Ghent University

Faculty of Sciences

Department of Applied Mathematics, Computer Science and Statistics

BLIND RESTORATION OF IMAGES WITH PENALTY-BASED DECISION MAKING. A CONSENSUS APPROACH

Luis Antonio González Jaime

PhD in Computer Science

Ghent, 11 June 2015

Promotor

prof. dr. Etienne E. Kerre

Co-promotors

prof. dr. Santiago Aja-Fernández (Universidad de Valladolid)

prof. dr. Mike Nachtegael

This work has been funded by the European Commission under the MIBISOC project (Grant Agreement: 238819), within the action Marie Curie Initial Training Network of the 7FP.

Agradecimientos

Después de haber escrito 200 folios de tesis creo que me enfrento a la sección más complicada. Es difícil ahora recordar a todos los que habéis hecho posible este trabajo. ¡Sois muchos!. Los que de una manera u otra habéis estado durante todo o parte del camino, apoyándome y dedicándome vuestro tiempo para que esto al final sea posible. Sobretodo a los ajenos a este proyecto que de forma altruista os habéis implicado. Todos sabéis que sois parte de este trabajo, y que sin vosotros no hubiera sido posible. En especial, dedicársela a mis padres y hermana, que saben que siempre han sido mi gran apoyo. Si no fuera por ellos esto nunca se hubiera hecho realidad. Gracias a ellos, sé que con constancia y dedicación se puede luchar contra las adversidades para lograr el fin que se persigue.

¡Gracias a todos por vuestro apoyo y comprensión!

"... y lo que yo querría es menos de la mitad
de lo que la mitad de ustedes se merece."
Bilbo Bolsón

Acknowledgements

After having written 200 pages of thesis, I think I am now facing the most complicated section. It is hard to mention all of you who have made this project possible, there are so many of you! Some of which have been with me during the whole thesis and others for just for part of it. You have supported me and dedicated your time to make this all possible. To those not related with the project who have been altruistically involved, you all know that you are part of this work and that without you, it would not have been possible. I especially dedicate this to my parents and my sister, who know they have always been my biggest support. If it had not been for them, this would never have become a reality. Thanks to them, I know that with perseverance and dedication you can fight against the odds to reach the end.

Thank you all for your support and understanding!

"I don't know half of you half as well as I should like;
and I like less than half of you half as well as you deserve."
Bilbo Baggins

Contents

1

Introduction

1.1 Motivation

Image processing has become especially important in recent years. The irruption of smartphones and a massive use of social networks have increased the use of images, and thus their needs and applications. Moreover, improvements in technology and acquisition have also increased the use of other image types, such as magnetic resonance images, ultrasounds, computer tomography, among others.

Despite of technological improvements, images still suffer from a wide range of degradations and artifacts that are unavoidable. These are mainly due to acquisition, processing or transmission, such as noise, interferences, motion blur, misfocus, or lens distortions. These degradations usually decrease the perceptual fidelity of the image, and also decrease the performance of the task for which it was created, what negatively influences in an adequate interpretation and analysis of the data, as well as other post-processing computations (e.g. image registration, segmentation). It comes therefore as no surprise that two of the most common signal processing tasks are *image enhancement* and *image restoration*, which are almost present in any image application.

Image restoration aims to estimate the uncorrupted image from a degraded one, what is usually achieved by modeling the system that degrades the image, and the subsequent application of a reverse procedure. Unlike image enhancement that improves the visual appearance of an image, or transforms it in a way that facilitates computerized analysis for a specific application.

Unfortunately for restoration methods, the *degradation system* is not always

possible to model, either because the information cannot be retrieved from the degraded image, or because the knowledge about the problem is limited and imprecise, what means that some of the parameters cannot be estimated. As a result, it may contribute in a wrong result of some restoration methods, or even be impossible to apply them.

Therefore, we propose to define a *framework* to deal with situations, where due to the lack of information, these restoration methods cannot be applied. For it, the missing information in the restoration method is substituted by a decision-making process. I.e., we model the degradation system using a *consensus methodology*, in such way that we use a set of possible solutions to select one, or a combination of them, as the solution that minimizes some error measure, and thus better approaches the degradation system.

