal to 1.

Let us now introduce the definition of a *geodesic distance*, [252,266] assuming that $\mathbb{R}^2$ is the Euclidean space, $\mathbb{W}$ is a planar subset of $\mathbb{R}^2$ and u, v are points belonging to set $\mathbb{W}$. A path from u to v is a continuous mapping $\mathcal{Q}$: $[a, b] \rightarrow \mathbb{W}$, such that $\mathcal{Q}(a) = u$ and $\mathcal{Q}(b) = v$. The point u is considered as the starting point, while v is the ending point on the path $\mathcal{Q}$, [51,84,152,261].

An increasing polygonal line $\mathbb{Q}$ on path $\mathcal{Q}$ is any polygonal line such that $\mathbb{Q} = \{\mathcal{Q}(q_k)\}_{k=0}^{\eta}$, $a = q_0 <, \ldots, < q_\eta = b$. The length $\mathcal{L}$ of the polygonal line $\mathbb{Q}$ is considered to be the total sum of its constitutive line segments $\mathcal{L}(\mathbb{Q}) = \sum_{k=1}^{\eta} \rho(\mathcal{Q}(q_{k-1}), \mathcal{Q}(q_k))$, where $\rho(u,v)$ is the distance between the points u and v, when a specific metric is adopted. A path $\mathcal{Q}$ from u to v is called

rectifiable, if and only if $\mathcal{L}(\mathbb{Q})$, where $\mathbb{Q}$ is an increasing polygonal line, is bounded. Its upper bound is called the length of the path $\mathcal{Q}$.

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

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 $\mathcal{H} = (\mathbf{x}, \mathcal{N})$ be defined by $\mathbf{x}$, which is the set of all points on the image domain Ω and a neighborhood relation $\mathcal{N}$ ($\leftrightarrow$) between the lattice points, [278].

A digital path $\mathbb{Q} = \{q_i\}_{k=0}^{\eta}$ defined on the lattice $\mathcal{H}$ is a sequence of neighboring points $(q_{k-1}, q_k) \in \mathcal{N}$. The length $\mathcal{L}(\mathbb{Q})$ of the digital path $\mathbb{Q}\{q_k\}_{k=0}^{\eta}$ is simply $\sum_{k=1}^{\eta} \rho^{\mathcal{H}}(q_{k-1}, q_k)$, where $\rho^{\mathcal{H}}$ denotes the distance between two neighboring points of the lattice $\mathcal{H}$ and the geodesic distance between q_0 and q_η is the minimal length of $\mathcal{L}(\mathbb{Q})$.

Constraining the paths to be totally included in a predefined set W yields the digital geodesic distance ρ^W. In this work $\mathcal{N}_\iota$-neighborhood system ($\iota = 4$ or $\iota = 8$) is considered, with a topological distance of 1 assigned to any neighboring points, and the set W will be the supporting window of appropriate size. All paths considered in this Chapter are included in the filtering window W, (Fig. 5.2).

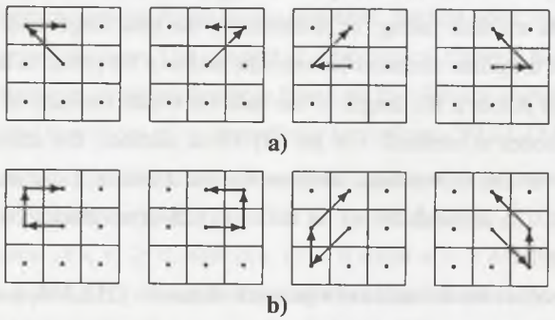


Fig. 5.2. Digital paths of a) length 2 and b) length 3, connecting two neighboring points within a predefined window W of size 3 × 3, when the 8-neighborhood system is applied

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 $q_0 = (i_{u_0}, j_{v_0})$. Its neighbors will be denoted as $q_1 = (i_{u_1}, j_{v_1})$, which means that the neighbors are the second points of all digital paths originating at q_0. Then the third point of a digital path starting at q_0

will be $q_2 = (i_{u_2}, j_{v_2})$ and so on, till the path reaches in η steps the ending point $q_\eta = (i_{u_\eta}, j_{v_\eta})$. In this way the sequences $i_{u_1}, \ldots, i_{u_\eta}$ and $j_{v_1}, \ldots, j_{v_\eta}$ uniquely define the digital path starting at u_{i_0}, v_{j_0} and ending at u_{i_η}, v_{j_η}. The set of all possible digital paths contained in W joining two points $u, v \in W$ will be denoted as $\Psi^W(u,v)$.

Two pixels u and v will be called connected (denoted as $u \Leftrightarrow v$), if there exists a digital path $\mathbb{Q}^W(u,v)$ contained in the set W starting from u and ending at v. If two pixels at positions q_0 and q_η are connected by a digital path $\mathbb{Q}^{W,\eta}\{q_0, q_1, \ldots, q_\eta\}$ of length η then let $\Lambda^{W,\eta}\{q_0, q_1, \ldots, q_\eta\}$ be a measure of the *connection cost* defined over the digital path linking the starting point q_0 and ending point q_η, (f is a nonnegative scalar function of vector variables)

$$\Lambda^{W,\eta}\{q_0, \ldots, q_\eta\} = f\{\mathbf{x}_{q_0}, \ldots, \mathbf{x}_{q_\eta}\} = f\left\{\mathbf{x}_{i_0,j_0}, \mathbf{x}_{u_{i_1},v_{j_1}}, \ldots, \mathbf{x}_{u_{i_\eta},v_{j_\eta}}\right\}. \tag{5.1}$$

The connection cost $\Lambda^{W,\eta}$ over a digital path can be seen as a measure of dissimilarity between color image pixels at points $q_0, q_1, \ldots, q_\eta$ forming a specific path linking q_0 and q_η, [76,251,390]. If a path joining two distinct points u, v, such that $\mathbf{x}_u = \mathbf{x}_v$ consists of 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* (DTOCS), [251,390] introduced for the gray scale images. For two given points q_k and q_{k-1}, $k = 1, 2, \ldots, \eta$, which are in a neighborhood relation, let the generalized distance between the two points be called connection cost defined on a hybrid spatial-color space discussed in [148,366]

$$\Lambda^{W,1}\{q_{k-1}, q_k\} = \|\mathbf{x}_{q_k} - \mathbf{x}_{q_{k-1}}\| + \tilde{\xi} \cdot \rho^W(u_k, u_{k-1}), \tag{5.2}$$

where $\tilde{\xi}$ establishes a proper weighting in the hybrid spatial-color space. The connection cost of a whole digital path $q_0, q_1, \ldots, q_\eta$ will be then

$$\Lambda^{W,\eta}\{q_0, q_1, \ldots, q_\eta\} = \sum_{k=1}^{\eta} \left(\|\mathbf{x}_{q_k} - \mathbf{x}_{q_{k-1}}\| + \xi \cdot \rho^W\{q_k, q_{k-1}\}\right). \tag{5.3}$$

As we will work with small filtering window, we will focus on the color space only, by setting $\tilde{\xi} = 0$. Similarly to the gray scale case, we will call the minimal connection cost $\Gamma^{W,\eta}(u,v)$ of a path of length η linking two points $u, v \in W$, the η-geodesic between the point u and v: $\Gamma^{W,\eta}(u,v) = \min\{\Lambda(\gamma), \gamma \in \Psi^{W,\eta}\}$.

In this way the η-geodesic is defined as the path of length η, 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 u and $v \in W$, then we obtain the generalized multichannel geodesic distance between these points: $\Gamma^W(u,v) = \min_\eta\{\Gamma^{W,\eta}(u,v)\} =$

$\min\{\Lambda(\pi), \pi \in \mathbb{Q}^{W,\eta}(u,v), \eta \in \mathbb{N}\}$. $\Gamma^W(u,v)$ defines the multidimensional distance transform, which is a generalization of the DTOCS introduced in [390] for the gray scale images, [313].

In general, two distinct pixel's locations on the image lattice can be connected by many paths, (see Fig. 5.2). Moreover the number of possible geodesic paths of certain length η connecting two distinct points depends on their locations, length of the path and the neighborhood system used, [50, 51, 172].

5.2 General Filter Framework

In this work, general fuzzy filtering structure proposed in [236, 240, 242, 313, 377, 379] will be used. The general form of the fuzzy adaptive filters presented here is defined as a weighted average of input vectors inside the processing window W

$$\mathbf{y} = \sum_{k=1}^{N} \psi_k^* \mathbf{x}_k = \sum_{k=1}^{N} \psi_k \mathbf{x}_k \Big/ \sum_{k=1}^{N} \psi_k. \tag{5.4}$$

The relationship between the pixel under consideration $\mathbf{x}_1$ and each pixel in the window should be reflected in the decision on 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. The choice of a specific form of the similarity function yields different filters of specific properties, which can be applied for a wide range of low level vision tasks.

Let us now define a similarity function ψ, analogous to a membership function used in fuzzy systems, between two pixels connected by all possible digital paths leading from u to v

$$\psi^{W,\eta}(u,v) = \sum_{k=1}^{\omega} f\left\{\Lambda_k^{W,\eta}(u,v)\right\}, \tag{5.5}$$

where ω is the number of all paths connecting u and v, $\Lambda_k^{W,\eta}(u,v)$ is a dissimilarity value along a specific path k from the set of all ω possible paths leading from u to v and $f(\cdot)$ is a smooth function of $\Lambda_k^{W,\eta}$. By definition $\psi^{W,\eta}(u,v)$ returns a value evaluated over all routes linking the starting point u with the endpoint v.

The smooth function $f : (0\ \infty) \to \mathbb{R}$ should satisfy the following conditions: f is a decreasing in $(0;\infty)$, f is convex in $(0;\infty)$, $f(0) = 1$, $f(\chi) \to 0$, when $\chi \to \infty$. Several functions satisfying the above conditions have been proposed in the literature, [170, 245, 246, 304, 313, 324, 378]. However, the shape of the function is not of great importance and for the impulsive noise removal good results are obtained using the exponential form of the function $f(\cdot)$, [41]. Therefore we assume

$$\psi^{W,\eta}(u,v) = \sum_{k=1}^{\omega} \exp\left[-\beta \cdot \Lambda_k^{W,\eta}(u,v)\right], \tag{5.6}$$

where β is the filter design parameter. For $\eta = 1$ and a square (3×3) window W the similarity function ψ is defined according to (5.3) as $\psi^{W,1}(u,v) = \exp\{-\beta\|\mathbf{x}_u - \mathbf{x}_v\|\}$, and then if $\mathbf{x}_u = \mathbf{x}_v$, $\Lambda^{W,1}(u,v) = 0$, $\psi^{W,1}(u,v) = 1$, and for $\|\mathbf{x}_u - \mathbf{x}_v\| \to \infty$, $\psi^{W,1} \to 0$, [240, 313, 328, 404].

A normalized form of the similarity function is defined as

$$\psi^*(u,v) = \frac{\psi^{W,\eta}(u,v)}{\sum\limits_{w \Leftrightarrow u} \psi^{W,\eta}(u,w)}, \tag{5.7}$$

where $v \Leftrightarrow u$ denotes all points v connected by digital paths with u which are contained in W.

Assuming that the pixel $\mathbf{x}_u$ is the pixel under consideration, with $\mathbf{x}_v$ representing the pixel included in the supporting element W, which is connected to $\mathbf{x}_u$ via a digital path, the proposed filter output $\mathbf{y}_u$ is given as

$$\mathbf{y}_u = \sum_{v \Leftrightarrow u} \psi^*(u,v) \cdot \mathbf{x}_v, \quad \psi^*(u,v) = \frac{\psi^{W,\eta}(u,v)}{\sum\limits_{w \Leftrightarrow u} \psi^{W,\eta}(u,w)}. \tag{5.8}$$

The filter output is the weighted average of all points $\mathbf{x}_v$ connected by digital paths with the pixel $\mathbf{x}_u$. As the pixel $\mathbf{x}_v$ is the ending point of a path leading from u, therefore this filter structure is called DPA-*Last* (DPAL) as v is the last point on the path, (Fig. 5.3b).

5.2.1 Digital Paths Approach Filter Class

Another possible filtering scheme takes into account the similarity between the starting point q_0 and point q_1 crossed by a digital path connecting pixel q_0 and its neighbor q_1 with all points q_η, which can be reached in η steps from q_0. The aim of taking into account the points $q_2, \ldots, q_\eta$ when calculating the filter output is to explore not only the direct neighborhood of q_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 q_0, passing q_1 and then visiting successive points, till the path reaches length η. In this case, the similarity function takes the form

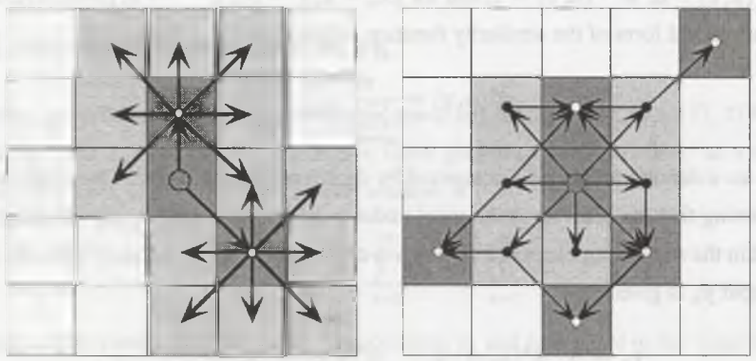
$$\psi^{W,\eta}(u,v) = \psi^{W,\eta}(q_0,q_1) = \sum_{\{q_2^*,q_3^*,\ldots,q_\eta^*\}} f\left(\Lambda^{W,\eta}\{q_0,q_1,q_2^*,q_3^*,\ldots,q_\eta^*\}\right), \tag{5.9}$$

where $\{q_0, q_1, q_2^*, \ldots, q_\eta^*\}$ denotes all paths originating at $u = q_0$ crossing $v = q_1$ end ending at q_η^*, which are totally included in W and $f(\cdot)$ is a smooth function of $\Lambda^{W,\eta}$.

By analogy to the previous Section, the exponential function will be used, and then the similarity function takes the form

$$\psi^{W,\eta}(u,v) = \psi^{W,\eta}(q_0, q_1) = \sum_{\{q_2^*, q_3^*, \ldots, q_\eta^*\}} \exp\left[-\beta \cdot \Lambda^{W,\eta}\left\{q_0, q_1, q_2^*, \ldots, q_\eta^*\right\}\right], \tag{5.10}$$

where β is the smoothing parameter.



a) b)

*Fig. 5.3. In the DPAF and DPAL filters, the weights are assigned to the pixels surrounding the central pixel $\mathbf{x}_1$ and are determined in different ways. In the DPAF approach (**a**), the weights in (5.13) 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, (5.14). In the DPAL approach, the weights are obtained by exploring all digital paths leading from the central pixel to the pixels contained in W (**b**) and then a weighted average of all pixels from W is calculated, (5.18)*

A normalized form of the similarity function can be defined as

$$\psi^*(x,y) = \psi^*(q_0, q_1) = \frac{\sum_{\{q_2^*, q_3^*, \ldots, q_\eta^*\}} \exp\left[-\beta \cdot \Lambda^{W,\eta}\left\{q_0, q_1, q_2^* \ldots, q_\eta^*\right\}\right]}{\sum_{\{q_1^*, q_2^*, \ldots, q_\eta^*\}} \exp\left[-\beta \cdot \Lambda^{W,\eta}\left\{q_0, q_1^*, q_2^* \ldots, q_\eta^*\right\}\right]}, \tag{5.11}$$

where $\{q_0, q_1, q_2^*, \ldots, q_\eta^*\}$ denotes a path joining $u = q_0$ and q_η, crossing $v = q_1$, whereas $\{q_0, q_1^*, q_2^*, \ldots, q_\eta^*\}$ do not necessarily cross $v = q_1$ when joining q_0 and q_η.

Assuming that the pixel $\mathbf{x}_u$ at the position $u = q_0$ is the pixel under consideration, with $\mathbf{x}_v$ representing the pixel at $v = q_1$, the filter output $\mathbf{y}_u$ is given as

$$\mathbf{y}_u = \mathbf{y}_{q_0} = \sum_{v \Leftrightarrow u} \psi^*(u,v) \cdot \mathbf{x}_v = \sum_{v \leftrightarrow u} \psi^*(u,v) \cdot \mathbf{x}_v = \sum_{q_1^* \leftrightarrow q_0} \psi^*(q_0, q_1^*) \cdot \mathbf{x}_{q_1^*}, \tag{5.12}$$

and combining this with (5.11) gives

$$\mathbf{y}_u = \mathbf{y}_{q_0} = \sum_{q_1^* \leftrightarrow q_0} \frac{\sum_{\{q_2^*, q_3^*, \ldots, q_\eta^*\}} \exp\left[-\beta \cdot \Lambda^{W,\eta}\left\{q_0, q_1^*, q_2^*, \ldots, q_\eta^*\right\}\right]}{\sum_{\{q_1^*, q_2^*, \ldots, q_\eta^*\}} \exp\left[-\beta \cdot \Lambda^{W,\eta}\left\{q_0, q_1^*, q_2^* \ldots, q_\eta^*\right\}\right]} \cdot \mathbf{x}_{q_1^*} = \sum_{q_1^* \leftrightarrow q_0} \psi^*(q_0, q_1^*) \cdot \mathbf{x}_{q_1^*}. \tag{5.13}$$

Using the notation from Chapter 4, we can formulate (5.13) as

$$\mathbf{y} = \sum_{k=2}^{N} \psi_k^* \mathbf{x}_k, \tag{5.14}$$

where ψ_k^*, the *normalized weighting coefficients*, play the role of the generalized conductivity coefficients from Section 4.1 and $\mathbf{x}_k$ are the neighbors of $\mathbf{x}_1$, which is the central pixel in the filter mask W.

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

$$\mathbf{y} = (1-\lambda^*)\mathbf{x}_1 + \lambda^* \sum_{k=2}^{N} \psi_k^* \mathbf{x}_k, \quad \text{or} \quad \mathbf{x}_1^{t+1} = (1-\lambda^*)\mathbf{x}_1^t + \lambda^* \sum_{k=2}^{N} \psi_k^* \mathbf{x}_k^t. \tag{5.15}$$

Using the relation $\lambda^* = \lambda \sum_{k=1}^{N} c_k$, (4.27) and taking paths consisting of one step only, the classical form of the anisotropic diffusion scheme defined by (4.23) can be obtained.

Figures 5.4 and 5.5 show the dependence of PSNR on the λ^* and β values for the color *LENA* image contaminated by Gaussian, impulsive and mixed noise for the classical multichannel PM anisotropic diffusion scheme (Section 4.3) and the proposed DPAF, (DPA-*First*) filter defined by (5.14). Especially interesting is the behavior of the plots as a function of λ^*. As can be seen, for images contaminated by a noise process of high intensity, the maximum of PSNR is obtained for λ^* very close to 1, which means that it is favorable to omit the central pixel while calculating the weighted average in (5.14). This was already noticed in [290,300], (2.13), where the central pixel was not taken into the averaging process, which is equivalent to setting $\lambda^* = 1$. That is why we set $\lambda^* = 1$ in (5.14) to define the new DPAF filter, (5.13), (5.14). The superiority of this approach over the classical scheme is clearly seen in Fig. 5.4 and 5.5, where especially for highly corrupted images, the difference in terms of PSNR is quite significant, (see also Tab. 5.4 and 5.5).

In a similar way the DPAL filter can be defined as

$$\mathbf{y}_u = \mathbf{x}_{q_0} = \frac{\sum_{\{q_1^*, q_2^*, q_3^*, \ldots, q_\eta^*\}} \exp\left[-\beta \cdot \Lambda^{W,\eta}\left\{q_0, q_1^*, q_2^*, \ldots, q_\eta^*\right\}\right] \cdot \mathbf{x}_{q_\eta^*}}{\sum_{\{q_1^*, q_2^*, \ldots, q_\eta^*\}} \exp\left[-\beta \cdot \Lambda^{W,\eta}\left\{q_0, q_1^*, q_2^* \ldots, q_\eta^*\right\}\right]} = \sum_{q_\eta^*} \psi^*(q_0, q_\eta^*) \cdot \mathbf{x}_{q_\eta^*}, \tag{5.16}$$

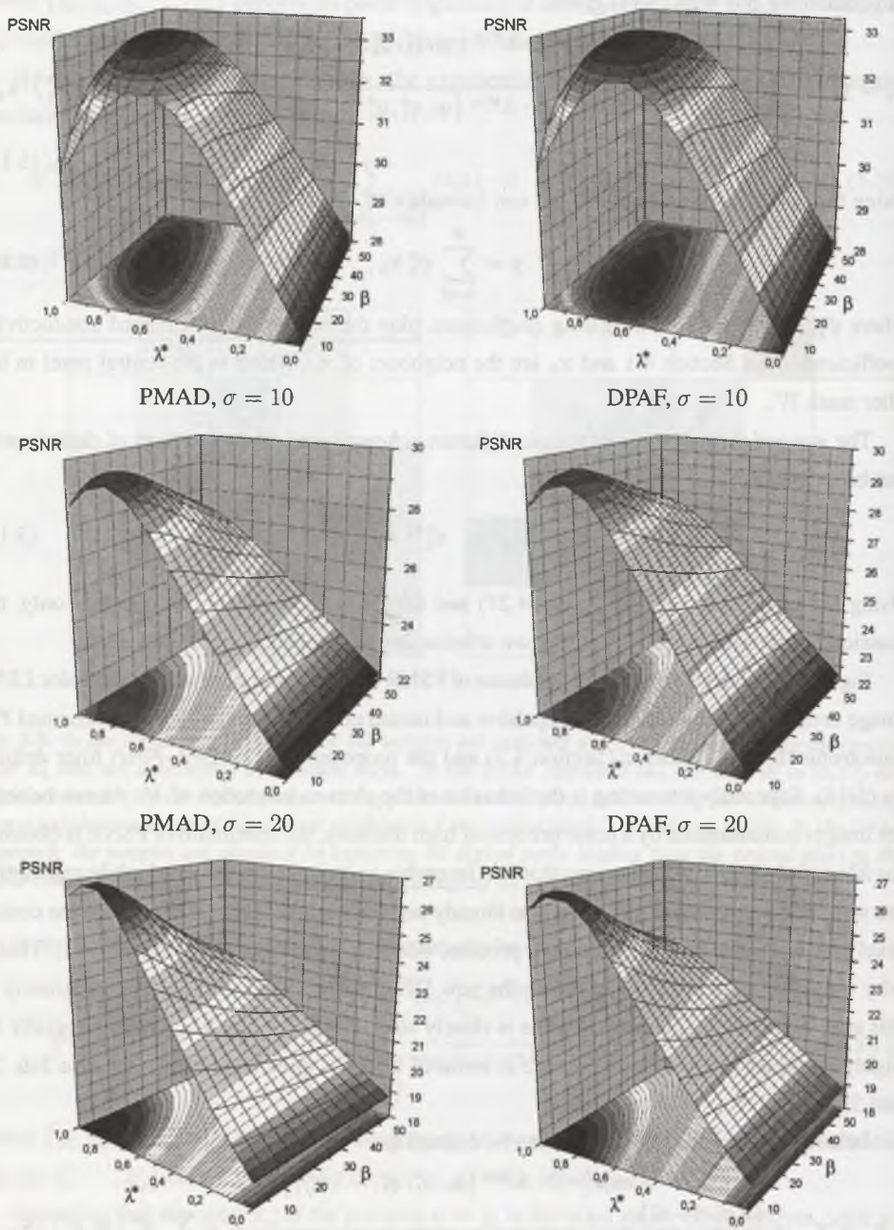


Fig. 5.4. Dependence of the efficiency of the PMAD (left) and DPAF (right) on the λ^* and β parameters for the color image LENA contaminated with Gaussian noise of $\sigma = 10$, $\sigma = 20$ and $\sigma = 30$

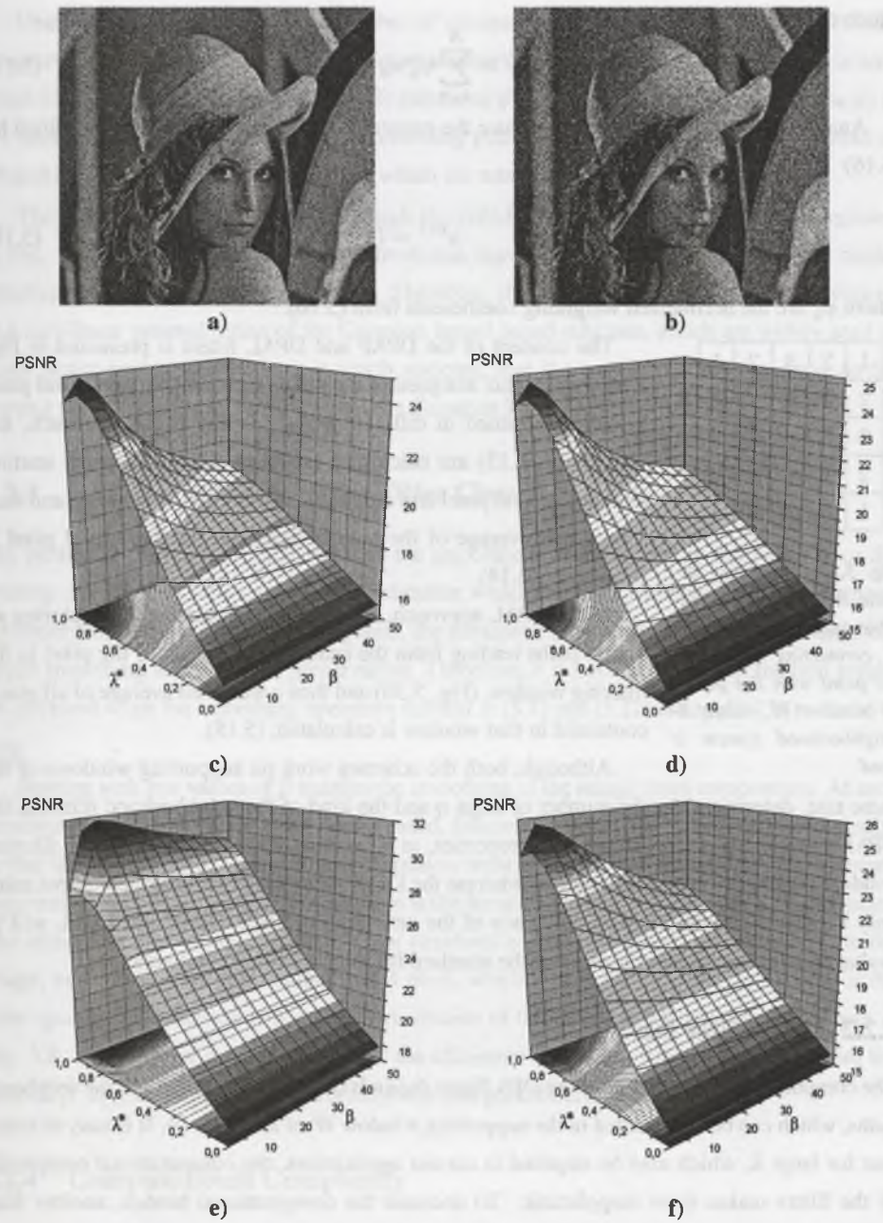


*Fig. 5.5. Dependence of the efficiency of the PMAD filter and the DPAF on the λ^* parameter for the color image LENA contaminated with: **a)** impulsive noise, ($p = 0.12$, NM2), **b)** mixed noise, ($\sigma = 30$, $p = 0.12$, NM2). Below the results obtained with the PMAD filter: **c), d)** and with the DPAF: **e), f)**, ($\eta = 2$). As expected the maximum of PSNR is achieved for λ^* close to 1*

which can be written as

$$\mathbf{y} = \sum_{k=2}^{N} \psi_k^* \mathbf{x}_k \,. \tag{5.17}$$

Analogously to (5.15), we can introduce the general and iterative form of DPAL defined by (5.16)

$$\mathbf{y} = (1-\lambda^*)\mathbf{x}_1 + \lambda^* \sum_{k=2}^{N} \psi_k^* \mathbf{x}_k \,, \qquad \mathbf{x}^{t+1} = (1-\lambda^*)\mathbf{x}_1^t + \lambda^* \sum_{k=2}^{N} \psi_k^* \mathbf{x}_k^t \,, \tag{5.18}$$

where ψ_k^* are the normalized weighting coefficients from (5.16).

1	2	3	2	1
2	2	4	2	2
3	4	•	4	3
2	2	4	2	2
1	2	3	2	1

Fig. 5.6. The array depicts the number of possible paths of length $\eta = 2$ connecting the center point with the points of window W, when 8-neighborhood system is used

The concept of the DPAF and DPAL filters is presented in Fig. 5.3. The weights assigned to the pixels surrounding the central pixel $\mathbf{x}_1$ are determined in different ways. In the DPAF approach, the weights in (5.13) are calculated exploring all digital paths starting from the central pixel and crossing its neighbors, (Fig. 5.3a) and then a weighted average of the nearest neighbors of the central pixel is calculated, (5.14).

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. 5.3b) and then a weighted average of all pixels contained in that window is calculated, (5.18).

Although, both the schemes work on supporting windows of the same size, determined by the number of steps η and the kind of the neighborhood relation, the DPAL has more powerful smoothing properties, as it involves all the pixels from the filtering window W 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 will be evaluated and compared with some of the standard filtering techniques in Section 5.3.

5.2.2 Fast Filter Design

The computational complexity of the DPA filters depends on the path length η and the number of paths, which can be constructed in the supporting window W of size $(k \times k)$. It is easy to notice 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 window W is set to (3×3) independently of the digital path length η. 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 the parameter β used in (5.6), is very small ($\beta \to 0$). Then the weights in (5.8) reduce to $\psi^*(u,v) = \omega(u,v)/\Omega$, where $\omega(u,v)$ is the number of digital paths of length η connecting points u and v, and Ω denotes the number of all possible digital paths starting from u, which are totally included in W.

The convolution mask obtained through the DPAL framework, when $\beta \to 0$ is depicted in Fig. 5.6. The examination of the convolution masks reveals their similarity to the masks obtained through Gaussian kernels, [23]. Therefore, the DPAL and also DPAF can be viewed as a non-linear *generalization* of the Gaussian kernel based schemes, which are widely used in many image processing tasks. It is worth noticing, that if we allow the path to return to the starting point, the approximated form of the Gaussian kernel can be obtained.

5.2.3 Iterative Behavior of the Filter Class

The parameter β in (5.6), (5.10) regulates the smoothness of the similarity function. Since the filtering structure of (5.4) is a regression estimator, which enables a smooth interpolation among the observed, noise-corrupted image pixels, the parameter β provides the required balance between smoothing and the detail preservation. Therefore, it is not surprising that the best results are obtained when the smoothing operators defined in (5.8) and (5.12) are applied in an iterative way.

Starting with low values of β enables the smoothing of the image noise components. At each iteration step, the parameter β can be increased, following a procedure, mathematically similar to that used in simulated annealing optimization algorithm. In particular, β can be increased exponentially $\beta(n) = \beta(n-1)\cdot\alpha$, where n is the iteration number and α is a design parameter. The increasing of β causes that after a few iterations no further changes are introduced to the image, as for high β the filter output is that pixel, which lies on the geodesic digital path in the color space. The influence of α on the performance of the DPAL and FDPA filters is shown in Fig. 5.8. The value of α is not critical for the efficiency of the new filter class, and α from the interval $[1,2]$ guarantees *fast filter convergence* and good filtering results.[1]

5.2.4 Computational Complexity

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

[1]Note that the increasing of the β parameter is equivalent to the 'cooling' procedure applied in Section 4.4 for the construction of the FB diffusion scheme.

power and processing time. A general framework to evaluate the computational requirements of image filtering algorithms based on a fixed processing window is given in [38,243].

The requirement of this approach is that the filter window W is symmetric $(k \times k)$ and contains k^2 vector samples of dimension m. In most image processing applications a value of $k = 3$ is considered.

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

Tab. 5.1. Number of possible simple digital paths Ω in dependence on path length η

The computational complexity of a specific filter is given in terms of the total execution time needed for a complete filtering cycle. The total execution time T is calculated as $T = \sum w_\varrho \cdot \varrho$, where ϱ is the number of particular operations required for a complete cycle, and w_ϱ is the relative operation weight.

In the analysis of the filters the following operations are used: AD (additions), MU (multiplications), DI (divisions), SQ (square roots), CO (comparisons), AR (arc cosines) and EX (exponents). Mostly w_{AD} is assumed to be 1, while other w_ϱ 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 a fixed window, the methodology presented in [37,38,243] cannot be directly applied to evaluate the new filters' complexity. The complexity of the proposed filters depends mostly on the number of possible digital paths, which in turn depends on the path's type and its length. For a given path of length η, the number of simple paths Ω can be easily evaluated. Table 5.1 depicts the number of possible paths corresponding to the DPA and FDPA filters, [295,313,378,379].

FILTER	AD	MU	DI	SQ	EX	CO	AR	TOTAL
DPA_2	947	228	56	112	56	—	—	1399
DPA_3	8827	1478	368	1104	368	—	—	12145
$FDPA_2$	403	100	24	48	24	—	—	599
$FDPA_3$	1139	230	56	168	56	—	—	1649
$FDPA_2^*$	169	22	24	9	24	—	—	**248**
$FDPA_3^*$	721	24	56	9	56	—	—	866
$VMF_{3\times3}$	186	63	—	21	—	8	—	**278**
$VMF_{5\times5}$	855	330	—	110	—	24	—	1319
$BVDF_{3\times3}$	375	210	21	21	—	8	21	656
$BVDF_{5\times5}$	1970	1100	110	110	—	24	110	3424
$DDF_{3\times3}$	540	282	21	42	—	8	21	914
$DDF_{5\times5}$	2785	1455	110	220	—	24	110	4704

Tab. 5.2. Number of elementary operations needed for a complete processing cycle

The complexity of the DPA and FDPA filters can be determined as follows:

1. *Filtering of 1 pixel requires the computation of all weights $\psi^{W,\eta}$ (see point **2**), $m\cdot(\Omega-1)$ additions and $m \cdot \Omega$ multiplications.*

2. *Computation of all weights $\psi^{W,\eta}$ requires the computation of all similarity functions $\psi^{W,\eta}$ (see point **3**), Ω divisions and $(\Omega - 1)$ additions.*

3. *Computation of all similarity functions $\psi^{W,\eta}$ requires Ω computations of the distance $\Lambda_j^{W,\eta}$ (see point **4**), $(\Omega - 1)$ additions, Ω multiplications and Ω computations of an exponent.*

4. *Computation of one distance $\Lambda_j^{W,\eta}$ along path j requires η computations of the Euclidean distance, (if the L_2 metric is used) and $(\eta - 1)$ additions.*

5. *Computation of one particular Euclidean distance requires m multiplications, $2m$ additions and 1 square root.*

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

$$T = (2\eta m\Omega + \Omega p + m\Omega - m - 2) \cdot AD + (\Omega + m\Omega + 2\eta) \cdot MU + \Omega \cdot DI + \Omega\eta \cdot SQ + \Omega \cdot EX\,. \quad (5.19)$$

It should be emphasized at this point that the computational complexity analysis of the new filter is based on a 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.

The analysis of the FDPA filtering algorithm reveals that the L_2 distance should be evaluated η times for each path of length η. If the total number of paths in the supporting window is Ω, 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 (3×3) 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, [38,176] can be applied in the DPA or FDPA filter design.

Table 5.2 summarizes the total number of operation for different filters, with DPA_η denoting the basic DPA filter of length η, $FDPA_\eta$ denoting straightforward application of FDPA algorithms and $FDPA_\eta^*$ signifying 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 the filters based on angular distances.

5.3 Efficiency of the 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. 5.3 using artificial and natural color images corrupted by Gaussian and mixed Gaussian and impulsive noise.

The use of nonlinear filters in color image processing is motivated primarily by their good performance 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.

To quantitatively evaluate the behavior of the proposed algorithms, two 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 (60×60) containing a square of size (30×30) was generated, (Fig. 5.7a). 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 (90×90) contains a top-cut pyramid, which is used to simulate a *ramp-edge*, (Fig. 5.7c). The test image *SQUARE* was corrupted by multivariate impulsive noise following the model NM2 given by (1.10) with the degree of contamination $p = 0.1$, (Fig. 5.7b). The test image *PYRAMID* was corrupted by mixed impulsive noise with $p = 0.1$ and $\sigma = 20$, (Fig. 5.7d).

The standard *Digital Paths Approach* (DPAF, DPAL) and the *Fast Digital Paths Approach* (FDPA) algorithms were compared in terms of objective quality criteria with the VMF, AMF and PMAD and other filtering techniques listed in Tab. 5.3.

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 (3×3). Anisotropic diffusion filter used in the experiments denoted as PMAD is a vector implementation of the PM anisotropic diffusion, (Section 4.1) which utilizes the conductivity function c_1 defined by (4.6), [112, 228]. For the PMAD 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 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 fixed experimental parameter values were used.

In case of images corrupted with Gaussian noise, the AMF as expected gave better results than the VMF, but it blurred heavily the image edges. The classical PM anisotropic diffusion

gives good results for images corrupted with Gaussian noise of low intensity, but it requires many iterations till its performance can be comparable with the new filter class in terms of objective quality criteria. In case of images distorted by strong Gaussian noise, the PMAD approach is not able to suppress the spikes, which leads to a poor overall performance of this filter, (see Figs. 5.9, 5.10).

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 blotches, [80]. The 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 β in (4.6) we can smooth the noise out, but then the PMAD 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, (see Figs. 5.9, 5.10 and 5.15f), 5.18d). The results obtained using the DPAF, DPAL and FDPA filters confirm their good properties in case of images corrupted by both impulsive and Gaussian noise.

The new filtering structure gives **satisfying results** both in flat regions and also at image edges, (see Figs. 5.9, 5.10 and also 5.15). 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 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. 5.12).

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, leaving the edges of the objects almost unchanged.

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. 5.3. The test images were contaminated by additive Gaussian noise of $\sigma = 30$ and also by mixed noise ($p = 0.12$, NM2, $\sigma = 30$). As the results for *LENA* and *PEPPERS* are consistent, only the results obtained with *LENA* image are reported.

The SNR, PSNR, NMSE and NCD image quality measures defined in Chapter 1 were used for the comparisons. The results obtained using the new filtering techniques are compared with the filtering algorithms from Tab. 5.3 in Tabs. 5.4, 5.5. For the denoising of both contaminated

LENA images with the new filtering techniques, predefined parameter values were used: path length $\eta = 2$, $\beta = 10$, $\alpha = 1.25$. For all evaluated filters 10 iterations were performed and the best result in terms of PSNR are presented in Tabs. 5.4, 5.5.

Figure 5.8 depicts the efficiency of the proposed algorithms, (DPAL and FDPA) in terms of the NCD quality measure, as a function of the design parameters α and β. 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 α ensures quick convergence of the proposed filters to a stable state and as can be seen in Fig. 5.8, good results can be obtained for any α in the range $[1, 2]$.

Tables 5.4 and 5.5 indicate that the new filters yield especially good results in the case of images corrupted by the Gaussian and mixed Gaussian and impulsive noise. In addition to the 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 useful for various computer vision applications, (see Figs. 5.13, 5.14, 5.15, 5.18, 5.19).

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. 5.12).

The comparison of the new filters efficiency with some of the standard filters is presented in Fig. 5.11, 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 GDF technique, presented in [398, 399], which is based on the gradient norm described in Section 4.3. The visual comparison between the FDPA and the GDF introduced in [398, 399] is shown in Fig. 5.17.

The high efficiency of the proposed filter class is also confirmed by Figs. 5.18, 5.19, Fig. 5.16 (removal of raster structure), Fig. 5.13 (restoration of artworks, [313, 337]) and Fig. 5.14 (microarray image denoising, [190, 303, 348, 378], Fig. 1.7).

In conclusion, from the results listed in the Tables and shown in the Figures, it can be observed 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 noise removal algorithms, [295,306,313–315,333,348,377,377,404].

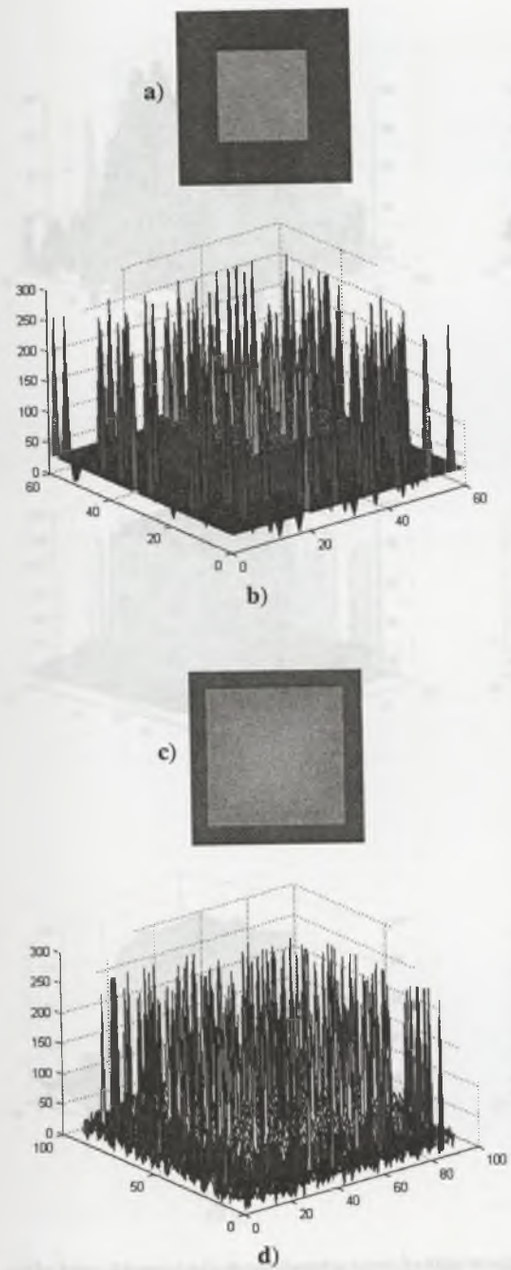


Fig. 5.7. Test image SQUARE (a), corrupted by impulsive noise, (green channel) (b), test image PYRAMID (c), corrupted by mixed noise, (green channel) (d)

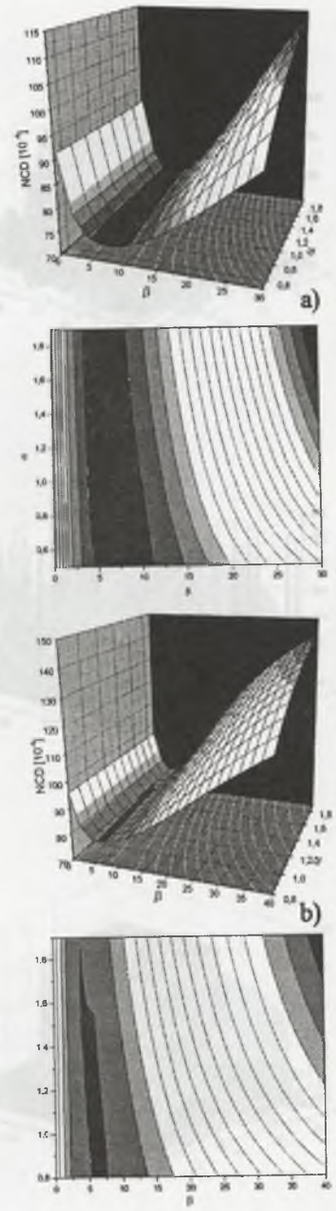


Fig. 5.8. Efficiency of the a) DPAL and b) FDPA filters in terms of NCD and their dependence on α and β, ($\eta = 2$) for color LENA corrupted by mixed noise ($p = 0.12$, NM2, $\sigma = 30$), ($n = 3$)

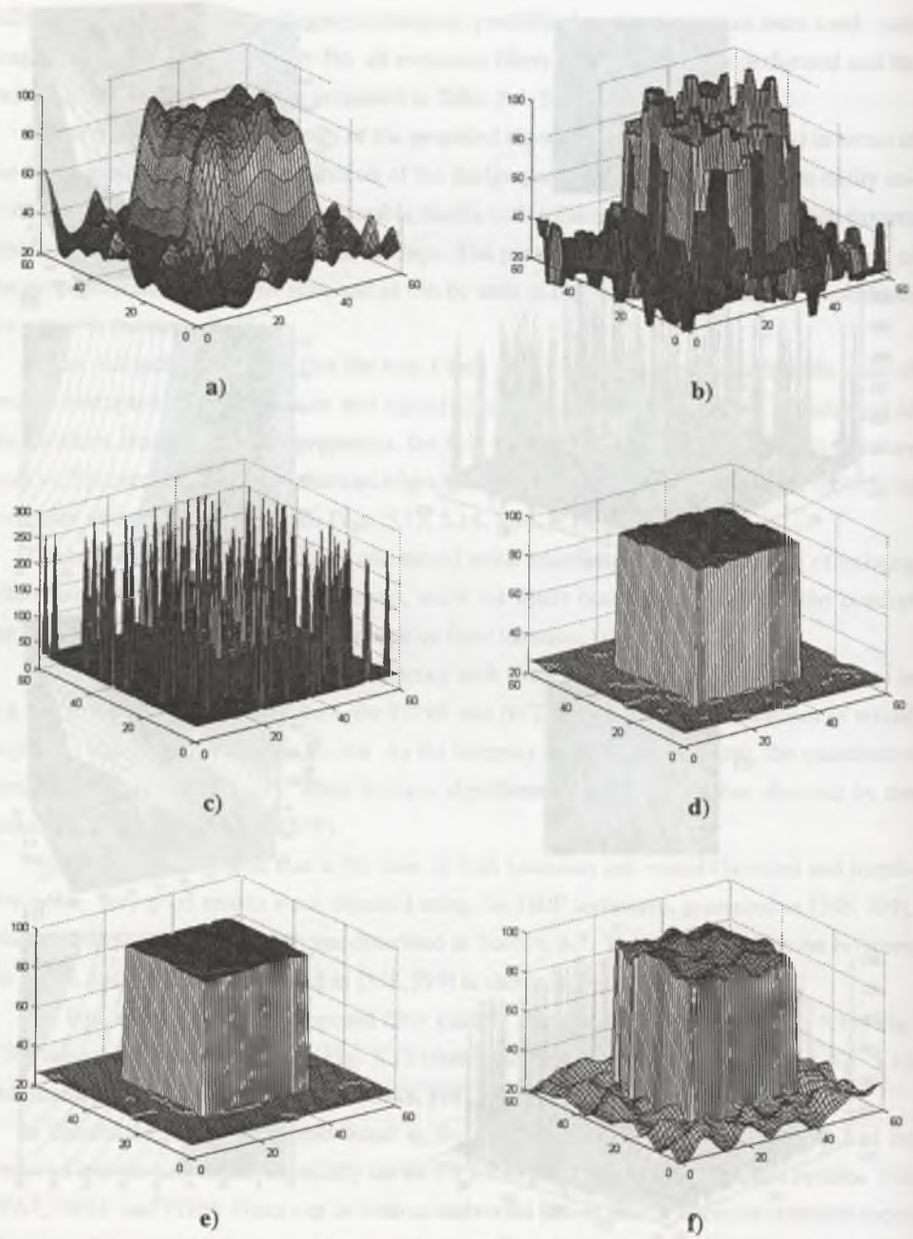


*Fig. 5.9. Three-dimensional representation of the results of noise attenuation in the green channel of the SQUARE test image corrupted by impulsive noise, using the standard and novel techniques: **a**) AMF, **b**) VMF, **c**) PM-AD, **d**) FDPA, **e**) DPAL and **e**) DPAF, (five iterations, $\eta = 2$)*