Within decision-making methodologies, we focus on *penalty-based decision making* to conform the framework as it presents a good compromise when the best solution is not known *a priori*, several solutions can contribute positively in a better one, and the set of solutions is from a diverse nature. Moreover, this is carried out on a fuzzy environment, what means that the set of input solutions are fuzzy. Fuzzy sets add the ability to model and reason with uncertainty, providing greater flexibility to represent the uncertainty resulting from the lack of knowledge. However, unlike fuzzy restoration methods that usually use a single method to model the uncertainty of the entire problem, and fuzzy fusion methods that usually use a single criterion to aggregate the set of input solutions. Penalty-based decision making allows to use different restoration methods, as well as fusing a set of solutions using different actions according to the desirability in the problem. That in some cases cannot be done by using a single fusion criterion, and a single fuzzy restoration method.

As there exist too many degradations that may affect an image, we focus on image restoration methods contaminated by noise. Concretely, on those methods that model the noise distribution of an image from a set of parameters. Such that we apply consensus methodology on those approaches that fail when some of these parameters are not properly estimated, or the noise type does not fit the noise distribution for which the method was developed. However, this methodology is interesting for its adaptability to a wide range of problems in image processing, both in image restoration and other fields as segmentation and image reduction, but not only. Moreover, as it also allows to use different methods as

input solutions, we can work with scenarios where we do not know beforehand which method is better to use.

Some areas of application

The proposed consensus framework have different applications. Some of these applications are introduced in this thesis for noise reduction. Specifically:

- There exist many noise reduction methods for a specific noise distribution [4, 13, 20, 58, 67, 77, 89, 90, 95, 114], however all of them fail in their performance with images owning a noise distribution for which these algorithms are not optimal; or when exist a mismatch in the assumed noise model. It would be desirable to have a blind noise suppression algorithm being able to deal with various noise distributions, or combinations of them. However this is a complex issue due to the different nature of the images, in other words, different images may get affected by different noise sources. Therefore, a *blind noise reduction method* could be defined with the help of the proposed methodology, in such way that a consensus solution can be reached starting with a set of various restoration methods for different noise distributions.

- Those techniques that demand large amount of data, in order to reduce the acquisition time, also reduce the temporal averaging; as a consequence, the noise is increased and may not strictly follow the initial degradation model. One consequence is the spatially-dependent variance of the noise, as in the case of images that uses *sensitiviy encoding (SENSE)* as a reconstruction method [5, 8, 53, 85]. What it may cause the impossibility of using stationary noise reduction methods. Thus, using the proposed framework and a restoration method for stationary noise, the uncertainty of spatially-dependent noise can be addressed.

1.2 Objectives

The main objective of this thesis is providing a *framework (or methodology)* to work with missing information in image restoration. Such that those noise reduction methods that originally cannot be applied because of a slight misfit in the

data, or any uncertainty in the input parameters, can be adapted using it. For it, the following sub-objectives are extracted:

1. **Study of the applicability of decision making in the field of image processing, in particular in image noise reduction.** Decision making has been already used in different areas, such as operational research, artificial intelligence or management, as well as in other applications of image processing. We especially focus on penalty-based decision making methodologies and their fields of application within image processing.

2. **Definition of a theoretical framework based on penalty functions to work with missing information in image restoration methods.** This framework should also incorporate the possibility of applying on matrices (images), subsets of these matrices (pixel regions), or matrix elements (pixels). What it allows to use different decision criteria depending on the region, where these regions can be of any shape, and share any characteristic. For it, the use of penalty functions over cartesian product of lattices is studied. Furthermore, the definition of penalty functions makes it easily adaptable to a wide range of problems while some conditions are fulfilled.

3. **Applicability of the penalty-based decision making framework (consensus methodology) to real problems in noise reduction.** For it, various noisy scenarios are studied, where a new method, or a modification of an existing one, is carried out to deal with missing information. We intend to show that using this framework when missing information exists, it can obtain similar results to the case in which this uncertainty is nonexistent. Then, the next objectives are carried out for each approach:

   (a) *Selection of the working scenario in image restoration and the method to adapt using consensus methodology.* Usually, a noise restoration method that presents some limitations when there exists some lack of information.

   (b) *Definition of the new method based on the consensus framework to deal with the uncertainty.* As well as the definition of the required functions and parameters for the consensus methodology.

   (c) *Validation of the proposed method with synthetic and real images.* The method is compared with the original image when it is available. In

those cases where the degradation model can be recovered, the consensus performance is compared to the original method. Moreover, for those other cases where it can also be compared with existing methods, a representative selection from the *State of the art* is carried out, proving that the proposed method can obtain similar or better results than existing methods.

1.3 Thesis structure

The thesis is divided in 8 chapters grouped in parts, and an appendix. The content of them is indicated below:

- Chapter 1 corresponds to this introduction.
- Part I gives an overview of all the necessary materials used in this thesis.
  - Chapter 2 provides an introduction to image processing methods focused on image restoration, as well as the noise distributions that usually affect images. To conclude, the similarity measures used to evaluate the performance of the presented approaches on Part III are introduced.
  - Chapter 3 presents an introduction to fuzzy logic and fuzzy set theory. It begins with the definition of fuzzy sets, and some basic concepts and their operations. Followed by the introduction to fuzzy logic, to finish with the definition of $\mathcal{L}$-Fuzzy sets. $\mathcal{L}$-Fuzzy sets and lattice theory are the basis of the proposed consensus decision-making methodology.
- Part II: Chapter 4 shows the relationship that exists between *fuzzy decision-making* and image processing. Then, the necessary tools to define *penalty-based decision making* are introduced, a concrete approach of fuzzy decision-making. Among these tools highlights the family of aggregation functions used in the aggregation phase, and the definition of penalty functions over cartesian product of lattices for the exploitation phase. Finally, a complete overview of the proposed consensus methodology is shown.
- Part III introduces in more detail various applications for image noise reduction with *consensus methodology*. Namely,

  - In Chapter 5, a first approach shows that consensus is a powerful tool for noise reduction with undetermined noise distribution, because the noise is not known, the noise model does not follow the initial assumptions, or the image contains different noise sources that cannot be properly modeled. The solution is obtained through penalty-based decision making using a set of filtered images, where these images are previously filtered by different filters optimized for a specific noise.
  - In Chapter 6, a new approach is presented to deal with non-stationary Gaussian noise. We show the goodness of consensus offering the chance to use parametric restoration methods when any of the parameters cannot be properly estimated or the data do not strictly fit the underlying model. In the case of the chosen filter *linear minimum mean square error (LMMSE)*, the confidence measure is estimated through a consensus methodology applied over a set of matrices obtained using different parameter configurations. This consensus estimation allows to calculate the noise pattern that disturbs the original image, to finally obtain similar filtered results to the case in which all the parameters are accurately known before.
  - In Chapter 7, a last approach is proposed for spatial non-stationary noise in MRI. A clear example of this kind of noise can be found in parallel MRI acquisitions that uses *Sensitiviy Encoding (SENSE)* as a reconstruction process, but not only. We adapt the LMMSE estimator optimized for stationary Rician noise to the case of spatial non-stationary Rician noise through *penalty-based decision making*. In this case, the calculus of the confidence matrix used by the LMMSE estimator is carried out by a consensus decision process. The performance of the new proposed filter shows similar behaviour to the optimal case assuming a noise pattern perfectly known.

- Part IV exposes in Chapter 8 the discussion and general conclusions of this research, as well as possible future research lines. This chapter also includes the list of publications derived from this work, showing that *penalty-based decision making* is an option for *blind restoration of images*.

- Appendix A collects the probability distributions and their moments used for the Chapters 6 and 7.

I

Background

2

Introduction to image restoration

Today, images are almost everywhere in our daily life. Hundreds of applications where we use them: in our mobiles, newspapers, internet, camera surveillance and so on. There is almost no area where images are not present. Interestingly, despite this there is not a universally accepted image processing definition accepted among authors. So we decided to use a widely accepted one: any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image [45].

Several processes can be applied to images. We can categorize them into three types: low-, mid-, and high-level processes [45]. Low-level processes involve primitive operations such as image pre-processing to reduce noise, contrast enhancement, and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images. Mid-level processing on images involves tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are features extracted from those images (e.g., edges, contours, and the identity of individual objects). Finally, higher-level processing involves "making sense" of a set of recognized objects (e.g. a hand taking an object, the position of a table...), and performing the cognitive functions normally associated with vision.