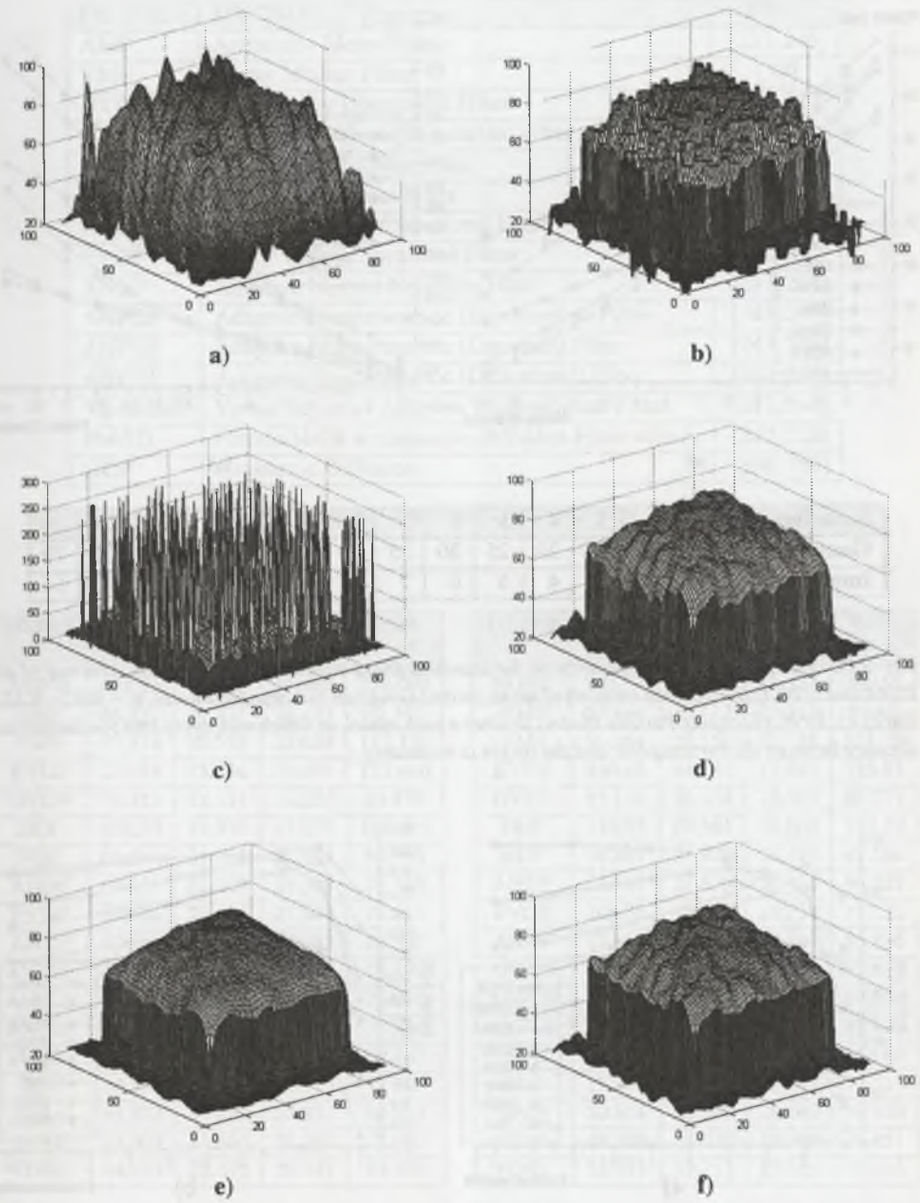
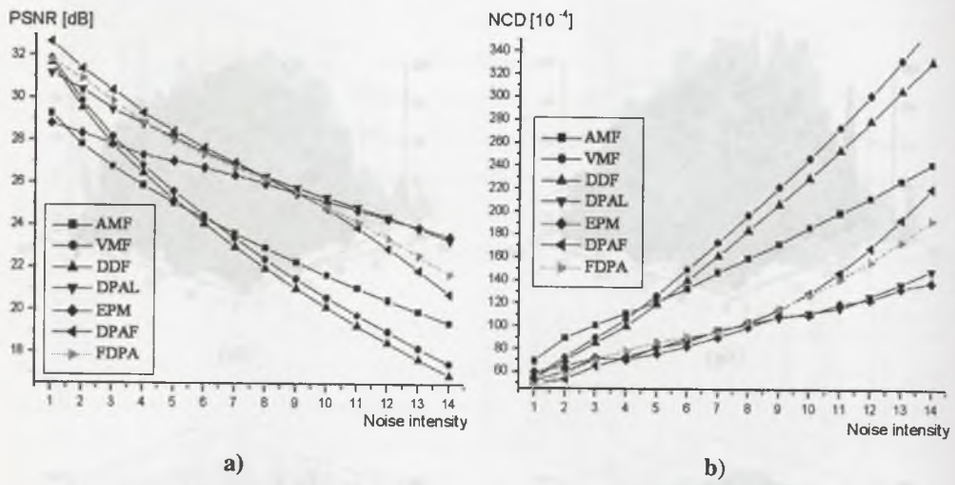


*Fig. 5.10. Three-dimensional representation of the results of noise attenuation in the the green channel of the PYRAMID test image corrupted by mixed Gaussian and impulsive noise using the standard and new techniques: **a**) AMF, **b**) VMF, **c**) PM-AD, **d**) FDPA, **e**) DPAL and **f**) DPAF, (five iterations, $\eta = 2$)*



Noise intensity	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gaussian σ	5	10	15	20	25	30	35	40	45	50	55	60	65	70
Impulsive [%]	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Fig. 5.11. Comparison of the efficiency of the standard filters with the new filter class in terms of a) PSNR and b) NCD for different amounts of noise, (mixed Gaussian and impulsive noise, $p = 0.01 - 0.12$, NM2), c). EPM, (Escaping Particle Model) denotes a path model, in which with every step the Euclidean distance between the current point and the origin is increasing

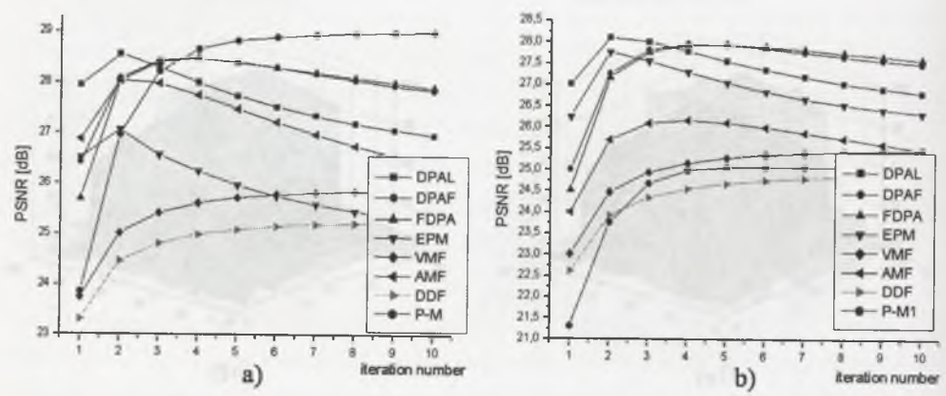


Fig. 5.12. Plots of the PSNR in subsequent iterations for various filters applied to color LENA image contaminated with Gaussian noise of $\sigma = 30$ (a) and mixed noise, ($\sigma = 30$, $p = 0.12$, NM2) (b)

FILTER	METHOD	REF.
AMF	Arithmetic Mean Filter	[231, 246]
VMF	Vector Median Filter	[19]
BVDF	Basic Vector Directional Filter	[395, 397]
GVDF	Generalized Vector Directional Filter	[397]
DDF	Directional-Distance Filter	[138]
HDF	Hybrid Directional Filter	[106]
AHDF	Adaptive Hybrid Directional Filter	[106]
FVDF	Fuzzy Vector Directional Filter	[240]
ANNF	Adaptive Nearest Neighbor Filter	[237, 238]
ANPEF	Adaptive Nonparametric (Exponential) Filter	[242, 246]
ANPGF	Adaptive Nonparametric (Gaussian) Filter	[242, 246]
ANPDF	Adaptive Nonparametric (Directional) Filter	[242, 246]
VBAMMF	Vector Bayesian Adaptive Median/Mean Filter	[242, 246]
PMAD	Perona-Malik Anisotropic Diffusion Filter with c_1	[227, 228]
GDF	Geometric Diffusion	[398, 399]

Tab. 5.3. Filters taken for comparison with the proposed noise reduction techniques

FILTER	NMSE $[10^{-3}]$	SNR [dB]	PSNR [dB]	NCD $[10^{-4}]$
NONE	420.55	13.762	18.860	250.090
AMF	66.452	21.775	26.873	95.347
VMF	87.314	20.589	25.688	117.170
BVDF	279.54	15.536	20.634	117.400
GVDF	76.713	21.151	26.250	84.876
DDF	100.50	19.979	25.077	108.960
HDF	66.584	21.766	26.865	92.769
AHDF	60.166	22.206	27.305	91.369
FVDF	57.466	22.406	27.504	77.111
ANNF	63.341	21.983	27.082	82.587
ANPEF	60.396	22.190	27.288	76.896
ANPGF	60.443	22.187	27.285	76.890
ANPDF	58.389	22.337	27.435	78.486
PMAD	41.434	23.826	28.925	69.482
GDF	34.530	24.618	29.753	72.100
DPAF	42.873	23.678	28.813	82.814
DPAL	43.005	23.665	28.800	77.932
FDPA	44.913	23.476	28.611	84.918

Tab. 5.4. Comparison of the efficiency of the new algorithms with various techniques from Tab. 5.3, using the LENA standard color image corrupted by Gaussian noise of $\sigma = 30$, [295]

FILTER	NMSE $[10^{-3}]$	SNR [dB]	PSNR [dB]	NCD $[10^{-4}]$
NONE	905.93	10.429	15.528	305.55
AMF	97.444	20.112	25.211	95.80
VMF	96.464	20.156	25.255	121.79
BVDF	336.46	14.731	19.829	123.93
GVDF	91.118	20.404	25.503	89.277
DDF	110.62	19.561	24.660	113.39
HDF	74.487	21.279	26.378	97.596
AHDF	68.563	21.639	26.738	96.327
FVDF	108.76	19.635	24.734	111.22
ANNF	75.652	21.212	26.310	86.836
ANPEF	90.509	20.433	25.532	97.621
ANPGF	90.523	20.432	25.531	97.603
ANPDF	74.203	21.296	26.394	85.026
PMAD	339.55	14.691	19.790	113.65
GDF	59.371	22.264	27.363	77.510
DPAF	50.804	22.941	28.040	76.076
DPAL	49.999	23.010	28.109	72.851
FDPA	53.573	22.711	27.809	78.666

Tab. 5.5. Comparison of the new algorithms with the techniques from Tab. 5.3 using the LENA color image corrupted by mixed Gaussian and impulsive noise ($\sigma = 30$, $p = 0.12$, NM2), [295]

*Fig. 5.13. Illustrative example of the application of the DPAF filter for the noise removal in artworks, [313, 337]: **a**) color image and below its zoomed part, **b**) the result of the DPAF filtering*

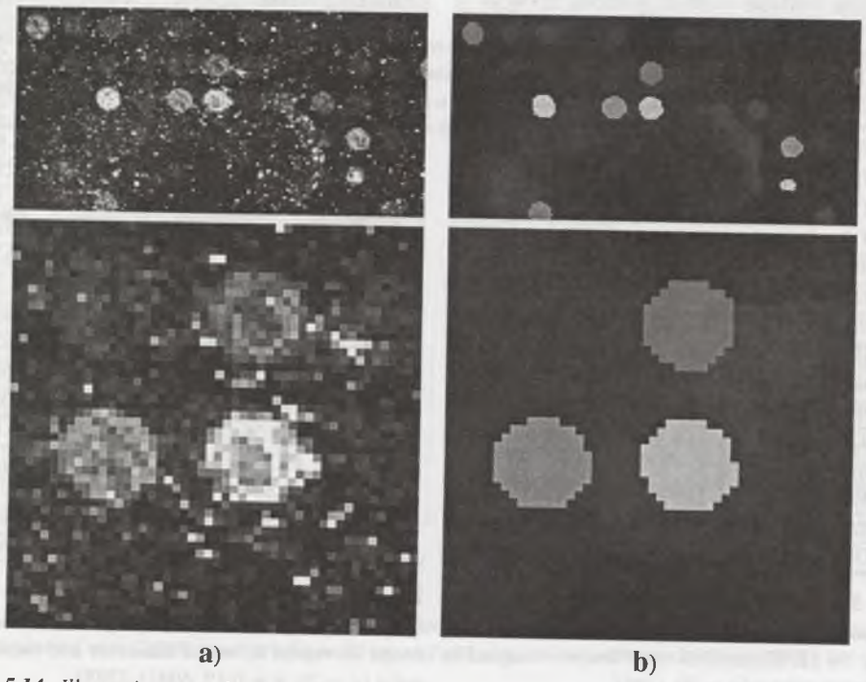


*Fig. 5.14. Illustrative example of the efficiency of DPAL filter for noise removal in cDNA microarrays, (see Fig. 1.7 d): **a**) red channel image and below its zoomed part, **b**) the result of the DPAL filtering*

*Fig. 5.15. Color test images LENA **a**) and PEPPERS **b**) with depicted regions of interest **c**). The chosen image regions were contaminated by mixed impulsive ($p = 0.12$, NM2) and Gaussian noise of $\sigma = 30$ (NM5), **d**) and then restored with the DPAF method (**e**) and with the VMF (**f**)*

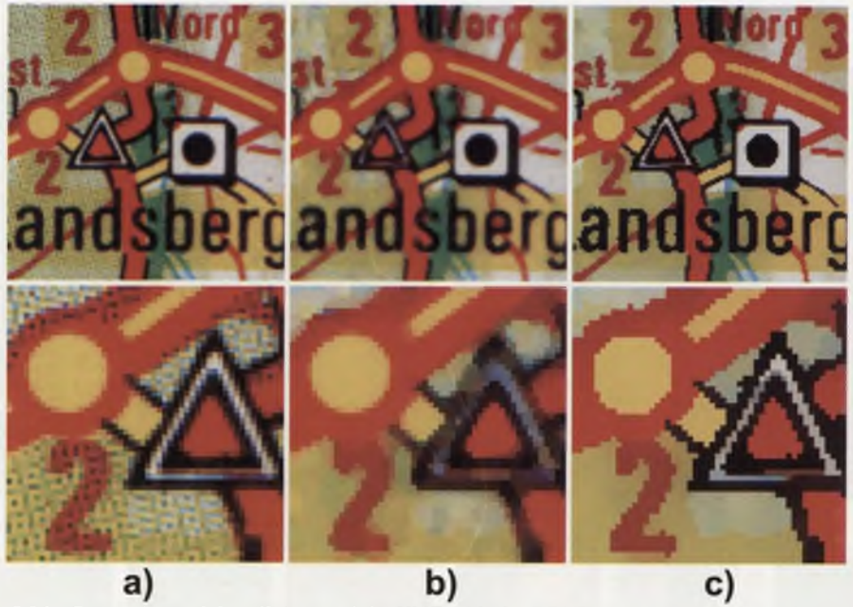


Fig. 5.16. Comparison of the efficiency of the VMF with the DPAF: ***a****) test image (part of a scanned map),* ***b****) VMF, (3 × 3 mask),* ***c****) DPAF, ($\beta = 20, \alpha = 1.25, \eta = 2$, 3 iterations)*

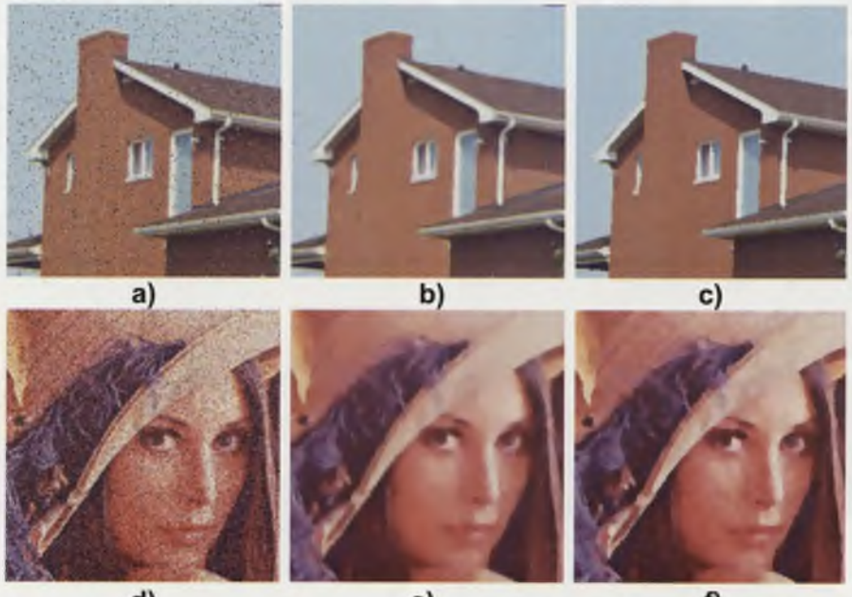


Fig. 5.17. Comparison of the GDF, [398, 399] with the DPAF: ***a****) test image HOUSE contaminated with impulsive noise ($p = 0.1$, NM2),* ***b****) GDF, [398, 399],* ***c****) DPAF,* ***d****) test image LENA contaminated with mixed impulsive ($p = 0.1$, NM2) and Gaussian noise of $\sigma = 30$,* ***e****) GDF,* ***f****) DPAF, [313]*

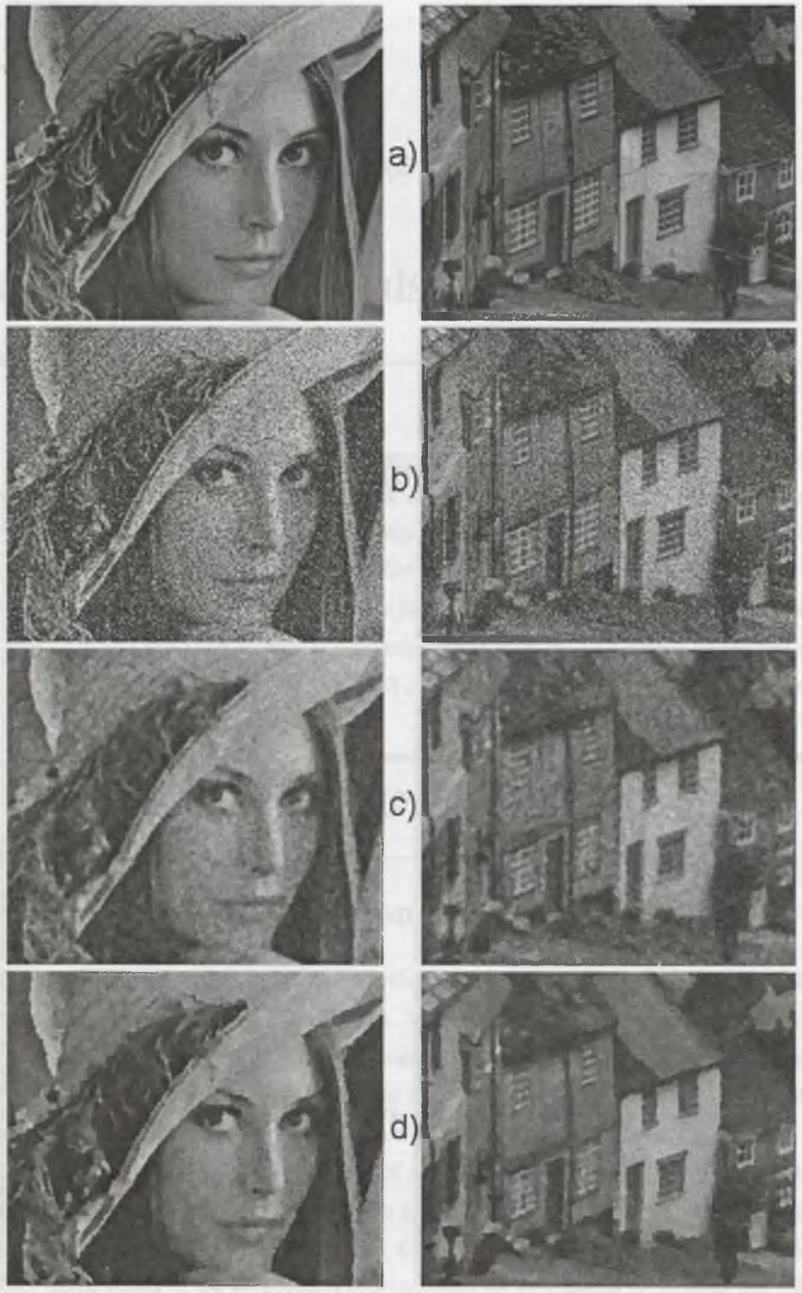


Fig. 5.18. Comparison of FDPA with VMF: ***a****) test images,* ***b****) images corrupted with mixed noise, ($\sigma = 30$, $p = 0.05$, NM1),* ***c****) VMF,* ***d****) FDPA, ($\beta = 10, \alpha = 1.25, \eta = 2, n = 5$), [306]*

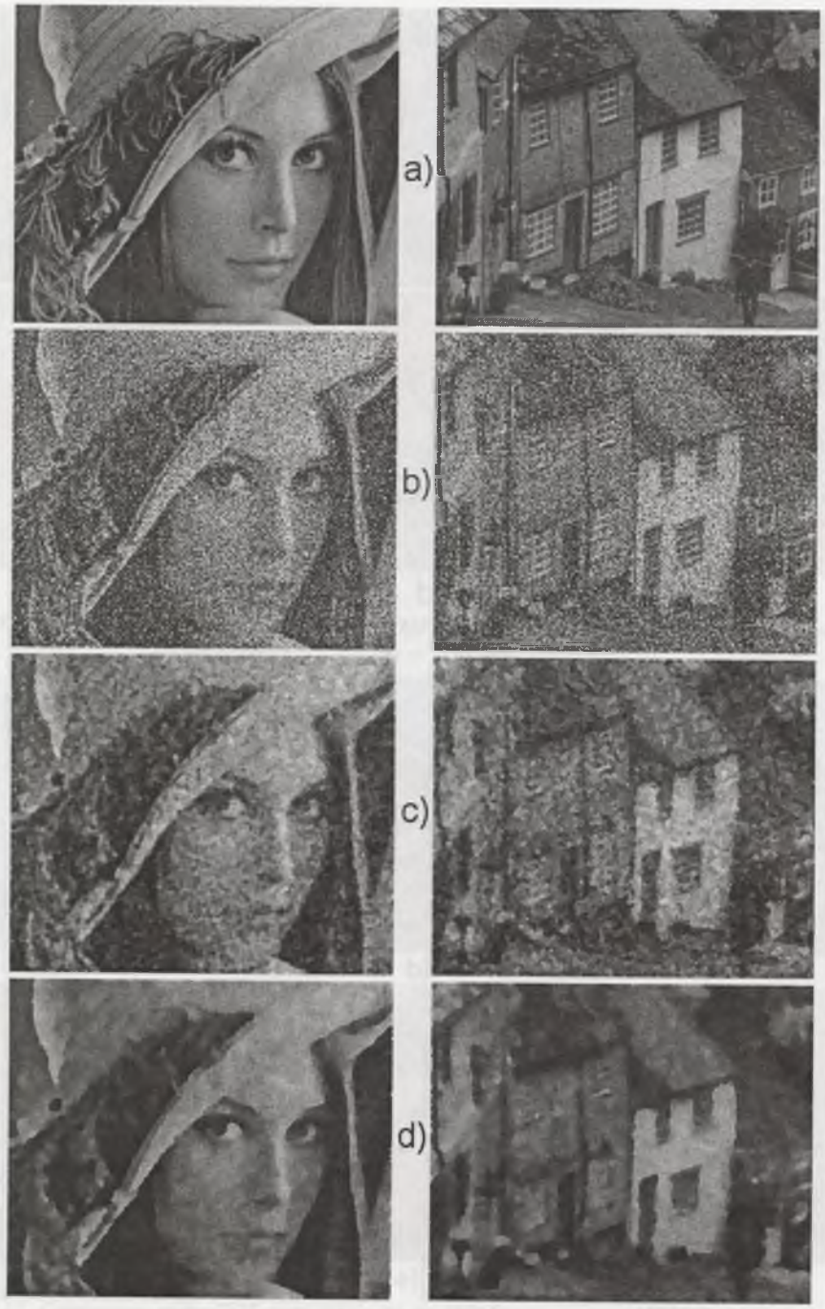


*Fig. 5.19. Comparison of FDPA with VMF: **a**) test images, **b**) images corrupted with mixed noise, ($\sigma = 60$, $p = 0.15$, NM1), **c**) VMF, **d**) FDPA, ($\beta = 10, \alpha = 1.25, \eta = 2, n = 5$), [306]*

6

Nonparametric Impulsive Noise Removal

In this Chapter the problem of nonparametric impulsive noise removal in multichannel images is addressed. A new class of filters, developed by the author of this monograph, based on the nonparametric probability density estimation of the sample data is presented and its relationship to the commonly used filtering techniques is investigated.

*The computational complexity of the new filter class is shown to be significantly lower than that of the Vector Median Filter. Extensive simulation experiments indicate that the presented filters **outperform** the VMF, as well as other techniques currently used to eliminate impulsive noise in color images.*

6.1 Nonparametric Estimation

APPLYING statistical pattern recognition techniques requires the estimation of the probability density function of the data samples. When designing a pattern recognition system, nonparametric classification is often used, because nonparametric techniques do not assume a particular form of the density function, since the underlying density of real data rarely fits common statistical models.

Density estimation describes the process of modelling the probability density function of a given sequence of sample values drawn from an unknown density distribution. The simplest form of the density estimation is the *histogram*: the sample space is first divided into a grid, then the density in the center of the grid cells is approximated by the number of samples that fall into one bin. The main disadvantage of the histogram is its strong dependence on the chosen width of the bins, the origin of the grid and in higher dimensions the sparse histogram occupancy.

Nonparametric density estimation avoids this disadvantage by placing a kernel function on every sample value in the sample space and then summing the values at each sample point. This results in a smooth density estimates that are not affected by an arbitrarily chosen partition of the sample space, [56, 91, 92, 107, 225, 265, 280, 286, 381, 389].

The nonparametric approach to estimating densities can be introduced by assuming that the color space occupied by the multichannel image pixels is divided into m-dimensional hypercubes. If h_N is the length of an edge of a hypercube, then its volume is given by $V_N = h_N^m$. If we are interested in estimating the number of pixels falling into the hypercube of volume V_N, then we can define the function

$$\phi(\mathbf{x}_k) = \begin{cases} 1, & \text{if } |x_{kj}| \leq \frac{1}{2}, \quad j = 1, \ldots, m, \\ 0, & \text{otherwise}, \end{cases} \tag{6.1}$$

which defines a unit hypercube centered in the origin, [92].

The function $\phi\left(\|\mathbf{x} - \mathbf{x}_k\| / h_N\right)$ is equal to unity if the pixel $\mathbf{x}_k$ falls within the hypercube V_N centered in $\mathbf{x}$ and is zero otherwise. The number of pixels in the hypercube with the length of edges equal to h_N is then

$$\Gamma_N = \sum_{k=1}^{N} \phi\left(\frac{\|\mathbf{x} - \mathbf{x}_k\|}{h_N}\right), \tag{6.2}$$

and the estimate of probability that a sample $\mathbf{x}$ is within the hypercube is $p_N = \Gamma_N/(NV_N)$, which gives

$$p_N(\mathbf{x}) = \frac{1}{NV_N} \sum_{k=1}^{N} \phi\left(\frac{\|\mathbf{x} - \mathbf{x}_k\|}{h_N}\right). \tag{6.3}$$

This estimate can be generalized by using a smooth kernel function $\mathcal{K}(\cdot)$ in place of $\phi(\cdot)$ in (6.1) and the width parameter h_N which satisfy

$$\mathcal{K}(\mathbf{x}) = \mathcal{K}(-\mathbf{x}),\ \mathcal{K}(\mathbf{x}) \geq 0,\ \int \mathcal{K}(\mathbf{x})\, d\mathbf{x} = 1, \quad \text{and} \quad \lim_{N\to\infty} h_N = 0,\ \lim_{N\to\infty} h_N^m = \infty. \tag{6.4}$$

The multivariate kernel density estimator in the m-dimensional case, can be defined as

$$p_N(\mathbf{x}) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{h_1 \cdots h_m} \mathcal{K}\left(\frac{|x_1 - x_{k1}|}{h_1}, \frac{|x_2 - x_{k2}|}{h_2}, \ldots, \frac{|x_m - x_{km}|}{h_m}\right), \tag{6.5}$$

with $\mathcal{K}$ denoting a multidimensional kernel function $\mathcal{K}\colon R^m \to R$, $h_1, \ldots, h_m$ signifying the bandwidths for each dimension and N being the number of samples in the filtering window W.

A common approach to build multidimensional kernel functions is to use a *product kernel*

$$\mathcal{K}(\chi_1, \ldots, \chi_m) = \prod_{j=1}^{m} \mathcal{K}(\chi_j), \quad \text{then} \quad p_N(\mathbf{x}) = \frac{1}{N} \sum_{k=1}^{N} \prod_{j=1}^{m} \left(\frac{|x_j - x_{kj}|}{h_j}\right). \tag{6.6}$$

The shape of the approximated probability density function depends h e a v i l y on the bandwidth chosen for the density estimation. Small values of h lead to spiky density estimates showing spurious features. On the other hand, too large values of h produce over-smoothed estimates that hide structural features of the estimated probability density.

Very often a special case of the generalized Gaussian function is taken to obtain the nonparametric estimate of the density probability, [6, 104, 242]

$$\mathcal{K}(x|\chi, h, \gamma) = \gamma_1^m \exp\left(-\frac{\gamma_2}{2}\left(\frac{|x - \chi|}{h}\right)^{\frac{2}{1+\gamma}}\right), \tag{6.7}$$

$$\gamma_1 = \left(\frac{(\Gamma(1.5(1+\gamma)))^{0.5}}{(1+\gamma)\Gamma(0.5(1+\gamma))^{0.5}}\right) h^{-1},\ \gamma_2 = \left(\frac{\Gamma(1.5(1+\gamma))}{\Gamma(0.5(1+\gamma))}\right)^{\frac{1}{1+\gamma}},\ \Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt, \tag{6.8}$$

if $\gamma = 0$ then we obtain the Gaussian, if $\gamma = 1$ then the double exponential distribution is obtained. For $\gamma \to -1$ the distribution tends to be rectangular and for $\gamma \in (-1, 1)$ intermediate symmetrical distributions are obtained. In the multivariate case, for $\gamma = 0$ we obtain

$$\mathcal{K}(\mathbf{x}) = \frac{1}{(h\sqrt{2\pi})^m} \exp\left(-\frac{\|\mathbf{x} - \chi\|^2}{2h^2}\right), \tag{6.9}$$

and the density estimate of the unknown probability density function at $\mathbf{x}$ is determined as a sum of kernel functions placed at each sample $\mathbf{x}_k$ belonging to the window W

$$p_N(\mathbf{x}, h) = \frac{1}{N\left(h\sqrt{2\pi}\right)^m} \sum_{k=1}^{N} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_k\|^2}{2h^2}\right). \tag{6.10}$$

The smoothing parameter h depends on the local density estimate of the sample data and its form is of great importance for the nonparametric estimator, [26, 134, 140, 351, 363]. It can be made adaptive and then

$$p_N(\mathbf{x}) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{h(\mathbf{x}_k)} \mathcal{K}\left(\frac{\|\mathbf{x} - \mathbf{x}_k\|}{h(\mathbf{x}_k)}\right). \tag{6.11}$$

The resulting variable smoothing parameter depends on the local density estimate of the pixels in the filter window, [101–103, 287, 364, 372]. An efficient method to make the estimator adaptive was proposed in [241]

$$h(\mathbf{x}_k) = N^{-\frac{m}{\tau}} \sum_{j=1}^{N} \|\mathbf{x}_j - \mathbf{x}_k\|, \tag{6.12}$$

where τ is a design parameter.

In [56] the results of Parzen were extended to the multivariate data and assuming the Gaussian kernel, the formulas for the h parameter, which gives the optimal estimation with respect

to the mean squared error were provided. For the color image samples ($m = 3$), the optimal h^* is obtained using the formula, [103,287,365]

$$h^* = 0.53\hat{\sigma} N^{-\frac{1}{7}}, \tag{6.13}$$

where $\hat{\sigma}^2$ is the estimation of the Gaussian noise variance.

Choosing the Gaussian kernel function for $\mathcal{K}$ and assuming the lack of correlation between the channels, the *rule of thumb* for the optimal bandwidth is according to [128]

$$h^* = (4/(m+2))^{-\frac{1}{m+4}} \hat{\sigma} N^{-\frac{1}{m+4}}, \tag{6.14}$$

where $\hat{\sigma}$ denotes the approximation of the standard deviation of the samples. In one dimensional case (6.14) reduces to the well known, *rule of thumb* of Scott, [69,280,286,408]

$$h^* = 1.06 N^{-\frac{1}{5}} \hat{\sigma}. \tag{6.15}$$

A version which is more robust against outliers in the sample set can be constructed if the interquartile range is used as a measure of dispersion, instead of the variance, [127,286]

$$h^* = 0.79 \varrho N^{-\frac{1}{5}}, \tag{6.16}$$

where ϱ is the inter-quartile range. Another robust estimate of the optimal bandwidth is

$$h^* = 0.9 \mathcal{R} N^{-\frac{1}{5}}, \quad \mathcal{R} = \min(\hat{\sigma}, \varrho/1.34). \tag{6.17}$$

Generally the simplified rule of choosing the optimal bandwidth h can be written as

$$h_1^* = C\, \hat{\sigma}\, N^{-\frac{1}{m+4}}, \tag{6.18}$$

where C is an appropriate weighting coefficient.

From the maximum likelihood principle and assuming the independence of the data samples, we can write the likelihood of drawing the complete dataset as the product of the densities

$$\mathcal{L}^*(h) = \prod_{j=1}^{N} p_N(\mathbf{x}_j, h) = \prod_{j=1}^{N} \frac{1}{N} \sum_{k=1}^{N} \frac{1}{(h\sqrt{2\pi})^m} \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|^2}{2h^2}\right). \tag{6.19}$$

As this likelihood function attains a global maximum at $h = 0$, in [93] a modified approach has been proposed

$$\mathcal{L}^*(h) = \left[\prod_{j=1}^{N} \frac{1}{N} \sum_{\substack{k=1 \\ k \neq j}}^{N} \frac{1}{(h\sqrt{2\pi})^m} \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|^2}{2h^2}\right)\right]^{\frac{1}{m}}. \tag{6.20}$$

This function has one maximum for h, which can be found by setting the derivative of the logarithm of $\mathcal{L}^*(h)$ to zero

$$\frac{\partial \log \mathcal{L}^*(h)}{\partial h} = \frac{1}{N} \sum_{j=1}^{N} \frac{\sum_{i \neq j}^{N} \frac{\|\mathbf{x}_j - \mathbf{x}_k\|^2}{h^3} \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|^2}{2h^2}\right)}{\sum_{k \neq j}^{N} \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|^2}{2h^2}\right)} - \frac{m}{h} = 0. \tag{6.21}$$

A crude but fast way to obtain an approximate solution to (6.21) is assuming that the density estimate of (6.6) on a certain location $\mathbf{x}$ in the feature space is determined by the nearest sample only, [158,365]. In this way

$$\frac{\partial \log \mathcal{L}^*(h)}{\partial h} = \frac{1}{N} \sum_{j=1}^{N} \frac{\|\tilde{\mathbf{x}}_j - \mathbf{x}_k\|^2}{h^3} = \frac{m}{N}, \tag{6.22}$$

which leads to

$$h^* = \sqrt{\frac{1}{mN} \sum_{j=1}^{N} \|\tilde{\mathbf{x}}_j - \mathbf{x}_j\|^2}, \tag{6.23}$$

where $\tilde{\mathbf{x}}_j$ represents the nearest neighbor of the sample $\mathbf{x}_j$. In this work, a more general version

$$h_2^* = C \sqrt{\frac{1}{mN} \sum_{j=1}^{N} \|\tilde{\mathbf{x}}_j - \mathbf{x}_j\|^2}, \tag{6.24}$$

with C being a tuning coefficient, will be used.

6.2 Similarity Based Filter Class

Let us assume a filtering window W containing N monochrome image pixels $\{x_1, \ldots, x_N\}$ and let us define the similarity function $\psi : [0, \infty) \to \mathbb{R}$ which is non-ascending and convex in $[0; \infty)$ and satisfies $\psi(0) = 1$, $\psi(\infty) = 0$. The similarity between two pixels of the same intensity should be 1 and the similarity between pixels with minimal and maximal gray scale values should be close to 0. A monotonically decreasing function $\psi(x_k, x_j)$ of the form $\psi(x_k, x_j) = \psi(|x_k - x_j|)$ can easily satisfy the three required conditions.

Let us additionally define the cumulated sum Ψ of similarities between a given pixel and all other pixels belonging to the filtering window W. For the central pixel we introduce Ψ_1 and for the neighbors of x_1 we define Ψ_k as

$$\Psi_1 = \sum_{j=2}^{N} \psi(x_1, x_j), \quad \Psi_k = \sum_{j=2, j \neq k}^{N} \psi(x_k, x_j), \quad k = 2, \ldots, N, \tag{6.25}$$

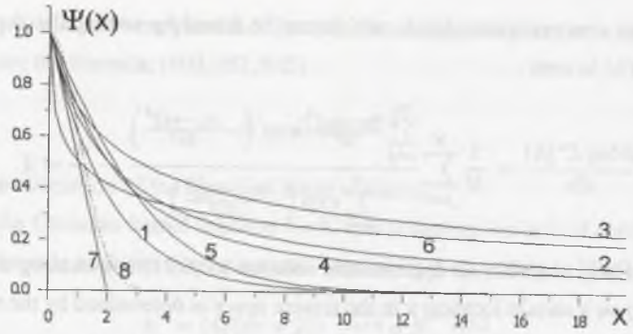
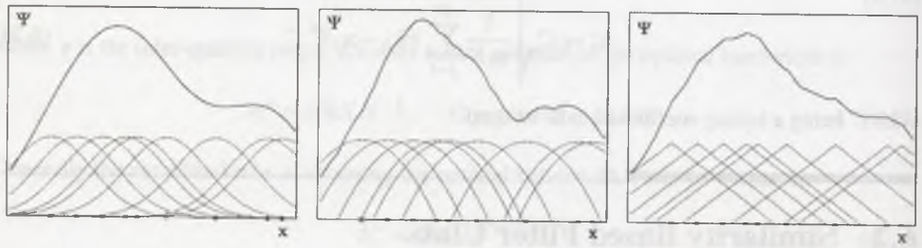


Fig. 6.1. Similarity functions used as kernels of the nonparametric estimation, (6.26 - 6.28)

which means that for x_k we do not take into account the similarity between x_k and x_1, which is the main idea of the proposed algorithm.

The omission of the similarity value $\psi(x_k, x_1)$ when calculating Ψ_k, privileges the central pixel x_1, as Ψ_1 contains $(N-1)$ similarities $\psi(x_1, x_k)$ and Ψ_k, for $k > 1$ has only $(N-2)$ similarity values, as the central pixel x_1 is excluded from the calculation of the sum Ψ_k[1], [294, 304, 305, 309, 356].



a) b) c)

Fig. 6.2. Dependence of the cumulative similarity values Ψ on the pixels' gray scale value for a window containing a set of samples with intensities $\{15, 24, 33, 41, 45, 55, 72, 90, 95\}$ *using the Gaussian (a), Epanechnikov (b) and the Triangle kernel (c)*

In the construction of the new filter, the reference pixel x_1 in the window W is replaced by one of its neighbors if $\Psi_1 < \Psi_k$, $k = 2, \ldots, N$. If this is the case, then x_1 is replaced by that x_{k^*} for which $k^* = \arg\max \Psi_k$. In other words, x_1 is detected as being corrupted if $\Psi_1 < \Psi_k$, $k = 2, \ldots, N$ and is replaced by its neighbors x_k, which maximizes the sum of similarities Ψ between all the pixels of W excluding the central pixel, [312].

The basic assumption is that a new pixel must be taken from the samples belonging to W, (introducing new pixels, which do not occur in the filtering window is prohibited, like in the

[1]We assume that $\psi(x_j, x_k) = 0$ for $j = k$.

VMF). For this purpose, the function ψ must be convex, which means that in order to find a maximum of the sum of similarity functions Ψ, it is sufficient to determine the values of Ψ in points $x_1, x_2, \ldots, x_N$ only, [293, 309, 354].

The presented approach, can be applied in a straightforward way to the multichannel images, [53, 101, 309, 372]. Now we can use the similarity function defined as $\psi\{\mathbf{x}_k, \mathbf{x}_j\} = \psi(\|\mathbf{x}_k - \mathbf{x}_j\|)$, where $\|\cdot\|$ denotes the specific vector norm and in exactly the same way we can maximize the total similarity function Ψ for the vectorial case.

Several convex functions were examined, [293, 296, 310, 329, 357] in order to compare the presented approach with the standard filters used in color image processing listed in Tab. 6.1. Good results (Tab. 6.2, Fig. 6.3), were obtained when applying the following similarity functions, which can be treated as kernels of nonparametric density estimation, (Figs. 6.1, 6.23), [280, 286]

$$\psi_1(x) = \exp\left\{-\frac{x}{h},\right\}, \; \psi_2(x) = \frac{1}{1 + x/h}, \; h \in (0; \infty), \tag{6.26}$$

$$\psi_3(x) = \frac{1}{(1+x)^h}, \; \psi_4(x) = 1 - \frac{2}{\pi}\arctan\left(\frac{x}{h}\right), \; \psi_5(x) = \frac{2}{1 + \exp\left\{\frac{x}{h}\right\}}, \; h \in (0; \infty) \tag{6.27}$$

$$\psi_6(x) = \frac{1}{1 + x^h}, \; \boxed{\psi_7(x) = \begin{cases} 1 - \frac{x}{h}, & \text{if} \quad x \le h, \\ 0, & \text{if} \quad x > h, \end{cases}} \; \psi_8(x) = \exp\left\{-\left(\frac{x}{h}\right)^2\right\}, h \in (0; \infty). \tag{6.28}$$

It is interesting to note, that remarkably good results were achieved for the simplest, linear similarity function $\psi_7(x)$, (Figs. 6.1, 6.2c), 6.3, Tab. 6.2), which allows to construct a fast impulsive noise removal algorithms, [305, 310, 330, 334, 336, 358].

In the multichannel case, we have

$$\Psi_1 = \sum_{j=2}^{N} \psi_7(\rho\{\mathbf{x}_1, \mathbf{x}_j\}), \quad \Psi_k = \sum_{j=2, j \neq k}^{N} \psi_7(\rho\{\mathbf{x}_k, \mathbf{x}_j\}), \tag{6.29}$$

where $\rho\{\mathbf{x}_j, \mathbf{x}_k\} = \|\mathbf{x}_j - \mathbf{x}_k\|$ and $\|\cdot\|$ is the L_2 vector norm, as it yields best results, (Tab. 6.3). Applying the linear similarity function ψ_7 we obtain

$$\psi(\mathbf{x}_j, \mathbf{x}_k) = \begin{cases} 1 - \rho(\mathbf{x}_j, \mathbf{x}_k)/h, & \text{for} \quad \rho(\mathbf{x}_j, \mathbf{x}_k) < h, \\ 0, & \text{otherwise.} \end{cases} \tag{6.30}$$

Then we have from (6.29) and (6.30) assuming that $\rho(\mathbf{x}_j, \mathbf{x}_k) < h$, for $j, k = 1, \ldots, N$

$$\Psi_1 = N - 1 - \frac{1}{h}\sum_{j=2}^{N} \rho(\mathbf{x}_1, \mathbf{x}_j), \; \Psi_k = \sum_{\substack{j=2, \\ j \neq k}}^{N} \left(1 - \frac{\rho(\mathbf{x}_k, \mathbf{x}_j)}{h}\right) = N - 2 - \frac{1}{h}\sum_{j=1}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j). \tag{6.31}$$

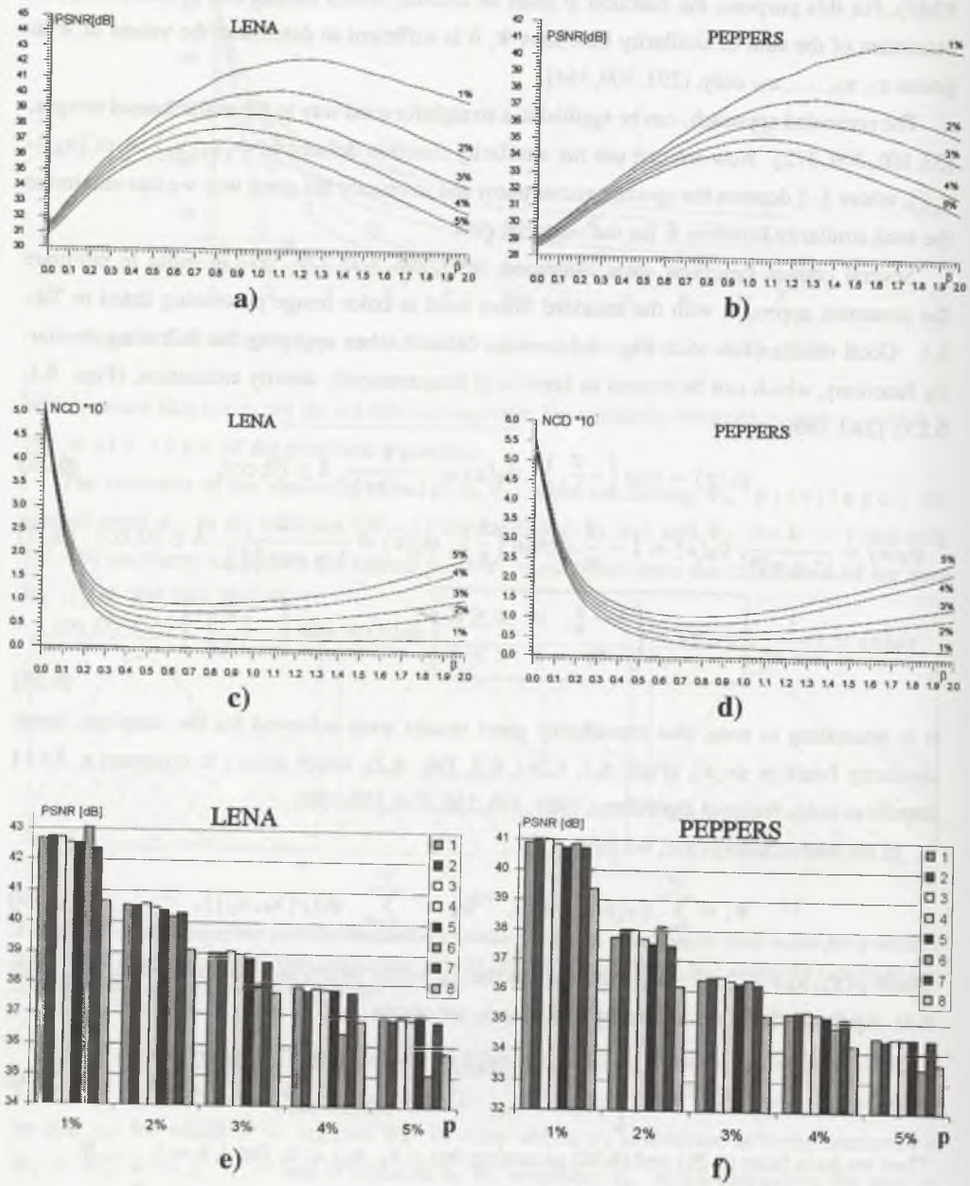


Fig. 6.3. Dependence of the efficiency of the nonparametric filter, based on the linear kernel ψ_7 (6.28) on the h parameter in terms od PSNR and NCD (a - d) for the LENA and PEPPERS color test images corrupted by impulsive noise, (NM4). Below the comparison of the filter efficiency using different kernels, (6.26 - 6.28)

Notation	Filter	Ref.
AMF	Arithmetic Mean Filter	[246]
VMF	Vector Median Filter	[19]
BVDF	Basic Vector Directional Filter	[395]
GVDF	Generalized Vector Directional Filter	[138]
DDF	Directional-Distance Filter	[137]
HDF	Hybrid Directional Filter	[106]
AHDF	Adaptive Hybrid Directional Filter	[106]
FVDF	Fuzzy Vector Directional Filter	[240]
ANNF	Adaptive Nearest Neighbor Filter	[237]

Tab. 6.1. Filters taken for the comparisons with the proposed adaptive noise removal technique

METHOD	NMSE $[10^{-4}]$	RMSE	PSNR
AMF	82.863	12.903	25.917
VMF	23.304	6.842	31.427
BVDF	29,074	7.643	30.466
DDF	24.003	6.944	31,288
HDF	22.845	6.775	31.513
AHDF	22,603	6,739	31.559
FVDF	26.755	7.331	30.827
ANNF	31.271	7.926	30.149

ψ	NMSE $[10^{-4}]$	RMSE	PSNR
$\psi_1(x)$	4.959	3.157	38.145
$\psi_2(x)$	5.398	3.294	37.776
$\psi_3(x)$	9.574	4.387	35.288
$\psi_4(x)$	5.064	3.190	38.054
$\psi_5(x)$	4.777	3.099	38.307
$\psi_6(x)$	11.024	4.707	34.675
$\psi_7(x)$	**4.693**	**3.072**	**38.384**
$\psi_8(x)$	5.056	3.163	38.137

Tab. 6.2. Comparison of the new algorithm, based on different kernel functions with the standard techniques, using LENA color image contaminated by 5% impulsive noise, ($p = 0.05$, NM1)

LENA NORM	NMSE 10^{-4}	RMSE	PSNR [dB]	h
L_1	5.042	3.183	38.074	6.58
L_2	4.659	3.060	38.417	6.35
L_∞	5.304	3.265	37.854	6.50

PEPPERS NORM	NMSE 10^{-4}	RMSE	PSNR [dB]	h
L_1	9.236	3.888	36.337	10.14
L_2	8.426	3.713	36.736	9.37
L_∞	9.960	4.038	36.008	9.24

Tab. 6.3. Best results obtained with the new algorithm with ψ_7 kernel for the LENA and PEPPERS images using different Minkowski norms, (NM1, $p = 0.04$)

In this way the difference between Ψ_1 and Ψ_k, $(k > 1)$ is

$$\Psi_1 - \Psi_k = N - 1 - \frac{1}{h}\sum_{j=2}^{N}\rho(\mathbf{x}_1, \mathbf{x}_j) - \left[N - 2 - \frac{1}{h}\sum_{j=2}^{N}\rho(\mathbf{x}_k, \mathbf{x}_j)\right] = \tag{6.32}$$

$$= 1 - \frac{1}{h}\sum_{j=2}^{N}\left[\rho(\mathbf{x}_1, \mathbf{x}_j) - \rho(\mathbf{x}_k, \mathbf{x}_j)\right], \tag{6.33}$$

$$\boxed{\Psi_1 - \Psi_k > 0 \quad \text{if} \quad h > \sum_{j=2}^{N}\left[\rho(\mathbf{x}_1, \mathbf{x}_j) - \rho(\mathbf{x}_k, \mathbf{x}_j)\right], \quad k = 2, \ldots, N.} \tag{6.34}$$

If this condition is satisfied, then the central pixel $\mathbf{x}_1$ is considered as not disturbed by the noise process, otherwise the pixel $\mathbf{x}_k$ for which the cumulative similarity value Ψ attains

maximum, replaces the central noisy pixel. In this way, the filter changes the central pixel only when it is detected to be noisy and preserves the original undistorted image structures, [336, 355].

The construction of the new filter is presented in Fig. 6.5 for the gray scale case and in Fig. 6.4 for the two-dimensional data. In the example provided by Fig. 6.5, (see also Fig. 6.2), the supporting window W contains 9 pixels of intensities $\{15, 24, 33, 41, 45, 55, 72, 90, 95\}$, (their special arrangement in W is not relevant). Each of the graphs from **a)** to **i)** shows the dependence of Ψ_1 and $\Psi_{/1}$ on the gray scale value, ($\Psi_{/1} < \Psi_1$), where $\Psi_{/1}$ denotes the cumulative similarity value with rejected central pixel x_1, on the sample's intensity. Graph **a)** shows the plot of Ψ_1 and $\Psi_{/1}$ for $x_1 = 15$, plot **b)** for $x_1 = 24$ and so on till plot plot **i)**, which shows the graphs of Ψ_1 and $\Psi_{/1}$ for $x_1 = 95$. The central pixel will be replaced in cases: **(a)**, **(b)**, **(f)** - **(i)**, as in those cases there exists a pixel x_k for which $\Psi_1 < \Psi_k$. The continuous plots show that the extremum of the similarity function $\Psi_{/1}$ is always obtained at points $x_k \in W$, which is an important feature of this algorithm. Because the function $\Psi_{/1}$ is convex, the maximum can be found by calculating the similarity values in N points only, which makes the algorithm computationally attractive.

It is easy to observe that the construction of the new filter is similar to the standard VMF, [294, 310, 329, 332, 358]. Instead of the function R_k in (3.3), a modified cumulative distance function can be used

$$\boxed{R_k = \begin{cases} -h + \sum_{j=2}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j), & \text{for} \quad k = 1, \\ \sum_{j=2}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j), & \text{for} \quad k = 2, \ldots, N, \end{cases}} \tag{6.35}$$

and in the same way as in the VMF, the central vector $\mathbf{x}_1$ in W is being replaced by $\mathbf{x}_{k^*}$ such that $k^* = \arg\min R_k$. It is easy to notice that the above construction is equivalent to the condition expressed in (6.34). Now, instead of maximizing the cumulative similarity Ψ_k, the modified cumulative distance R_k is minimized. In this way, the condition for retaining the original image pixel is: $R_1 \leq R_k$, $k = 2, \ldots, N$, which leads to the rule of retaining $\mathbf{x}_1$

$$-h + \sum_{j=2}^{N} \rho(\mathbf{x}_1, \mathbf{x}_j) \leq \sum_{j=2}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j), \quad k = 2, \ldots, N, \tag{6.36}$$

$$\boxed{R_1 \leq R_k \quad \text{if} \quad h \geq \sum_{j=2}^{N} [\rho(\mathbf{x}_1, \mathbf{x}_j) - \rho(\mathbf{x}_k, \mathbf{x}_j)], \quad k = 2, \ldots, N.} \tag{6.37}$$

The main characteristic of the new filter construction is the *rejection* of the central pixel $\mathbf{x}_1$, when calculating R_k, $k > 1$, [294, 330, 334, 335]. This scheme, based on the *leave-one-out* technique, is the most important feature of this algorithm. As the central pixel is suspected to

be noisy, it is not taken into consideration, when calculating the distances associated with the neighbors of $\mathbf{x}_1$. In this way the filter replaces the central pixel only when it is detected to be corrupted, while retaining the original undisturbed image structures. [2]

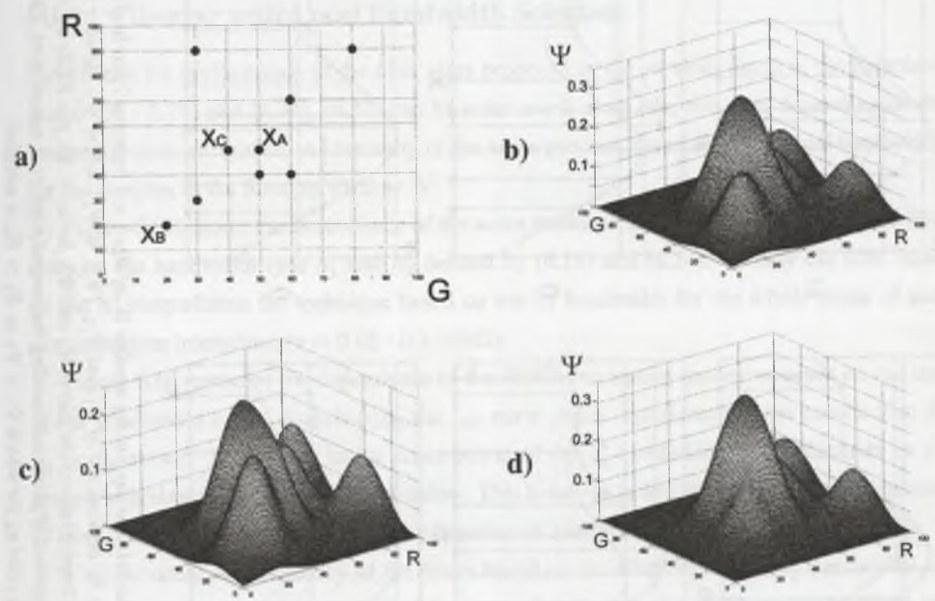


*Fig. 6.4. Impulsive noise removal technique in the 2D case. Fig. **a)** depicts the arrangement of pixels in W and Fig. **b)** their nonparametric probability density estimation. Figs. **c)** and **d)** present the density plots for the cases when the central pixels x_A and x_B are removed from W. It can be seen that in the first case **c)** the pixel $x_1 = x_A$ will be retained and in the second case **d)** the pixel $x_1 = x_B$ will be replaced by x_A. The pixel x_A will be preserved, as in Fig. **c)** the plot attains its maximum at x_C, but this maximum is less than the maximum for x_A in Fig. **b)**. Regarding sample x_B, its rejection causes that the maximum is attained at x_A and this pixel will replace the central pixel x_B*

As it can be easily observed, the parameter h in (6.34) and (6.35) strongly influences the intensity of the filtering process. The fraction of pixels replaced by the new filter is a decreasing function of h. The value of h has to be set by the designer, which can be seen as a drawback of the presented technique, as some knowledge on the image structure and impulsive noise intensity is required.

As already noticed, the VMF has the disadvantage of replacing too many uncorrupted image pixels. This is improved in the new filter design by setting appropriate h values, which forces the filter to preserve uncorrupted pixels, but still enables to remove corrupted ones. The subject of automatic setting of h value is addressed in the next Section.

[2] Note that similar techniques, based on the rejection of the central pixel, were described in Section 2.1.

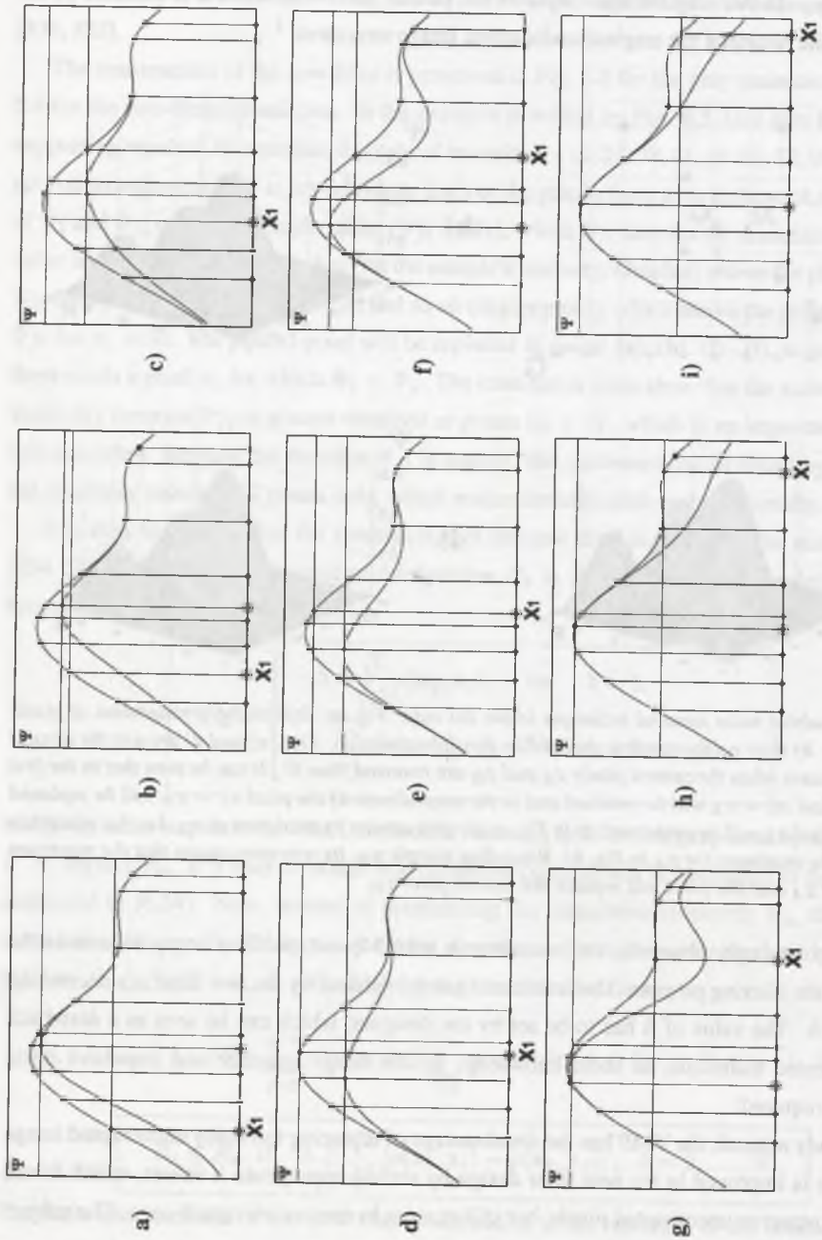


Fig. 6.5. Illustration of the new filter construction using the Gaussian kernel. The supporting window W of size 3×3 contains 9 pixels of intensities (15, 24, 33, 41, 45, 55, 72, 90, 95). Each of the graphs from a) to i) shows the dependence of Ψ_1 and $\Psi_{/1}$, ($\Psi_{/1} < \Psi_1$), where $\Psi_{/1}$ denotes the cumulative similarity value with rejected central pixel x_1 on the gray scale value. Graph a) shows the plot of Ψ_1 and $\Psi_{/1}$ for $x_1 = 15$, plot b) for $x_1 = 24$ and so on till plot plot i) shows the graphs of Ψ_1 and $\Psi_{/1}$ for $x_1 = 95$. The arrangement of pixels surrounding the central pixel x_1 is not relevant. The central pixel will be replaced in cases: (a), (b), (f - i), as in those cases there exists a pixel x_k for which $\Psi_1 < \Psi_k$ or $R_1 > R_k$ is satisfied

6.3 Adaptive Filter Design

6.3.1 Filtering with Local Bandwidth Selection

To enhance the performance of the filter class proposed in the previous Section, the parameter h in (6.26 - 6.28) and (6.34), (6.37) can be determined in an adaptive way, depending on the image structure, properties and intensity of the noise process, by establishing local bandwidths for the samples in the filtering window W.

Figure 6.6a) shows the dependence of the noise attenuation capability of the proposed filter class on the bandwidth type h_1^* and h_2^* defined by (6.18) and (6.24). Clearly the filter based on the h_2^* outperforms the technique based on the h_1 bandwidth for the whole range of used contamination intensities, ($p = 0.01$ - 0.1, NM2).

Figure 6.6b) presents the dependence of the PSNR restoration quality measure on the kind of the Minkowski norm. Surprisingly, the L_∞ norm yields significantly better results than the L_1 or L_2 norms. This is due to the construction of the h_2^* bandwidth, which depends on the nearest neighbor in the sliding filter window. This behavior is advantageous, as the calculation of the L_∞ norm is much faster than the evaluation of distances determined by L_1, L_2 norms.

Unfortunately, the efficiency of the filters based on the adaptive h_1^* and h_2^* bandwidths are dependent, (especially for very small noise contamination) on the coefficient C in (6.18) and (6.24). Figure 6.6c) shows the dependence of PSNR for the filter based on h_2^* as a function of C in (6.24). For low noise intensity, the parameter C should be significantly larger than for the case of images corrupted by heavy noise process. However, setting C to 4 is an acceptable trade-off, as can be seen in Figure 6.6d), which depicts the efficiency of the proposed filter in comparison with VMF, AMF and BVDF. It can be observed, that although the $C = 4$ is not an optimal setting for the whole range of tested noise intensities, nevertheless the described adaptive filter yields much better results than the traditional techniques. This is also testified by Fig. 6.7, which compares the filtering results obtained by the filter based on adaptive h_2^* bandwidth, ($C = 4$) with the performance of the *reference* VMF filter.

Another drawback of the presented filter class is the high computational complexity of the algorithms, caused by the need of adaptive calculation of the bandwidth for the changing set of pixels in the moving filtering window. Although the calculation of the L_∞ is very fast, however the calculation of $(N-1)$ distances for each pixel position is time consuming and can pose problems, especially in real time applications. Therefore a filter structure based on global bandwidth, determined once for the whole image, is presented in the next Section.

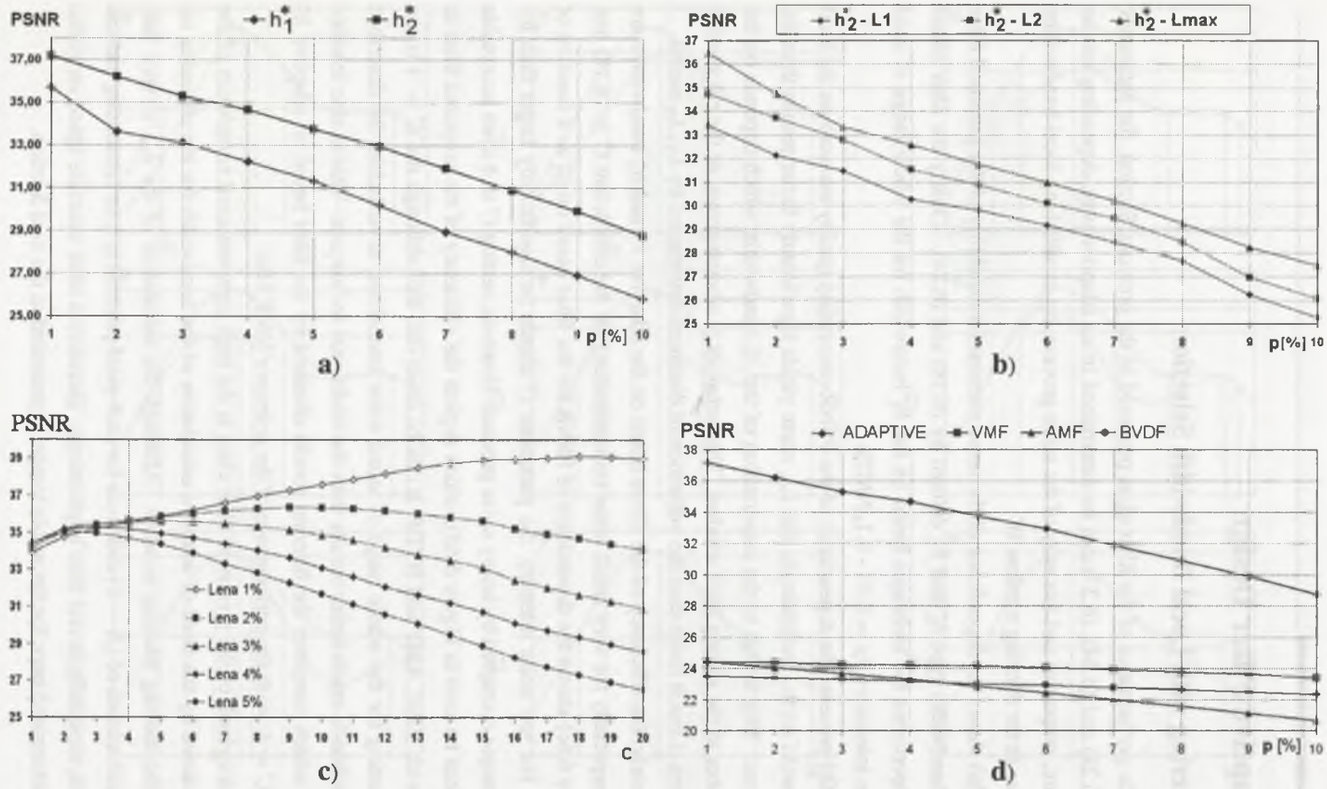


Fig. 6.6. Dependence of the optimal efficiency of the proposed adaptive filtering scheme using the bandwidth h_1^* (6.18) and h_2^* (6.24) - (a), beside the dependence of PSNR on the kind of Minkowski norm for the bandwidth h_2^* - (b), below dependence of PSNR on the tuning parameter C in (6.24) - (c) and the comparison of results obtained using the h_2^* bandwidth, (L_∞, $C = 4$), with the standard VMF and BVDF - (d), (tests were performed on color image LENA corrupted by impulsive noise NM2)

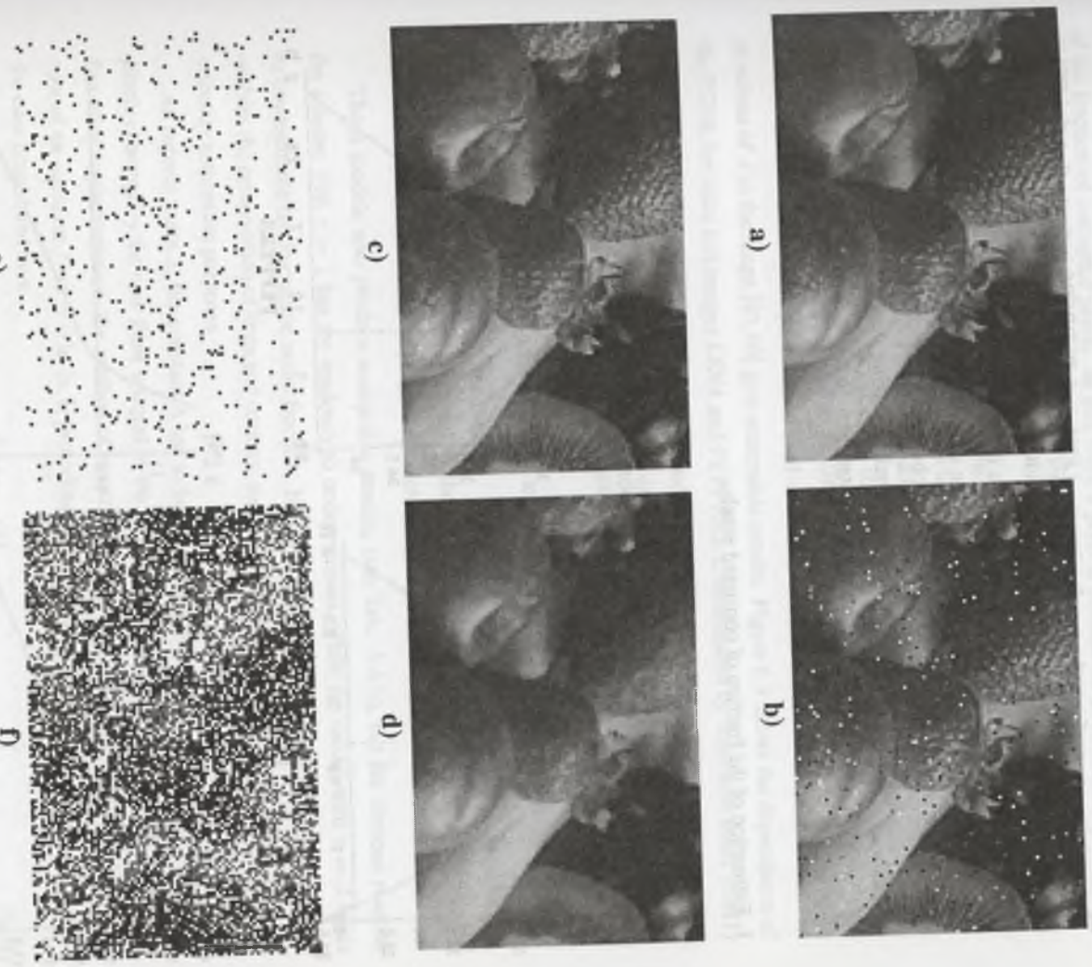


Fig. 6.7. Illustration of the efficiency of the adaptive filter design based on the h_2^* bandwidth with C parameter set to 4 in comparison with the VMF: a) part of the test image FRUITS, b) test image corrupted by impulsive noise, ($p = 0.02$, NM2), c) adaptive filter output, d) VMF output, e) and f) depict the difference (multiplied by factor 5) between the original and c) and d) respectively

6.3.2 Technique Based on Global Kernel Bandwidth

The experiments performed on color images *LENA* and *PEPPERS* indicate that the PSNR reaches its maximum for that value of the bandwidth parameter h, that leads to a number of pixel replacements equal to the number of noisy pixels in the noise-corrupted image, [294, 297, 310]. Figure 6.8, which shows the performance of the adaptive filter and depicts the fraction of replaced pixels as a function of h, validates this observation. Such filter behavior suggests that superior filtering results can be obtained by globally adapting the bandwidth h of the nonparametric scheme to the image structure and noise statistics. In this way the *Adaptive Nonparametric Filter* (ANPF) works as follows:

1. Estimation of the fraction of corrupted pixels,
2. Finding optimal, g l o b a l value of h,
3. Final filtering using the obtained optimal, global value of h.

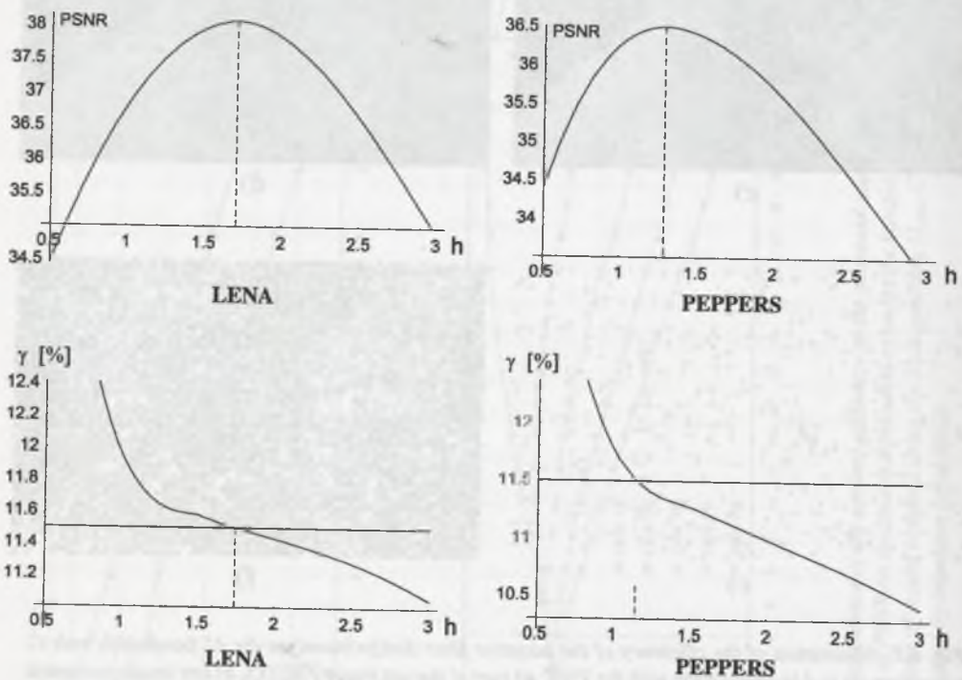


Fig. 6.8. Dependence of the filtering results on the h bandwidth using the Gaussian kernel, for the LENA and PEPPERS image with 11.5% of corrupted pixels, (NM1, $p = 0.04$), below the dependence of the fraction of pixels replaced by the filter on the h value for the noisy LENA and PEPPERS images

In most applications the noise intensity is unknown and we need to find a robust estimator of the fraction of corrupted pixels. In this work a pixel is considered to be undamaged by the noise process, if among its eight neighbors, there e x i s t a t l e a s t τ p i x e l s which, are *close* to it.

Two pixels are declared to be *close* if the L_2 distance between them, in the RGB color space, is less than a predefined constant d. As has been experimentally evaluated, this estimator works correctly, even for images with quite different structures. Table 6.4 shows the result of the estimation of the noise intensity p, using the described estimator for two test color images *LENA* and *PEPPERS*, with $\tau = 1, 2, 3$ and different fractions of the corrupted pixels p, (NM1).

The value of the distance parameter d used in the construction of the estimator is not critical, as values of d in the range $[40, 60]$ give acceptable results. Figure 6.9 shows the dependence of the PSNR for color test images *LENA* and *PEPPERS*, contaminated by impulsive noise (NM1) on the τ and d parameters. As can be seen good results are obtained for $\tau = 2$ and $d \in [40, 60]$, ($N = 9$). This is also confirmed by Fig. 6.10, which presents the filtering efficiency dependence on the parameter d for $\tau = 2$, (3×3 filter mask).

One can also use such estimators as:

◇ a pixel is considered to be undamaged, if among eight of its neighbors, there exist at least one, ($\tau = 1$) which is close to it,

◇ a pixel is considered to be undamaged by the noise process, if among eight of its neighbors, there exist at least three, ($\tau = 3$) which are close to it.

These models also produce acceptable results, (see Tab. 6.4 b), but for obvious reasons the scheme with $\tau = 1$ has the tendency to underestimate, while the model with $\tau = 3$ tends to overestimate the impulsive noise fraction. It is also easy to observe that the value of $\tau = 2$ enables the preservation of lines and corners, and therefore this parameter was used for the noise intensity estimation purposes.

As regards point **2**, the constant h has to be set for that value, for which the percentage of pixels changed by the new filter is equal to the estimated noise fraction p. In order to design a fast filter implementation, the method of bisection can be used. This method allows to find the root of an equation $f(x) = 0$ in $[a, b]$ providing that $f(x)$ is continuous and $f(a) \cdot f(b) < 0$. In the case considered here

$$f(h) = \gamma(h) - p, \tag{6.38}$$

where $\gamma(h)$ is the fraction of pixels changed by the filter, dependent on h.

Although the algorithm may be of infinite length and may not converge to the optimal value of h, it always provides a good approximation of the optimal h. To initiate the process, a starting interval [a,b] and a predefined number of iterations should be provided by the designer.

real p $\tau = 2$	estimated p (*LENA*)	estimated p (*PEPPERS*)
0.01	0.0113	0.0122
0.02	0.0206	0.0216
0.05	0.0500	0.0510
0.10	0.0980	0.0986
0.20	0.1942	0.1964
0.40	0.3972	0.3973
0.70	0.7501	0.7504

a)

real p LENA	estimated p $\tau = 1$	estimated p $\tau = 3$
0.01	0.0099	0.0158
0.02	0.0192	0.0253
0.05	0.0476	0.0547
0.10	0.0933	0.0103
0.20	0.1821	0.2016
0.40	0.3541	0.4301
0.70	0.5981	0.8472

b)

*Tab. 6.4. Comparison of the real and estimated fractions of the noisy pixels for $d = 50$: **a)** $\tau = 2$, (LENA and PEPPERS, NM4) **b)** $\tau = 1$ and $\tau = 3$, (LENA, NM4), [310]*

For a wide range of the fractions of noisy pixels (from $p = 0.01$ to more than 0.5, NM1) and various standard color images used for the evaluation purposes $f(0)f(4) < 0$ holds, so a long enough interval is: $a = 0$, $b = 4$, (see Fig. 6.8), [309].

In order to avoid the increase of the computational complexity caused by the estimator, the following solution is recommended. For finding the optimal value of h, using the method of bisection, not the whole image should be used, but only a small part of it, (we assume that the noise process is stationary). For example, if an image is composed of 500×500 pixels, taking randomly placed 25×25 square gives 625 pixels, which is enough for the purpose of the estimation and determination of the optimal h value. On the other hand, it is only 0.25% of the image pixels, so due to estimation and finding of the h value, (eight iterations) filtering time is extended only by about 2%, [310].

For the evaluation of the efficiency of the proposed filter, a number of simulations with different noise models presented in Section 1.3.2 were carried out. The results obtained with the ANPF were compared with a set of standard noise reduction methods listed in Tab. 6.1. The *Root of the Mean Squared Error* (RMSE), *Signal to Noise Ratio* (SNR), *Peak Signal to Noise Ratio* (PSNR), *Normalized Mean Squared Error* (NMSE) and *Normalized Color Difference* (NCD) were used for the comparisons, [242, 246].

The simulation results shown in Tab. 6.5, obtained using the noise model NM1, show that the new filter framework **excels significantly** over the standard techniques, widely used in many multichannel image denoising applications. The ANPF efficiency was also compared with different filtering techniques using the NM2 noise model, (Fig. 6.11) and its **superiority** over traditional techniques was again confirmed. The satisfying results presented in Tab. 6.5, Fig. 6.11 are also verified by Fig. 6.12, where the described filter has been compared with VMF, BVDF and DDF using noise model NM4 and the PSNR as the quality measure indicator.

LENA	NMSE [10^{-4}]	RMSE	PSNR [dB]
AMF	79.317	12.627	26.105
VMF	18.766	6.142	32.365
BVDF	24.587	7.030	31.192
DDF	18.872	6.159	32.340
HDF	18.610	6.116	32.401
AHDF	18.310	6.067	32.472
FVDF	22.251	6.688	31.625
ANNF	26.800	7.340	30.817
ANPF	**4.659**	**3.060**	**38.417**

PEPPERS	NMSE [10^{-4}]	RMSE	PSNR [dB]
AMF	108.650	13.338	25.629
VMF	27.570	6.719	31.585
BVDF	47.944	8.860	29.182
DDF	28.179	6.793	31.490
HDF	26.819	6.627	31.705
AHDF	26.430	6.579	31.768
FVDF	33.337	7.388	30.760
ANNF	45.115	8.595	29.446
ANPF	**8.426**	**3.713**	**36.736**

Tab. 6.5. Comparison of the efficiency of the ANPF with the standard techniques, (Tab. 6.1) using the LENA and PEPPERS standard color images, (NM1, $p = 0.04$)

Another good property of the new adaptive filter is that the new filter can be applied in an iterative way and that after the second or third iteration no further filtering is performed, (the PSNR is not decreasing, as it is in the case of VMF), which indicates that the new filter reaches very quickly its root, (see Fig. 6.13).

The good performance of the proposed adaptive filtering design is also confirmed by subjective, visual comparison with the VMF presented in Figs. 6.14 - 6.20, using different noise corruption schemes, [297,354,355,357]. It can be easily observed, that the new filter has a good ability to distinguish between the corrupted and undisturbed pixel images, which is especially visible when evaluating the filters' estimation errors in Figs. 6.14e, f) and 6.15e, f). As shown in Fig. 6.14 the new adaptive filter can be also successfully applied to gray scale images.

The adaptive nonparametric algorithm presented in this Section is based on the the concept of the *similarity* between pixels, *nonparametric estimation* and the *leave-one-out scheme*, but can also be seen as a *modification* and *improvement* of the commonly used *Vector Median Filter*. The computational complexity of the new filter is significantly lower than that of the VMF, especially when the 4-neighborhood system is applied. The presented comparison shows that the new filter outperforms the VMF, as well as other standard procedures used in color image processing in terms of objective and subjective quality measures.

The proposed algorithm is simple and fast and can be easily implemented. The proposed robust method of the estimation of noise intensity, enables the tuning of the filter design parameter h to the image structure and noise statistics. Thus, this filtering technique can be applied in many applications, in which fast and reliable removal of impulses is required with minimal image quality degradation.

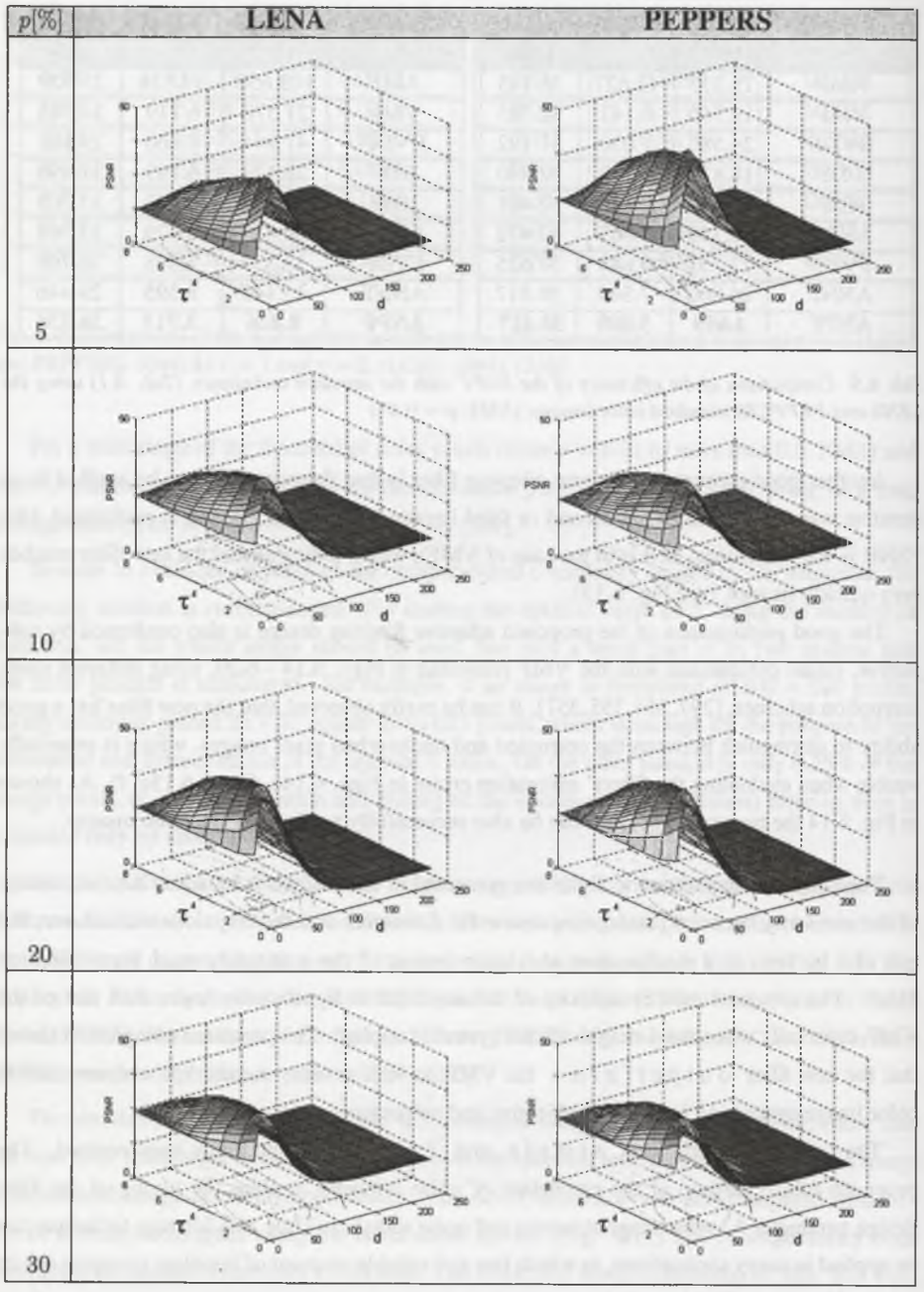


Fig. 6.9. PSNR dependence on the number of close neighbors τ and the distance parameter d. For the evaluations LENA and PEPPERS images contaminated with noise NM4 were used. Good results are obtained for $\tau = 2$, $(N = 9)$ and $d \in [40, 60]$

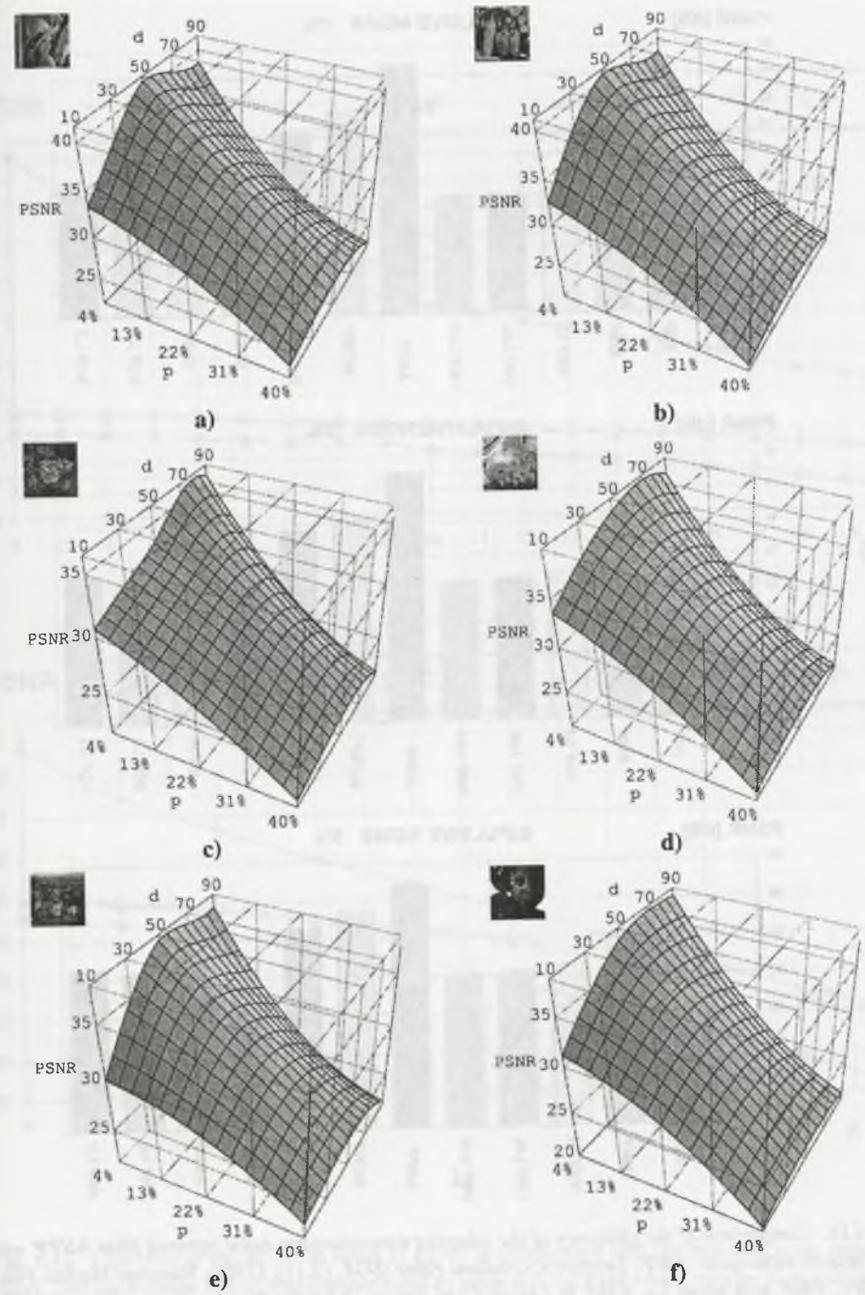


Fig. 6.10. PSNR dependence for the number of close neighbors $\tau = 2$ on the distance parameter d for the color test images (LENA, PEPPERS, MONARCH, FRUITS, GOLDHILL, GIRL, a - f), contaminated with impulsive noise NM4

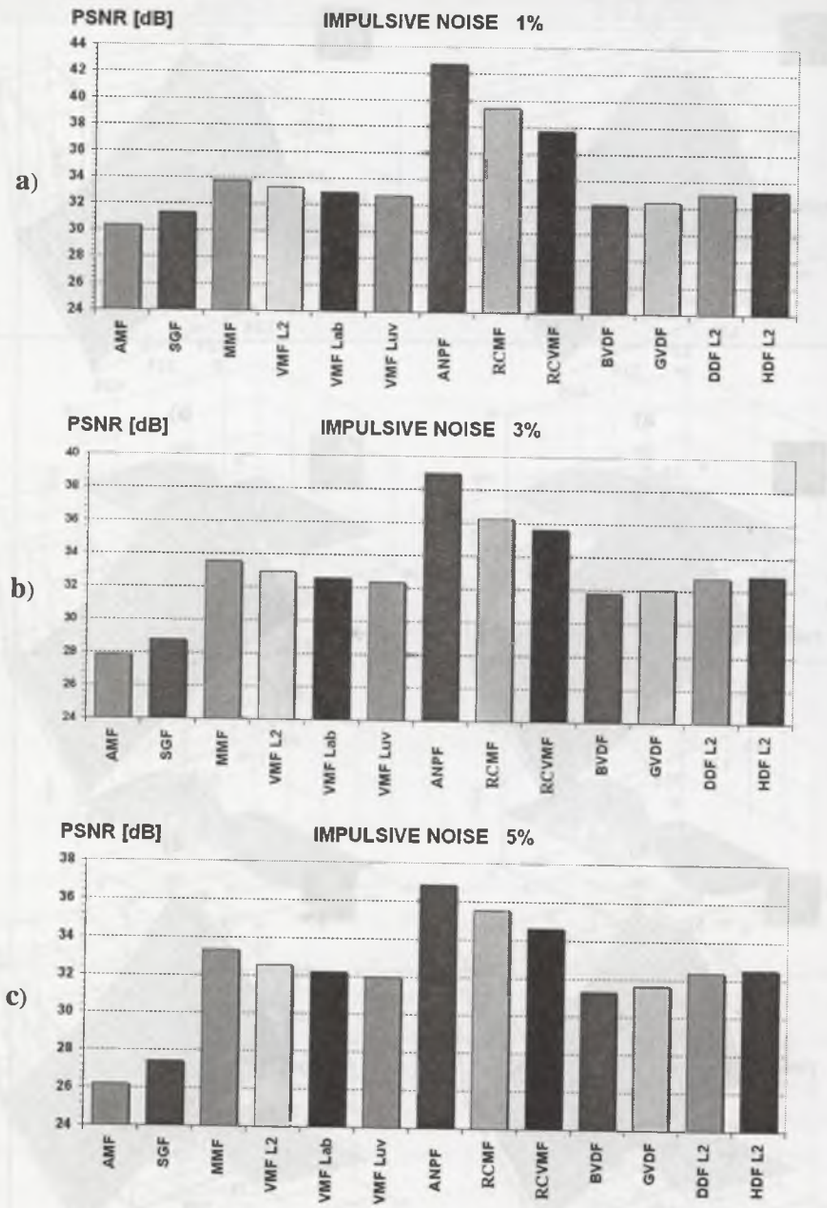


Fig. 6.11. Comparison of the efficiency of the adaptive nonparametric noise removal filter **ANPF** *with the standard techniques:* **AMF**, *Symmetric Gradient Filter (***SGF***, (2.11), [170]), Marginal Median Filter (***MMF***),* **VMF** *with norm L_2,* **VMF** *in Lab and Luv spaces, Adaptive Nonparametric Filter (***ANPF***), marginal Rank Conditioned Median Filter (***RCMFm***, $\tau = 3$, (Fig. 2.5 b)), Rank Conditioned Vector Median Filter (***RCVMF***, $\tau = 3$, (3.18)),* **BVDF**, **GVDF**, **DDF** *and* **HDF** *with norm L_2, (NM2)*