We should keep in mind that vision is the most advanced of our senses, so it is not surprising that images play one of the more important roles in human perception. However, unlike humans, who are limited to the visual band of the

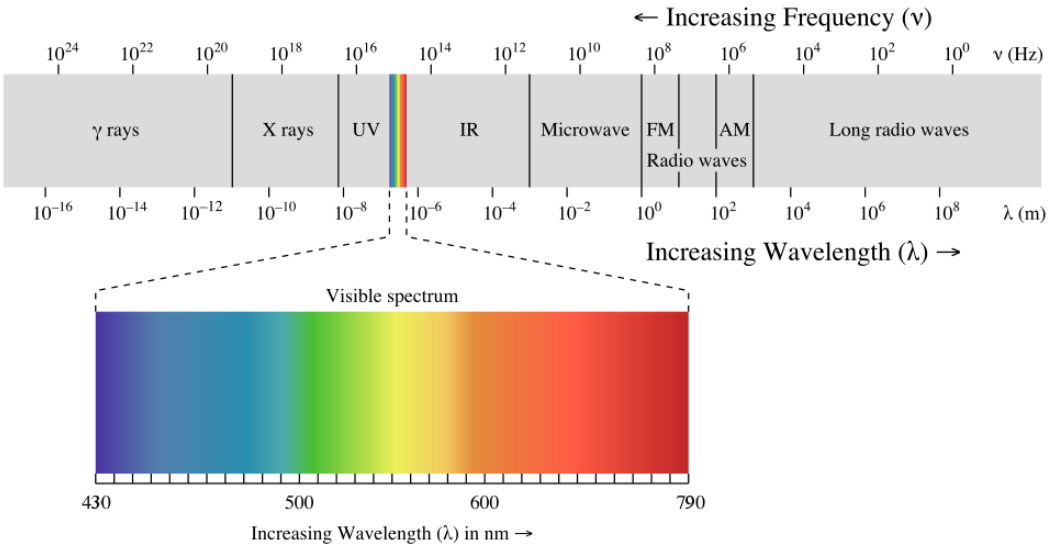


Figure 2.1: Electromagnetic spectrum. Courtesy image by Philip Ronan, under *Creative Commons license*.

electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum (see Figure 2.1), ranging from γ-rays (highest energy) to radio waves (lowest energy). They can operate on images generated by sources that humans are not accustomed to associate with images. These include ultrasound and electron microscopy. Thus, digital image processing encompasses a wide and varied field of applications.

Therefore, there are numerous ways to acquire images, but our objective in all is the same: to generate images from sensed data. The output of most sensors is a continuous voltage waveform whose amplitude and spatial behaviour are related to the physical phenomenon being sensed. To create an image we need to convert the continuous sensed data into digital form.

2.1 Image categorization according to its source

One of the simple ways to categorize the vast amount of image processing applications is by its *source* (e.g., visual, X-ray, radio waves, and so on). The electromagnetic energy spectrum is the principal energy source for images, although it is not the only one. Other important sources of energy include acoustic, ultra-

sonic, and electronic (in the form of electron beams used in electron microscopy). Therefore, we introduce the source imaging categories that are relevant in this text, explaining how these are generated and the areas in which they are applied.[1]

The most used images based on radiation are from the visible spectrum, that is the portion of the EM spectrum that is visible to the human eye (see Figure 2.1). A typical human eye will respond to wavelengths from about 430 to 790 nm [45]. In terms of frequency, this corresponds to a band in the vicinity of 380-700 THz. The infrared band lies just out of the human vision, distinguishing between near-, mid- and far-infrared. Sometimes near-infrared, also based on light reflection, is used in conjunction with visual imaging. The images from the visible spectrum are mainly built with sensors sensible to visible light that convert the electromagnetic signal into a digital value, for instance, one of the most known sensors are the charged-coupled device (CCD) used in photo and video cameras, although they are not the only ones. Hence it is not surprising that visible imaging is the most familiar and the most used by far to all the others in terms of application. Examples of its application are optical character recognition (OCR), where images of typewritten or printed text can be automatically converted into text; or factory quality control, where images can be used to check the liquid level from a bottle, up to check if the components of an assembled object are correctly placed.