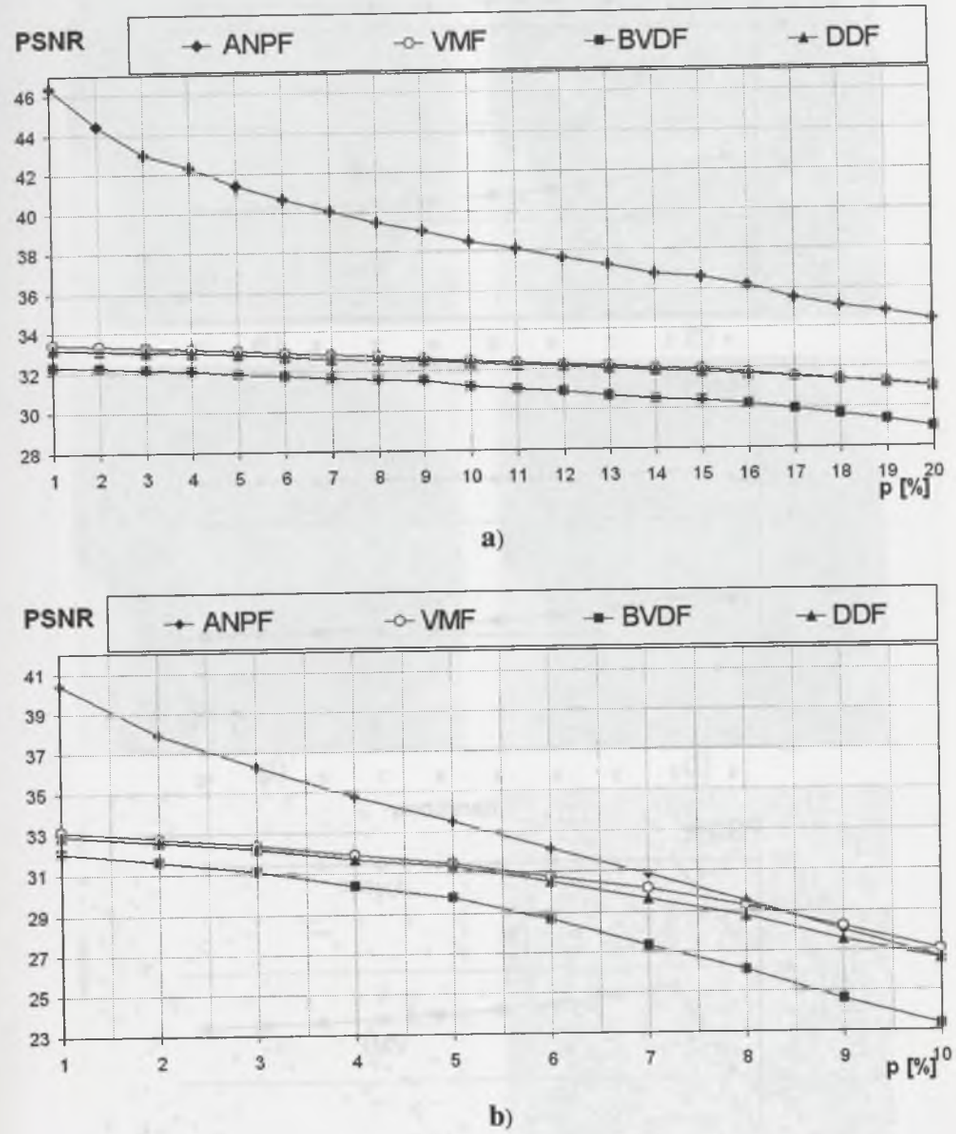


Fig. 6.12. Efficiency of the ANPF in terms of PSNR in comparison with the standard noise reduction filters. Test color image LENA *was contaminated by noise process NM4 with p ranging from 0.01 to 0.2*

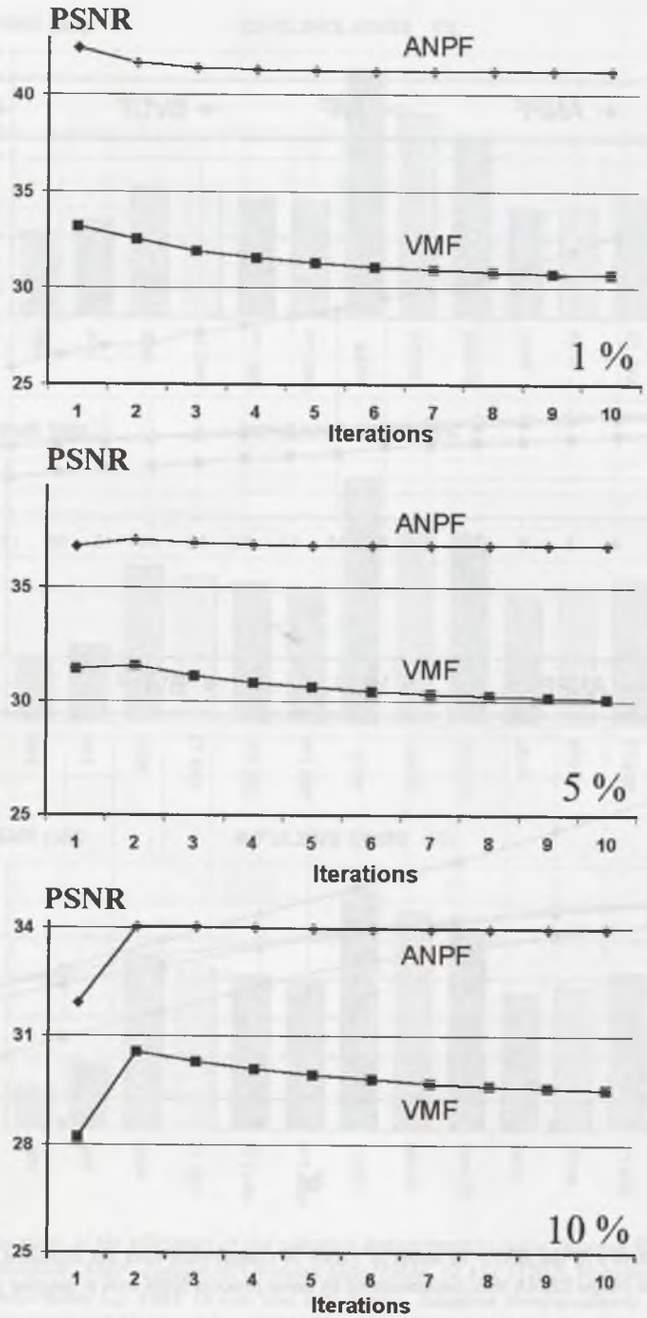


Fig. 6.13. Dependence of the noise reduction efficiency of the ANPF on the number of iterations, (LENA color image contaminated with NM4)

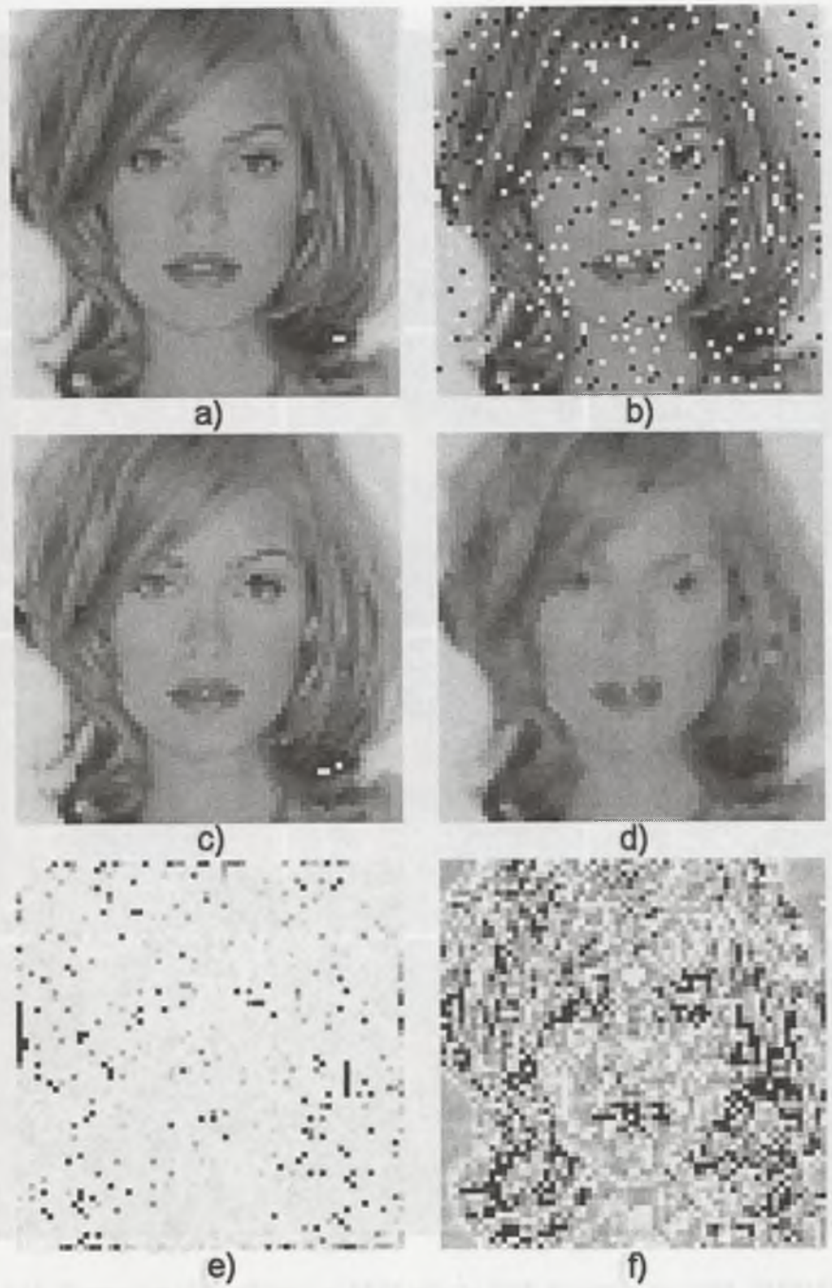


*Fig. 6.14. Comparison of the efficiency of the VMF and ANPF: **a**) gray test image, **b**) image contaminated by 2% impulse noise (NM1), **c**) image filtered with ANPF, **d**) VMF output and below **e**), **f**) the absolute difference between the original and filtered image for both the ANPF, (left) and the VMF, (right)*

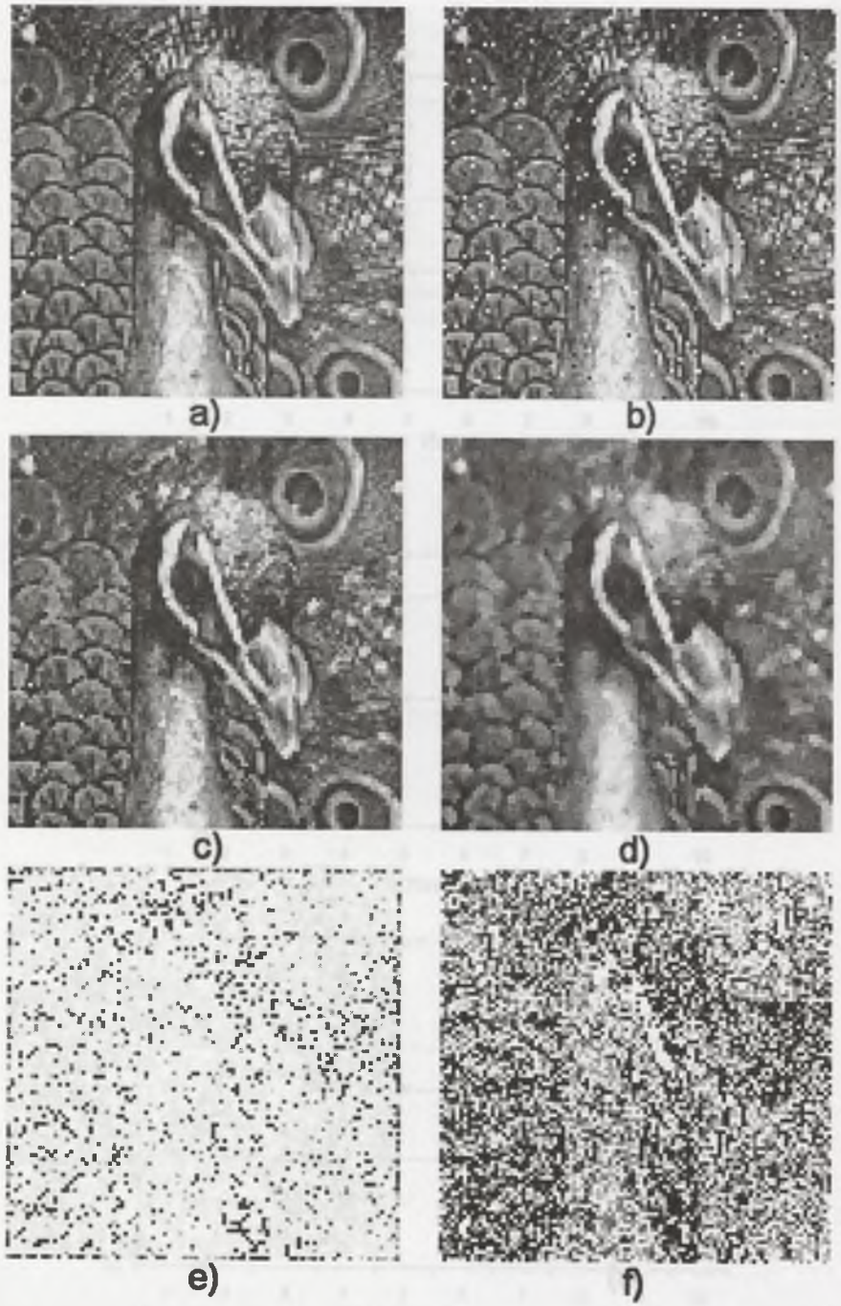


Fig. 6.15. Comparison of the efficiency of the ANPF and VMF, (blue channel): a) test color image, b) noisy image, ($p = 0.02$, NM1), c) image filtered with ANPF, d) VMF output and below e), f) the absolute difference between the original and filtered image for both ANPF, (left) and VMF, (right)

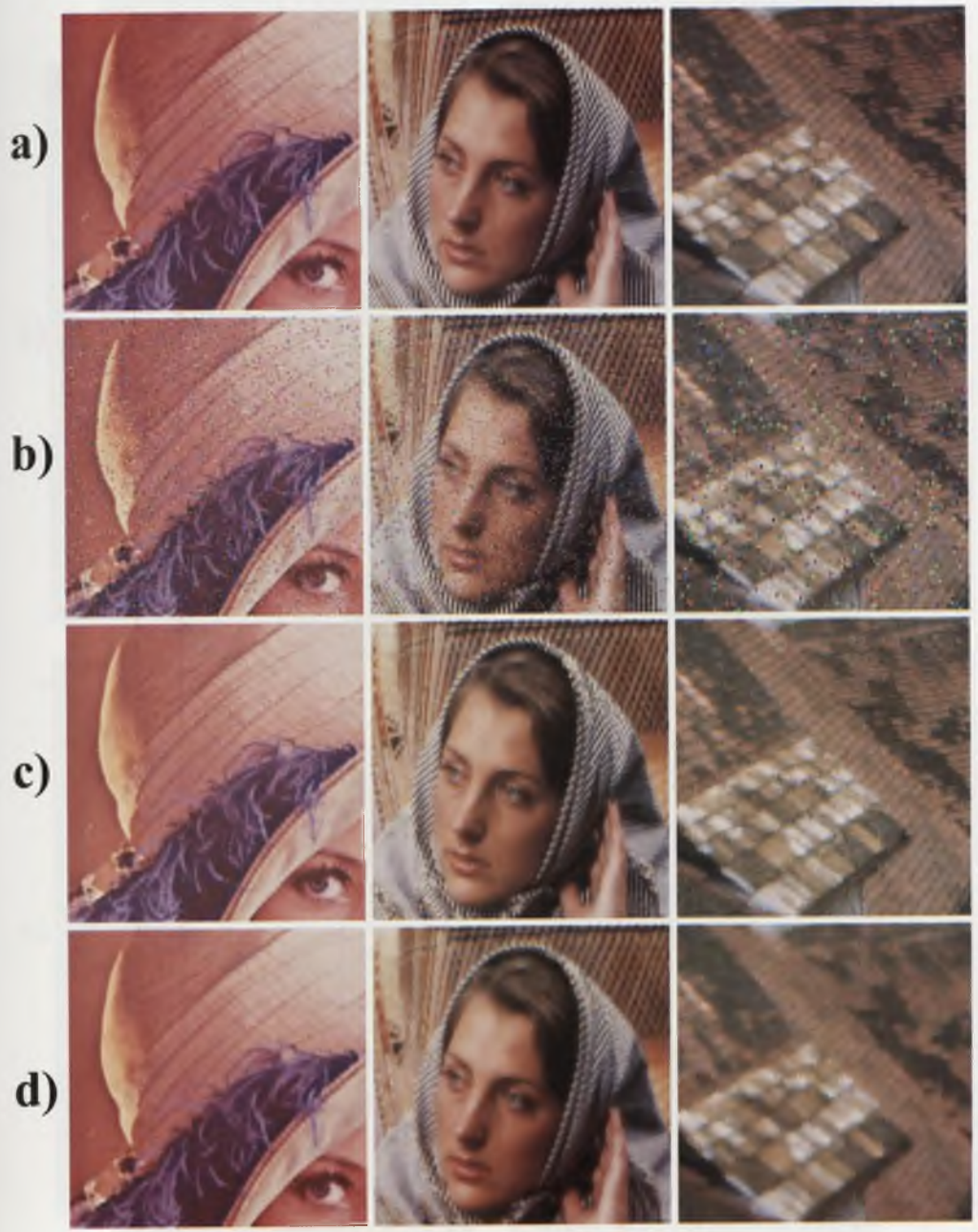


Fig. 6.16. Comparison of the efficiency of ANPF with the VMF: a) parts of the LENA, BARBARA and GOLDHILL images, b) images contaminated by 2% impulsive noise (NM1), c) images restored using the ANPF, d) the result of the filtering with the VMF

*Fig. 6.17. Efficiency of the ANPF, **a**) test image PORTRAIT [402], **b**) image degraded by $p = 0.04$ impulse noise (NM3, with $p_1 = p_2 = p_3 = 0.2$, $p_4 = 0.4$), **c**) ANPF output, **d**) VMF output*

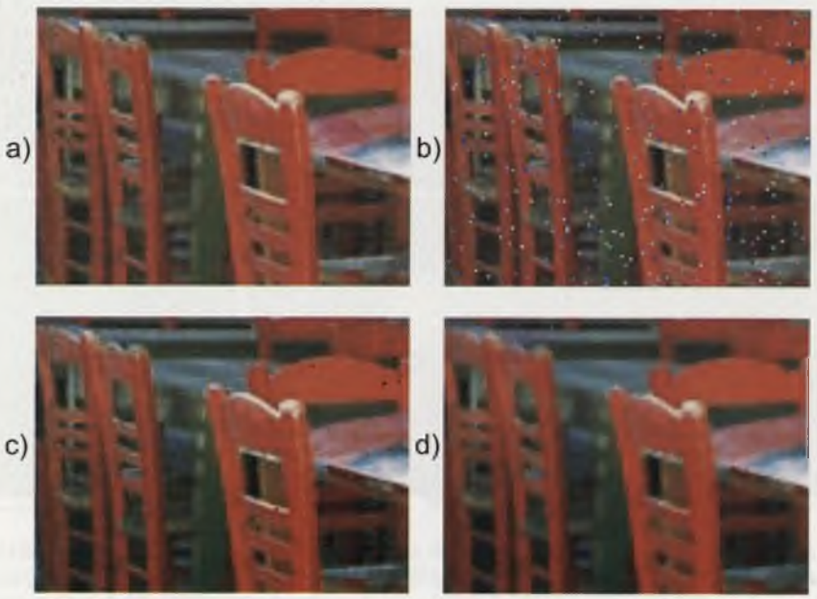


*Fig. 6.18. Comparison of the efficiency of the ANPF with the VMF, **a**) test image CAFE [402], **b**) image degraded by $p = 0.03$ impulse noise (NM2), **c**) image filtered using the ANPF, **d**) VMF output*

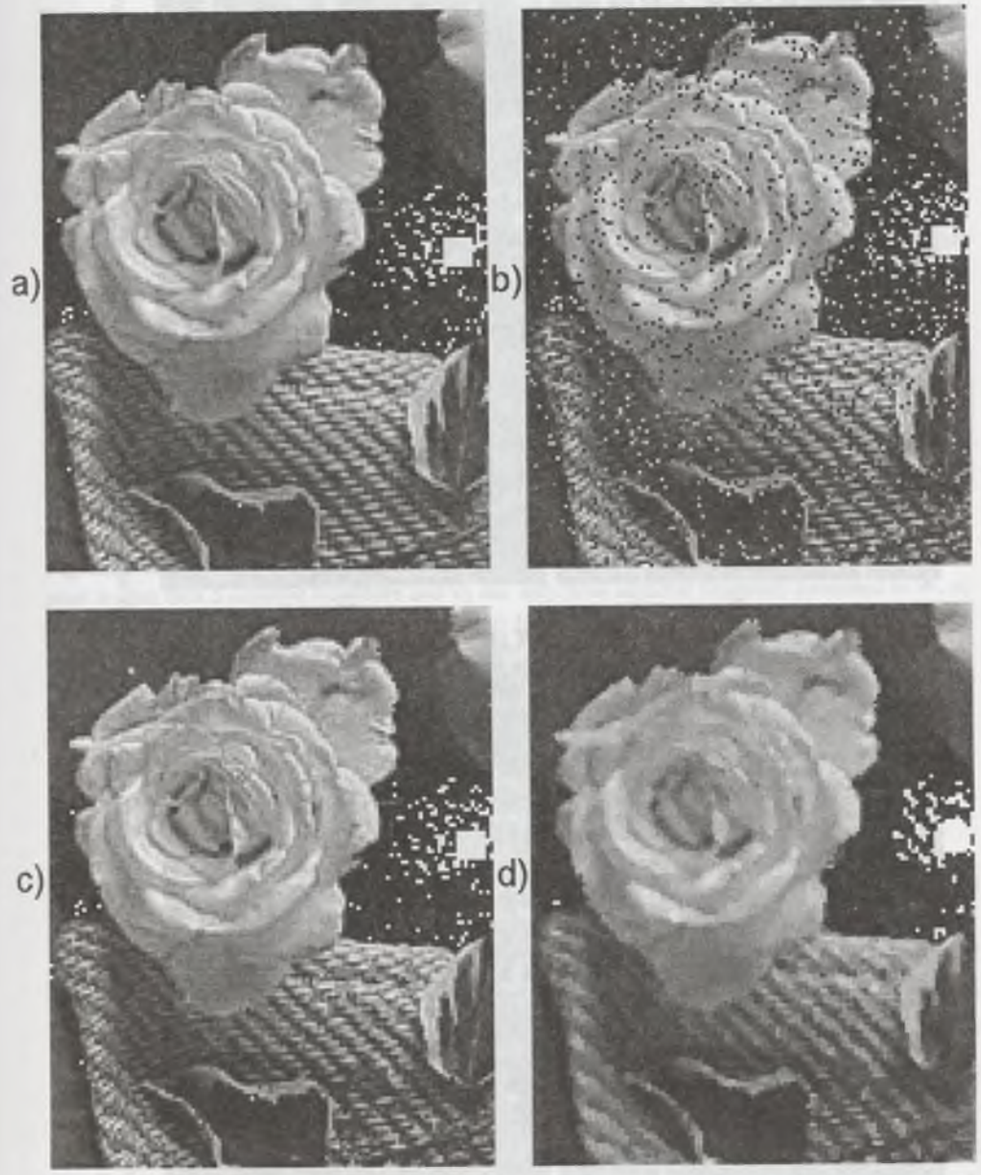


*Fig. 6.19. Comparison of the efficiency of the VMF and the ANPF, (red channel), **a**) test image ROSE [402], **b**) image contaminated by 6% impulsive noise (NM3, $p = 0.06$, with $p_4 = 1$), **c**) image filtered using the proposed adaptive technique, **d**) output of the VMF*

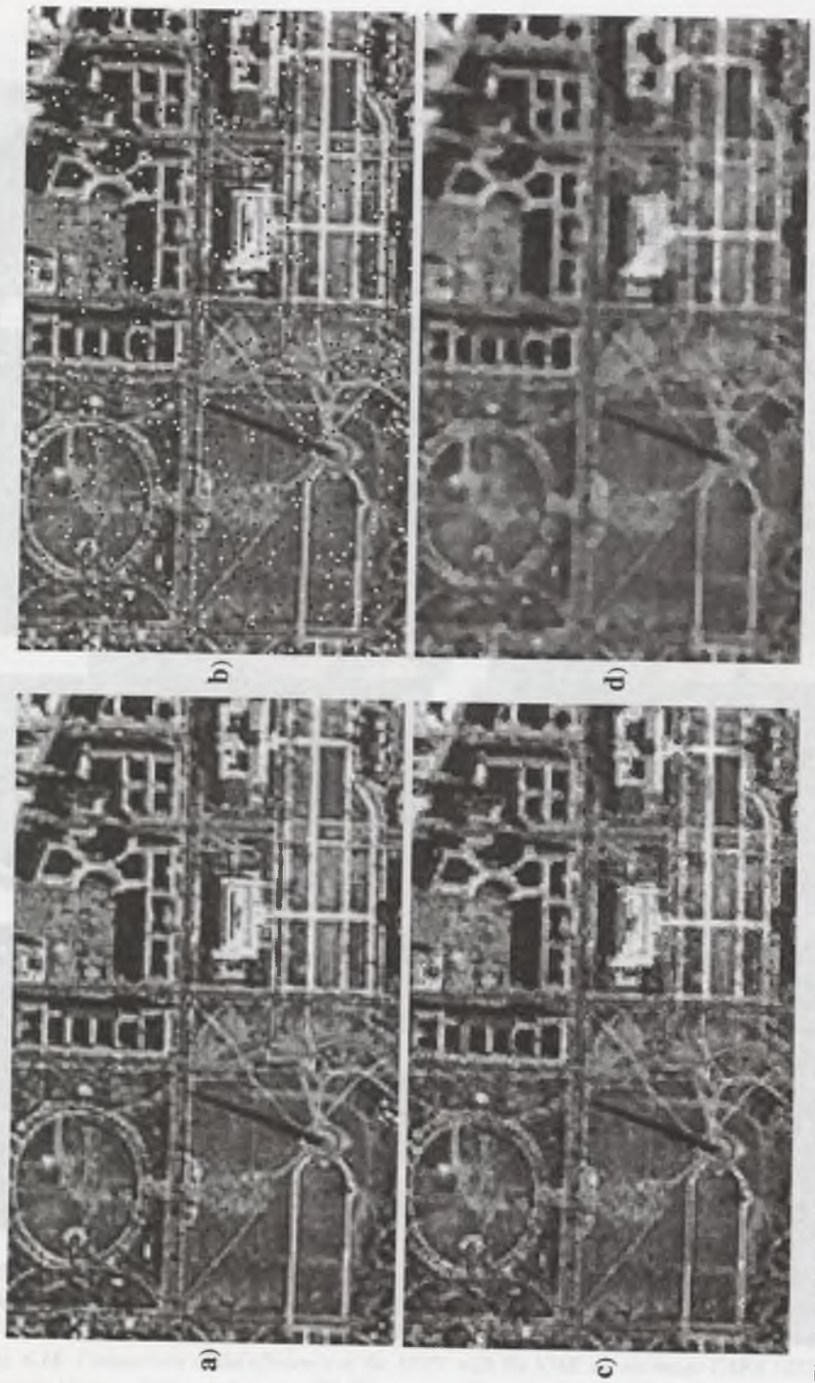


Fig. 6.20. Illustration of the efficiency of ANPF: a) test aerial image (green channel), b) images corrupted by 10% impulsive noise (NM1), c) new filter output, d) effect of median filtering (3 × 3 mask), [347]

6.3.3 Fast Nonparametric Filter Design

The proposed here fast filter design is a modification of the filtering framework presented in the previous Section and is based on the idea of the comparison of the image pixels contained in a filter window W with their adjacent pixels, (direct neighbors). The filter output is the pixel in W, which is most similar to its direct neighborhood contained in W. Therefore, the estimated probability density function serves as a measure of similarity in the chosen color space, [331, 350, 353, 356, 357, 359]. If a pixel is similar to its neighborhood, the probability density estimation for that pixel results in a relatively large value. Noisy pixels on the other hand are almost always outliers from the cluster formed by adjacent pixels and therefore the probability density estimation for those pixels results in relatively small values, [343].

x_2	x_3	x_4
x_5	$\boxed{x_1}$	x_6
x_7	x_8	x_9

a)

x_2	x_3	
$\boxed{x_5}$	x_1	
x_7	x_8	

b)

$\boxed{x_2}$	x_3	
x_5	x_1	

c)

15	24	95
33	**72**	90
41	45	55

d)

15	24	
33	72	
41	45	

e)

15	24	
33	72	

f)

Fig. 6.21. Illustration of the adjacency relation: a) the central pixel x_1 has 8 neighbors in W, b) the pixel x_5 has then 5 adjacent neighbors and x_2 has only three adjacent neighbors contained in W, c). Beside an example of the filtering window with gray scale intensities related to Fig. 6.22 is shown (d - f)

Given a set of noisy image samples $x_1, x_2, \ldots, x_N$ from the filter window W, let $\sim$ denotes the adjacency relation between two pixels contained in W. Assuming the 8-neighborhood system, the central pixel has 8 adjacent neighbors, the pixels in the corners of W have 3 adjacent neighbors and the remaining pixels have 5 adjacent neighbors determined by the $\sim$ relation, (Fig. 6.21). The sum of similarity values for the sample x_k is then determined as

$$\Psi(x_k) = \sum_{x_j \sim x_k} \mathcal{K}\left(\frac{\|x_j - x_k\|}{h_j}\right). \tag{6.39}$$

The filter output is defined as that x_k for which $\Psi(x_k)$ is maximal, (see Fig. 6.22). The total similarity value $\Psi(x_k)$ is not normalized to bandwidth and number of sample values. The reason is that the values of $\Psi(x_k)$ for different x_k are only used for comparison among each other and omission of the normalization results in a significant performance gain, as it privileges the central sample, which has the largest number of neighbors, (Figs. 6.21a, d).

The bandwidth in (6.39) can be determined according to (6.18) and hence depends on the standard deviation $\hat{\sigma}$. Since $\hat{\sigma}$ is computed using only pixels from the filter window, the bandwidth is very sensitive to noise and may vary over a big range of values. As an option an experimentally chosen fixed value can be used as bandwidth to avoid this effect, (Fig. 6.24), [358,362].

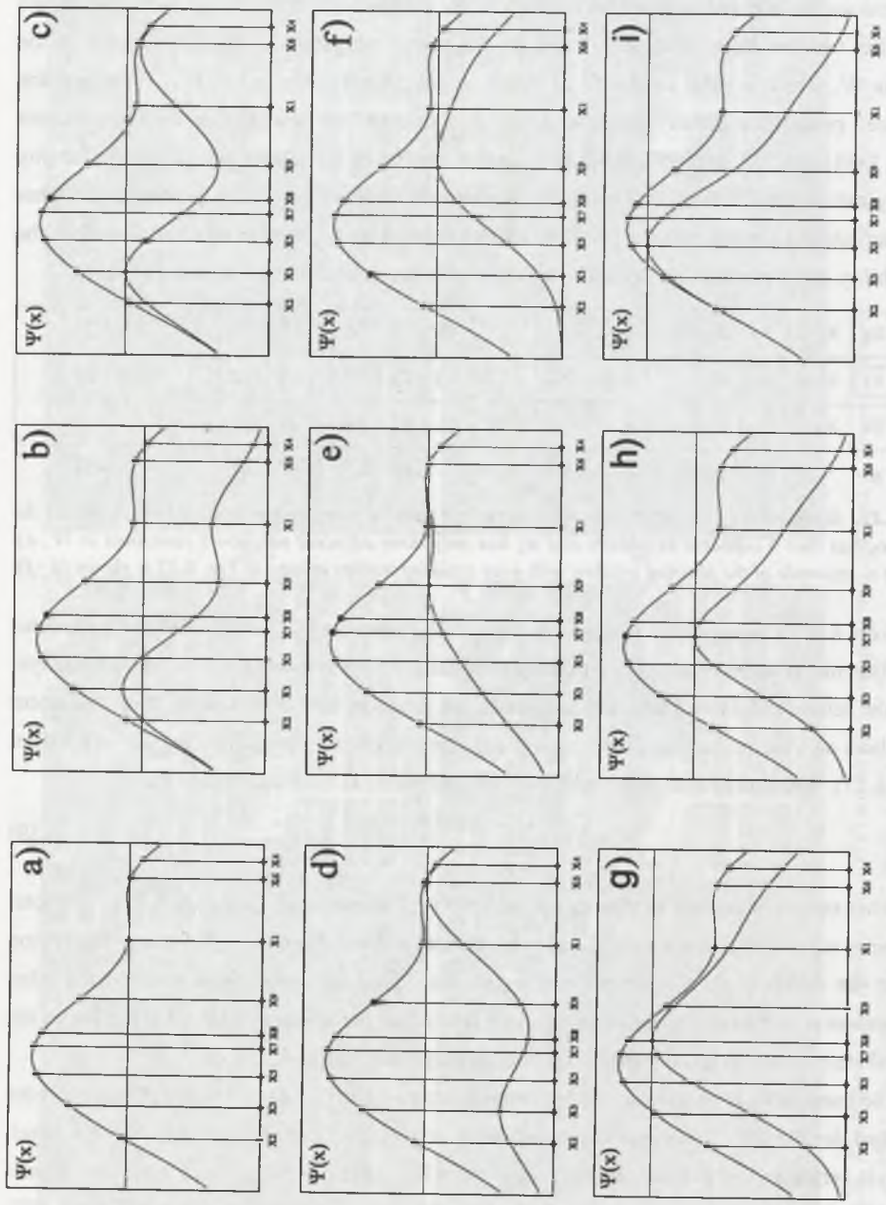


Fig. 6.22. Density estimation plots for the case of a filtering window consisting of pixels shown in Fig. 6.21, a) and additional plots of the probability density estimation using the ~ neighborhood concept for x_1 (b), x_2 (c), . . ., x_8 (i) samples. As can be seen the highest value of the probability density estimation is obtained for x_8, (plot i) and this pixel will replace the central noisy pixel x_1

p	0.05	0.05	0.05	0.10	0.10	0.10
FILTER	MAE	MSE	NCD	MAE	MSE	NCD
Noisy	2.54	393.3	0.0415	5.10	790.2	0.0838
VMF	**3.27**	**31.2**	**0.0387**	**3.42**	**34.2**	**0.0400**
BVDF	3.81	39.8	0.0400	3.95	44.2	0.0412
DDF	3.39	32.8	0.0389	3.51	35.4	0.0400
HDF	3.42	31.2	0.0399	3.55	33.9	0.0412
$\mathcal{G}$, L_2, A	0.79	11.5	0.0093	0.98	20.2	0.0125
$\mathcal{G}$, $h = 55$	0.42	11.8	0.0051	0.79	20.8	0.0100
$\mathcal{G}$, L_1, A	0.82	14.8	0.0101	1.16	24.9	0.0149
$\mathcal{E}$, L_2, A	1.17	15.3	0.0138	1.23	21.7	0.0151
$\mathcal{L}$, L_2, A	0.43	10.6	0.0055	0.84	34.2	0.0128
$\mathcal{T}$, L_2, A	0.45	14.0	0.0063	0.96	50.8	0.0159

***Tab. 6.6.** Filtering results achieved using the test image LENA contaminated by impulsive noise (NM2) using different kernels with adaptive (A) and globally determined bandwidth h and different norms, ($\mathcal{G}$ denotes the Gaussian kernel, $\mathcal{E}$ the kernel of Epanechnikov, $\mathcal{L}$ the Laplacian kernel and $\mathcal{T}$ the linear, Triangle kernel, see Fig. 6.23)*

For the evaluation purposes, the color test image *LENA* was corrupted with 1 to 10 percent impulsive noise, (NM2). The filter quality was measured using the *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE) and the *Normalized Color Difference* (NCD).

Tab. 6.6 and Fig. 6.25 show the results of a quantitative comparison between the described fast filter scheme and the VMF as well as the BVDF, HDF and DDF, (Tab. 6.1). For experiments with fixed bandwidth an experimental value of $h = 55$ was chosen (Gaussian kernel), which brought subjectively good results, (see Fig. 6.24). As can be seen from Tab. 6.6, the noise reduction capability depends to some extent on the choice of the filter kernel, (see Fig. 6.23) and again good results were obtained for the Triangle kernel. Apart from the sometimes up to a few times lower MAE and NCD values, compared with the vector median filter, the new fast filter shows enormous improvements in detail preservation, (Figs. 6.26, 6.27).

The always very low values of MAE and NCD show that the new filter is **clearly superior** to VMF, BVDF and DDF in terms of detail preservation for all applied filter settings. Another advantage of the proposed filtering class is its very low computational complexity when compared to the VMF. For the VMF, the calculation of 36 distances between pixels are needed, whereas the new filter structure with fixed bandwidth requires only 20 different distances, which makes the new filter class interesting for real-time applications The remarkably good results for the probability density estimation with fixed bandwidth can be used for very fast filtering, as in this case there is no need to determine adaptively the variance of samples in W.

Kernel	PDF $f_h(\chi)$
Gaussian ($\mathcal{G}$)	$\frac{1}{\sqrt{2\pi}h}e^{-\frac{\chi^2}{2h^2}}$
Epanechnikov ($\mathcal{E}$)	$\left[\frac{3h^2}{6h^2-2}\left(1-\frac{\chi^2}{h^2}\right)\right]^+$
Laplace ($\mathcal{L}$)	$\frac{1}{2h}e^{-\left\|\frac{\chi}{h}\right\|}$
Triangular ($\mathcal{T}$)	$\left[\frac{h}{2h-1}\left(1-\left\|\frac{\chi}{h}\right\|\right)\right]^+$

$$[f_h(\chi)]^+ = \begin{cases} f_h(\chi) & \text{if } |\chi/h| < 1, \\ 0 & \text{otherwise,} \end{cases}, \quad f_h(0) = 1, \quad \int_{-\infty}^{+\infty} f_h(\chi)\, d\chi = 1.$$

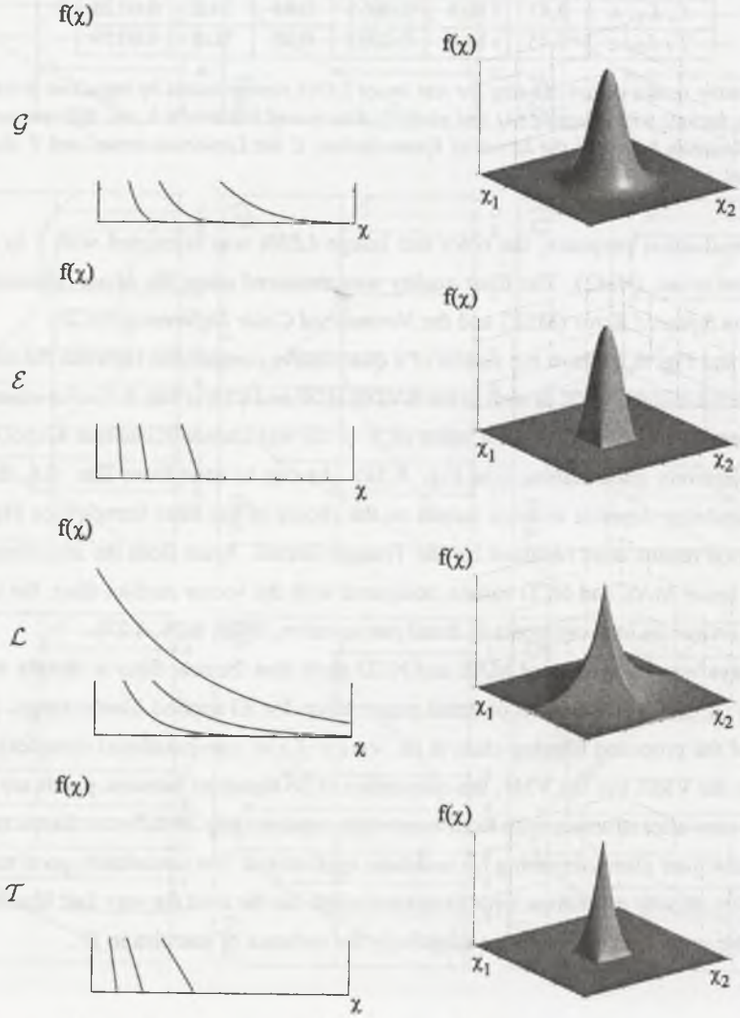


Fig. 6.23. Nonparametric 1D kernels and their 2D realizations, (6.6)

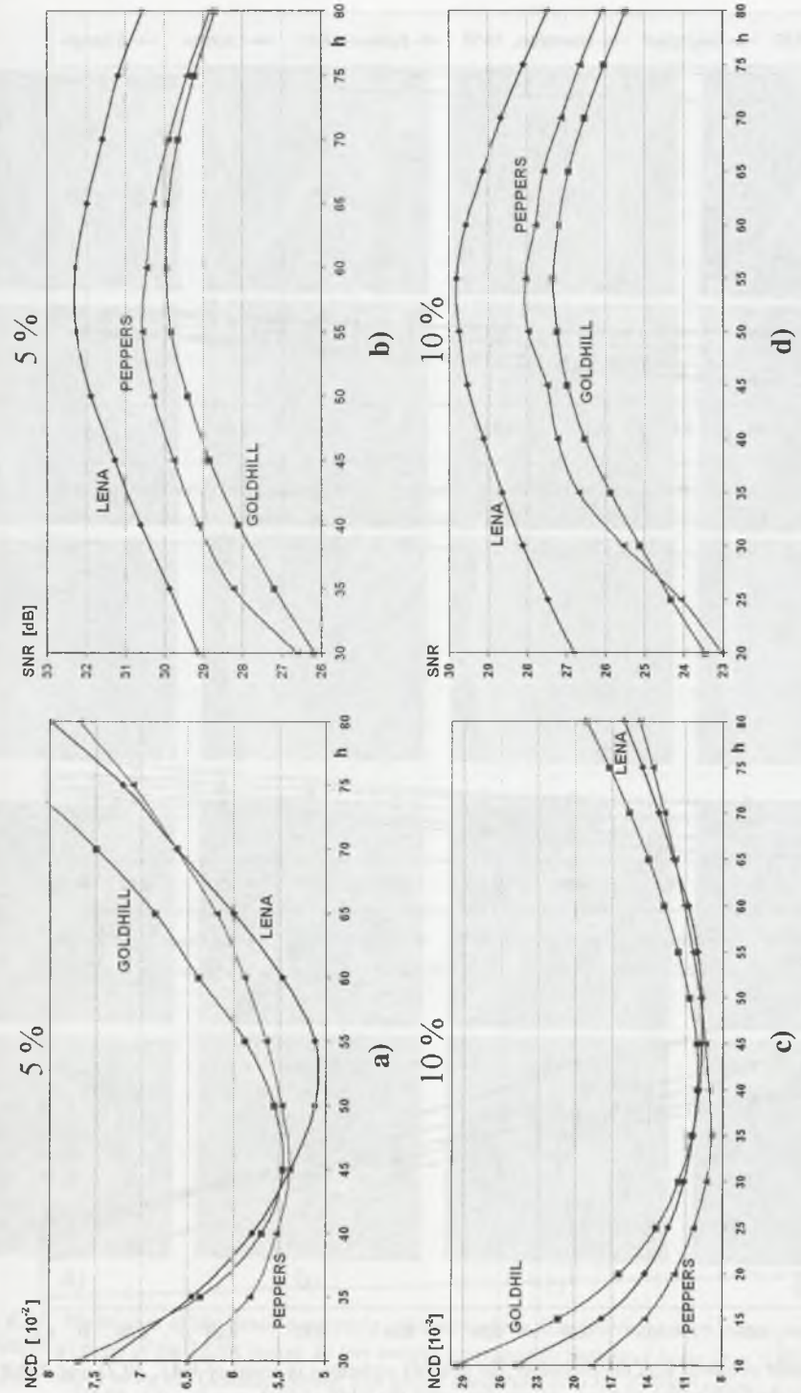


Fig. 6.24. Dependence of the efficiency of the proposed fast filter (NCD and SNR) on the global, experimentally established global bandwidth of the Gaussian kernel for the test images LENA, PEPPERS, GOLDHILL contaminated by 5% and 10% impulsive noise, (NM2). The value of $h = 55$ was used for the comparisons with the standard filtering techniques presented in Tab. 6.6

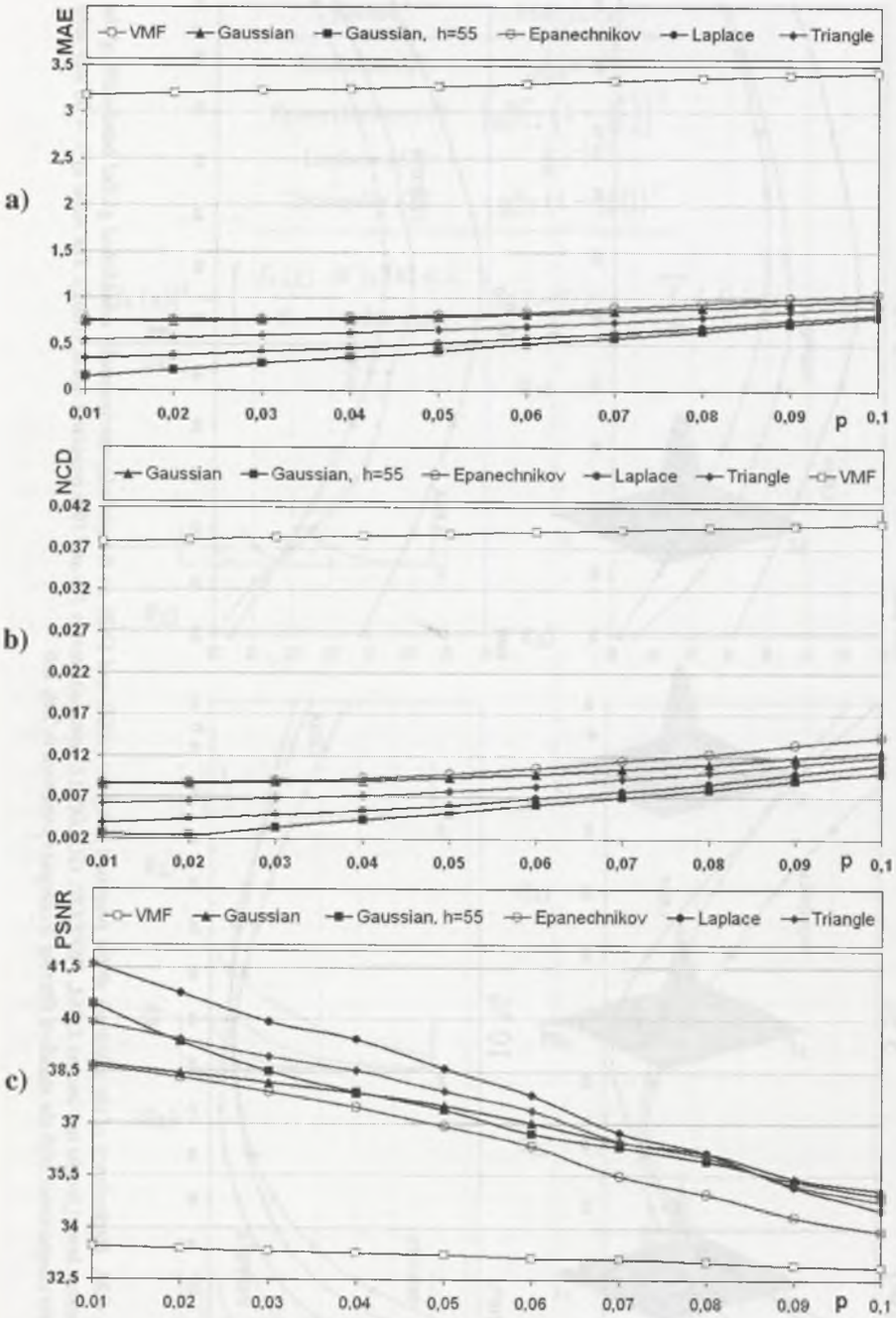


Fig. 6.25. Results obtained with the proposed fast filtering technique in terms of MAE, NCD and PSNR. The plots show the filter performance in comparison with the VMF, (LENA, $p = 0 - 0.1$, NM2)

*Fig. 6.26. Illustration of the detail preserving efficiency of the fast filtering design in comparison with the VMF: **a**) parts of the LENA image, **b**) test images corrupted by impulsive noise ($p = 0.05$, NM2), **c**) fast filter output using the Gaussian kernel and global bandwidth $h = 55$, **d**) VMF output*

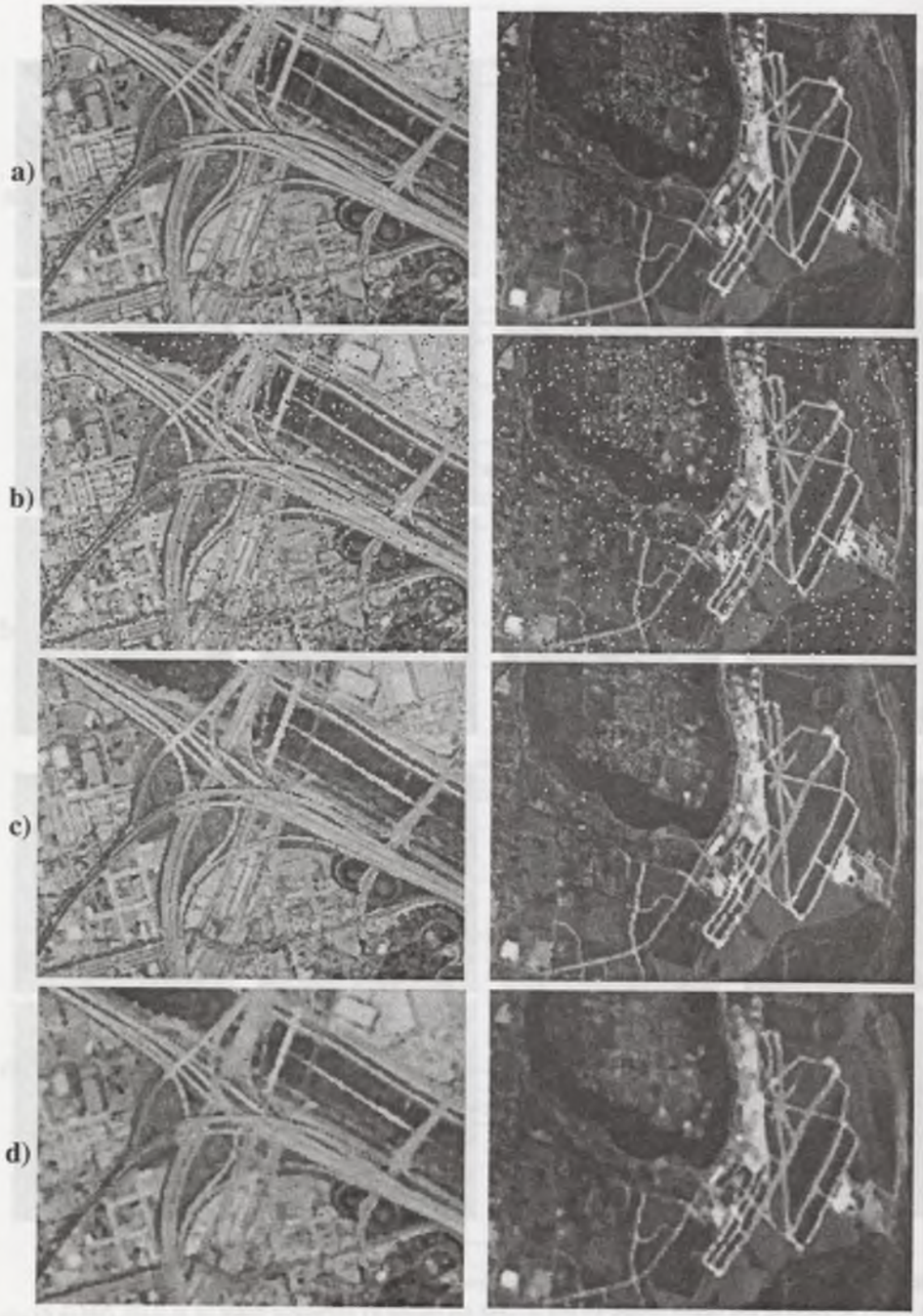


*Fig. 6.27. Illustration of the fast filter efficiency: **a**) test aerial images (green channel), **b**) images corrupted by 10% impulsive noise (NM1), **c**) fast filter output, **d**) VMF output, [346]*

7

Adaptive Vector Median Based Techniques

In this Chapter a class of Weighted Vector Directional Filters (WVDF) and Sigma Vector Median Filters (SVMF), which are based on the selection of the output sample from the multichannel input set, are presented. The WVDF output minimizes the sum of weighted angular distances to other input samples from the filtering window. Dependent on the weighting coefficients, the class of the WVDFs can be designed to perform a number of smoothing operations with different properties, that can be applied for specific filtering scenarios. The optimized WVDFs are able to remove image noise, while maintaining image details preservation capabilities and sufficient robustness for a variety of signal and noise statistics.

The multichannel SVMF is a novel adaptive filtering technique based on the robust order statistic concepts and simplified statistical measures of vectors' dispersion. The simulation results indicate that the presented algorithms are computationally attractive, yield good performance and are able to preserve salient image features, while efficiently suppressing impulsive noise.

7.1 Weighted Vector Directional Filters

BASED on the magnitude of vectors, filtering techniques process the color image according to its brightness, whereas operating on the directionality of vectors, image filters take into account the chromatic properties of the input samples, [196, 200, 246]. Therefore, the filtering

techniques operating on the directional domain of color images are able to preserve their chromaticity. Since the human visual system is sensitive to changes in color and edge information (indication of the shape and boundary of objects in the image), color chromaticity preservation along with the noise attenuation is a fundamental property required in many applications such as television image denoising, virtual restoration of artworks, satellite image processing, old movie restoration, surveillance applications and many others, [1, 8, 39, 49, 157, 200].

Recently introduced *Weighted Vector Directional Filters* (WVDF), [15, 16, 68, 97, 187, 189] employ non-negative real weight coefficients $\{\psi_1, \psi_2, \ldots, \psi_N\}$ associated with the input vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$. These filters pass to the output the vector $\mathbf{y} \in W$, which minimizes the aggregated weighted angular distance to other samples belonging to W.

This angular minimization approach is useful for the directional data such as color image data. In [218] it has been proven that in the case of color images, filtering schemes based on the directional processing may achieve better performance in terms of the color chromaticity preservation than approaches operating on the vectors' magnitude.

Let us consider the aggregated weighted distance A_k associated with the input vector $\mathbf{x}_k$

$$A_k = \sum_{j=1}^{N} \psi_j \, a(\mathbf{x}_k, \mathbf{x}_j), \qquad k = 1, 2, \ldots, N, \tag{7.1}$$

where $a(\mathbf{x}_k, \mathbf{x}_j)$ denotes the angle between vectors $\mathbf{x}_k = (x_{k1}, x_{k2}, x_{k3})$ and $\mathbf{x}_j = (x_{j1}, x_{j2}, x_{j3})$. The ordered sequence of $A_1, A_2, \ldots, A_N$ is given as $A_{(1)} \leq A_{(2)} \leq \ldots \leq A_{(N)}$ and the ordering of $A_{(k)}$ implies the same ordering of the input set $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, which results in the ordered set $\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(N)}$, where $\mathbf{x}_{(k)}$ is associated with $A_{(k)}$. In this way, the WVDF output is defined as the lowest order statistic $\mathbf{x}_{(1)}$, which is equivalent to the sample minimizing the cumulated angular distance. From this algorithm structure, it is evident that the WVDF output is restricted to the dynamic range of the input samples and thus, it can never introduce new samples.

Let us assume that $\mathbf{x}_{(1)}$ is the *minimum* vector and $\mathbf{x}_{(N)}$ is the *maximum* vector of the input set W. The WVDF output $\mathbf{y}(\boldsymbol{\psi}, W)$ is a function of the weight vector $\boldsymbol{\psi} = \{\psi_1, \psi_2, \ldots, \psi_N\}$ and it can be expressed as the sample $\mathbf{y}$ minimizing

$$\mathbf{y}(\boldsymbol{\psi}, W) = \arg \sum_{j=1}^{N} \psi_j \, a(\mathbf{y}, \mathbf{x}_j). \tag{7.2}$$

Then, the following is valid:

- the WVDF filter has N independent parameters, since its output $\mathbf{y}(\boldsymbol{\psi}, W)$ depends on the weight vector $\boldsymbol{\psi}$,
- the WVDF output corresponds to one of the local minima of $A(\mathbf{y})$,

- the WVDF output $\mathbf{y}(\boldsymbol{\psi}, W)$ is always one of the samples of W and therefore it cannot introduce any new outliers and color artifacts.

Each setting of the weight coefficients represents a unique filter, which can be used for specific purposes. Using an optimization scheme, (Fig. 7.1) the weight coefficients can follow the statistics and structural context of the desired signal and can be adapted in a required manner.

The purpose of image filtering is to estimate the desired signal $\mathbf{o}$ as precisely as possible. To measure the similarity between the original image $\mathbf{o}$ and the filtered image $\mathbf{y}$, a number of different objective measures, based on the difference in the statistical distributions of the pixel values, can be utilized, [37, 38, 64]. One of the most popular criteria is the Minkowski norm given by

$$e_i = \|\mathbf{o}_i - \mathbf{y}_i\|_\gamma = \left(\sum_{k=1}^{m} |o_{ik} - y_{ik}|^\gamma \right)^{1/\gamma}, \tag{7.3}$$

where γ denotes the norm parameter and o_{ik} and y_{ik} are the k-th elements of the original image pixel $\mathbf{o}_i$ and the filter output $\mathbf{y}_i$, $(i = 1, 2, \ldots, Q)$, respectively. The error criterion (7.3) expresses the loss in performance or error of the filtering operation.

As in most image processing problems, a *cost function*, which depends on the mostly unavailable original image and the filter output, will be used to penalize errors during the filtering procedure. It is natural to assume that if one penalizes filtering errors through the cost function, then the optimal solution is the function of the inputs, that minimizes the expected average loss $E\{\|\mathbf{o} - \mathbf{y}\|_\gamma\}$, where $E\{\cdot\}$ signifies the statistical expectation.

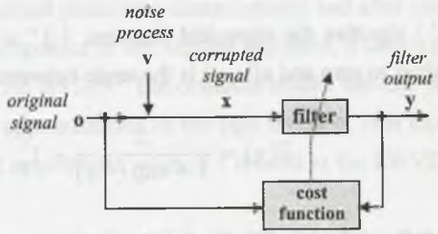


Fig. 7.1. Filtering problem, in which a filter is based on the minimization of the cost function

With the constraint of non-negative weights, keeping the aggregated measure A_k in (7.1) positive, the optimization problem with inequality constraints can be expressed as, [17, 188, 189]

$$\boxed{\text{minimize } J(\boldsymbol{\psi}, W) \quad \text{with subject to} \quad \psi_k \geq 0, \quad \text{for } k = 1, 2, \ldots, N.} \tag{7.4}$$

Thus, the setting of the WVDF weight coefficients depends on the cost function $J(\boldsymbol{\psi}, W)$, which can be defined in many ways. It has been observed in [24], that J_1 and J_2 criteria are useful in environments corrupted by impulsive noise and describe well the detail preservation and noise attenuation capabilities of the optimized filters. Therefore, in this work the J_1 and J_2 cost functions are used

$$J_1(\boldsymbol{\psi}, W) = E\{\|\mathbf{o} - \mathbf{y}\|_1\}, \qquad J_2(\boldsymbol{\psi}, W) = E\{\|\mathbf{o} - \mathbf{y}\|_2\}. \tag{7.5}$$

7.1.1 Angular Sigmoidal Optimization

Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ be the input set of the m-channel samples and let $\mathbf{o}$ be the desired (original or noise-free) image. Let us assume that each input sample $\mathbf{x}_k \in W$ is associated with the non-negative real weight ψ_k, for $k = 1, 2, \ldots, N$. Then, we can modify the sigmoidal optimization presented in Section 2.3, so that it can work within the multichannel framework, [186, 189]

$$\psi_k(i+1) = \{\psi_k(i) + 2\epsilon\Upsilon[\mathbf{o}(i) - \mathbf{y}(i)]\,\Xi(\Upsilon[\mathbf{x}_k(i) - \mathbf{y}(i)])\}^+ , \tag{7.6}$$

where $\mathbf{y}(i)$ is the the output of *the sigmoidally optimized WVDF* (SWVDF) scheme related to the actual weight coefficients $\psi_1(i), \psi_2(i), \ldots, \psi_N(i)$ at sample position i, $(i = 1, \ldots, Q)$ and ϵ denotes the iteration constant, (adaptation step-size), [424]. The notation $\mathbf{x}_k(i)$ describes the input sample with the k-th position in the filter window W centered in $\mathbf{x}(i) = \mathbf{x}_1$, $\Upsilon(\cdot)$ denotes the transformation

$$\Upsilon(\mathbf{u} - \mathbf{v}) = \mathcal{S} \cdot a(\mathbf{u}, \mathbf{v}), \quad \text{where} \quad \mathcal{S}(\mathbf{u}, \mathbf{v}) = \begin{cases} +1 & \text{if} \quad \|\mathbf{u}\| \geq \|\mathbf{v}\|, \\ -1 & \text{if} \quad \|\mathbf{u}\| < \|\mathbf{v}\|, \end{cases} \tag{7.7}$$

$\Xi(\cdot)$ signifies the sigmoidal function, $\{\cdot\}^+$ is a projection operation which sets the negative values to zero and $a(\mathbf{u}, \mathbf{v})$ is the angle between vectors $\mathbf{u}$, $\mathbf{v}$

$$\Xi(\chi) = \frac{2}{1 + \exp(-\chi)} - 1, \quad \{\chi\}^+ = \begin{cases} 0, & \text{if } \chi < 0, \\ \chi, & \text{otherwise.} \end{cases} \tag{7.8}$$

7.1.2 Linear Optimization

Let us now consider the generalized linear approximation of the *sign* function $\Xi(\cdot)$ in (7.8). The extension of the algorithm based on the linear approximation of the sign function from the scalar to the vector case, requires the determination of the maximum and the minimum of the vector valued input set W and also the substitution of the absolute difference between two scalar samples with the angle between two multichannel vectors.

Let the uppermost ranked sample $\mathbf{x}_{(N)}$ represents the *maximum sample* of the vector valued input set W and let the lowest ranked vector $\mathbf{x}_{(1)}$ minimizing the sum of weighted angles to other input samples represents the *minimum input sample*. Thus, the update of the weight coefficients in the adaptive WVDF scheme, based on the linear approximation of the sign function (LWVDF) can be expressed as, [177, 189]: $\psi_k(i+1) = \Big\{\psi_k(i) + 2\epsilon\Big[\Upsilon(\mathbf{x}_{(N)} - \mathbf{x}_{(1)}) +$

$$-2a(\mathbf{o}(i), \mathbf{x}_k(i))\Big] - \sum_{j=1}^{N} \psi_j[\Upsilon(\mathbf{x}_{(N)} - \mathbf{x}_{(1)}) - 2a(\mathbf{x}_k(i), \mathbf{x}_j(i))]\Big\}^+, \tag{7.9}$$

where $k, j = 1, 2, \ldots, N$, and ϵ is the positive adaptation step-size.

The restrictions of both adaptation algorithms (7.6) and (7.9) follow the WM optimization framework. The adaptation step-size ϵ should be set to a certain small value and the achieved weight coefficients cannot be negative. For that reason, the negative weights are projected to zero. The starting weight vector $\psi(1)$ in the iterative scheme of weights finding may be set to arbitrary positive values, however all weights in the starting vector should have an equivalent importance. Moreover, the proposed optimization schemes require a learning signal like in the WM optimization.

In order to adapt the WVDF weight coefficients to the signal and noise statistics, *LENA* color images contaminated with 2%, 5%, 10%, 15% and 20% impulsive noise were used as the training sets, (NM4). All filtering results were obtained with a 3×3 square window, $(N = 9)$ and the proposed SWVDF and LWVDF optimization started with the same initial weighting vector $\psi(1) = [1, 1, \ldots, 1, 1]$, which corresponds to the BVDF operation.

The achieved optimization results are shown in Figs. 7.2, 7.3 (image restoration quality measures) in dependence on the value of the iteration constant ϵ, which ranged from 10^{-5} to 10^3. In the case of the SWVDF filter, the most appropriate ϵ was found, (Fig. 7.2) to be around 0.1. For smaller ϵ the SWVDF provides worse detail preserving characteristics and after some critical point, which depends on the statistical properties of the training sequence, it converges to an operation close to that one performed by the BVDF. [1] The obtained results indicate that the performance of the WVDF based on linear approximation of the sign function, (LWVDF) decreases with increasing value of ϵ. The most appropriate value of ϵ related to the LWVDF, (Fig. 7.3) is found to be around 0.01.

Numerical results and comparisons are presented in Tab. 7.1. In this Table the componentwise MMF filter [430], standard VMF [19], BVDF [397] and DDF [138], Fuzzy Vector Directional Filter (FVDF) [240, 245], GVDF [395, 397] and two Hybrid Vector Filters (HDF and AHDF) [106] were compared in terms of performance with the optimized, (LWVDF and SWVDF) and non-optimized filters, (WVDF_1, WVDF_2) with weights:

$$\begin{array}{|c|c|c|}\hline 2 & 1 & 2 \\ \hline 1 & 3 & 1 \\ \hline 2 & 1 & 2 \\ \hline \end{array} \quad (\mathbf{WVDF}_1), \qquad \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 4 & 5 & 4 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad (\mathbf{WVDF}_2).$$

The obtained results confirm that the proposed WVDF framework can be designed to provide an **excellent trade-off** between noise attenuation and signal-detail preserving characteristics and the proposed technique **outperforms** the standard filtering schemes in terms of the commonly used objective measures.

[1] Note that for $\epsilon = 0$ the iterations do not change the initial values of ψ, and for $\psi_k = 1, k = 1, \ldots, N$ the BVDF structure is retained.

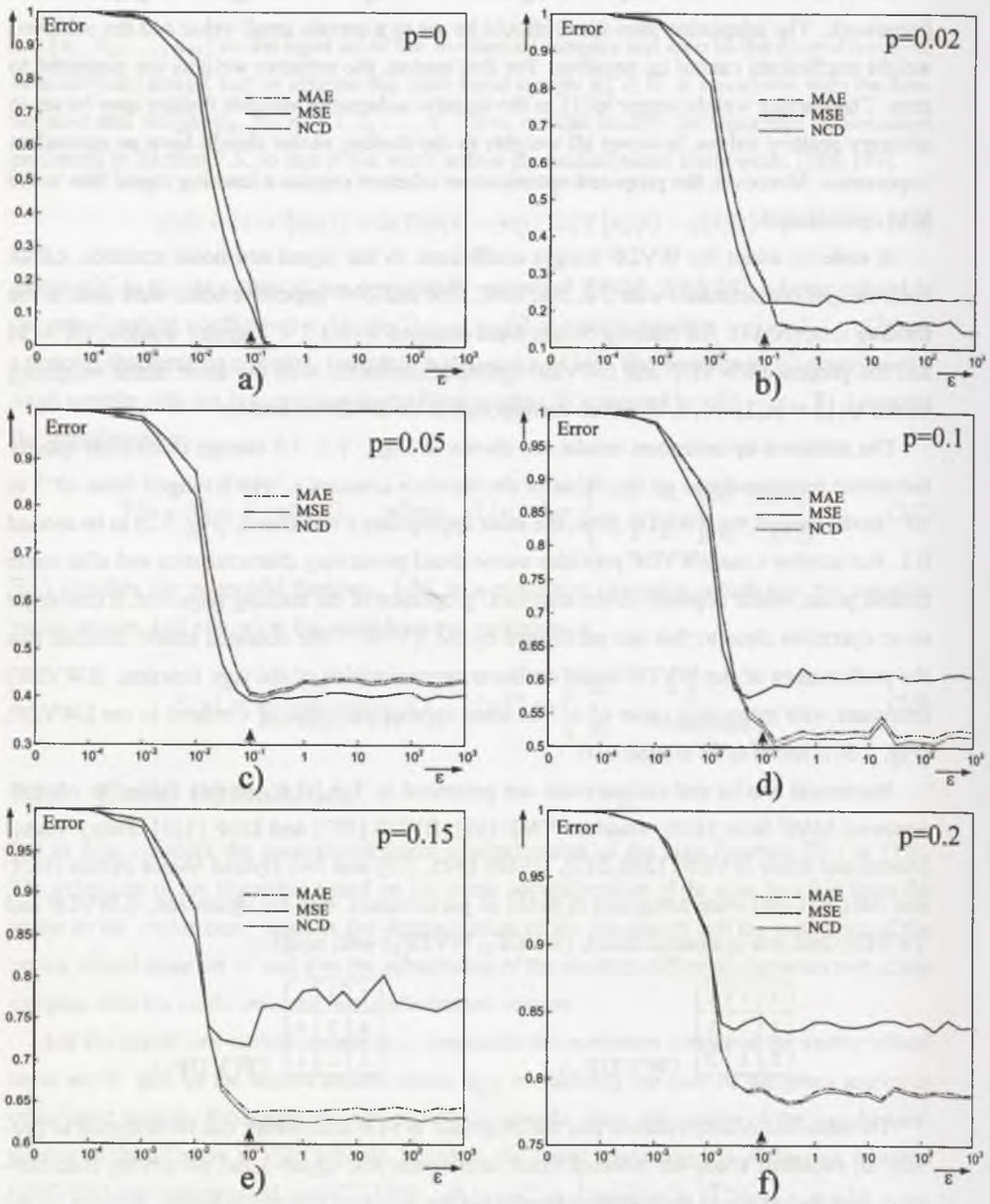


Fig. 7.2. Efficiency of the WVDF sigmoidal optimization, (SWVDF) expressed through normalized quality measures dependent on the iteration step-size ϵ. The training set was delivered by the image LENA with: a) no corruption, b) 2% impulsive noise, c) 5% impulsive noise, d) 10% impulsive noise, e) 15% impulsive noise, f) 20% impulsive noise, (NM4). Note that $\epsilon = 0$ characterizes the BVDF output

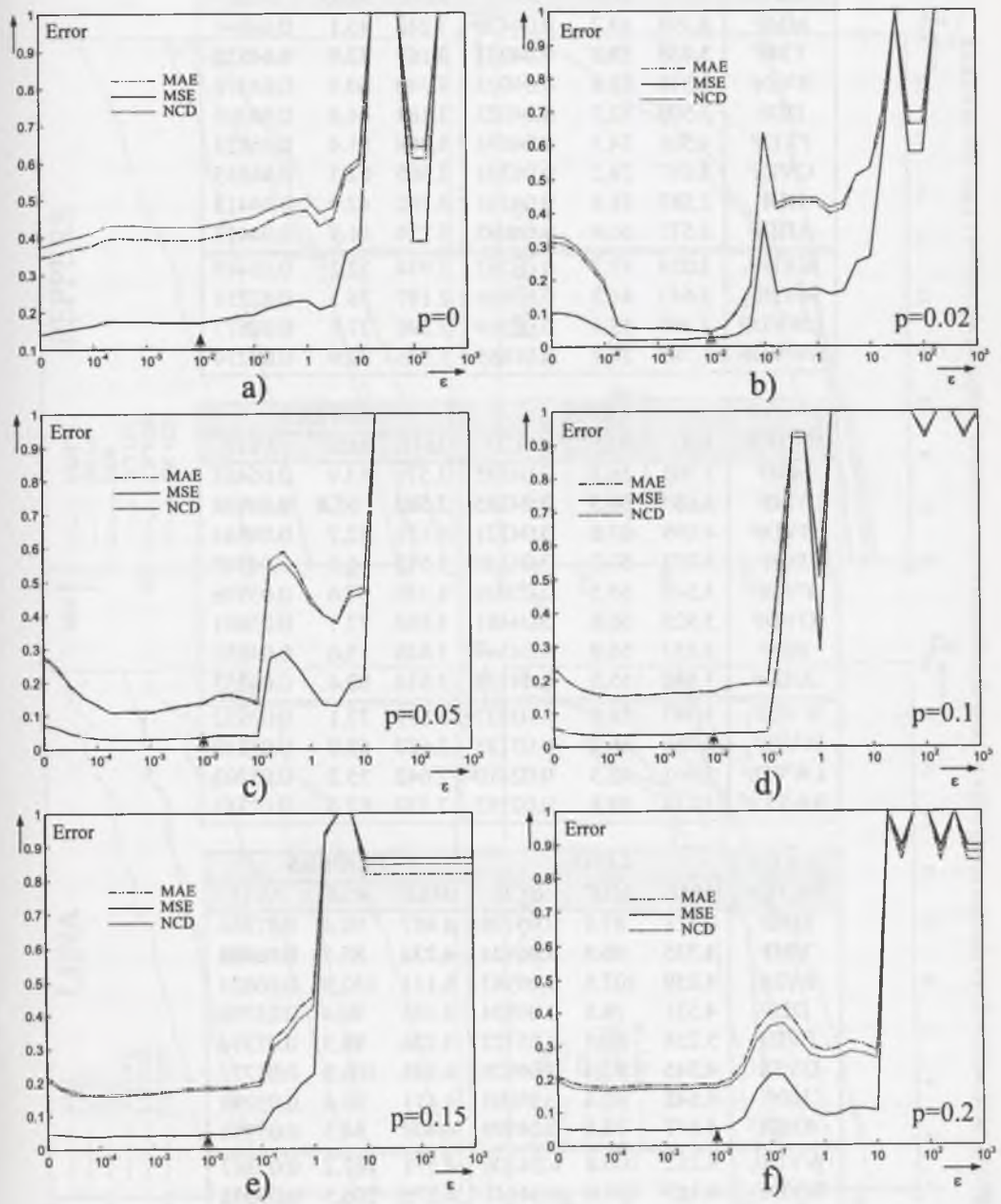


Fig. 7.3. Efficiency of the WVDF linear optimization, (LWVDF) expressed through the normalized objective quality measures dependent on the iteration step-size ϵ. The training set was obtained through the image LENA with: a) no corruption, b) 2% impulsive noise, c) 5% impulsive noise, d) 10% impulsive noise, e) 15% impulsive noise, f) 20% impulsive noise, (NM4)

$p = 0.05$	LENA			PEPPERS		
FILTER	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*
MMF	3.394	49.7	0.04420	3.248	43.1	0.04841
VMF	**3.430**	**50.8**	**0.04031**	**3.169**	**43.9**	**0.04520**
BVDF	3.818	58.6	0.04073	3.740	60.7	0.04378
DDF	3.509	52.3	0.04023	3.182	44.6	0.04309
FVDF	4.301	54.3	0.04834	4.068	51.4	0.05522
GVDF	3.697	59.2	0.04301	3.605	62.5	0.04855
HDF	3.587	51.8	0.04101	3.282	42.9	0.04413
AHDF	3.573	50.4	0.04095	3.274	41.9	0.04413
$WVDF_1$	3.054	47.7	0.03267	2.974	52.2	0.03449
$WVDF_2$	2.643	41.5	0.02826	2.197	38.1	0.02751
LWVDF	2.399	33.4	0.02569	2.296	37.6	0.02677
SWVDF	1.783	24.2	0.01885	1.876	33.9	0.02274

$p = 0.1$	LENA			PEPPERS		
FILTER	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*
MMF	3.703	56.8	0.04893	3.579	53.9	0.05463
VMF	**3.687**	**56.5**	**0.04285**	**3.503**	**55.0**	**0.04935**
BVDF	4.099	67.6	0.04321	4.151	82.7	0.04844
DDF	3.733	57.3	0.04240	3.512	56.6	0.04749
FVDF	4.540	59.5	0.05029	4.370	61.6	0.05946
GVDF	3.925	66.8	0.04481	3.862	72.7	0.05091
HDF	3.857	56.9	0.04344	3.626	53.6	0.04855
AHDF	3.840	55.5	0.04339	3.614	52.4	0.04853
$WVDF_1$	3.347	58.2	0.03537	3.399	77.1	0.03932
$WVDF_2$	2.989	56.3	0.03138	2.659	65.9	0.03249
LWVDF	2.661	42.5	0.02810	2.642	55.2	0.03103
SWVDF	2.114	39.8	0.02192	2.330	67.3	0.02745

$p = 0.2$	LENA			PEPPERS		
FILTER	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*
MMF	4.521	87.9	0.06198	4.487	91.4	0.07266
VMF	**4.335**	**80.3**	**0.04924**	**4.232**	**85.7**	**0.06008**
BVDF	4.859	107.8	0.04987	5.111	152.9	0.06024
DDF	4.321	78.8	0.04834	4.254	90.4	0.05796
FVDF	5.258	80.4	0.05722	5.226	98.3	0.07394
GVDF	4.345	83.4	0.04928	4.395	106.5	0.05771
HDF	4.548	80.4	0.05003	4.411	86.4	0.05998
AHDF	4.547	79.5	0.04999	4.409	84.5	0.05996
$WVDF_1$	4.212	106.8	0.04306	4.571	167.2	0.05317
$WVDF_2$	4.113	131.6	0.04141	4.275	206.5	0.05033
LWVDF	3.466	92.3	0.03533	3.824	148.6	0.04467
SWVDF	3.345	136.1	0.03333	4.064	234.1	0.04648

Tab. 7.1. Comparison of the LWVDF and SWVDF filters with standard techniques using LENA and PEPPERS images corrupted with impulsive noise of $p = 0.05$, $p = 0.1$ and $p = 0.2$, (NM4), [189]

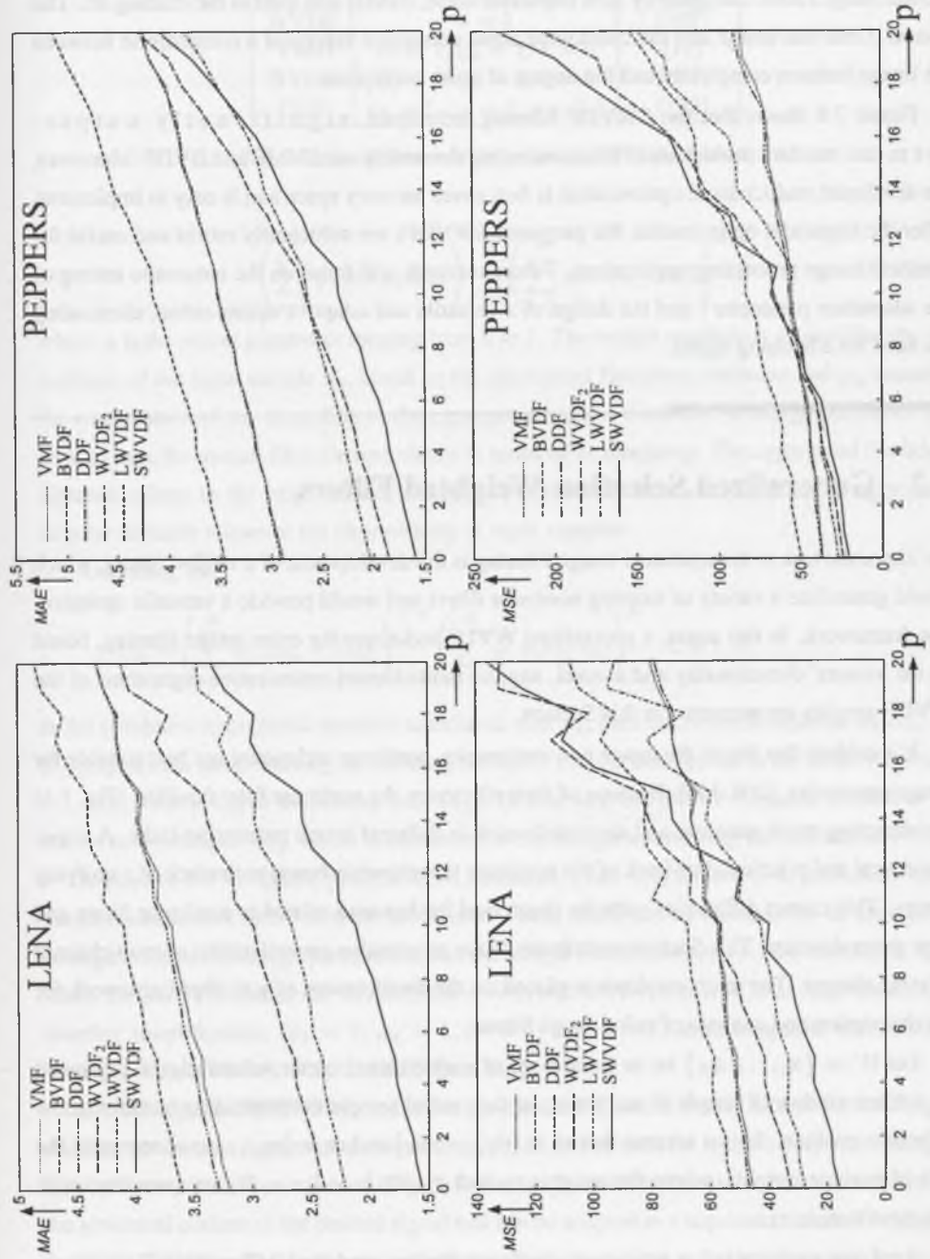


Fig. 7.4. Performance of the relevant methods for increasing impulsive noise corruption, (NM4) of test images LENA and PEPPERS

To achieve the robust weighting coefficients used for the evaluation presented in Tab. 7.1, the test image *LENA* corrupted by 10% impulsive noise, (NM4) was used as the training set. The reason is that this image and the considered noise corruption represent a compromise between the image features complexity and the degree of noise corruption.

Figure 7.4 shows that the SWVDF filtering techniques significantly outperform the standard multichannel filters including the widely used VMF and BVDF. Moreover, the developed multichannel optimization is fast, saves memory space and is easy to implement. After the sigmoidal optimization, the proposed SWVDFs are sufficiently robust and useful for practical image processing applications. Future research will focus on the automatic setting of the adaptation parameter ϵ and the design of a versatile self-adaptive optimization, eliminating the need for a learning signal.

7.2 Generalized Selection Weighted Filters

An important task in the nonlinear image filtering is the development of a unified theory, which would generalize a variety of existing nonlinear filters and would provide a versatile optimization framework. In this sense, a generalized WVDF technique for color image filtering, based on the vectors' directionality and a novel, angular multichannel optimization algorithms of the WVDF weights are presented in this Section.

It is evident that due to the image non-stationarity, nonlinear techniques are best suitable for image processing, [208,360]. Because of their efficiency, the nonlinear filter families, (Fig. 1.4) are attracting much attention and are widely used in different image processing tasks. A major theoretical and practical drawback of the nonlinear techniques is however the lack of a unifying theory. This causes difficulties with the theoretical background related to nonlinear filters and their generalization. This Section contributes to the progressive generalization of multichannel filtering classes. The main emphasis is placed on the development of a unified framework for the description and analysis of color image filters.

Let $W = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be as usual a set of multichannel vector valued samples spanned by a filter window of length N and let $\mathbf{x}_1$ be the central sample corresponding to the window reference position. Let us assume that $\boldsymbol{\psi} = [\psi_1, \ldots, \psi_N]$ and $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_N]$ represent the sets of positive weights, where the weights ψ_k and φ_k, for $k = 1, \ldots, N$, are associated with the input sample $\mathbf{x}_k$.

Applying a minimization procedure, similar to the one used for VMF or BVDF, the generalized *Selection Weighted Vector Filters* (SWVF) output is the sample $\mathbf{x}_{(1)} \in W$ minimizing:

Filter	SWVF Parameters	Reference
WVMF	$\kappa = 0$	[407]
WVDF	$\kappa = 1$	[189]
VMF	$\psi_k = 1, \varphi_k = 1, \kappa = 0$	[19]
BVDF	$\psi_k = 1, \varphi_k = 1, \kappa = 1$	[397]
DDF	$\psi_k = 1, \varphi_k = 1, \kappa = 0.5$	[138]

Tab. 7.2. Special cases of the proposed SWVF framework

$$\left(\sum_{k=1}^{N} \psi_k \, \rho(\mathbf{x}_{(1)}, \mathbf{x}_k)\right)^{1-\kappa} \left(\sum_{k=1}^{N} \varphi_k \, a(\mathbf{x}_{(1)}, \mathbf{x}_k)\right)^{\kappa}, \tag{7.10}$$

where κ is the power parameter ranging from 0 to 1. The weight coefficient ψ_k signifies the importance of the input sample $\mathbf{x}_k$, based on the aggregated Euclidean distances and φ_k measures the contribution of $\mathbf{x}_k$, according to the aggregated angular distances. A design parameter κ is used to tune the overall filter characteristics in terms of its efficiency. The aggregated Euclidean distance relates to the brightness of the vectors under consideration, whereas the aggregated angular distance relates to the chromaticity of input samples.

Assuming, that

$$D_k = \left(\sum_{j=1}^{N} \psi_j \, \|\mathbf{x}_k - \mathbf{x}_j\|_\gamma\right)^{1-\kappa} \left(\sum_{j=1}^{N} \varphi_j \, a(\mathbf{x}_k, \mathbf{x}_j)\right)^{\kappa}, \; k = 1, 2, \ldots, N, \tag{7.11}$$

is the combined aggregated measure associated with $\mathbf{x}_k$, then the ordered sequence of $D_1, \ldots, D_N$ implies the same ordering of the input set $\mathbf{x}_1, \ldots, \mathbf{x}_N$, which results in the ordered set $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(N)}$, where $\mathbf{x}_{(k)}$ is associated with $D_{(k)}$. In this way, the SWVF output is defined as the lowest order statistic $\mathbf{x}_{(1)}$, which is equivalent to the sample minimizing the expression (7.10).

A class of SWVF filters, [184,313,341] includes, (Tab. 7.2) a number of previously introduced multichannel filters as their subclasses. These filters can be obtained by an appropriate configuration of the design parameter κ and the weight coefficients $\psi_1, \ldots, \psi_N$ and $\varphi_1, \ldots, \varphi_N$. Thus, the SWVF includes the WVMF, (for $\kappa = 0$) and WVDF, (for $\kappa = 1$) as basic subclasses. Another simplification, ($\psi_k = 1$, $\varphi_k = 1$, for $k = 1, 2, \ldots, N$) leads to special cases such as VMF ($\kappa = 0$), BVDF ($\kappa = 1$) and DDF ($\kappa = 0.5$).

In this way the SWVF filters constitute a wide class of multichannel filters. Each setting of the filter parameters represents a unique filter, which can be used for specific purposes. Using an appropriate optimization scheme, (Fig. 7.1) the weight coefficients can follow the statistic and the structural content of the desired signal and can be adapted in a required manner. To simplify the SWVF optimization and to provide better illustration of the weights adaptation, let us assume the equivalence between the weight vectors $\boldsymbol{\psi}$ and $\boldsymbol{\varphi}$ so that $\psi_k = \varphi_k$, for $k = 1, 2, \ldots, N$.

Thus, we will make use of the weight coefficients $\psi_1, \psi_2, ..., \psi_N$ only. These coefficients can be adaptively determined, using the generalized multichannel sigmoidal optimization approach of the standard WM filters, [189, 423].

The development of nonlinear multichannel filters, [246] requires the determination of the distance between multichannel samples and the sample ordering based on the aggregated distances. This operation requires additionally the *polarity* of the distance measure between two multichannel samples.

Let us consider the **generalized difference** between two vectors $\mathbf{u}$ and $\mathbf{v}$, (7.7)

$$\Upsilon(\mathbf{u}-\mathbf{v}) = \mathcal{S}(\mathbf{u},\mathbf{v})\,[\rho(\mathbf{u},\mathbf{v})]^{1-\kappa}\,[a(\mathbf{u},\mathbf{v})]^{\kappa}\,, \tag{7.12}$$

where $\mathcal{S}(\cdot) \in \{-1, 1\}$ is a polarity function given by (7.7). Note that the *polarity function* introduced here, preserves the sign of the difference between the scalar image samples u and v, since for scalar case, i.e. $m = 1$ and $\kappa = 0$, the magnitude of $\mathbf{u}$ and $\mathbf{v}$ is equivalent to u and v, respectively.

Given an input set $W = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ and a weight vector $\boldsymbol{\psi} = [\psi_1, \psi_2, \ldots, \psi_N]$, we denote the SWVF output as $\mathbf{y} = \mathbf{y}(\boldsymbol{\psi}, W)$. The loss in performance (error in the filtering operation) can be defined as

$$e = |\Upsilon(\mathbf{o}-\mathbf{y})|\,. \tag{7.13}$$

One of the natural ways of choosing the weight coefficients $\psi_1, \psi_2, \ldots, \psi_N$ is to require that their choice should minimize the average cost or loss function. Therefore, the cost function of the SWVF filtering is defined as

$$J_{SWVF}(\boldsymbol{\psi}, W) = E\left\{|\Upsilon(\mathbf{o}-\mathbf{y})|\right\}\,. \tag{7.14}$$

With the constraint of non-negative weights, keeping the aggregated measure (7.11) positive, the optimization problem can be expressed as

$$\boxed{\text{minimize } J_{SWVF}(\boldsymbol{\psi})\,, \text{ with subject to } \psi_k \geq 0\,, \quad \text{for } k = 1, 2, \ldots, N\,.} \tag{7.15}$$

During the optimization procedure, the sliding filtering window is moved over the image domain and the weight coefficients ψ_k, for $k = 1, 2, \ldots, N$ are adjusted by adding the contribution of the samples multiplied by a certain regulation factor ϵ

$$\psi_k(i+1) = \left\{\psi_k(i) + 2\epsilon\Upsilon\left[\mathbf{o}(i) - \mathbf{y}(i)\right]\Xi\left(\Upsilon\left[\mathbf{x}_k(i) - \mathbf{y}(i)\right]\right)\right\}^{+}\,, \tag{7.16}$$

where $\mathbf{y}(i)$ is the the output of the sigmoidally optimized SWVF filter related to the actual weight vector $\boldsymbol{\psi}$ at image position i and $\Xi(\cdot)$ denotes the sigmoidal function.

This iterative algorithm determines the weight coefficients with respect to the filter weights obtained at the previously processed image sample. If $\Upsilon[\mathbf{o}(i) - \mathbf{y}(i)]$ is zero, then the filter holds the detail preserving properties and all weights coefficients remain unchanged. If $\Upsilon[\mathbf{x}_k(i) - \mathbf{y}(i)]$ is zero, the input sample $\mathbf{x}_k$ possesses the same noise attenuation and detail preserving capability as $\mathbf{y}(i)$ and the corresponding weight ψ_k remains also unchanged. In the rest of cases, $\Upsilon[\mathbf{o}(i) - \mathbf{y}(i)]$ and $\Upsilon[\mathbf{x}_k(i) - \mathbf{y}(i)]$ influence the weight update in terms of the trade-off between the noise smoothing and the signal-detail preservation. Note that the initial weight vector $\boldsymbol{\psi}$ can be set to arbitrary positive values, but the best choice is to start the weight adaptation with equal weights, e.g. $\psi_k = 1$, for $k = 1, 2, \ldots, N$, corresponding to the robust smoothing functions such as VMF, BVDF and DDF.

It is clear that the availability of original (training) signal $\mathbf{o}_i$, $i = 1, \ldots, Q$ in (7.16) is essential in the development of the new filter class. However, noise-free (training) samples may not be available in practical image processing applications. In such cases, the proposed scheme can be optimized using training sets available from other natural images. Upon completion of the training, the filters can be applied to real images, corrupted by an unknown noise process.

Another possibility is to replace the desired signal $\mathbf{o}(i)$ with the input central sample $\mathbf{x}(i)$

$$\psi_k(i+1) = \left\{\psi_k(i) + 2\epsilon\Upsilon[\mathbf{x}(i) - \mathbf{y}(i)]\Xi\left(\Upsilon\left[\mathbf{x}_k(i) - \mathbf{y}(i)\right]\right)\right\}^{+}\,. \tag{7.17}$$

This approach is useful, when the underlying noise probability is low and strong detail preserving characteristics of the SWVF filters are required. A different form of the SWVF scheme can be obtained if a robust and easily achieved estimate $\mathbf{y}^*$, e.g. marginal median filter (MMF) or sample average (AMF) is used instead of $\mathbf{x}(i)$, [189]

$$\psi_k(i+1) = \left\{\psi_k(i) + 2\epsilon\Upsilon\left[\mathbf{y}^*(i) - \mathbf{y}(i)\right]\Xi\left(\Upsilon\left[\mathbf{y}^*_k(i) - \mathbf{y}(i)\right]\right)\right\}^{+}\,, \tag{7.18}$$

where $\mathbf{y}^* = (y_1^*, y_2^*, ..., y_N^*)$ is the MF of the input set W, $y_j^* = MED\{x_{1j}, x_{2j}, ..., x_{Nj}\}$.

Fig. 7.5 shows the adaptation capability of the proposed SWVF scheme (7.16) started with the initial weighting vector $\boldsymbol{\psi} = [1, 1, \ldots, 1, 1]$. These results are obtained using for training the test image *LENA* corrupted by 5% and 10% impulsive noise, (NM4). Objective criteria like MAE, MSE and NCD are expressed in dependence on the regularization factor ϵ, which ranged from 10^{-12} to 10^3 and the design parameter κ. The obtained results indicate that the SWVF adaptation, as expected, depends strongly on ϵ. For very small values of ϵ, the SWVF provides worse detail preserving characteristics and performs the smoothing operation similar to the DDF. For evaluation purposes the ϵ value equal to 0.1 was taken and the obtained weighting coefficients are shown in Tab. 7.3. It can be observed that the κ value has small influence on the weights and that the weighting coefficients are approaching 1 for increasing noise intensity, which means that the schemes converge to BDF, VMF or DDF depending on κ settings.

The performance of the proposed methods, namely $SWVF_1$ defined by (7.16) and trained using the *LENA* image contaminated by noise process NM4, with $p = 0.1$, (see Fig. 7.6) and self-adaptive $SWVF_2$ given by (7.18), is compared in Tab. 7.4, using the weights presented in Tab. 7.3, with the standard filtering methods such as MMF, VMF, BVDF, DDF, HDF and AHDF, (Tab. 7.4).

It can be observed that the standard filtering techniques such as VMF and MMF suppress well impulses present in the image, however their excessive smoothing capability results in edge blurring. Since the DDF combines the properties of both VMF and BVDF, it can achieve better results than the BVDF and VMF.