Other source imaging is X-rays, that is one of the oldest sources of EM radiation used for imaging. The best known use of X-rays is medical diagnostics, but they also are used extensively in industry and other areas, like astronomy. X-rays for medical and industrial imaging are generated using an X-ray tube, which is a vacuum tube with a cathode and anode. The cathode is heated, using free electrons to be released. These electrons flow at high speed to the positively charged anode. When the electrons strike a nucleus, energy is released in the form of X-ray radiation. The energy (penetrating power) of X-rays is controlled by a voltage applied across the node, and by a current applied to the filament in the cathode. Computerized axial tomography (CAT) is another important use of X-rays in medical imaging. CAT is a process in which a ring of detectors encircles an object (or patient) and an X-rays source, concentric with the detector ring, rotates about the object. The X-rays pass through the object and are collected at the opposite end by the corresponding detectors in the ring. As the source

[1] As it is not the main purpose of this work to explain all types of image applications, we refer the interested reader for a more extensive study to the Chapters 1 and 2 from R.C. González and R.E. Woods [45].

rotates, this procedure is repeated. Tomography consists of algorithms that use the sensed data to build an image that represents a "slice" through the object. Motion of the object in a direction perpendicular to the ring of detectors produces a set of such slices, which constitute a three-dimensional (3-D) rendition of the inside of the object. In industrial processes, techniques similar to the ones just discussed, but generally involving higher-energy X-rays, are applicable.

At the other end of the spectrum (γ-rays), the main applications of imaging in the radio band are in medicine and astronomy. In medicine, radio waves are used in magnetic resonance imaging (MRI), also known as nuclear magnetic resonance imaging (NMRI) and magnetic resonance tomography (MRT). MRI is a medical imaging technique used in radiology to investigate the anatomy and function of the body in both health and disease, where some of the main advantages of this technique is that it allows us to diagnose on in vivo tissue, and unlike X-ray and CAT, MRI presents no known biological hazards. This technique places a patient in a powerful magnet, such as an MRI scanner, and passes radio waves through his or her body in short pulses. Then the water molecules in the body, which have small particles called protons, work like tiny magnets that are very sensitive to magnetic fields. Then the protons in the body line up in the same direction due to the magnetic field, in the same way that a magnet can pull the needle of a compass. Short bursts of radio waves are then sent to certain areas of the body, knocking the protons out of alignment. When the radio waves are turned off, the protons realign and in doing so send out radio signals, which are picked up by receivers. These signals provide information about the exact location of the protons in the body. They also help to distinguish between the various types of tissue in the body, because the protons in different types of tissue realign at different speeds and produce distinct signals. The time taken for the protons to fully relax is measured in two ways. The first is the time taken for the magnetic vector to return to its resting state and the second is the time needed for the axial spin to return to its resting state. The first is called T1 relaxation, the second is called T2 relaxation. In the same way that millions of pixels on a computer screen can create complex pictures, the signals from the millions of protons in the body are combined to create a detailed 3-D volumen of the inside of the body, being able to produce images in any plane.

Although imaging in the electromagnetic spectrum is dominant by far, there are a number of other imaging modalities that also are important. Imaging

using "sound" finds application in geological exploration, industry and medicine. Geological applications use sound in the low end of the sound spectrum (hundreds of Hz) while imaging in other areas use ultrasound (millions of Hz). The most important commercial applications of image processing in geology are in mineral and oil exploration. Nonetheless, ultrasound imaging finds its best applications in medicine, especially in obstetrics, where unborn babies are imaged to determine the health or their development. Ultrasound images are generated using the following basic procedure:

1. The ultrasound system (a computer, ultrasound probe consisting of a source and receiver, and a display) transmits high-frequency (1 to 5 MHz) sound pulses into the body.
2. The sound waves travel into the body and hit a boundary between tissues (e.g., between fluid and soft tissue, soft tissue and bone). Some of the sound waves are reflected back to the probe, while some travel on further until they reach another boundary and get reflected.
3. The reflected waves are picked up by the probe and related to the computer.
4. The machine calculates the distance from the probe to the tissue or organ boundaries using the speed of sound in tissue (1540 m/s) and the time of each echo's return.
5. The system displays the distances and intensities of the echoes on the screen, forming a two-dimensional image.