The comparison of the results presented in Tab.7.4 shows that the SWVF framework can be designed to *outperform* the standard filtering schemes in terms of the quality criteria and significantly improves the performance of the multichannel filters based on the directional processing. As can be derived from Tab. 7.4, the $SWVF_1$ scheme given by (7.16) provides significantly better results, which justifies the usage of standard images artificially corrupted with impulsive noise for training purposes. It can be observed, (Fig. 7.6) that there is only a slight dependence of the quality criteria values on the contamination intensity of the LENA image used for the training purposes, which is a great advantage of the proposed scheme.

The $SWVF_2$ defined by (7.18), although self-adaptive, also outperforms the generic techniques, like VMF, BDV and DDF, which shows that even without the training, the proposed scheme yields better results than the traditional methods. The only drawback is the increased computational load associated with the iterative search for the optimal weighting coefficients.

	$p = 0.05$			$p = 0.1$			$p = 0.15$			$p = 0.2$		
	0.11	0.35	0.12	0.10	0.40	0.19	0.17	0.47	0.17	0.18	0.48	0.30
$\kappa = 0$	0.15	1.00	0.18	0.18	1.00	0.17	0.23	1.00	0.20	0.28	1.00	0.25
	0.08	0.38	0.12	0.18	0.39	0.14	0.20	0.47	0.15	0.27	0.48	0.20
	0.16	0.46	0.15	0.12	0.52	0.14	0.22	0.58	0.16	0.24	0.61	0.33
$\kappa = \frac{1}{2}$	0.23	1.00	0.26	0.21	1.00	0.25	0.29	1.00	0.21	0.37	1.00	0.31
	0.10	0.50	0.20	0.15	0.49	0.18	0.20	0.56	0.17	0.25	0.63	0.26
	0.11	0.48	0.25	0.19	0.46	0.27	0.25	0.60	0.31	0.37	0.74	0.56
$\kappa = 1$	0.22	1.00	0.23	0.24	1.00	0.29	0.35	1.00	0.30	0.47	1.00	0.46
	0.23	0.45	0.12	0.18	0.56	0.16	0.31	0.57	0.26	0.42	0.70	0.44

Tab. 7.3. SWVF weights obtained using (7.16) in dependence on noise contamination p, (NM4) and κ

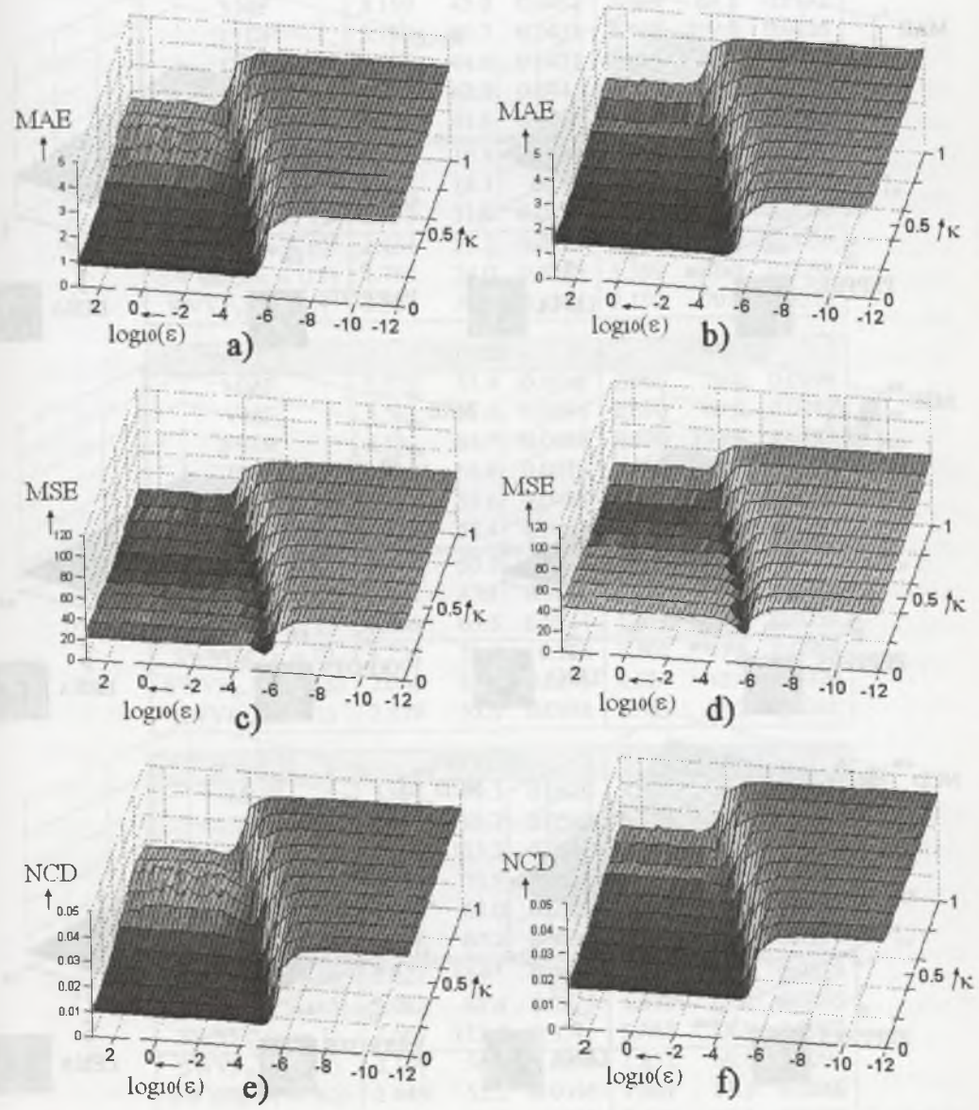


Fig. 7.5. SWVF weight adaptation expressed through the objective image quality measures in dependence on the regularization factor ε and parameter κ. The training set was obtained through the LENA image with: (a, c, e) 5% impulsive noise ($p = 0.05$, NM4) and (b, d, f) 10% impulsive noise ($p = 0.10$, NM4)

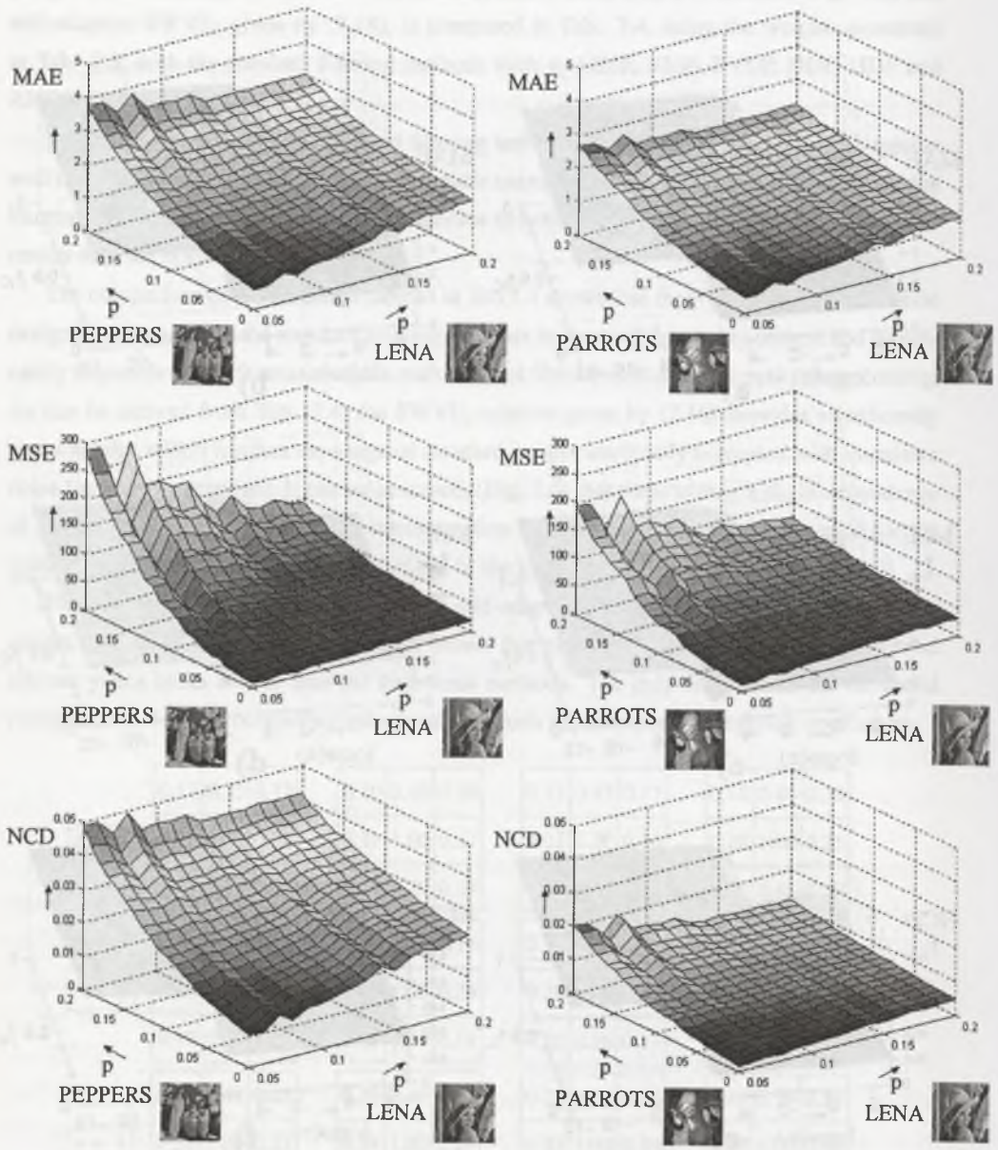


Fig. 7.6. Robustness of the $SWVF_1$ scheme in terms of MAE, MSE and NCD for test images PEPPERS and PARROTS. The adaptive scheme (7.16) was performed using the LENA image as a training set. The images were contaminated impulsive noise, (NM4, p ranging from 0 to 0.2). Note the slight dependence of the quality criteria on the contamination intensity of the training image

$p = 0.05$	PEPPERS			PARROTS		
Filter/Criterion	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*
MMF	3.248	43.1	0.0484	2.718	63.1	0.0170
VMF	3.169	43.9	0.0452	2.669	64.2	0.0132
BVDF	3.740	60.7	0.0438	3.460	109.0	0.0116
DDF	3.182	44.6	0.0431	2.645	65.3	0.0117
HDF	3.282	42.9	0.0441	2.786	65.7	0.0122
AHDF	3.274	41.9	0.0441	2.771	63.5	0.0121
$SWVF_1$, ($\kappa = 0$)	0.995	19.9	0.0138	0.903	27.1	0.0042
$SWVF_1$, ($\kappa = 0.5$)	0.962	18.1	0.142	0.745	18.5	0.0033
$SWVF_1$, ($\kappa = 1$)	1.595	31.0	0.0193	1.373	43.2	0.0046
$SWVF_2$, ($\kappa = 0$)	1.454	21.2	0.0204	1.256	30.9	0.0056
$SWVF_2$, ($\kappa = 0.5$)	1.783	24.0	0.0255	1.399	36.0	0.0058
$SWVF_2$, ($\kappa = 1$)	2.522	39.3	0.0295	2.199	70.0	0.0070

$p = 0.1$	PEPPERS			PARROTS		
MMF	3.579	53.9	0.0546	2.960	70.0	0.0198
VMF	3.503	55.0	0.0494	2.890	69.6	0.0142
BVDF	4.151	82.7	0.0484	3.630	113.5	0.0127
DDF	3.512	56.6	0.0475	2.839	69.7	0.0128
HDF	3.626	53.6	0.0485	3.002	69.9	0.0132
AHDF	3.614	52.4	0.0485	2.999	68.6	0.0131
$SWVF_1$, ($\kappa = 0$)	1.460	50.7	0.0203	1.267	47.2	0.0067
$SWVF_1$, ($\kappa = 0.5$)	1.381	43.1	0.0196	1.021	29.8	0.0049
$SWVF_1$, ($\kappa = 1$)	2.068	65.5	0.0244	1.611	53.6	0.0058
$SWVF_2$, ($\kappa = 0$)	1.754	33.3	0.250	1.501	41.6	0.0069
$SWVF_2$, ($\kappa = 0.5$)	2.068	35.1	0.0295	1.624	43.4	0.0070
$SWVF_2$, ($\kappa = 1$)	2.879	58.5	0.0338	2.385	75.2	0.0081

$p = 0.15$	PEPPERS			PARROTS		
MMF	3.996	70.3	0.0620	3.275	80.9	0.0236
VMF	3.858	68.7	0.0540	3.178	80.0	0.0158
BVDF	4.598	113.2	0.0532	3.883	125.2	0.0144
DDF	3.844	70.8	0.0518	3.070	76.7	0.0143
HDF	3.992	68.0	0.0530	2.786	65.7	0.0122
AHDF	3.994	67.2	0.0530	2.771	63.5	0.0121
$SWVF_1$, ($\kappa = 0$)	2.221	114.1	0.0299	1.942	104.1	0.0129
$SWVF_1$, ($\kappa = 0.5$)	2.088	88.4	0.0275	1.539	67.2	0.0086
$SWVF_1$, ($\kappa = 1$)	2.667	113.0	0.0311	2.065	88.9	0.0087
$SWVF_2$, ($\kappa = 0$)	2.189	56.1	0.0309	1.857	60.9	0.0095
$SWVF_2$, ($\kappa = 0.5$)	2.448	55.1	0.0346	1.881	53.3	0.0086
$SWVF_2$, ($\kappa = 1$)	3.346	93.3	0.0390	2.643	89.5	0.0099

Tab. 7.4. Comparison of the efficiency of the SWVF techniques: SWF_1 (7.16) and SWF_2 (7.18), with the standard filters using color test images PEPPERS and PARROTS contaminated by impulsive noise of intensity $p = 0.05$, $p = 0.1$ and $p = 0.15$, (NM4)

7.3 Sigma Vector Median Filters

It is commonly known that the standard techniques capable of low-pass filtering and operating on a fixed supporting window blur and eliminate salient image features. To keep the noise-free samples unchanged during the filtering operation, noise reduction techniques have to be constructed, to increase the degree of freedom in the filter design, by introducing tuning parameters into its structure, such as in the case of multichannel weighted filtering schemes, [177,189,407].

Another way, is to incorporate the structural information to the filter design realized by adaptively changing the direction of the filter operating sub-window, [129,130] or to deal with the image samples of similar intensities, which form digital paths on the image domain, [72, 295,379]. In the case of impulsive noise environments, the most popular and computationally efficient approaches are related to the switching-based filtering, [1,2,29]. In such a scheme, the switching rule changes between the nonlinear mode, which smoothes out noisy samples and the identity operation, which leaves the uncorrupted samples unchanged.

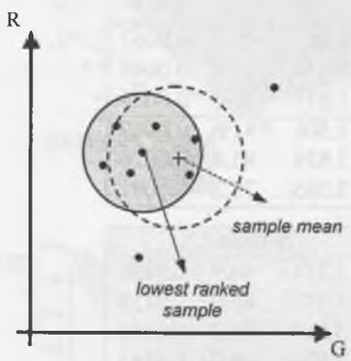


Fig. 7.7. The concept of the sigma filtering in the 2-dimensional case, in which the radius of the circles is related to the variance of the samples multiplied by the adjusting parameter θ

Sigma Vector Median Filter (SVMF) takes advantages of the switching-based filtering and can be seen as an adaptive extension of the *Rank Conditioned Vector Median Filter* (RCVMF) presented in Section 3.2, (3.18). In addition to this concept, the introduced adjusting parameter, that allows to detect the noisy samples, extends the degree of freedom of the novel multichannel filter, [360]. This filtering approach is useful for detection and removal of impulsive noise in a wide range of applications, in which the preservation of the desired structures and color information is of importance.

The switching based filtering is related mostly to the gray scale imaging, [42,68,96,166,429]. The extension of these algorithms to color images may be problematic especially in terms of flexibility to accommodate the algorithms for a variety of window shapes, [68,429] computational complexity, [429] or the number of switching levels, [96].

The proposed SVMF method is based on the robust order statistic theory and on the approximation of the multivariate dispersion computed using the input multichannel samples. Its unique and distinguishing element is the statistical operator servicing as the control of switching

between the robust VMF and the identity operation. The input central sample is considered to be noisy if it lies outside the range, (Fig. 7.7) formed by the approximated multivariate dispersion of the input multichannel samples. To increase the degree of freedom in such a design, the proposed method utilizes approximation of the multivariate dispersion multiplied by a certain regulation parameter θ. Note that a similar concept was applied to gray scale images, [135,166] and the filter output was defined as a weighted mean of the input samples lying within the standard deviation of the central pixel value, (2.14).

The measure of the multivariate samples' dispersion is very often defined using the variance-covariance matrix Σ of the samples $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ defined as $\Sigma = E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T]$, where E is the expected value operator and $\bar{\mathbf{x}}$ denotes the arithmetic mean of the vector samples. The dispersion matrix Σ is square, symmetric and usually of full rank.

In many applications it is very useful to use a scalar value capturing the multivariate data dispersion. One of the ways of introducing such a scalar measure is the so called generalized variance $|\Sigma|$ defined as the determinant of the Σ matrix, which can be calculated as the product of the eigenvalues of Σ, [417]. The idea is to measure the volume occupied by the multivariate variables in the color space. The multivariate dispersion can also be given as a sum of the eigenvalues of the variance-covariance matrix, (total variance), [281]. The former plays an important role in the maximum likelihood estimation and model selection and the latter is used as a measure of variation in principal components analysis, [10,205,212,289]. These dispersion measures well describe the samples' variability, but their drawback is that they are computationally very expensive and thus inappropriate for image processing. That is why, simple but effective dispersion measures based on the samples mean and vector median, are used in this work: multivariate variance measured from the samples mean and variance measured from the samples median, [289].

To avoid the computational difficulties connected with the calculation of variance-covariance matrices of multichannel samples, the proposed method utilizes the approximation of the variance of the vector data, [197,360]. Let ν be the approximation of the multivariate variance of the vectors contained in a supporting window W of size N, given by

$$\nu = \frac{D_{(1)}}{N-1}, \tag{7.19}$$

where $D_{(1)}$ is the distance measure minimizing the generalized distance, (3.21)

$$\left(\sum_{j=1}^{N} a(\mathbf{x}_k, \mathbf{x}_j)\right)^{\kappa} \left(\sum_{j=1}^{N} \rho(\mathbf{x}_k, \mathbf{x}_j)\right)^{1-\kappa}, \tag{7.20}$$

i.e. the measure associated with the VMF for $\kappa = 0$, (3.3) and with the BVDF for $\kappa = 1$, (3.19).

Let us now assume that $\kappa = 0$, then

$$D_{(1)} = R_{(1)} = \sum_{j=1}^{N} \|\mathbf{x}_{(1)} - \mathbf{x}_j\|_\gamma, \tag{7.21}$$

where $\mathbf{x}_{(1)}$ is the output of the VMF. The local variance approximation

$$\nu = \left(\sum_{j=1}^{N} \|\mathbf{x}_{(1)} - \mathbf{x}_j\|_\gamma\right) \Big/ (N-1), \tag{7.22}$$

represents the mean distance between the vector median $\mathbf{x}_{(1)}$ and all other samples contained in W. The division of the smallest aggregated distance $R_{(1)}$ by $(N-1)$, (number of distances from $\mathbf{x}_{(1)}$ to all samples from W), ensures that the dispersion measure is non-dependent on the filtering window size. Then, the output of the *Sigma Vector Median Filter*, (SVMF) is defined as, [360, 361]

$$\mathbf{y}_{SVMF_1} = \begin{cases} \mathbf{x}_{(1)}, & \text{for } R_1 \geq \xi_1, \\ \mathbf{x}_1, & \text{otherwise}, \end{cases} \tag{7.23}$$

where $\mathbf{y}_{SVMF_1}$ is the proposed SVMF output, R_1 is the cumulated distance from the central pixel $\mathbf{x}_1$ to all other pixels contained in W and ξ_1 is the threshold value given by

$$\xi_1 = R_{(1)} + \theta_1 \nu = \frac{N-1+\theta_1}{N-1} R_{(1)}, \tag{7.24}$$

where ν is the approximated variance (7.19) and θ_1 is the tuning parameter used to adjust the smoothing properties of the proposed SVMF$_1$ method.

The switching scheme (7.23) can be rewritten as

$$\mathbf{y}_{SVMF_1} = \begin{cases} \mathbf{x}_{(1)}, & \text{for } R_1 \geq \frac{N-1+\theta_1}{N-1} R_{(1)}, \\ \mathbf{x}_1, & \text{otherwise}. \end{cases} \tag{7.25}$$

If the distance measure R_1 of the central sample $\mathbf{x}_1$ is greater or equal to the threshold ξ_1, then the central sample is most probably noisy and is being replaced with the lowest ranked vector $\mathbf{x}_{(1)}$, (Fig. 7.7). If the accumulated distance R_1 of the central sample $\mathbf{x}_1$ is less than the threshold ξ_1, then the central sample is declared to be similar to other input samples, which indicates that it is most probably noise-free and no filtering operation is performed.

In order to follow both concepts sketched in Fig. 7.7, it is possible to modify the decision stage and to replace the lowest ranked vector with the sample mean, which leads to a much faster algorithm, as instead of the calculation of 36 distances needed for VMF, only 9 distances between the samples and their mean are required . Then, the approximation of the variance is given by

$$\nu_{\bar{\mathbf{x}}} = \frac{R_{\bar{\mathbf{x}}}}{N}, \quad \text{where} \quad R_{\bar{\mathbf{x}}} = \sum_{j=1}^{N} \|\bar{\mathbf{x}} - \mathbf{x}_j\|_\gamma, \tag{7.26}$$

is the aggregated distance between the multichannel input samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ and the sample mean $\bar{\mathbf{x}}$. In such case, the output of the modified vector sigma filter (SVMF$_2$) is defined as

$$\mathbf{y}_{SVMF_2} = \begin{cases} \mathbf{x}_{(1)}, & \text{for } R_1 \geq \xi_2, \\ \mathbf{x}_1, & \text{otherwise}, \end{cases} \qquad \xi_2 = R_{\bar{\mathbf{x}}} + \theta_2 \nu_2 = \frac{N+\theta_2}{N} R_{\bar{\mathbf{x}}}, \tag{7.27}$$

where ξ_2 is the threshold value and θ_2 is the adjusting parameter like in the SVMF$_1$ scheme.

It is clear that the proposed method will perform the identity operation for any value of θ_1, if the lowest ranked vector $\mathbf{x}_{(1)}$ is identical with the central sample $\mathbf{x}_1$. If $\mathbf{x}_{(1)} \neq \mathbf{x}_1$ then the SVMF$_1$ output is a root, if

$$\theta_1 \geq \theta_1^* = \left(\frac{R_{(N)}}{R_{(1)}} - 1\right)(N-1), \tag{7.28}$$

which means that an additional increasing of θ does not influence the filter properties.

It is worth noticing that the filtering scheme is scale and bias invariant. Let us consider the input set $W_1 = \{{}^1\mathbf{x}_j, j = 1, \ldots, N\}$ and the modified input set $W_2 = \{{}^2\mathbf{x}_j, j = 1, \ldots, N\}$ achieved by adding the vector constant $\mathbf{k}$ to the input set W_1 multiplied by scalar constant k, i.e. ${}^2\mathbf{x}_i = k \cdot {}^1\mathbf{x}_i + \mathbf{k}$. It can be easily shown that the addition of a vector constant has no influence on the filter properties, since vector distances $R_1^{W_1}, R_2^{W_1}, \ldots, R_N^{W_1}$ are the same as $R_1^{W_2}, R_2^{W_2}, \ldots, R_N^{W_2}$. The multiplication of the input set by a constant, has also no influence on the switching condition, since

$$R_1^{W_2} = k R_1^{W_1}, \quad R_{(1)}^{W_2} = k R_{(1)}^{W_1}, \tag{7.29}$$

and then the conditions

$$R_1^{W_1} \geq \frac{N-1+\theta_1}{N-1} R_{(1)}^{W_1}, \quad R_1^{W_2} \geq \frac{N-1+\theta_1}{N-1} R_{(1)}^{W_2}, \tag{7.30}$$

are equivalent to (7.23). Thus, the decision stage of the proposed method is scale and bias non-dependent.

The proposed *Sigma Vector Median Filters* are computationally efficient, since they perform practically the same set of operations as their non-adaptive special case VMF, [19, 37]. The comparison of the construction of the SVMF$_1$ and the VMF techniques shows that both schemes need to compute the aggregated distances and search for their minimum. The switching rule requires division, multiplication and addition, however in the case of noise-free samples, no additional processing is necessary. If an outlier is detected, the reminder of operations is the same as in the case of standard VMF. In the case of the SVMF$_2$ scheme, the computational efficiency is more advantageous, since in the case of noisy samples this filter does not have to perform the time consuming ordering operations performed both in the SVMF$_1$ and VMF schemes.

Using (7.19) and (7.20) and taking $\kappa = 0.5$ and $\kappa = 1$ the *Sigma Directional Distance Filter* (SDDF) and *Sigma Basic Vector Directional Filter* (SBVDF) are obtained. The efficiency of the schemes, based on the data dispersion measured from the sample minimizing the accumulated sum of appropriate distances or from the samples centroid is presented in Tabs. 7.5, 7.6 and in Fig. 7.8. As can be observed the parameter κ in (7.20) has small influence on the filters' efficiency and therefore $\kappa = 0$ or $\kappa = 1$ should be used for erasing impulsive pixels.

The dispersion measure based on (7.19) is as expected more robust to the impulsive noise corruption than the measure of data variation measured from the centroid of samples, (7.26). However, although the filters based on (7.26) are inferior to the filters using (7.19), the $SVMF_2$ schemes are extremely fast and are much better suitable for the impulsive noise reduction than the traditional filters like VMF, BVDF, DDF, HDF, (see Tab. 7.6).

The comparison of the results obtained with both proposed methods, shows that the sub-optimal value of θ_2 used in the $SVMF_2$ scheme is larger than that θ_1 of the $SVMF_1$ approach. This observation is also confirmed by Fig. 7.9, which depicts the results achieved using the test image *LENA* corrupted by impulsive noise with the intensity ranging from $p = 0.01$ to $p = 0.20$, (NM4).

As can be easily observed, the optimal parameters θ_1 and θ_2 are decreasing with the amount of corrupted pixels, (Figs. 7.5, 7.8, 7.9) as more and more pixels have to be replaced with the VMF, DDF or BVDF according to the type of dispersion model. This indicates that some kind of more advanced adaptive design is needed to automatically adjust the θ parameter to the noise corruption process.

The efficiency assessment of the described filter class provided using the objective quality measures, (Tabs. 7.5, 7.6) and also evaluated visually, (Figs. 7.10, 7.11) confirm the good performance of the presented filtering techniques and their usefulness for the impulsive noise removal in color images.

Noise (p)	0.05			0.10			0.15		
Criterion	MAE	RMSE	NCD	MAE	RMSE	NCD	MAE	RMSE	NCD
Noisy	3.18	22.22	0.04158	6.32	31.28	0.08256	9.58	38.53	0.12408
VMF	3.29	5.63	0.03881	3.44	5.87	0.04011	3.59	6.17	0.04145
BVDF	3.82	6.35	0.04006	3.95	6.64	0.04115	4.09	6.99	0.04236
DDF	3.41	5.77	0.03906	3.53	5.99	0.04012	3.70	6.26	0.04136
HDF	3.45	5.64	0.04007	3.58	5.87	0.04125	3.74	6.18	0.04259
SVMF, $\theta = 1$	1.22	**3.94**	0.01374	1.16	**4.14**	**0.01357**	**1.23**	**4.74**	**0.01539**
SVMF, $\theta = 4$	**0.41**	4.06	**0.00611**	**0.90**	7.34	0.01477	1.67	11.32	0.02869
SVMF, $\theta = 8$	0.62	7.67	0.01031	1.77	14.05	0.02886	3.48	20.68	0.05604

Tab. 7.5. SVMF filtering results, (LENA, $p = 0.05$, $p = 0.1$ $p = 0.15$, NM2)

$p = 0.05$	LENA			PEPPERS			PARROTS		
FILTER	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*
VMF	3.430	50.8	0.0403	3.169	43.9	0.0452	2.669	64.2	0.013
BVDF	3.818	58.6	0.0407	3.740	60.7	0.0438	3.460	109.0	0.0116
DDF	3.509	52.3	0.0402	3.182	44.6	0.0431	2.645	65.3	0.0117
FVDF	4.301	54.3	0.0483	4.068	51.4	0.0552	3.802	94.5	0.0147
GVDF	3.587	55.3	0.0420	3.433	57.9	0.0453	3.036	93.6	0.0126
HDF	3.857	56.9	0.0434	3.282	42.9	0.0441	2.786	65.7	0.0122
$SVMF_1$	0.777	18.3	0.0082	0.729	16.5	0.0090	0.699	27.8	0.0027
$SVMF_2$	0.980	21.4	0.0103	0.878	18.2	0.0107	0.840	31.1	0.0031
$SBVDF_1$	0.805	19.1	0.0089	0.789	22.7	0.0113	0.694	34.3	0.0026
$SBVDF_2$	1.054	27.6	0.0115	0.987	34.4	0.0129	0.875	44.8	0.0032
$SDDF_1$	0.731	16.4	0.0080	0.649	14.6	0.0096	0.545	21.2	0.0024
$SDDF_2$	0.948	19.9	0.0105	0.816	17.9	0.0111	0.678	25.9	0.0027

$p = 0.1$	LENA			PEPPERS			PARROTS		
FILTER	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*
VMF	3.687	56.5	0.0428	3.503	55.0	0.0494	2.890	69.6	0.0142
BVDF	4.099	67.6	0.0432	4.151	82.7	0.0484	3.630	113.5	0.0127
DDF	3.733	57.3	0.0424	3.512	56.6	0.0475	2.839	69.7	0.0128
FVDF	4.540	59.5	0.0503	4.370	61.6	0.0592	3.984	98.1	0.0155
GVDF	3.925	66.8	0.0448	3.785	73.4	0.0492	3.188	96.2	0.0137
HDF	3.857	56.9	0.0434	3.626	53.6	0.0486	3.002	69.9	0.0132
$SVMF_1$	0.959	25.9	0.0105	0.941	27.4	0.0117	0.862	35.4	0.0041
$SVMF_2$	1.123	28.3	0.0121	1.063	29.0	0.0133	1.016	40.6	0.0047
$SBVDF_1$	1.048	33.1	0.0105	1.155	56.7	0.0135	0.941	47.4	0.0035
$SBVDF_2$	1.311	48.6	0.0131	1.533	99.1	0.0174	1.129	67.1	0.0045
$SDDF_1$	0.913	23.3	0.0098	0.895	30.2	0.0117	0.703	25.9	0.0030
$SDDF_2$	1.094	28.3	0.0118	1.103	44.0	0.0142	0.843	33.5	0.0037

$p = 0.2$	LENA			PEPPERS			PARROTS		
FILTER	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*	*MAE*	*MSE*	*NCD*
VMF	4.335	80.3	0.0492	4.232	85.7	0.0601	3.448	91.9	0.0174
BVDF	4.859	107.8	0.0499	5.111	152.9	0.0602	4.183	140.0	0.0165
DDF	4.321	78.8	0.0483	4.254	90.4	0.0579	3.386	91.2	0.0161
FVDF	5.258	80.4	0.0572	5.226	98.3	0.0739	4.016	118.1	0.0175
GVDF	4.345	83.4	0.0493	4.562	122.4	0.0586	3.450	100.9	0.0174
HDF	4.548	80.4	0.0500	4.411	86.4	0.0599	3.594	92.7	0.0169
$SVMF_1$	1.816	77.6	0.0212	1.898	97.3	0.0251	1.618	90.0	0.0116
$SVMF_2$	1.928	75.4	0.0221	1.995	94.2	0.0266	1.803	96.3	0.0128
$SBVDF_1$	2.232	122.9	0.0203	2.676	199.8	0.0275	1.907	126.3	0.0092
$SBVDF_2$	2.708	171.0	0.0245	3.638	332.1	0.0385	2.568	196.1	0.0128
$SDDF_1$	1.803	77.5	0.0192	1.953	109.9	0.0239	1.417	74.9	0.0087
$SDDF_2$	2.034	91.3	0.0212	2.389	151.6	0.0289	1.672	91.6	0.0101

Tab. 7.6. Comparison of the sigma filters using the LENA, PEPPERS and PARROTS color images corrupted by impulsive noise of intensity $p = 0.05$, $p = 0.1$ and $p = 0.2$, (NM4)

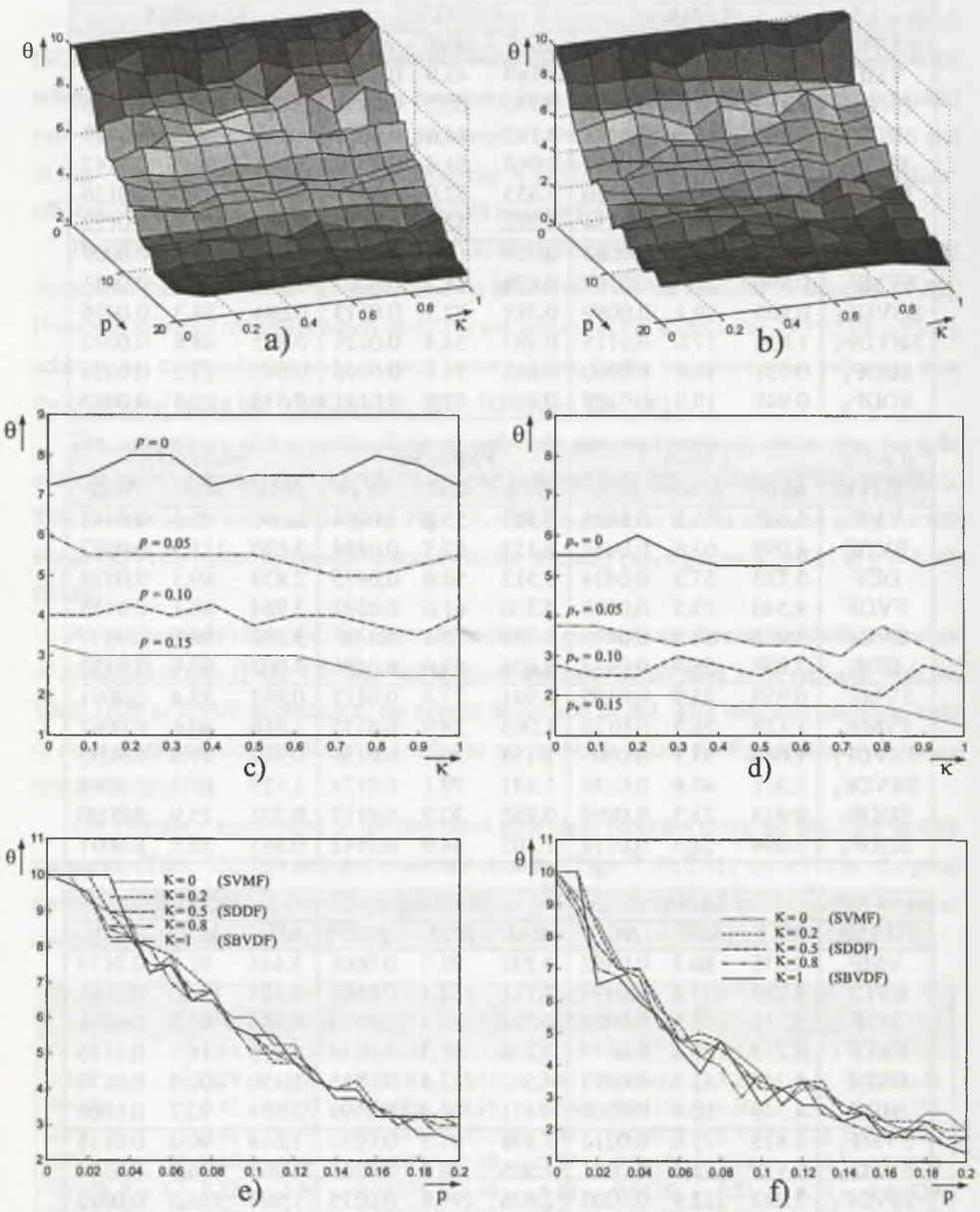


Fig. 7.8. Dependence of the objective quality measures of the SDDF on the tuning parameters θ, κ and impulsive noise intensity p, (NM4) for the LENA image: (a) MAE, (b) MSE. Below the details of achieved results in dependence on impulsive noise probability p, (NM4) according to : (c) MAE, (d) MSE are shown. At the bottom the dependence of the optimal θ value on the noise intensity for various filter classes is depicted in terms of MAE: (e) and MSE: (f)

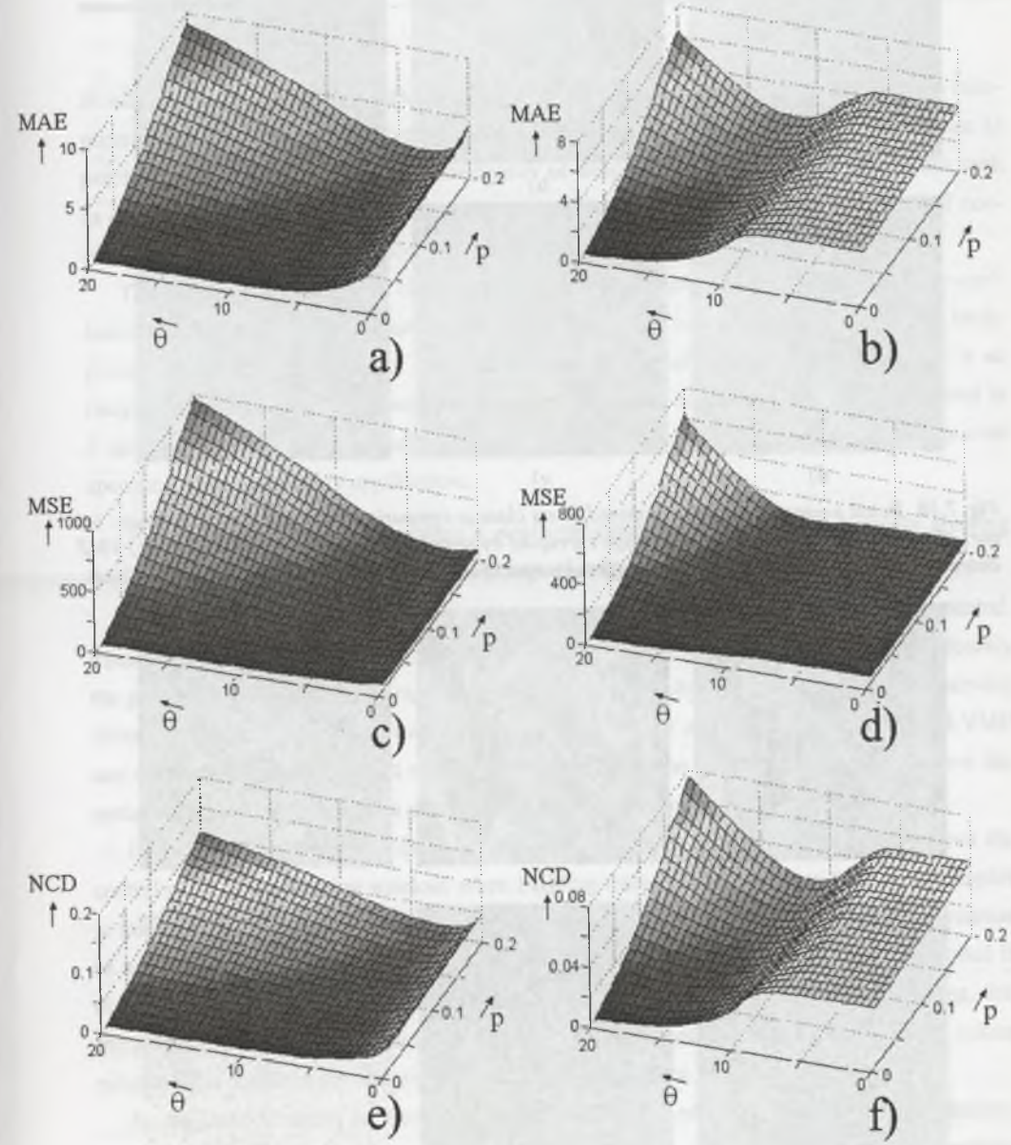


Fig. 7.9. Performance of $SVMF_1$ and $SVMF_2$ techniques in dependence on adjusting parameters θ_1, θ_2 and impulsive noise intensity p, (NM4) using the test image LENA: (a, c, e) $SVMF_1$ and (b, d, f) $SVMF_2$

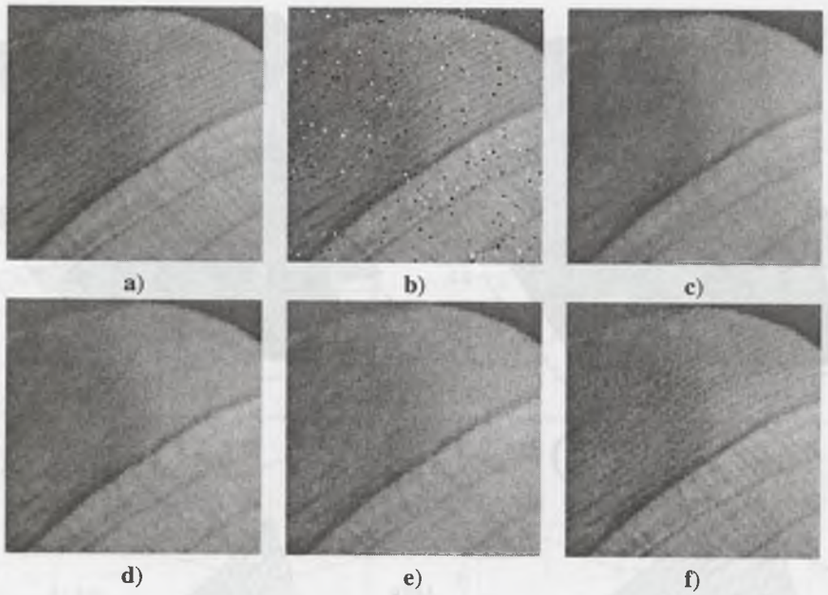


Fig. 7.10. Detail preservation of the proposed filters class in comparison with standard techniques: (a) part of original test image LENA, (b) image corrupted by impulsive noise, ($p = 0.04$, NM2), (c) VMF output, (d) DDF output, (e) BVDF output, (f) the output of the SVMF for $\theta = 6$, [360, 361]

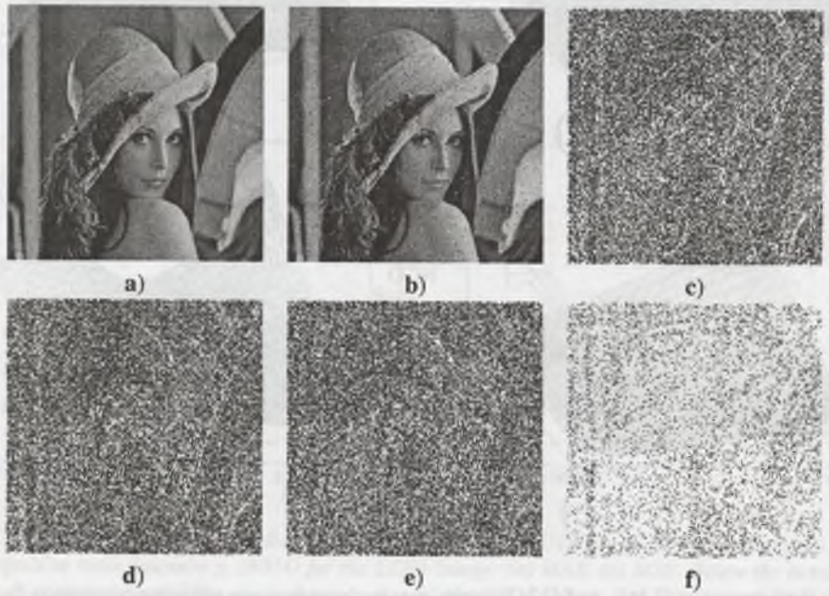


Fig. 7.11. Filtering errors expressed as the difference between the original and filtered image: (a) test image LENA, (b) image corrupted with impulse noise, ($p = 0.04$, NM2), (c) VMF filtering error, (d) DDF filtering error, (e) BVDF filtering error, (f) filtering error of the SVMF with $\theta = 6$, [360, 361]

Summary

Nonlinear image processing methods continue to grow in popularity and the advances in computing performance have accelerated the process of moving from theoretical explorations to practical implementations. The nonstationarity of images, the significance of visual cues such as edges and the nonlinearity of human visual system, all contribute to the importance of *nonlinear methods* in imaging applications.

The presented work can be characterized as a *monograph* of the author's *original contributions* to the dynamic and expanding field of *multichannel image processing* put on the background of the state of the art of the *noise removal* in digital images. This *monograph* is an *integration* of techniques proposed by the author in various scientific publications scattered in a variety of journal papers and referred conference proceedings, and is oriented towards a wide spectrum of *contemporary applications*.

This monograph details the author's *most important contributions* to the rapidly growing field of *nonlinear noise reduction* in *multichannel images*:

In the third Chapter the *modified weighted median filter framework* has been presented. This *new technique* simplifies the structure of the *weighted medians* and improves *significantly* the properties of the *central weighted vector median filter* by enhancing its *detail preserving* abilities. Future research will focus on the development of a robust modified weighted VMF and on the development of fast methods of the optimization of its parameters, to achieve the optimal filtering efficiency for a given image and noise scenario.

In the next Chapter, the robust *anisotropic diffusion* filtering scheme, which *ignores* the central pixel of the filtering window, when building the *weighted average* of the input samples is introduced. This improvement allows to use the *anisotropic technique* for the *suppression* of strong Gaussian and heavy tailed noise, as the influence of the central, corrupted pixel is *diminished* by an appropriate setting of the conductivity coefficients. It is worth noticing, that the proposed structure is a *generalization* of the previous *nonlinear adaptive techniques*, whose robustness is based on the *rejection* of the central pixel of the filtering window.

In the same Chapter, an iterative *forward* and *backward anisotropic diffusion* technique, based on the *unsharp masking* concept has been described. This method enables to construct new families of filters, able to remove strong Gaussian noise and to enhance the image edges. This new approach to the problem of *noise reduction* and *image enhancement* is very flexible, as the designer can model the conductivity coefficients taking the derivatives of the classical

flux functions with appropriate setting of the parameters. The efficiency of the newly developed filter class can be increased by *neglecting* the influence of the central pixel, which should enable the enhancement of images contaminated by impulsive noise.

The next Chapter has been devoted to the development of a *powerful class* of filters based on the *digital paths* concepts and *fuzzy similarity measures* among pixels in neighborhood relation. This novel technique, which utilizes the *connection* between image pixels, is an extension of the *adaptive noise reduction filtering* and *anisotropic diffusion techniques*, presented in Chapter 2 and 3 respectively, and is shown to have *advantages* over the *traditional methods*. Extensive simulations revealed that the proposed filtering framework *significantly* excels over the standard methods and can be applied for the removal of both Gaussian and impulsive noise.

The family of *detail-preserving* impulsive noise removal techniques described in Chapter 6, which make use of the concepts of *similarity* between the neighboring pixels, elements of the *nonlinear regression* and *nonparametric probability density estimation* theory, is shown to possess *excellent* noise reduction and detail preserving capabilities. The presented filter class is also *computationally efficient*, especially when using a simplified filtering structure with a global bandwidth parameter. The *excellent efficiency*, coupled with the low computational burden, makes this filtering class interesting for a wide range of real time applications.

Another *powerful* method of *impulsive noise removal* has been presented in the next Chapter. This method, utilizing the *switching filtering concept*, is based on the simplified measure of the samples' dispersion and is able to efficiently detect noisy samples, while preserving salient image features. Future work will be focused on the *adaptation procedure*, which would enable the automatic setting of the switching threshold parameter.

The advantages brought by the modification of the weighted vector median described in Chapter 3, can be fully exploited by the *iterative optimization procedure* presented in the last Chapter. The optimization method is based on the *classical optimization* of the *weighted medians* and its *extension* to the *multivariate case* is accomplished through the introduction of the *polarity function*, which assigns a *polarity* to the distance between two image samples. This *optimization method* is shown to be *quite efficient* as the *optimized weighted vector directional distance filters* yield much better results as their static counterparts.

The author of this monograph hopes that the presented state of the art and the original contributions to the expanding and challenging field of color image enhancement will be useful in various applications, in which the *noise removal* with the preservation of salient image features is of vital importance. For a deeper investigation of the presented methods an extensive bibliography has been prepared.

* * *

Bibliography

[1] J. Abbas, M. Domański, Median-based filters with prediction error processing for video restoration, Proc. of ISCAS, May 28-31, Geneva, Switzerland, 689-692, 2000.

[2] J. Abbas, M. Domański, Decision-based median-related filters for image and video restoration, Proc. of Int. Conf. on Signals and Electronic Systems (ICSE), Ustroń, Poland, 17-20 October, 181-186, 2000.

[3] 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 Acoustic, Speech and Signal Processing, 4, 2371-2374, 1995.

[4] E. Abreu, Signal-dependent rank-ordered mean (SD-ROM) filter, in [208], 2001.

[5] S.T. Acton, Multigrid anisotropic diffusion, IEEE Trans. on Image Processing, 7, 3, 280-291, 1998.

[6] M.A. Ajzerman, E.M. Braverman, L.I. Rozonoer, "Rozpoznawanie Obrazów - Metoda Funkcji Potencjalnych", WNT, Warszawa 1976.

[7] L. Alparone, M. Barni, F. Bartolini, V. Cappellini, Adaptively weighted vector-median filters for motion-fields smoothing, Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2267-2270, May 7-10, 1996.

[8] H. Altunbasak, H.J. Trussell, Colorimetric restoration of digital images, IEEE Trans. on Image Processing, 10, 393-402, 2001.

[9] L. Alvarez, F. Guichard, P.L. Lions, J.M. Morel, Image selective smoothing and edge detection by nonliner diffusion, SIAM J. Numer. Anal., 29, 3, 845-866, 1992.

[10] T.W. Anderson, "An Introduction to Multivariate Statistical Analysis", New York, Wiley, 2nd edition, 1994.

[11] D. Androutsos, K.N. Plataniotis, A.N. Venetsanopoulos, A novel vector-based approach to color image retrieval using a vector angular-based distance measure, Computer Vision and Image Understanding, 1-2, 75, 46-58, 1999.

[12] G.R. Arce, N.C. Gallagher, T.A. Nodes, Median filters: theory and applications, in "Advances in Computer Vision and Image Processing", (T.S. Huang, Ed.), II, JAI Press, 1986.

[13] G.R. Arce, Multistage order statistic filters for image sequence processing, IEEE Trans. on Signal Processing, 39, 1146-1163, 1991.

[14] G.R. Arce, T.A. Hall, K.E. Barner, Permutation weighted order statistic filter lattices, IEEE Trans. on Image Processing, 4, 1070-1083, 1995.

[15] G.R. Arce, A generalized weighted median filter structure admitting negative weights, IEEE Trans. on Signal Processing, 46, 3195-3205, 1998.

[16] G. R. Arce, J. Paredes, Image enhancement with weighted medians, in [208], 27-67, 2001.

[17] A. Asano, K. Itoh, Y. Ichioka, Optimization of the weighted median filter by learning, Optics Letters, 16, 3, 168-170, 1991.

[18] J. Astola, P. Haavisto, Y. Neuovo, Median type filters for color signals, Proc. of the Int. Symp. on Circuits and Systems, Helsinki, June 7-9, 2, 1753-1756, 1988.

[19] J. Astola, P. Haavisto, Y. Neuvo, Vector median filters, Proc. of the IEEE, 78, 4, 678-689, 1990.

[20] J. Astola, P. Kuosmanen, "Fundamentals of Nonlinear Digital Filtering", CRC Press, 1997.

[21] J. Astola, D. Akopian, O. Vainio, S. Agaian, New digit-serial implementations of stack filters, Signal Processing, 61, 181-197, 1997.

[22] J. Astola, D. Akopian, A time-recursive implementation of threshold boolean filters, IEEE Trans. on Circuits and Systems - II, 46, 139-148, 1999.

[23] V. Aurich, J. Weule, Non-liner gaussian filters performing edge preserving diffusion, DAGM-Symposium 1995, Proceedings in Informatik Aktuell, Springer, 538-545, 1995.

[24] I. Avcibas, B. Sankur, K. Sayood, Statistical evaluation of image quality measures, Journal of Electronic Imaging, 11, 206-223, 2002.

[25] G.J. Awcock, R. Thomas, "Applied Image Processing", Macmillan, 1995.

[26] G.A Babich, O.I. Camps, Weighted Parzen windows for pattern classification, IEEE Trans. on PAMI, 18, 5, 567-570, 1996.

[27] P. Bakker, L.J. van Vliet, P.W. Verbeek, Edge preserving orientation adaptive filtering, Proc. of the 5th Annual Conf. of the Advanced School for Computing and Imaging (ASCI), Heijen, Netherlands, 207-213, June 15-17, 1999.

[28] D. Barash, A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation, IEEE Trans. on PAMI, 24, 6, 844-847, 2002.

[29] A.J. Bardos, S.J. Sangwine, Selective vector median filtering of colour images, Proc. of the 6th IEE Int. Conf. on Image Processing and its Applications, Dublin, Ireland, 708-711, 1997.

[30] K.E. Barner, G.R. Arce, Permutation filters: a class of nonlinear filters based on set permutations, IEEE Trans. on Signal Processing, 42, 4, 782-798, 1994.

[31] K.E. Barner, R.C. Hardie, Extended permutation filters and their application to edge enhancement, IEEE Trans. on Image Processing, 5, 855-867, 1996.

[32] K.E. Barner, G.R. Arce, Design of permutation order-statistic filters through group colorings, IEEE Trans. on Circuits and Systems, 44, 7, 531-548, 1997.

[33] K.E. Barner, R.C. Hardie, Spatial-rank order selection filter, in [208], 2001.

[34] K.E. Barner, G.R Arce, "Nonlinear Signal and Image Processing", CRC Press, Boca Raton, 2004

[35] V. Barnett, The ordering of multivariate data, Journal of the Royal Statistical Society, A 139, III, 318-355, 1976.

[36] V. Barnett, T. Lewis, "Outliers in Statistical Data", Wiley, 1994.

[37] M. Barni, V. Cappellini, A. Mecocci, Fast vector median filter based on Euclidean norm approximation, IEEE Signal Processing Letters, 1, 6, 92-94, 1994.

[38] M. Barni, V. Cappellini, On the computational complexity of multivariate median filters, Signal Processing, 71, 45-54, 1998.

[39] M. Barni, F. Bartolini, V. Capellini, Image processing for virtual restoration of artworks, IEEE Multimedia, 7, 34-37, April-June 2000.

[40] M. Bartkowiak, M. Domański, Vector median filters for processing of color images in various color spaces, Proc. of the 5th Int. Conf. on Image Processing and its Applications, IEE, Edinburgh, UK, 833-836, 1995.

[41] M. Basu, M. Su, Image smoothing with exponential functions, Int. Journal of Pattern Recognition and Artificial Intelligence, 15, 4, 735-752, 2001.

[42] A. Beghdadi, K. Khellaf, A noise-filtering method using a local information measure, IEEE Trans. on Image Processing, 6, 879-882, 1997.

[43] R. Berstain, Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images, IEEE Trans. on Circuits and Systems, 34, 1275-1291, 1987.

[44] J.C. Bezdek, S.K. Pal (Eds.), "Fuzzy Models for Pattern Recognition", IEEE Press, 1992.

[45] S. Biswas, N.R. Pal, S.K. Pal, Smoothing of digital images using the concept of diffusion process, Pattern Recognition, 29, 3, 497-510, 1996.

[46] M.J. Black, G. Sapiro, D.H. Marimont, D. Heeger, Robust anisotropic diffusion, IEEE Trans. on Image Processing, 7, 3, 421-432, 1998.

[47] M.J. Black, G. Sapiro, Edges as outliers: anisotropic smoothing using local image statistics, Lecture Notes in Computer Science, 1682, 259-270, 1999.

[48] P. Blomgren, T.F. Chan, Color tv: total variation methods for restoration of vector-valued images, Special Issue on Partial Differential Equations and Geometry Driven Diffusion in Image Processing and Analysis, IEEE Trans. on Image Processing, 7, 3, 304-309, 1998.

[49] C. Boncelet, Image noise models, in "Handbook of Image and Video Processing", (Ed. A. Bovik), Academic Press, 2000.

[50] G. Borgefors, Distance transformations in arbitrary dimensions, Computer Vision, Graphics and Image Processing, 27, 321-345, 1984.

[51] G. Borgefors, Distance transformations in digital images, Computer Vision, Graphics and Image Processing, 34, 344-371, 1986.

[52] R.M. Boynton, "Human Color Vision", Halt, Rinehart and Winston, 1990.

[53] P.A. Bromiley, N.A. Thacker, P. Courtney, Colour image segmentation by non-parametric density estimation in colour space, Proc. of the British Machine Vision Conf. (BMVC), Manchester, July 2001.

[54] D.R.K. Brownrigg, The weighted median filter, Commun. of Assoc. Computers, 807-818, 1984.

[55] P.D. Burns, Image signal modulation and noise characteristics of charge-coupled device imagers, Proc. of SPIE, 1071, 144-152, 1989.

[56] T. Cacoullos, Estimation of a multivariate density, Ann. Inst. Stat. Math., 18, 179-189, 1966.

[57] J. Canny, A computational approach to edge detection, IEEE Trans. on PAMI, 8, 6, 679-698, 1986.

[58] R.A. Carmona, S. Zhong, Adaptive smoothing respecting feature directions, IEEE Trans. on Image Processing, 7, 3, 353-358, 1998.

[59] K.R. Castleman, "Digital Image Processing", Prentice Hall, Toronto, 1996.

[60] F. Catté, F. Dibos, G. Koepfler, Image selective smoothing and edge detection by nonliner diffusion, SIAM Journal of Numer. Anal., 1, 29, 182-193, 1992.

[61] F. Catté, P.L. Lions, J.M. Morel, T. Coll, Image selective smoothing and edge detection by nonlinear diffusion - II, SIAM Journal of Numer. Anal., 29, 3, 845-866, 1992.

[62] T.F. Chan, J. Shen, L. Vese, Variational PDE models in image processing, Notices of the American Mathematical Society, 1, 50, 14-26, 2003.

[63] P. Charbonnier, G. Aubert, M. Blanc-Ferraud, M. Barlaud, Two deterministic half-quadratic regularization algorithms for computed imaging, Proc. of IEEE Int. Conf. on Image Processing (ICIP), 168-172, Austin (Texas), November 1994.

[64] D. Chaudhuri, C.A. Murthy, B.B. Chaudhuri, A modified metric to compute distances, Pattern Recognition, 25, 7, 667-677, 1992.

[65] K. Chen, Bit-serial realizations of a class of nonlinear filters based on positive boolean functions, IEEE Trans. on Circuits and Systems, 36, 785-794, 1989.

[66] C.H. Chen, L.F. Pau, P.S.P. Wang, "Handbook of Pattern Recognition and Computer Vision", World Scientific, 1993.

[67] T. Chen, K.K. Ma, L.H. Chen, Tri-state median filter for image denoising, IEEE Trans. on Image Processing, 8, 12, 1834-1838, 1999.

[68] T. Chen, H.R. Wu, Adaptive impulse detection using center-weighted median filters, IEEE Signal Processing Letters, 8, 1, 1-3, 2001.

[69] H. Chen, P. Meer, Robust computer vision through kernel density estimation, Lecture Notes in Computer Science, LNCS 2350, 236-250, 2002.

[70] 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.

[71] A. Chydzinski, B. Smolka, Self-avoiding walks on the lattice Z^2 with 8 neighbourhood systems, Applicationes Mathematicae, 28, 2, 169-180, 2001.

[72] M. Ciuc, R.M. Rangayyan, T. Zaharia, V. Buzuloiu, Adaptive neighborhood for color image filtering, Proc. of SPIE, 3304, 277-286, 1998

[73] C. Connoly, T. Flies, A study of efficiency and accuracy in the transformation from RGB to CIELAB color space, IEEE Trans. on Image Processing, 6, 1046-1048, 1997.

[74] N. Cooharojananone, K. Aizawa, Cumulative angular distance measure for color indexing, IEICE Trans. on Inf. and Systems, E84-D, 4, 537-540, 2001.

[75] E.J. Coyle, J.H. Lin, M. Gabbuoj, Optimal stack filtering and the estimation and structural approaches to image processing, IEEE Trans. on Acoustics, Speech, and Signal Processing, 37, 2037-2066, 1989.

[76] O. Cuisenaire, Distance transformations: fast algorithms and applications to medical image processing, PhD Thesis, Université catholique de Louvain, Belgium, October 1999.

[77] C.S. McCurnin, L.C. Schooley, G.R. Sims, Charge-coupled-device signal processing models and comparisons, Journal of Electronic Imaging, 2, 100-107, 1993.

[78] H.A. David, "Order Statistics", Wiley, New York, 1981.

[79] E.R. Davies, Accuracy of multichannel median filter, Electronics Letters, 36, 25, 2000.

[80] E.R. Davies, Image distorsions produced by mean, median and mode filters, IEE Digest 1998/284, 1-5, Colloquium on Non-Linear Signal and Image Processing, May, 1998.

[81] L.S. Davis, A. Rosenfeld, Noise cleaning by iterated local averaging, IEEE Trans. on Syst. Man. Cybern., 8, 705-710, 1978.

[82] R. Deriche, O. Faugeras, Les edp en traitement des images et vision par ordinateur, Technical Report 2697 of INRIA, 1995.

[83] M. Desbrun, M. Meyer, P. Schröder, A.H. Barr, Anisotropic feature-preserving denoising of height fields and bivariate data, Proc. of Graphics Interface, Montréal, Québec, Canada, 145-152, 15-17 May 2000.

[84] R. Diestel, "Graph Theory", Electronic Edition, Springer Verlag, New York, 2000.

[85] Y. Dodge, V. Rousson, Multivariate L_1 mean, Metrika, 49, 127-134, 1999

[86] M. Domański, Zaawansowane Techniki Kompresji Obrazów i Sekwencji Wizyjnych, WPP, Poznań 1998, 2000.

[87] M. Domański, J.K. Abbas, A family of efficient nonlinear filters for video restoration, Machine Graphics & Vision, 9, 1-2, 353-361, 2000.

[88] M. Domański, J.K. Abbas, Median-based filters with prediction error processing for video restoration, IEEE Int. Symp. on Circuits and Systems, Geneva, 4, 689-692, 2000.

[89] M. Domański, J.K. Abbas, Decision-based median-related filters for image and video restoration, Int. Conf. on Signals and Electronic Systems (ICSES), Ustroń, Poland, 17-20, October 2000.

[90] J. Dopazo, Microarray data processing and analysis, in "Microarray Data Analysis", (Eds. S.M. Lin, K.F. Johnson), Kluwer Academic, 2002.

[91] R.O. Duda, P.E. Hart, "Pattern Classification and Scene Analysis", Wiley, New York, 1973.

[92] R.O. Duda., P.E. Hart, D.G. Stork, "Pattern Clasiffication", Wiley, New York, 2001.

[93] R.P.W. Duin, On the choice of smoothing parameters for Parzen estimators of probability density functions, IEEE Trans. on Comput. 25, 1175-1179, 1976.

[94] S.A. Durrani, The distance-weighted k-nearest neighbor rule, IEEE Trans. on Systems, Man and Cybernetics, 15, 630-636, 1977.

[95] M.B. Eisen, P.O. Brown, DNA arrays for analysis of gene expression, Methods in Enzymology, 303, 179-205, 1999.

[96] H.L. Eng, K.K. Ma, Noise adaptive soft-switching median filter, IEEE Trans. on Image Processing, 10, 2, 242-251, 2001.

[97] M. Fischer, J.L. Paredes, G.R. Arce, Weighted median image sharpeners for the world wide web, IEEE Trans. on Image Processing, 11, 7, 717-727, 2002.

[98] B. Fischl, E.L. Schwartz, Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing, IEEE Transactions on PAMI, 19, 4, 342-352, 1997.

[99] W. Förstner, Image preprocessing for feature extraction in digital intensity, color and range images, Proc. of the Summer School on Data Analysis and the Statistical Foundations of Geomatics, Crete, Greece, May 25-30, 1998.

[100] M.F.A. Fortier, D. Ziou, C. Armenakis, S. Wang, Nouvelles perspectives en détection de contours: Textures et images multispectrales, Proc. of Vision Interface, Trois-Rivières, Canada, 19-21 May, 1999.

[101] A. Fotinos, N. Laskaris, S. Fotopoulos, Fuzzy color filtering using potential functions, Int. Conf. on Digital Signal Processing (DSP), Santorini, Greece, July 2-4, 1997.

[102] A. Fotinos, N. Laskaris, S. Fotopoulos, fuzzy color filtering using relative entropy, Proc. of EUSIPCO, Rhodes, Greece, 809-812, 8-11 September 1998.

[103] F. Fotinos, N. Laskaris, G. Economou, S. Fotopoulos, Multidimensional fuzzy filtering using statistical parameters, Multidimensional Systems and Signal Processing, 10, 415-424, 1999.

[104] K. Fukunaga, "Introduction to Statistical Pattern Recognition", Academic Press, London, 1990.

[105] M. Gabbouj, E.J. Coyle, N.C. Gallager, An overview of median and stack filtering, Circuit Syst. and Signal Processing, 11, 1, 7-45, 1992.

[106] M. Gabbouj, F.A. Cheickh, Vector median - vector directional hybrid filter for colour image restoration, Proc. of EUSIPCO, Trieste, Italy, September 10-13, 879-881, 1996.

[107] L. Gajek, M. Kałuszka, "Wnioskowanie Statystyczne, Modele i Metody", WNT, Warszawa, 2000.

[108] N.C. Gallagher, Median filters: a tutorial, Proc. of ISCAS, 1737-1745, 1988.

[109] C. Garnica, F. Boochs, M. Twardochlib, A new approach to edge-preserving smoothing for edge extraction and image segmentation, Proc. of IAPRS, 32, Amsterdam, 2000.

[110] S. Geman, G. Reynolds, Constrained restauration and the recovery of discontinuities, IEEE Trans. on PAMI, 14, 3, 367-383, 1992.

[111] A.T. Georgiadis, "Adaptive equalisation for impulsive moise environments", PhD Thesis, The University of Edinburgh, 2000.

[112] G. Gerig, R. Kikinis, O. Kuebler, F. Jolesz, Nonlinear anisotropic filtering of MRI data, IEEE Trans. on Medical Imaging, 11, 2, 221-232, 1992.

[113] G. Gilboa, Y. Zeevi, N. Sochen, Anisotropic selective inverse diffusion for signal enhancement in the presence of noise, Proc. of ICASSP, Istanbul, Turkey, I, 211-224, June 2000.

[114] G. Gilboa, Y. Zeevi, N. Sochen, Signal and image enhancement by a generalized forward-and-backward adaptive diffusion process, Proc. of EUSIPCO, Tampere, Finland, September 2000.

[115] G. Gilboa, Y. Zeevi, N. Sochen, Resolution enhancement by forward-and-backward nonlinear diffusion processes, Proc. of the Nonlinear Signal and Image Processing Conf. (NSIP), Baltimore, Maryland, June 2001.

[116] G. Gilboa, N. Sochen, Y.Y. Zeevi, A forward-and-backward diffusion process for adaptive image enhancement and denoising, IEEE Trans. on Image Processing, 11, 7, 689-703, July 2002.

[117] J. Gomes, L. Velho, "Image Processing for Computer Graphics", Springer Verlag, 1997.

[118] R.C. Gonzalez, R.E. Woods, "Digital Image Processing", Addisson Wesley, Reading MA, 1992.

[119] P.J. Green, Bayesian reconstruction from emission tomography data using a modified em algorithm, IEEE Trans. on Med. Imaging, 9, 84-93, 1990.

[120] S. Guillon, P. Baylou, M. Najim, N. Keskes, Adaptive non-linear filters for 2D and 3D image enhancement, Signal Processing, 67, 237-254, 1998.

[121] P. Haavisto, P. Heinonen, Y. Neuovo, Vector FIR-Median hybrid filters for multispectral signals, Electronics Letters, 24, 7-8, 1988.

[122] A. Ben Hamza, H. Krim, Image denoising: a nonlinear robust statistical approach, IEEE Trans. on Signal Processing, 49, 12, 3045-3054, 2001.

[123] 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, 1, 2, 197-209, 1991.

[124] R.C. Hardie, C.G. Boncelet, LUM filters: a class of rank-order-based filters for smoothing and sharpening, IEEE Trans. on Signal Processing, 41, 3, 1061-1076, 1993.

[125] R.C. Hardie, K.E. Barner, Rank conditioned rank selection filters for signal restoration, IEEE Trans. on Image Processing, 3, 2, 192-206, 1994.

[126] R.C. Hardie, C.G. Boncelet, Gradient-based edge detection using nonlinear edge enhancing prefilters, IEEE Trans. on Image Processing, 4, 1572-1577, 1995.

[127] W. Häerdle, "Smoothing Techniques With Implementations in S", Springer, New York, 1991.

[128] W. Härdle, M. Müller, Multivariate and Semiparametric Kernel Regression, Sonderforschungsbereich 373, 1997-26, Humboldt Universitaet Berlin, 1997.

[129] Y. Hashimoto, Y. Kajikawa, Y. Nomura, Directional difference-based switching median filters, Electronics and Communications in Japan, III, 85, 3, 22-32, 2002.

[130] Y. Hashimoto, Y. Kajikawa, Y. Nomura, Directional difference filters: its effectiveness in the postprocessing of noise detectors, Electronics and Communications in Japan, III 3, 85, 2, 74-82, 2002.

[131] K.J. Hermiston, D.M. Booth, NIRS and objective quality measures, Lecture Notes in Computer Science, LNCS 1689, 385-394, 1999.

[132] J. Immerkaer, Use of blur-space for deblurring and edge-preserving noise smoothing, IEEE Trans. on Image Processing, 10, 6, 837-840, 2001.

[133] B. Jähne, "Practical Handbook on Image Processing for Scientific Applications", CRC Press, Boca Raton, FL, 1997.

[134] B. Jeon, D.A. Landgrebe, Fast Parzen density estimation using clustering-based branch and bound, IEEE Trans. on PAMI, 16, 9, 950-954, 1994.

[135] A.R. Jiménez, R. Ceres, J.L. Pons, A new adaptive filter and quality evaluation index for image restoration, Pattern Recognition, 34, 457-467, 2001.

[136] T.E. Jenkins, "Optimal Sensing Techniques and Signal Processing", Prentice Hall, 1987.

[137] D.G. Karakos, P.E. Trahanias, Combining vector median and vector directional filters: the directional-distance filters, Proc. of the IEEE Int. Conf. on Image Processing (ICIP), Washington DC, 171-174, October 1995.

[138] D. Karakos, P.E. Trahanias, Generalized multichannel image filtering structures, IEEE Trans. on Image Processing, 6, 7, 1038-1045, 1997.

[139] A.J. Kasinski, Smoothing noisy images without destroying predefined feature carriers, Lecture Notes in Computer Science, LNCS 1296, 519-526, 1997.

[140] V. Katkovnik, I. Shmulevich, Kernel density estimation with adaptive varying window size, Pattern Recognition Letters, 23, 1641-1648, 2002

[141] V. Kayargadde, J.B. Martens, An objective measure for perceived noise, Signal Processing, 49, 3, 187-206, 1996.

[142] S.L. Keeling, R. Stollberger, Nonlinear anisotropic diffusion filtering for multiscale edge enhancement, Inverse Problems, 18, 175-190, 2002.

[143] L. Khriji, M. Gabbouj, Vector median-rational hybrid filters for multichannel image processing, IEEE Signal Processing Letters, 6, 7, 186-190, 1999.

[144] L. Khriji, M. Gabbouj, Adaptive fuzzy order statistics-rational hybrid filters for color image processing, Fuzzy Sets and Systems, 128, 35-46, 2002.

[145] L. Khriji, M. Gabbouj, Rational-based adaptive fuzzy filters, Int. Journal of Computational Cognition, 2, 1, 113-132, 2004.

[146] H.M. Kim, B. Kosko, Fuzzy Filters for Impulsive Noise, in B. Kosko, (Ed.), "Fuzzy Engineering", 251-275, Prentice-Hall, Englewood Cliffs, New Jersey, 1997.

[147] B.B. Kimia, K. Siddiqi, Geometric heat equation and nonlinear diffusion of shapes and images, Computer Vision and Image Understanding, 64, 3, 305-322, 1996.

[148] R. Kimmel, R. Malladi, N. Sochen, Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images, Int. Journal of Computer Vision, 39, 2, 111-129, 2000.

[149] C.O. Kiselman, Regularity properties of distance transformations in image analysis, Computer Vision and Image Understanding, 64, 3, 390-398, 1996.

[150] R. Klette, P. Zamperoni, "Handbuch der Operatoren für die Bildverarbeitung", Vieweg, Wiesbaden, 1992.

[151] S.J. Ko, Y.H. Lee, Center weighted median filters and their application to image enhancement, IEEE Trans. on Circuits and Systems, 38, 9, 984-993, 1991.

[152] V. Kober, M. Morozov, J. Alvarez-Borrego, Nonlinear filters with spatially connected neighborhoods, Optical Engineering, 40, 6, 971-983, 2001.

[153] A. Kokaram, "Motion Picture Restoration", Springer-Verlag, Berlin, 1998.

[154] T.Y. Kong, A. Rosenfeld, "Topological Algorithms for Digital Image Processing", Elsevier, Amsterdam, 1996.

[155] A. Koschan, M. Abidi, A comparison of median filter techniques for noise removal in color images, Proc. of the 7th German Workshop on Color Image Processing, D. Paulus, J. Denzler (Eds.), Erlangen, Germany, Report of University of Erlangen-Nurnberg, Institute of Computer Science, 34, 15, 69-79, October 2001.

[156] C. Kotropoulos, I. Pitas, Multichannel L filters based on marginal data ordering, IEEE Trans. on Signal Processing, 42, 10, 2581-2595, 1994.

[157] C. Kotropoulos, I. Pitas, "Nonlinear Model-Based Image and Video Processing and Analysis", Willey, 2001.

[158] M.A. Kraaijveld, A Parzen classifier with an improved robustness against deviations between training and test data, Pattern Recognition Letters, 17, 679-689, 1996.

[159] K. Krissian, G. Malandain, N. Aayache, Directional anisotropic diffusion applied to segmentation of vessels in 3D images, Technical Report 3064, INRIA, 1996.

[160] K. Krissian, Flux-based anisotropic diffusion applied to enhancement of 3D angiograms, IEEE Trans. on Medical Imaging, 21, 11, 1-4, 2002.

[161] D.T. Kuan, A.A. Sawchuk, T.C. Strand, P. Chavel, Adaptive noise smoothing filter for images with signal-dependent noise, IEEE Trans on PAMI, 7, 2, 165-177, 1985.

[162] P. Kuosmanen, Statistical analysis and optimization of stack filters, Acta Polytechnica Scandinavica, EI77, 1994.

[163] M. Kuwahara, K. Hachimura, S. Eiho, M. Kinoshita, "Digital Processing of Biomedical Images", Plenum Press, New York, 187-203, 1976.

[164] J.S. Lee, Digital image enhancement and noise filtering by use of local statistics, IEEE Trans. on PAMI, 2, 2, 165-168, 1980.

[165] J.S. Lee, Refined filtering of image noise using local statistics, Computer Graphics and Image Processing, 15, 380-389, 1981.

[166] J.S. Lee, Digital image smoothing and the sigma filter, Computer Vision, Graphics, and Image Processing, 24, 255-269, 1983.

[167] L. Levi, Unsharp masking and related image enhancement techniques, Computer Graphics and Image Processing, 3, 163-177, 1974.

[168] X. Li, T. Chen, Nonlinear diffusion with multiple edginess thresholds, Pattern Recognition, 27, 1029-1037, 1994.

[169] X. Li, D. Lu, Y. Pan, Color restoration and image retrieval for Donhunag fresco preservation, IEEE Multimedia, 7, 38-42, 2000.

[170] R.S. Lin, Y.C. Hsueh, Multichannel filtering by gradient information, Signal Processing, 80, 2, 279-293, 2000.

[171] P.Y Liu, H.X Li, Fuzzy techniques in image restoration research - a survey, Int. Journal of Computational Cognition, 2, 2, 131-149, 2004.

[172] G. Lohman, "Volumetric Image Analysis", Wiley & Teubner, 1998.

[173] A. Lopes, R. Touzi, E. Nezry, Adaptive speckle filters and scene heterogeneity, IEEE Trans. on Geoscience and Remote Sensing, 28, 6, 992-1000, 1990.

[174] D.B. Lopez, F.H. Mendoza, J.M. Ramirez, Noise in color digital images, Proc. of the Midwest Symp. on Circuits and systems, 403-406, August 1999.

[175] M. Lucassen, P. Bijl, The CIE94 colour difference formula for describing visual detection thresholds in static noise, Second European Conference on Color in Graphics, Imaging and Vision (CGIV), Aachen, Germany, 8-11, April 5-8, 2004.

[176] L. Lucat, P. Siohan, Vector-Median type filters and fast-computation algorithms, IEEE Int. Symp. on Circuits and Systems, Hong Kong, June 9-12, 1997.

[177] L. Lucat, P. Siohan, D. Barba, Adaptive and global optimization methods for weighted vector median filters, Signal Processing: Image Communications, 17, 509-524, 2002.

[178] R. Lukac, S. Marchevsky, Boolean expression of LUM smoothers, IEEE Signal Processing Letters, 8, 292-294, 2001.

[179] R. Lukac, S. Marchevsky, LUM smoother with smooth control for noisy image sequences, EURASIP Journal of Applied Signal Processing, 2, 110-120, 2001.

[180] R. Lukac, Adaptive impulse noise filtering by using center-weighted directional information, Proceedings of the CGIV Conference, Poitiers, France, 86-89, 2002.

[181] R. Lukac, Adaptive LUM smoother for noisy watermarked images, Pattern Recognition and Image Analysis, 12, 448-454, 2002.

[182] R. Lukac, Binary LUM smoothing, IEEE Signal Processing Letters, 9, 400-403, 2002.

[183] R. Lukac, Color image filtering by vector directional order-statistics, Pattern Recognition and Image Analysis, 12, 3, 279-285, 2002.

[184] R. Lukac, Optimised directional distance filter, Machine Graphics and Visions: Special Issue on Colour Image Processing and Applications, 11, 2/3, 311-326, 2002.

[185] R. Lukac, B. Smolka, P. Galajda, Hearth dynamics filtered by optimised adaptive median, Proceedings of the International Federation for Medical & Biological Engineering, 2nd European Medical & Biological Engineering Conference (EMBEC), Vienna, Austria, III, 2, 954-955, December 4-8, 2002.

[186] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Sigmoidal weighted vector directional filter, Lecture Notes in Computer Science, LNCS 2626, 418-427, 2003.

[187] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, P. Zavarsky, Angular Multichannel Sigma Filter, Proceedings of the 28th IEEE International Conference on Acoustic, Speech & Signal Processing (ICASSP), Hong Kong, III, 745-748, April 6-10, 2003.

[188] R. Lukac, B. Smolka, T. Terpo, Genetic optimization of weighted vector directional filters, Pattern Recognition and Image Analysis, 14, 1, 4049, 2004.

[189] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopulos, Selection weighted vector directional filters, Computer Vision and Image Understanding, 94, 140-167, 2004.

[190] R. Lukac, B. Smolka, K.N. Plataniotis, A.N Venetsanopoulos, New approach to the impulsive noise removal in microarray images, Proceedings of 5th IFAC Symposium 2003 on Modelling and Control in Biomedical Systems, Melbourne, Australia, 145-150, (Elsevier), August 21-23, 2003.

[191] R. Lukac, K.N. Plataniotis, A.N. Venetsanopoulos, R. Bieda, B. Smolka, Color edge detection techniques, in "Signaltheorie und Signalverarbeitung, Akustik und Sprachakustik, Informationstechnik", Studientexte zur Sprachkommunikation, W.E.B. Universität Verlag, Dresden, 21-47, 29, 2003.

[192] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, P. Zavarsky, Angular multichannel sigma filter, Proceedings of the ICASSP, Hong Kong, III, 745-748, April 2003.

[193] R. Lukac, B. Smolka, Application of the adaptive center-weighted vector median framework for the enhancement of cDNA microarray images, International Journal of Applied Mathematics and Computer Science, 13, 101-115, 2003.

[194] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Entropy vector median filter, Lecture Notes in Computer Science, LNCS 2652, 1117-1125, 2003.

[195] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Generalized adaptive vector sigma filters, Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Baltimore, USA, 537-540, July 2003.

[196] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Entropy vector median filter, Lecture Notes in Computer Science, LNCS 2652, 1117-1125, 2003.

[197] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, A variety of multichannel sigma filters, Proceedings of SPIE 5146, 244-253, 2003.

[198] R. Lukac, B. Smolka, A. Swierniak, K.N. Plataniotis, A.N. Venetsanopoulos, Weighted vector directional filters optimized by genetic algorithms, Lecture Notes in Computer Science, LNCS 3019, 595-600, 2003.

[199] R. Lukac, B. Smolka, A. Swierniak, K.N. Plataniotis, A.N. Venetsanopoulos, Adaptive noise reduction in DNA microarray images based on the center-weighted vector median filters, Lecture Notes in Computer Science, LNCS 3019, 714-721, 2003.

[200] R. Lukac, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Color image filtering and enhancement, IEEE Signal Processing Magazine, Special Issue on Color Image Processing, (invited paper), to appear in 2004.

[201] J. Maeda, T. Iizawa, T. Ishizaka, C. Ishikawa, Y. Suzuki, Segmentation of natural images using anisotropic diffusion and linking of boundary edges, Pattern Recognition, 31, 12, 1993-1999, 1998.

[202] N. Madras, G. Slade, "The Self-Avoiding Walk", Boston, Birkhäuser, 1993.

[203] M. Mahy, L. van Eyckden, A. Oosterlinck, Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV, Color Research and Application, 19, 105-121, April 1994.

[204] O.L. Mangasarian, D.R. Musicant, Robust linear and support vector regression, IEEE Trans. on PAMI, 22, 9, 950-955, 2000.

[205] K.V. Mardia, J.T. Kent, J.M. Bibby, "Multivariate Analysis", Academic Press, 1979.

[206] G.A. Mastin, Adaptive filters for digital image noise smoothing: an evaluation, Computer Vision, Graphics and Image Processing, 31, 103-121, 1985.

[207] M.J. McDonnell, Box-filtering techniques, Computer Graphics and Image Processing, 17, 65-70, 1981.

[208] S. Mitra, J. Sicuranza, "Nonlinear Image Processing, (Communications, Networking and Multimedia)", Academic Press, 2001.

[209] M.S. Moore, S.K. Mitra, Performance analysis of the two-state signal-dependent rank order mean filter, Proc. of the 10th IS&T/SPIE Conf. on Nonlinear Image Processing, San Jose, CA, January 1999.

[210] M.S. Moore, M. Gabbouj, S.K. Mitra, Vector SD-ROM filter for removal of impulse noise from color images, Proc. of the Second EURASIP Conf. (ECMCS), Kraków, 24-26 June, 1999.

[211] J.M. Morel, S. Solimini, "Variational Methods in Image Segmentation", Birkhäuser, Boston, MA, 1995.

[212] S. Mustonen, A measure for total variability in multivariate normal distribution, Computational Statistics and Data Analysis, 23, 321-334, 1997.

[213] M. Nagao, T. Matsuyama, Edge preserving smoothing, Computer Graphics and Image Processing, 9, 394-407, 1979.

[214] M. Nagao, T. Matsuyama, "A Structural Analysis of Complex Aerial Photographs", Plenum Press, New York, 1980.

[215] A. Nieminen, P. Heinonen, Y. Neuovo, A new class of detail preserving filters for image processing, IEEE Trans. on PAMI, 9, 74-90, 1987.

[216] N. Nikolaidis, I. Pitas, Directional statistics in nonlinear vector field filtering, Signal Processing, 38, 3, 299-316, 1994.

[217] N. Nikolaidis, I. Pitas, Multichannel L filters based on reduced ordering, IEEE Trans. on Circuits and Systems for Video Technology, 6, 5, 470-482, 1996.

[218] N. Nikolaidis, I. Pitas, Nonlinear processing and analysis of angular signals, IEEE Trans. on Signal Processing, 46, 3181-3194, 1998.

[219] N. Nordström, Biased anisotropic diffusion - a unified regularization and diffusion approach to edge detection, Image and Vision Computing, 8, 4, 318-327, 1990.

[220] K. Öistämö, Y. Neuovo, Video signal processing using vector median, Proc. of SPIE, 1360, 1171-1183, 1990.

[221] S. Osher, L.I. Rudin, Feature-oriented image enhancement using shock filters, SIAM Journal on Numerical Analysis, 27, 4, 919-940, 1990.

[222] H. Palus, Colour spaces, in [271], 67-89, 1998.

[223] J. Park. L. Kurz, Image enhancement using the modified ICM method, IEEE Trans. on Image Processing, 5, 765-771, 1996.

[224] J.R. Parker, "Algorithms for Image Processing and Computer Vision", Wiley, New York, 1997.

[225] E. Parzen, On estimation of a probability density function and mode, Ann. of Math. Statistics, 33, 1065-1076, 1962.

[226] S. Peltonen, M. Gabbouj, J. Astola, Nonlinear filter design: methodologies and challenges, in Proc. of the IEEE Symp. on Image and Signal Processing and Analysis (ISPA), Pula, Croatia, 102-106, 2001.

[227] P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion, Proc. of IEEE Workshop on Computer Vision, 16-22, Miami, USA, 1987.

[228] P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion, IEEE Trans. on PAMI, 12, 629-639, 1990.

[229] A.E.K. Pienkowski, "Artificial Colour Perception using Fuzzy Techniques in Digital Image Processing", Verlag TÜV, Rheinland, Köln, 1989.

[230] I. Pitas, A.N. Venetsanopoulos, A new filter structure for the implementation of certain classes of image processing operations, IEEE Trans. on Circuits and Systems, 35, 6, 636-647, 1988.

[231] I. Pitas, A.N. Venetsanopoulos, "Nonlinear Digital Filters, Principles and Applications", Kluwer Academic Publishers, Boston, 1990.

[232] I. Pitas, P. Tsakalides, Multivariate ordering in color image processing, IEEE Trans. on Circuits and Systems for Video Technology, 1, 3, 247-256, 1991.

[233] I. Pitas, A.N. Venetsanopoulos, Order statistics in digital image processing, Proc. of the IEEE, 12, 80, 1893-1921, 1992.

[234] I. Pitas, "Digital Image Processing Algorithms", Prentice Hall, 1993.

[235] I. Pitas, "Digital Image Processing Algorithms and Applications", Wiley, 2000.

[236] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Colour image processing using fuzzy vector directional filters, Proc. of the IEEE Workshop on Nonlinear Signal and Image Processing (NSIP), Greece, 535-538, 1995.

[237] K.N. Plataniotis, D. Androutsos, V. Sri, A.N. Venetsanopoulos, A nearest-neighbour multichannel filter, Electronic Letters, 31, 1910-1911, 1995.

[238] 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, 6, 6, 699-703, 1996.

[239] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Adaptive multichannel filters for color image processing, Proc. of the Visual Communications Image Processing Conf., SPIE 2727, Part III, 1270-1279, 1996.

[240] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Fuzzy adaptive filters for multichannel image processing, Signal Processing Journal, 55, 1, 93-106, 1996.

[241] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Multichannel filters for image processing, Signal Processing: Image Communications, 9, 2, 143-158, 1997.

[242] K.N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, A.N. Venetsanopoulos, Color image processing using adaptive multichannel filters, IEEE Trans. on Image Processing, 6, 7, 933-950, 1997.

[243] K.N. Plataniotis, A.N. Venetsanopoulos, Vector filtering, in [271], 188-209, 1998.

[244] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Color image processing using adaptive vector directional filters, IEEE Trans. on Circuits and Systems II, 45, 10, 1414-1419, 1998.

[245] K.N. Plataniotis, D. Androutsos, A.N. Venetsanopoulos, Adaptive fuzzy systems for multichannel signal processing, Proc. of the IEEE, 87, 9, 1601-1622, 1999.

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

[247] I. Pollak, A.S. Willsky, H. Krim, Image segmentation and edge enhancement with stabilized inverse diffusion equations, IEEE Trans. on Image Processing, 9, 2, 256-266, 2000.

[248] C.A. Poynton, "A Technical Introduction to Digital Video", Prentice Hall, Toronto, 1996.

[249] M.K. Prasad, Y.H. Lee, Stack filters and selection probabilities, IEEE Trans. on Signal Processing, 42, 2628-2643, 1994.

[250] W. Pratt, "Digital Image Processing", New York, John Willey & Sons, 1991.

[251] F. Preteux, N. Merlet, New concepts in mathematical morphology: the topographical distance function, Proc. of SPIE, 1568, 66-77, 1991.

[252] F. Preteux, On a distance function approach for gray-level mathematical morphology, in "Mathematical Morphology in Image Processing", E.R. Dougherty (Editor), Marcel Dekker, New York, 1993.

[253] M. Proesmans, E. Pauvels, L. van Gool, Coupled geometry-driven diffusion equations for low-level vision, in [262], 1994.

[254] P. Pujas, M.J. Aldon, Estimation of the colour image gradient with perceptual attributes, in Proc. of the 9th Int. Conf. on Image Analysis and Processing (ICIAP), Lecture Notes in Computer Science, 1310, 103-110, September 1997.

[255] G. Ramponi, The rational filter for image smoothing, IEEE Signal Processing Letters, 3, 3, 63-65, 1996.

[256] G. Ramponi, C. Moloney, Smoothing speckled images using an adaptive rational operator, IEEE Signal Processing Letters, 4, 3, 68-71, 1997.

[257] H. Rantanen, M. Karlsson, P. Pohjala, S. Kalli, Color video signal processing with median filters, IEEE Trans. on Consumer Electronics, 38, 157-161, 1992.

[258] C.S. Regazoni, A. Teschioni, A new approach to vector median filtering based on space filling curves, IEEE Trans. on Image Processing, 6, 990-1001, 1997.

[259] F. Rhodes, Digital metrics generated by families of paths, Discrete Mathematics, 135, 265-277, 1994.

[260] 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, 2, 111-128, 1999.

[261] S. Rital, H. Cherifi, Similarity hypergraph representation for impulsive noise reduction, Proc. of the 4th EURASIP Conf. focused on Video/Image Processing and Multimedia Communications (EC-VIP-MC), Zagreb, Croatia, July 2-4, 539-544, 2003.

[262] B. Romeny ter Haar, "Geometry-Driven Diffusion in Computer Vision", Kluwer, Boston, MA, 1994.

[263] B. Romeny ter Haar, L. Florack, J. Koendernick, M. Viergever, Scale-space theory in computer vision, First Int. Conf. on Scale-Space, Utrecht, The Netherlands, July 2-4, 1997, Lecture Notes in Computer Science 1252, 1997.

[264] M. Ropert, F.M. Saint-Martin, D. Pele, A new representation of weighted order statistic filters, Signal Processing, 54, 201-206, 1996.

[265] M. Rosenblatt, Remarks on some nonparametric estimates of a density function, Annals of Mathematical Statistics, 27, 832-835, 1956.

[266] N. Rougon, F. Preteux, Differential geodesic mathematical morphology, Proc. of SPIE Conf. on Mathematical Modelling, Bayesian Estimation and Inverse Problems, Denver, CO, 18-23 July, 3816, 294-305, 1999.

[267] J.C. Russ, "The Image Processing Handbook", CRC Press, 1995.

[268] P. Saint-Marc, J.S. Chen, G. Medioni, Adaptive smoothing: a general tool for early vision, Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 618-624, San Diego, 1989.

[269] P. Saint-Marc, J.S. Chen, G. Medioni, Adaptive smoothing: a general tool for early vision, IEEE Trans. on PAMI, 13, 6, 514-529, 1991.

[270] G.I. Sanchez-Ortiz, D. Rueckert,P. Burger, Knowledge-based tensor anisotropic diffusion of cardiac magnetic resonance images, Medical Image Analysis, 3, 2, 1-25, 1999.

[271] S.J. Sangwine, R.E.N. Horne,(Eds.), "The Colour Image Processing Handbook", Chapman & Hall, Cambridge, 1998.

[272] G. Sapiro, D.L. Ringach, Anisotropic diffusion of multivalued images with applications to color filtering, IEEE Trans. on Image Processing, 5, 11, 1582-1586, 1996.

[273] G. Sapiro, Geometric partial differential equations and image analysis, in "Short Course on Mathematical Methods in Speech and Image analysis", Institute of Mathematics and it's Applications, University of Minnesota, September 14, 2000.

[274] G. Sapiro, "Geometric Partial Differential Equations and Image Analysis", Cambridge University Press, Cambridge, MA, 2001.

[275] A. Scher, V. Dias, F.R. Rosenfeld, Some new image smoothing techniques, IEEE Trans. on SMC, 10, 153-158, 1980.

[276] O. Scherzer, J. Weickert, Relations between regularization and diffusion filtering, Journal of Math. Imaging and Vision, 12, 1, 43-63, 2000.

[277] P. Scheunders, J. Sijbers, Multiscale anisotropic filtering of color images, Proc. of ICIP, Thessaloniki, Greece, III, 170-173, October, 2001.

[278] M. Schmitt, Lecture notes on geodesy and morphological measurements, Proc. of the Summer School on Morphological Image and Signal Processing, Zakopane, Poland, 36-91, 1995.

[279] M. Schulze, J.A. Pearce, Some properties of the two-dimensional pseudomedian filter, Proc. of SPIE, 1451, 48-57, 1991.

[280] D.W. Scott, "Multivariate Density Estimation", Wiley, New York, 1992.

[281] G.A.F. Seber, "Multivariate Observations", Wiley, New York, 1984.

[282] J. Shah, A common framework for curve evolution, segmentation and anisotropic diffusion, in Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (ICPR), 136-142, June 1996.

[283] G. Sharma, H.J. Trussell, Digital color imaging, IEEE Trans. on Image Processing, 6, 901-932, 1997.

[284] G. Sharma, H.J. Trussell, Figures of merit for color scanners, IEEE Trans. on Image Processing, 6, 990-1001, 1997.

[285] F.Y. Shih, J.J. Liu, Size-invariant four-scan Euclidean distance transformation, Pattern Recognition, 31, 11, 1761-1766, 1998.

[286] B.W. Silverman, "Density Estimation for Statistics and Data Analysis", London, Chapman and Hall, 1986.

[287] D. Sindoukas, N. Laskaris, S. Fotopoulos., Algorithms for color image edge enhancement using potential functions, IEEE Signal Processing Letters, 4, 9, 269-272, 1997.

[288] W. Skarbek, "Metody Reprezentacji Obrazów Cyfrowych", Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1993.

[289] C.G. Small, A survey of multidimendional medians, Intrenational Statistical Review, 58, 3, 263-277, 1990.

[290] S.M. Smith, J.M. Brady, SUSAN - a new approach to low level image processing, Int. Journal of Computer Vision, 23, 1, 45-78, 1997.

[291] B. Smolka, Combined forward and backward anisotropic diffusion filtering of color images, Lecture Notes in Computer Science, LNCS 2449, 314-321, 2002.

[292] B. Smolka, Efficient modification of the central weighted vector median filter, Lecture Notes in Computer Science, LNCS 2449, 166-173, 2002.

[293] B. Smolka, On the similarity based impulsive noise removal technique for multichannel images, Lecture Notes in Computer Science, LNCS 2749, 1066-1073, 2003.

[294] B. Smolka, Adaptive modification of the vector median filter, Machine Graphics & Vision, 11, 2-3, 327-350, 2002.

[295] B. Smolka, On the new robust algorithm of noise reduction in color images, Computers & Graphics, 27, 503-513, 2003.

[296] B. Smolka, On the nonparametric impulsive noise reduction in multichannel images, Lecture Notes in Computer Science, LNCS 2652, 979-985, 2003.

[297] B. Smolka, New class of filters for impulsive noise removal in color images, Lecture Notes in Computer Science, LNCS 2652, 970-978, 2003.

[298] B. Smolka, K. Wojciechowski, A new method of texture binarization, Lecture Notes in Computer Science, LNCS 1296, 629-636, 1997.

[299] B. Smolka, K. Wojciechowski, Contrast enhancement of badly illuminated images, Lecture Notes in Computer Science, LNCS 1296, 271-278, 1997.

[300] B. Smolka, K. Wojciechowski, Edge preserving probabilistic smoothing algorithm, Lecture Notes in Computer Science, LNCS 1689, 41-48, 1999.

[301] B. Smolka, K. Wojciechowski, Contrast enhancement of gray scale images based on the random walk model, Lecture Notes in Computer Science, LNCS 1689, 411-418, 1999.

[302] B. Smolka, K. Wojciechowski, Random walk approach to image enhancement, Signal Processing, 81, 3, 465-482, 2001.

[303] B. Smolka, M. Szczepanski, K. Wojciechowski, K.N. Plataniotis, On the noise attenuation in DNA microarray images, Journal of Medical Informatics and Technologies, 2, 19-27, 2001.

[304] B. Smolka, A. Chydzinski, K. Wojciechowski, K. Plataniotis, A.N. Venetsanopoulos, On the reduction of impulsive noise in multichannel image processing, Optical Engineering, 40, 6, 902-908, 2001.

[305] B. Smolka, M.K. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, Fast modified vector median filter, Lecture Notes in Computer Science, LNCS 2124, 570-580, 2001.

[306] B. Smolka, M. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, Random walk approach to noise reduction in color images, Lecture Notes in Computer Science, LNCS 2124, 527-537, 2001.

[307] B. Smolka, K.N. Plataniotis, On the coupled forward and backward anisotropic diffusion scheme for color image enhancement, Lecture Notes in Computer Science, LNCS 2383, 70-80, 2002.

[308] B. Smolka, R. Lukac, Adaptive impulse detection based approaches for the noise reduction in heart image sequences, Journal of Medical Informatics and Technologies, 7-9, 21-30, 2002.

[309] B. Smolka, K.N. Plataniotis, A. Chydzinski, M. Szczepanski, A.N. Venetsanopulos, K. Wojciechowski, Self-adaptive algorithm of impulsive noise reduction in color images, Pattern Recognition, 35, 1771-1784, 2002.

[310] B. Smolka, R. Lukac, A. Chydzinski, K.N. Plataniotis, K. Wojciechowski, Fast adaptive similarity based impulsive noise reduction filter, Real Time Imaging, 9, 261-276, 2003.

[311] B. Smolka, K. Czubin, J.Y. Hardeberg, K.N. Plataniotis, M. Szczepanski, K. Wojciechowski, Towards automatic redeye effect removal, Pattern Recognition Letters, 24, 1767-1785, 2003.

[312] B. Smolka, A. Chydzinski, K.N. Plataniotis, A.N. Venetsanopoulos, New filtering technique for the impulsive noise reduction in color images, Mathematical Problems in Engineering, (in press).

[313] B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Nonlinear techniques for color image processing, in "Nonlinear Signal and Image Processing: Theory, Methods, and Applications", 445-505, Editors K.E. Barner , G.R. Arce, CRC Press, Boca Raton, FL., USA, 2004.

[314] B. Smolka, K.W. Wojciechowski, On the random walk approach to image enhancement, Proc. of the Second EURASIP Conf. (ECMCS), Kraków, 24-26 June, 1999.

[315] B. Smolka, K. Wojciechowski, M. Szczepanski, Random walk approach to image enhancement, Proc. of the Int. Conf. on Image Analysis and Processing (ICIAP), Venice, Italy, September 27-29, 1999.

[316] B. Smolka, M. Szczepanski, K. Wojciechowski, A. Swierniak, Random walk approach to the problem of impulse noise reduction, Proc. of the 5th Workshop on Colour Image Processing, Ilmenau, Germany, 7-8 October, 1999.

[317] B. Smolka, K. Wojciechowski, Edge preserving noise reduction algorithm, Proc. of the Second EURASIP Conf. (ECMCS), Kraków, 24-26 June, 1999.

[318] B. Smolka, K. Wojciechowski, M. Szczepanski, Detail preserving random noise reduction algorithm, Proc. of the 6th Int. Workshop on Systems, Signals and Image Processing (IWSSIP), Bratislava, Slovakia, June 2-4, 1999.

[319] B. Smolka, K. Wojciechowski, Detection of microcalcifications with the use of Levy flights model, Proc. of the Fifth Int. Workshop on Digital Mammography (IWDM), Toronto, Canada, June 11-14, 2000.

[320] B. Smolka, K. Wojciechowski, Edge preserving image smoothing based on self-avoiding random walk, Proc. of 15th Int. Conf. on Pattern Recognition (ICPR), Barcelona, Spain, Sept. 3-8, 2000.

[321] B. Smolka, K. Wojciechowski, New compression techniques for the presentation of printed documents on the WEB, Proc. of Electronic Imaging and the Visual Arts Conference (EVA), Moscow, 117-120, 2000.

[322] B. Smolka, A. Chydzinski, K. Wojciechowski, A. Swierniak, Self-avoiding random walk approach to image enhancement, Proc. of the 10th European Signal Processing Conference (EUSIPCO), Tampere, Finland, September 4-8, 2000.

[323] B. Smolka, K. Wojciechowski, K. Plataniotis, A.N. Venetsanopoulos, A new technique of the enhancement of scanned colour documents, Proc. of the Electronic Imaging and the Visual Arts Conference (EVA), Berlin, 178-189, 2000.

[324] B. Smolka, M.K. Szczepanski, K.W. Wojciechowski, A. Swierniak, Random walk approach to the problem of impulse noise reduction, Proc. of the 4th IFAC Symposium on Modelling and Control in Biomedical Systems, (Elsevier), Greifswald, Germany, 337-342, March 2000.

[325] B. Smolka, K. Wojciechowski, G. Zajaczkowski, From Gutenberg to Internet - Printed Documents on the WEB, Proc. of the Electronic Imaging and the Visual Arts Conference (EVA), Edinburgh, August 2000.

[326] B. Smolka, A. Swierniak, K. Wojciechowski, A. Chydzinski, Self-avoiding random walk approach to the enhancement of gel electrophoresis images, Proc. of the 10th European Signal Processing Conf. (EUSIPCO), September 4-8, Tampere, Finland, 2000.

[327] B. Smolka, M. Szlezak, K. Plataniotis, A. Venetsanopoulos, On the efficiency of the wavelet based compression techniques, Proc. of the Int. Conf. on Digital Imaging Technology (PICS), April 22-25, Montreal, Quebec, Canada, 267-270, 2001.

[328] B. Smolka, M. Szczepanski, K.N. Plataniotis, A. N. Venetsanopoulos, Random walk approach to noise reduction in color images, Proc. of IEEE Signal Processing Society, 2001 Int. Conf. on Image Processing (ICIP), Thessaloniki, Greece, October 7-10, 2001.

[329] B. Smolka, A. Chydzinski, K.N. Plataniotis, A.N. Venetsanopoulos, A similarity based approach to the reduction of impulsive noise in color images, IEEE Workshop on Nonlinear Signal and Image Processing (NSIP), Baltimore, Maryland, June 4-8, 2001.

[330] B. Smolka, M. Szczepanski, K.N. Plataniotis, A. N. Venetsanopoulos, On the fast modified vector median filter, Canadian Conf. on Electrical and Computer Engineering (CCECE), May 13-16, 2001, Toronto, Canada, 2001.

[331] B. Smolka, A. Chydzinski, M. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, On a new class of filters for the impulsive noise reduction in color images, Proc. of the 2nd Int. Symposium on Image and Signal Processing and Analysis (ISPA), June 19-21, Pula, Croatia, 2001.

[332] B. Smolka, M. Szczepanski, A. Chydzinsky, K.N. Plataniotis, A.N. Venetsanopoulos, new algorithm for impulsive noise reduction in color images, Proc. of Digital Signal Processing for Multimedia Communications and Services (ECMCS), Budapest, Hungary, 11-13 September 2001.

[333] B. Smolka, M. Szczepanski, A. Chydzinsky, K.N. Plataniotis A.N. Venetsanopoulos, on the class of noise reduction filters, Proc. of Digital Signal Processing for Multimedia Communications and Services (ECMCS), Budapest, Hungary, 11-13 September 2001.

[334] B. Smolka, M. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, Fast vector median filter modification, Proc. of the 10th Int. Conf. on System Modelling and Control (SMC), Zakopane, Poland, 21-25 May 2001.

[335] B. Smolka, M. Szczepanski, C. Canton-Ferrer, Efficient Algorithm for Impulsive Noise Reduction, Proc. of the Int. Conf. on Programmable Devices and Systems (PDS), Gliwice, Poland, 2001.

[336] B. Smolka, M. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, On the fast modification of the vector median filter, Proc. of ICPR, 931-934, Quebec City, Canada, 2002.

[337] B. Smolka, K.N. Plataniotis, M. Szczepanski, A.N. Venetsanopoulos, New efficient algorithm for the restoration of noisy color images, Proc. of the Electronic Imaging & Visual Arts (EVA), Florence, 100-103, 18-22 March, 2002.

[338] B. Smolka, R. Lukac, On the combined forward and backward anisotropic diffusion scheme for the multispectral image enhancement, Proc. of the Photogrammetric Computer Vision (PCV), Graz, Austria, September 9-13, 35, 3B, Commission III, 249-254, 2002.

[339] B. Smolka, M. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, Forward and backward anisotropic diffusion filtering for color image enhancement, Proc. of 14th Int. Conf. on Digital Signal Processing (DSP), Santorini, Greece, July 1-3, 2002.

[340] B. Smolka, C. Canton-Ferrer, M. Szczepanski, K. Wojciechowski, Evaluation of the efficiency of robust anisotropic diffusion schemes, Proc. of the Int. Conf. on Computer Vision and Graphics (ICCVG), Zakopane, Poland, 666-669, 25-29 September, 2002.

[341] B. Smolka, M. Studer, M. Stommel, K. Wojciechowski, Vector median based color gamma filter, Proc. of the Int. Conf. on Computer Vision and Graphics (ICCVG), Zakopane, Poland, 25-29 September, 2002.

[342] B. Smolka, M.K. Szczepanski, K.N. Plataniotis, A.N. Venetsanopoulos, On the modified weighted vector median filter, Proc. of Digital Signal Processing (DSP), Santorini, Greece, 2, 939-942, 2002.

[343] B. Smolka, M. Stommel, M. Studer, K. Wojciechowski, New class of noise reduction filters based on kernel density estimation, Proc. of the Int. Conf. on Computer Vision and Graphics (ICCVG), Zakopane, Poland, 25-29 September 2002.

[344] B. Smolka, R. Lukac, K.N. Plataniotis, New algorithm for noise attenuation in color images based on the central weighted vector median filter, Proc. of the 9th Int. Workshop on Systems, Signals and Image Processing (IWSSIP), Manchester, UK, 544-548, November 7-8, 2002.

[345] B. Smolka, M. Szczepanski, K.N. Plataniotis, C. Canton-Ferrer, On the efficiency of robust anisotropic diffusion filtering schemes, Proc. of the 9th Int. Workshop on Systems, Signals and Image Processing (IWSSIP), Manchester, U.K., Addendum 1-5, November7-8, 2002.

[346] B. Smolka On the fast nonparametric technique of multichannel image enhancement, 15-22, Spectral Imaging, Proc. of the Int. Workshop of the Carinthian Tech Research AG, (Ed. R. Leitner), 15-22, 3 April, Graz, 2003,

[347] B. Smolka Fast similarity based impulsive noise reduction technique for color images, Spectral Imaging, Proc. of the Int. Workshop of the Carinthian Tech Research AG, (Ed. R. Leitner), 23-30, 3 April, Graz, 2003,

[348] B. Smolka, K.N. Plataniotis, R. Lukac, and A.N. Venetsanopoulos, Noise reduction in DNA microarray images based on digital path approach, Proc. of 5th IFAC Symposium 2003 on Modelling and Control in Biomedical Systems, 17-22, (Elsevier), Melbourne, Australia, August 2003.

[349] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, On the forward and backward anisotropic diffusion framework, Proc. of the 2003 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP), Grado, Italy, June 8-11, 2003.

[350] B. Smolka, R. Lukac, K.N. Plataniotis, A.N. Venetsanopoulos, Modified anisotropic diffusion framework, Proc. of Visual Communication and Image Processing Conf. (VCIP), Lugano, Switzerland, July 8-11, 2003.

[351] B. Smolka, R. Lukac, K.N. Plataniotis, A.N. Venetsanopoulos, Application of kernel density estimation for color image filtering, Proc. of Visual Communication and Image Processing (VCIP), Lugano, Switzerland, July 8-11, 2003.

[352] B. Smolka, R. Bieda, R. Lukac, K.N. Plataniotis, Application of forward and backward anisotropic diffusion framework to color image denoising, Proc. of the 9th Int. Workshop on Systems, Signals and Image Processing (IWSSIP), Prague, Czech Republic, September 10-11, 2003.

[353] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, Kernel density estimation based impulsive noise reduction filter, Proc. of the 2003 Int. Conf. on Image Processing (ICIP), Barcelona, Spain, September 14-17, 2003.

[354] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, Similarity based impulsive noise removal in color images, Proc. of the 2003 Int. Conf. on Image Processing (ICIP), Barcelona, Spain, September 14-17, 2003.

[355] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, On the similarity based impulsive noise removal in color images. Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME), Baltimore, USA, I, 585-588, July 6-9, 2003.

[356] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, Robust retrieval of fine art paintings based on kernel density estimation, Proc. of the SPIE 5146, 137-148, 2003.

[357] B. Smolka, R.Lukac, Color image enhancement based on nonparametric density estimation, Proc. of the 4th EURASIP Conf. focused on Video/Image Processing and Multimedia Communications (EC-VIP-MC), Zagreb, Croatia, I, 271-276, July 2-4, 2003.

[358] B. Smolka, R. Lukac, Fast similarity based impulsive noise reduction technique for color images, Proc. of the 4th EURASIP Conf. focused on Video/Image Processing and Multimedia Communications (EC-VIP-MC), Zagreb, Croatia, I, 251-256, July 2-4, 2003.

[359] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, Nonparametric technique of noise reduction in color images, Proc. of SPIE, 5146, 254-265, 2003.

[360] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, Vector median based sigma filter, Proc. of the 2003 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP), Grado, Italy, June 8-11, 2003.

[361] B. Smolka, R. Bieda, R. Lukac, K.N. Plataniotis, Adaptive modification of the vector median filter, Proc. of the 9th Int. Workshop on Systems, Signals and Image Processing (IWSSIP), Prague, Czech Republic, September 10-11, 2003.

[362] B. Smolka, K.N. Plataniotis, R. Lukac, A.N. Venetsanopoulos, New class of impulsive noise reduction filters based on kernel density estimation, Proc. of the 28th IEEE Int. Conf. on Acoustic, Speech & Signal Processing (ICASSP), Hong Kong, III, 721-724, April 6-10, 2003.

[363] B. Smolka, M. Szczepanski, R. Lukac, A.N. Venetsanopoulos, Robust color image retrieval for the world wide web, Proc. of the IEEE Int. Conf. on Acoustic, Speech & Signal Processing (ICASSP), Montreal, Canada, May 2004.

[364] B. Smolka, A.N. Venetsanopoulos, R. Lukac, K.N. Plataniotis, Nonparametric Approach to Color Based Image Retrieval, Proceedings of the 12th European Signal Processing Conference, September 6-10, Vienna, Austria, 2004.

[365] B. Smolka, S. Budzan, A.N. Venetsanopoulos, R. Lukac, Nonparametric Technique of Impulsive Noise Removal Proceedings of the 12th European Signal Processing Conference, September 6-10, Vienna, Austria, 2004.

[366] N. Sochen, R. Kimmel, R. Malladi, A geometrical framework for low level vision, IEEE Trans. on Image Processing, 17, 3, 310-318, 1998.

[367] M. Sonka, V. Hlavac, R. Boyle, "Image Processing, Analysis and Machine Vision", International Thomson Computer Press, London 1993.

[368] A. Somboonkaew, S. Chitwong, F. Cheevasuvit, K. Dejhan, S. Mitatha, Segmentation on the edge preserving smoothing image, Proc. of 20th Asian Conf. on Remote Sensing, November 22-25, 1999, Hong Kong, China.

[369] M. Spann, A. Nieminen, Adaptive Gaussian weighted filtering for image segmentation, Pattern Recognition Letters, 8, 251-255, 1988.

[370] F. Spitzer, "Principles of Random Walk", D. van Nostrand Company, Princeton, 1975.

[371] V. Starovoitov, Towards a distance transform generalization, Proc. of the 9th Scandinavian Conf. on Image Analysis, (Ed. G. Borgefors), Uppsala, 499-506, 1995.

[372] H.M.G. Stokman, T. Gevers, Density estimation for color images, Journal of Electronic Imaging, 1, 10, 2001.

[373] T. Sun, M. Gabbouj, Y. Neuovo, Center weighted median filters, some properties and applications in image processing, Signal Processing, 35, 213-229, 1994.

[374] T. Sun, M. Gabbouj, Y. Neuvo, Adaptive L-filters with applications in signal and image processing, Signal Processing, 38, 331-344, 1994.

[375] K.K. Sung, A vector signal processing approach to color, M.S. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1992.

[376] J.S. Suri, D. Wu, J. Gao, S. Singh, S. Laxminarayan, A comparison of state-of-the-art diffusion imaging techniques for smoothing medical/non-medical image data, Proc. of the 15th Int. Conf. on Pattern Recognition (ICPR), Quebec, 11-15 August, 2002.

[377] M. Szczepanski, B. Smolka, K. Wojciechowski, K.N. Plataniotis, A. N. Venetsanopoulos, On the new filtering technique of noise attenuation in color images, Proc. of the 2nd Int. Symp. on Image and Signal Processing and Analysis (ISPA), June 19-21, 2001, Pula, Croatia.

[378] M. Szczepanski, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, Enhancement of the DNA microarray chip images, Proc. of Digital Signal Processing Conf. (DSP), Santorini, Greece, 2002.

[379] M. Szczepanski, B. Smolka, K.N. Plataniotis, A.N. Venetsanopoulos, On the geodesic paths approach to color image filtering, Signal Processing, 83, 1309-1342, 2003.

[380] M. Szczepanski, B. Smolka, K.N. Plataniotis, A.N.Venetsanopoulos, On the distance function approach to color image enhancement, Discrete Applied Mathematics, 139, 283-305, 2004

[381] R. Tadeusiewicz R., M. Flasiński, "Rozpoznawanie Obrazów", PWN, Warszawa, 1991.

[382] R. Tadeusiewicz, "Systemy Wizyjne Robotów Przemysłowych", WNT, Warszawa 1992.

[383] R. Tadeusiewicz, P. Korohoda, "Komputerowa Analiza i Przetwarzanie Obrazów", Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków, 1997.

[384] K. Tang, J. Astola, Y. Neuovo, Multichannel edge enhancement in color image processing, IEEE Trans. on Circuits and Systems for Video Technology, 4, 5, 468-479, 1994.

[385] K. Tang, J. Astola, Y. Neuvo, Nonlinear multivariate image filtering techniques, IEEE Trans. on Image Processing, 4, 788-798, 1995.

[386] B. Tang, G. Sapiro, V. Caselles, Color image enhancement via chromaticity diffusion, IEEE Trans. on Image Processing, 10, 701-707, 2001.

[387] T. Tasdizen, R. Whitaker, P. Burchard, S. Osher, Geometric surface smoothing via anisotropic diffusion of normals, in Proc. of IEEE Visualization, Boston, MA, USA, 125-132, 2002.

[388] L. Tenze, S. Carrato, G. Ramponi, An alignment algorithm for old motion pictures, IEEE Signal Processing Letters, 9, 10, 2002.

[389] S. Theodoris, K. Koutroumbas, "Pattern Recognition", Academic Press, 1998.

[390] P.J. Toivanen, New geodesic distance transforms for gray scale images, Pattern Recognition Letters, 17, 437-450, 1996.

[391] P.J. Toivanen, J. Vartiainen, A. Kaarna, Multidimensional median filtering of spectral images, Second European Conference on Color in Graphics, Imaging and Vision, (CGIV), Aachen, Germany, 398-402, April 5-8, 2004.

[392] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color image, Proc. of the IEEE Int. Conf. on Computer Vision, Bombay, India, 1998.

[393] F. Tomita, S. Tsuji, Extraction of multiple regions by smoothing in selected neighbourhoods, IEEE Trans. on Systems, Man and Cybernetics, 7, 107-109, 1977.

[394] F. Torkamani-Azar, K.E. Tait, Image recovery using the anisotropic diffusion equation, IEEE Trans. on Image Processing, 5, 11, 1573-1578, 1996.

[395] P.E. Trahanias, A.N. Venetsanopoulos, Vector directional filters: a new class of multichannel image processing filters, IEEE Trans. on Image Processing, 2, 4, 528-534, 1993.

[396] 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.

[397] P.E. Trahanias, D.G. Karakos, A.N. Venetsanopoulos, Directional processing of color images: theory and experimental results, IEEE Trans. on Image Processing, 5, 6, 868-881, 1996.

[398] D. Tschumperlé, R. Deriche, Constrained and unconstrained PDE's for vector image restoration, Proc. of the SCIA Conference, Bergen, Norway, June 11-14, 2001.

[399] D. Tschumperlé, R. Deriche, Diffusion PDE's on vector-valued images: local approach and geometric viewpoint, IEEE Signal Processing Magazine, 5, 19, 15-25, 2002.

[400] D. Tschumperlé, R. Deriche, Diffusion PDE's on vector-valued images: local approach and geometric viewpoint, IEEE Signal Processing Magazine, September 2002.

[401] K.P. Valavanis, J. Zheng, G. Paschos, A total color difference measure for segmentation in color images, Journal of Intelligent and Robotic Systems, 16, 269-313, 1996.

[402] M.I. Vardavoulia, I. Andreadis, P. Tsalides, A new vector median filter for colour image processing, Pattern Recognition Letters, 22, 675-689, 2001.

[403] A.N. Venetsanopoulos, K.N. Plataniotis, Multichannel image processing, Proc. of the IEEE Workshop on Nonlinear Signal and Image Processing (NSIP), 2-6, 1995.

[404] A.N. Venetsanopoulos, K.N. Plataniotis, M.K. Szczepanski, B. Smolka. On the geodesic paths approach to multichannel signal processing, Proc. of the IEEE Int. Symp. on Circuits and Systems, Scottsdale, Arizona, 26-29 May, 2002.

[405] C. Vertan, M. Malciu, V. Buzuloiu, V. Popescu, Median filtering techniques for vector valued signals, Proc. of the IEEE Int. Conf. on Image Processing (ICIP), Lausanne, Switzerland, I, 977-980, 16-19 September, 1996.

[406] C. Vertan, N. Boujemaa, V. Buzuloiu, A fuzzy color credibility approach to color image filtering, Proc. of IEEE Int. Conf. on Image Processing (ICIP), Vancouver, Canada, II, 808-811, 10-13 September, 2001.

[407] T. Viero, K. Oistämö, Y. Neuvo, Three-dimensional median-related filters for color image sequence filtering, IEEE Trans. on Circiuts and Systems for Video Technology, 4, 2, 129-142, 1994.

[408] M.P. Wand, M.C. Jones, "Kernel Smoothing", Chapman & Hall, 1995.

[409] D.C.C. Wang, A.H. Vagnucci, C.C. Li, Gradient inverse weighted smoothing scheme and the evaluation of its performance, Computer Graphics and Image Processing, 15, 167-181, 1981.

[410] D. Wang, Q. Wang, A weighted averaging method for image smoothing, Proc. of ICPR, 981-983, Paris, 1988.

[411] Z. Wang, D. Zhang, Progressive switching median filter for the removal of impulse noise from highly corrupted images, IEEE Trans. on Circuits and Systems II, 46, 78-80, 1999.

[412] A.B. Watson, J. Hu, J.F. McGovan, Digital video quality metric based on human vision, Journal of Electronic Imaging, 1, 10, 20-29, 2001.

[413] J. Weickert J, "Anisotropic Diffusion in Image processing", G.G. Teubner, Stuttgart, 1998.

[414] R. Whitaker, G. Gerig, Vector-valued diffusion, in [262], 93-134, 1994.

[415] R. Wichman, K. Oistamo, Q. Liu, M. Grundstrom, Y. Neuovo, Weighted vector median operation for filtering multispectral data, Proc. of the Conf.: Visual Communications and Image Processing (VCIP), 376-383, 1992.

[416] M. Wild, Idempotent and co-idempotent stack filters and min-max operators, Theoretical Computer Science, 603-631, 2003.

[417] S.S. Wilks, Certain generalizations in the analysis of variance, Biometrika, 24, 471-494, 1932.

[418] A.P. Witkin, Scale-space filtering, Proc. of the 4th Int. Joint Conf. on Artificial Intelligence, 1019-1022, 1983.

[419] G. Wyszecki, W.S. Stiles, "Color Science, Concepts and Methods, Quantitative Data and Formulas", Wiley, N.Y., 2nd Edition, 1982.

[420] R. Yang, L. Yin, M. Gabbouj, J. Astola, Y. Neuvo, Optimal weighted median filtering under structural constraints, IEEE Trans. on Signal Processing, 43, 591-604, 1995.

[421] L. Yaroslavsky, E. Murray, "Fundamentals of Digital Optics", Birkhäuser, Boston, 1996.

[422] L. Yin, J. Astola, Y. Neuvo, Adaptive stack filtering with application to image processing, IEEE Trans. on Signal Processing-II, 41, 162-184, 1993.

[423] L. Yin, Y. Neuvo, Fast adaptation and performance characteristics of FIR-WOS hybrid filters, IEEE Trans. on Signal Processing-II, 41, 1610-1628, 1994.

[424] L. Yin, R. Yang, M. Gabbouj, Y. Neuvo, Weighted median filters: a tutorial, IEEE Trans. on Circuits and Systems II, 43, 157-192, 1996.

[425] Y.L. You, W. Xu, A. Tannenbaum, M. Kaveh, Behavioral analysis of anisotropic diffusion in image processing, IEEE Trans. on Image Processing, 5, 11, 1539-1553, 1996.

[426] P. T. Yu, W. H. Liao, Weighted order statistics filters - their classification, some properties, and conversion algorithm, IEEE Trans. on Signal Processing, 42, 2678-2691, 1994.

[427] P. Zamperoni, "Methoden der Digitalen Bildsignalverarbeitung", Vieweg, Braunschweig, 1991.

[428] S. Di Zenzo, A note on the gradient of multi-image, Computer Vision Graphics Image Processing, 33, 116-125, 1986.

[429] S. Zhang, M.A. Karim, A new impulse detector for switching median filters, IEEE Signal Processing Letters, 9, 360-363, 2002.

[430] J. Zheng, K.P. Valavanis, J.M. Gauch, Noise removal from color images, Journal of Intelligent and Robotic Systems, 7, 257-285, 1993.

Abstract

NONLINEAR TECHNIQUES OF NOISE REDUCTION IN DIGITAL COLOR IMAGES

This monograph details the author's most important contributions to the rapidly growing field of nonlinear noise reduction in color images. Its content is structured into seven Chapters.

The first Chapter describes the fundamentals of color image processing and also presents the sources of image noise, describes their models and defines measures of the quality of image restoration.

The second Chapter is focused on the nonlinear adaptive schemes of noise reduction applied in gray scale imaging, which are very often extendable into the multichannel case.

Chapter 3 provides the state of the art in color image filtering and serves as a basis for the remaining Chapters, in which author's original contributions are presented.

In the next Chapter, the robust anisotropic diffusion filtering scheme, which ignores the central pixel of the filtering window, when building the weighted average of the input samples is introduced. This improvement allows to use the anisotropic technique for the suppression of strong Gaussian and heavy tailed noise, as the influence of the central, corrupted pixel is diminished by an appropriate setting of the conductivity coefficients. In this Chapter the iterative forward and backward diffusion technique is also presented.

Chapter 5 is devoted to the development of a powerful class of filters, based on the digital paths concepts and fuzzy similarity measures among pixels in neighborhood relation. This novel technique, which utilizes the connection between image pixels, instead of window based structures, is an extension of the adaptive noise reduction filtering and anisotropic diffusion techniques and is shown to have advantages over traditional methods. The extensive simulations reveal that the proposed filtering framework significantly excels over the standard methods and can be applied for the removal of both Gaussian and impulsive noise.

In the next Chapter the problem of nonparametric impulsive noise reduction in multichannel images is addressed. A new family of filters for noise attenuation elaborated by the author, based on the nonparametric probability density estimation of the sample data, is introduced and its relationship to commonly used filtering techniques is investigated.

The last Chapter deals with the adaptive optimization of the weighted vector median filters and also introduces the new technique based on the so called sigma-filtering. This novel adaptive technique is based on robust order statistic concepts and simplified statistical measures of vectors' dispersion.

Streszczenie

NIELINIOWE TECHNIKI REDUKCJI SZUMU W BARWNYCH OBRAZACH CYFROWYCH

Redukcja szumów jest jednym z najważniejszych etapów przetwarzania wstępnego obrazów cyfrowych. Efektywna filtracja sygnału wizyjnego warunkuje bowiem sukces dalszych etapów jego przetwarzania. Problem redukcji szumów jest szczególnie trudny w przypadku obrazów barwnych, albowiem nie została jak dotąd stworzona spójna teoria umożliwiająca bezpośrednią implementację dobrze poznanych filtrów eliminacji szumów w obrazach z poziomami szarości do poprawy jakości obrazów wielokanałowych.

W ciągu ostatnich lat zaproponowano liczne algorytmy redukcji szumów w obrazach barwnych. Najprostszą klasą są filtry liniowe, które mogą efektywnie usuwać addytywne szumy gaussowskie, jednakże nie są one zdolne do adaptacji do nieliniowości występujących w obrazie, co prowadzi do rozmywania krawędzi obiektów oraz innych, ważnych z punktu widzenia percepcji człowieka oraz dalszych etapów przetwarzania, struktur obrazu.

Aby poprawić efektywność filtracji szumów, na przestrzeni ostatnich lat zaproponowano różnorodne techniki nieliniowe, z których najpopularniejszą grupę stanowią filtry bazujące na statystykach porządkowych. Filtry rangowe, minimalizujące skumulowaną funkcję dystansową, są skuteczne w usuwaniu szumów impulsowych, jednakże ich wadą jest zbyt duża inwazyjność, manifestująca się w zastępowaniu nie tylko pikseli obrazu, które uległy kontaminacji, ale także pikseli oryginalnych, co prowadzi do destrukcji drobnych struktur obrazu o wielkości porównywalnej z wymiarami okna filtracyjnego. Dodatkową wadą tych filtrów jest ich nieskuteczność w redukcji szumu gaussowskiego.

Niniejsza monografia stanowi podsumowanie wysiłku badawczego autora w dziedzinie filtracji szumów występujących w barwnych obrazach cyfrowych. W pracy przedstawiono różnorodne klasy filtrów zaprojektowanych do eliminacji zakłóceń impulsowych, szumów gaussowskich oraz najbardziej degradujących obraz szumów mieszanych. Przedstawione w monografii algorytmy cechują się bardzo dobrą efektywnością, przewyższającą znacznie algorytmy standardowe, oraz niską złożonością obliczeniową, umożliwiającą ich zastosowanie w realizacjach praktycznych, szczególnie w systemach wizyjnych czasu rzeczywistego.

Rozdział pierwszy monografii stanowi wprowadzenie do problematyki przetwarzania barwnych obrazów cyfrowych. W rozdziale tym przedstawiono podstawowe koncepcje tworzenia wielokanałowego obrazu cyfrowego i jego filtracji, koncentrując się na problemie zakłóceń obrazu powstających w procesie jego akwizycji, przetwarzania, transmisji oraz przechowywania na nośnikach danych. W rozdziale tym wprowadzono modele szumów symulujących rzeczywiste zakłócenia oraz przedstawiono metody oceny jakości obrazów cyfrowych umożliwiające ewaluację efektywności różnorodnych metod redukcji artefaktów wywołanych przez zjawiska szumu.

W rozdziale drugim przedstawiono przegląd adaptacyjnych technik redukcji szumów gaussowskich, impulsowych oraz mieszanych w obrazach z poziomami szarości. W rozdziale tym omówiono algorytmy oparte na koncepcji nieliniowej średniej ważonej oraz dokonano przeglądu metod bazujących na statystykach porządkowych. Szczególną uwagę poświęcono ważonej medianie oraz iteracyjnym algorytmom wyznaczania optymalnych współczynników wagowych ze względu na zastosowanie tych metod do optymalizacji filtrów wektorowych przedstawionych w rozdziale siódmym.

Rozdział trzeci poświęcony jest omówieniu metod redukcji szumów występujących w barwnych obrazach cyfrowych. Szczegółowo opisano filtry oparte na statystykach porządkowych, transformacjach wykorzystujących koncepcje teorii zbiorów rozmytych, a także metody wykorzystujące estymację nieparametryczną. Szczególną uwagę poświęcono ważonej medianie wektorowej oraz zaproponowanej przez autora jej modyfikacji, prowadzącej do przyśpieszenia algorytmu oraz poprawy efektywności procesu filtracji.

Rozdział czwarty, nawiązujący do rozdziału drugiego, poświęcony jest dyfuzji anizotropowej, stanowiącej skuteczną metodę redukcji szumów gaussowskich. W rozdziale tym przedstawione zostały wyniki prac autora nad modyfikacją algorytmu dyfuzji anizotropowej, poprzez minimalizację wpływu centralnego piksela maski filtracyjnej, umożliwiającą także redukcję szumów impulsowych. W rozdziale tym opisano ponadto opracowaną przez autora metodę iteracyjną, opartą na technice nieostrego maskowania, wykorzystującą tak zwaną dyfuzję odwrotną do poprawy jakości obrazów, które uległy kontaminacji szumem gaussowskim.

Koncepcja minimalizacji wpływu centralnego piksela w masce filtracyjnej została rozwinięta w rozdziale piątym, w którym przedstawiono wyniki prac autora nad nową klasą filtrów opartych na ścieżkach cyfrowych i elementach teorii zbiorów rozmytych. Algorytmy redukcji szumów, wykorzystujące ideę eksploracji otoczenia centralnego piksela maski filtracyjnej przez ścieżki cyfrowe wyznaczające poprzez funkcję kosztu optymalne połączenia pikseli obrazu, cechują się świetną efektywnością redukcji szumów impulsowych, gaussowskich i mieszanych.

Opracowane przez autora metody stanowią uogólnienie i rozwinięcie dyfuzji anizotropowej przedstawionej w rozdziale czwartym i stanowią jego najbardziej znaczący wkład w rozwój nieliniowych metod redukcji szumów w barwnych obrazach cyfrowych.

Rozdział szósty poświęcony jest zastosowaniu estymacji nieparametrycznej do filtracji szumów impulsowych. W rozdziale tym przedstawiono ogólną koncepcję filtrów opartych na estymacie nieparametrycznej, wskazując na ich podobieństwo do mediany wektorowej, w przypadku gdy funkcja jądra ma postać funkcji liniowej. W rozdziale tym wprowadzono także rodzinę filtrów cechującą się dużą skutecznością w redukcji szumów impulsowych oraz zdolnością do zachowywania krawędzi obrazu i jego tekstury. Własności te osiągane są przez zaimplementowane mechanizmy adaptacyjne, dostosowujące parametry filtrów do struktur morfologicznych obrazu oraz poziomu jego zakłóceń. Na uwagę zasługuje mała złożoność obliczeniowa przedstawionych klas filtrów, pozwalająca na ich zastosowanie do przetwarzania obrazów w czasie rzeczywistym.

W rozdziale siódmym przedstawiono nowe metody optymalizacji ważonej mediany wektorowej za pomocą optymalizacji liniowej oraz sigmoidalnej, omówionej w rodziale drugim. Przedstawione metody optymalizacji, operujące zarówno na chrominancji, jak i na luminancji obrazu, prowadzą do wyznaczania optymalnych z punktu widzenia zadanej funkcji kosztu współczynników wektora wag. W rozdziale tym wprowadzono także adaptacyjną metodę eliminacji szumów impulsowych opartą na estymacji dyspersji elementów obrazu zawartych w oknie filtracyjnym. Ta nowa klasa filtrów, bazująca na koncepcji filtru typu sigma, charakteryzuje się dużą efektywnością redukcji szumów impulsowych oraz niską złożonością obliczeniową.

* *
*

Symbols

- a - angle
- A - accumulated angles
- α - parameter
- B - number of bits of an image channel
- β - parameter
- c - conductivity coefficient
- $\bar{c}$ - regularized conductivity coefficient
- C - parameter
- $\mathcal{C}$ - FB conductivity
- $\mathbf{C}_{\mathbf{x}}$ - chromaticity of $\mathbf{x}$
- δ - parameter
- ΔE - color difference
- $\mathbf{D}$ - vector direction
- $\mathcal{D}$ - nonempty set
- d - distance parameter
- e - estimation error
- $\mathcal{E}$ - energy
- ϵ - adaptation step-size
- E - statistical expectation
- η - digital path length
- f - decreasing function
- γ - parameter
- Γ_R - regularization function
- Γ - Gamma function
- $\Gamma^{W,\eta}$ - minimal connection cost
- Γ_N - number of pixels in a hypercube
- $\mathbf{G}$ - gradient
- g - gradient magnitude
- h - kernel bandwidth
- h_N - length of a hypercube edge
- $\mathcal{H}$ - digital lattice
- ι - neighborhood parameter
- (i, j) - discrete image plane coordinates
- i - pixel's position on the image domain
- κ - power parameter
- K_1, K_2 - image domain dimensions
- L - Minkowski norm
- $\mathcal{L}^*$ - likelihood
- $\mathcal{L}$ - digital path length
- l - image dimension
- λ - parameter
- Λ - connection cost measure
- m - number of image channels
- μ - median rank
- $M_{\mathbf{x}}$ - magnitude of vector $\mathbf{x}$
- n - iteration number

- N - number of pixels in W
- $\mathbb{N}$ - natural numbers
- $\mathcal{N}$ - neighborhood relation
- ν - local variance
- $\mathbf{o}$ - original, uncorrupted image
- Ω - image domain
- $\mathfrak{n}$ - number of digital paths in W
- p - probability of noise corruption
- $p_N(\mathbf{x})$ - probability of event $\mathbf{x}$
- ϕ - window function defining a hypercube
- Φ - flux magnitude
- ψ - weighting coefficient, similarity
- Ψ - cumulated similarities
- Ψ^W - digital paths contained in W
- ρ - distance
- $\mathbf{R}$ - correlation matrix
- $\mathbb{R}$ - real numbers
- Q - number of image pixels
- $\mathcal{Q}$ - continuous path
- $\mathbb{Q}$ - digital path
- q - point on a path
- S - normalizing constant
- $\mathcal{S}$ - sign function
- $\tilde{\mathcal{S}}$ - sigmoidal approximation of $\mathcal{S}$
- $\mathbb{S}^2$ - unit ball in RGB
- σ - standard deviation
- t - time
- T - total processing time
- τ - design parameter
- $\mathbb{T}^2$ - Maxwell triangle
- Υ - angular sign transformation
- ϱ - inter-quartile range
- ς - relation between pixels
- $\mathbf{v}$ - noise process
- V_N - hypercube volume
- W - filtering window
- $\mathbb{W}$ - planar subset of $\mathbb{R}^2$
- $\mathcal{W}$ - set of ordered samples from W
- (ξ, η) - continuous domain coordinates
- $\tilde{\xi}$ - weighting parameter
- Ξ - sigmoidal function
- x - noisy gray scale image
- $\mathbf{x}$ - noisy multichannel image
- $\mathbf{x}_k$ - k^{th} sample in W
- $\mathbf{x}_{(k)}$ - k^{th} sample in ordered set
- $\mathbf{x}_1$ - central pixel in W
- $\hat{\mathbf{x}}_k$ - nearest neighbor of $\mathbf{x}_k$
- $\mathbf{y}$ - filter output
- ζ - flux function
- $\mathbb{Z}$ - integer numbers
- $\star$ convolution
- $\times$ Cartesian product
- $\sim$ neighborhood relation in W
- $\prec$ vector ordering
- $\leftrightarrow$ neighborhood relation
- $\Leftrightarrow$ connectivity relation on a digital path
- $\|\cdot\|$ vector norm
- $<\cdot>$ expected value
- $\hat{x}$ mean of x

Acronyms

- AD - Anisotropic Diffusion
- AHDF - Adaptive Hybrid Directional Filter
- AMF - Arithmetic Mean Filter
- ANNF - Adaptive Nearest Neighbor Filter
- ANMF - Adaptive Nonparametric Multichannel Filter
- ANNMF - Adaptive Nearest Neighbor Multichannel Filter
- ANPDF - Adaptive Nonparametric Directional Filter
- ANPEF - Adaptive Nonparametric Exponential Filter
- ANPGF - Adaptive Nonparametric Gaussian Filter
- ANPF - Adaptive Nonparametric Filter
- BVDF - Basic Vector Directional Filter
- CLMMF - Crossing Level Median Mean Filter
- CIE - Commission Internationale de L'Eclairage
- CCD - Charge-Coupled Device
- CWM - Central Weighted Median
- CWVMF - Central Weighted VMF
- DDF - Directional Distance Filter
- DPA - Digital Paths Approach
- DPAL - DPA-*Last* Technique
- DPAF - DPA-*First* Technique
- DTOCS - Distance Transform on Curved Space
- EVMF - Extended Vector Median Filter
- FOVDF - Fuzzy Ordered Vector Directional Filter
- FB - Forward & Backward Diffusion
- FDPA - Fast Digital Paths Approach
- FVDF - Fuzzy Vector Directional Filter
- FWAF - Fuzzy Weighted Average Filters
- GDF - Geometric Diffusion Filter

- GVDF - Generalized Vector Directional Filter
- HDF - Hybrid Directional Filter
- HSV - Hue, Saturation, Value color space
- LUM - Lower-Upper-Middle
- LMS - Least Mean Squared
- LWVDF - Linearly Optimized WVDF
- MCWVMF - Modified CWVMF
- MED - Median
- MF - Median Filter
- MLE - Maximum Likelihood Estimate
- MMF - Marginal Median Filter
- NMSE - Normalized Mean Squared Error
- NCD - Normalized Color Difference
- PDE - Partial Derivative Equation
- PDF - Probability Density Function
- PM - Perona & Malik
- PMAD - Perona-Malik Anisotropic Diffusion
- PSNR - Peak Signal to Noise Ratio
- RCMFm - Rank Conditioned Median Filter (marginal)
- RCRS - Rank-Conditioned Rank-Selection
- RCVMF - Rank Conditioned Vector Median Filter
- RGB - Red, Green, Blue color space
- RMSE - Root Mean Squared Error
- ROF - Rank Order Filter
- SGF - Symmetric Gradient Filter
- SNR - Signal to Noise Ratio
- SBVDF - Sigma Basic Vector Directional Filter
- SDDF - Sigma Directional Distance Filter
- SD-ROM - Signal-Dependent Rank Ordered Mean
- SVMF - Sigma Vector Median Filter
- SWVF - Selection Weighted Vector Filter
- SWVDF - Sigmoidally Optimized WVDF
- TVMF - Thresholded Vector Median Filter
- VBAMMF - Vector Bayesian Adaptive Median-Mean Filter
- VMF - Vector Median Filter
- WM - Weighted Median
- WVMF - Weighted Vector Median Filter
- WVDF - Weighted Vector Directional Filters

Index

Wydano za zgodą
Rektora Politechniki Śląskiej

WYDAWNICTWO POLITECHNIKI ŚLĄSKIEJ
ul. Akademicka 5, 44-100 Gliwice, tel. (0-32) 237-13-81
http://loki.polsl.pl/wydawnictwo

Sprzedaż i Marketing
tel. (0-32) 237-18-48
wydawnictwo_mark@polsl.pl

Nakł. 100+50	Ark. wyd. 15	Ark. druk. 13,375	Papier offset. 70x100, 80 g
Oddano do druku 22.04.2004 r.		Podpisano do druku 22.04.2004 r.	Druk ukończ. w maju 2004 r.

Druk wykonano w Zakładzie Graficznym Politechniki Śląskiej w Gliwicach, ul. Kujawska 1
zam. 172/04

Książki Wydawnictwa można nabyć w księgarniach

GLIWICE

- Punkt Sprzedaży – ul. Akademicka 2 (237-17-87)
- „FORMAT" – Akademicka 5 (architektura i budownictwo)
- „LAMBDA" – ul. Akademicka 2 (237-21-40)
- Punkt Sprzedaży – ul. Akademicka 16 (automatyka, elektronika, informatyka)
- „ŻAK" – ul. Kaszubska

DĄBROWA GÓRNICZA

- „ANEKS" – ul. Ludowa 19A/III

ŁÓDŹ

- „POLITECHNIKA 100" – Żeromskiego 116 PŁ.

KATOWICE

- Punkt Sprzedaży – ul. Krasińskiego 8
- Hurtownia „DIK" – ul. Dulęby 7
- Hurtownia „JERZY" – ul. Słoneczna 24 (258-99-58)

TYCHY

- „I Ja Tours" ul. Piłsudskiego 10 (217-00-91 w.130)

ZABRZE

- Punkt Sprzedaży – ul. Roosevelta 26

KRAKÓW

- Techniczna – ul. Podwale 4
- Punkt Sprzedaży WND – AGH, Al. Mickiewicza 30

GDAŃSK

- EKO-BIS – ul. Dyrekcyjna 6 (058) 305-28-53

WARSZAWA

- Studencka – Pl. Politechniki 1
- Techniczna - ul. Kaliskiego 15
- Techniczna – ul. Świętokrzyska 14
- MDM – ul. Piękna 31

BIAŁYSTOK

- Dom Książki (Księgarnia 84) – ul. Dolistowska 3

POZNAŃ

- Księgarnia „POLITECHNIK" – ul. Piotrowo 3
- DOM Książki – Księgarnia Techniczna – ul. Półwiejska 14

Druk: Drukarnia Gliwice, ul. Zwycięstwa 27, tel. 230 49 50

Wydawnictwo Politechniki Śląskiej
44-100 Gliwice, ul. Akademicka 5
tel./fax (0-prefiks-32) 237-13-81
http://loki.polsl.pl/wydawnictwo
Sprzedaż i Marketing:
tel. (0-prefiks-32) 237-18-48,
e-mail: wydawnictwo_mark@polsl.pl