In a typical ultrasound image, millions of pulses and echoes are sent and received each second. Furthermore, the probe can be moved along the surface of the body and angled to obtain various views.

2.2 Image representation: spatial and frequency domain

Images that are seen in nature are *analog*, which means they are continuous signals in the space/time domain. Therefore, an image can be depicted as a two-dimensional function $f(x, y)$, where x and y are spatial coordinates, and the

value of f in any point (x, y) is proportional to the brightness of the image at that point, and of course, can take any possible value.

However, digital image processing needs that the image $f(x, y)$ is digitized in order to be understood by a computer. Then, this new image can be represented as a two-dimensional signal with discrete space coordinates that take values from a discrete set of values. These coordinates are not necessarily the values of the physical coordinates when the image was sampled.

Therefore, a digital image, assuming that it is a monochrome (or greyscale) image, can be seen as a matrix whose row and column indices identify a point in the image, and the value of the corresponding element of the matrix indicates the grey level or intensity in that point. This is what is known as *pixel*, abbreviation of *picture elements*. In Figure 2.2 is shown one possible representation of a digital image.

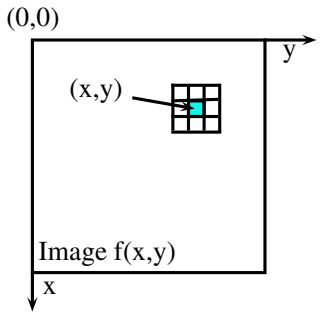


Figure 2.2: Digital image.

We represent an $N \times M$ image as

$$f = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M-1) \\ f(1,0) & f(1,1) & \cdots & f(1,M-1) \\ \vdots & \vdots & & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix} \tag{2.1}$$

Putting $a_{i,j} = f(i, j)$ we get a matrix representation:

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,M-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,M-1} \\ \vdots & \vdots & & \vdots \\ a_{N-1,0} & a_{N-1,1} & \cdots & a_{N-1,M-1} \end{bmatrix} \tag{2.2}$$

An image can also be represented as a vector, $\mathbf{v}$. For instance, a column vector of size $NM \times 1$ is formed by letting the first N elements of $\mathbf{v}$ be the first column of A, the next N elements be the second column, and so on. Alternatively, we can use the rows instead of the columns of A to form such a vector.

On the other hand, we can observe that the origin of a digital image is at the top left, with the positive x-axis increasing downward and the positive y-axis increasing to the right. This is due to a conventional representation based on the fact that many image displays (e.g., TV) sweep an image starting at the top left and moving to the right one row at a time.

In case the image is a colour image, the information in each coordinate (x, y) is coded according to the colour space that it is used, creating a multichannel image. These colour spaces provide a method for representing, organizing and manipulating colours. Most of these models are based on the combination of three primary colours to obtain any colour of nature.

The most known colour spaces are *RGB, CMYK, HSB, Lab, YCbCr*, which differ in the construction of chromatic circles that uses each one. For example, the HSB is based on hue-saturation-brightness, the CMYK uses subtractive colour mixing *Cyan-Magenta-Yellow-Black*, or RGB is based on additive mixture of primary light *red*, *green* and *blue*.

So far, a digital image is introduced as a matrix, whose pixels represent a proportional value of the intensity in the image, what is considered as spatial domain.

Nonetheless, the same image can be considered as a non-periodic function, and be defined in another two-dimensional space whose edges are determined for the amplitude and the frequency for each direction in the image [45]. This is known as the frequency domain.

Like one-dimensional signal processing, that transforms the spatial domain into the frequency domain, the two-dimensional version of the Fourier transform is used in 2-D. Furthermore, it has the same properties as its homologue in 1-D.

Then, in digital imaging the *Discrete Fourier Transform (DFT)* is used, whose expression is shown in Eq. (2.3):

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \cdot e^{-j2\pi(ux/N + vy/M)}, \tag{2.3}$$

where $f(x, y)$ is a digital image of size $N \times M$. As in the 1-D case, Eq. (2.3)

must be evaluated for values of the discrete variables u and v in the ranges $u = 0, 1, \ldots, N-1$ and $v = 0, 1, \ldots, M-1$.

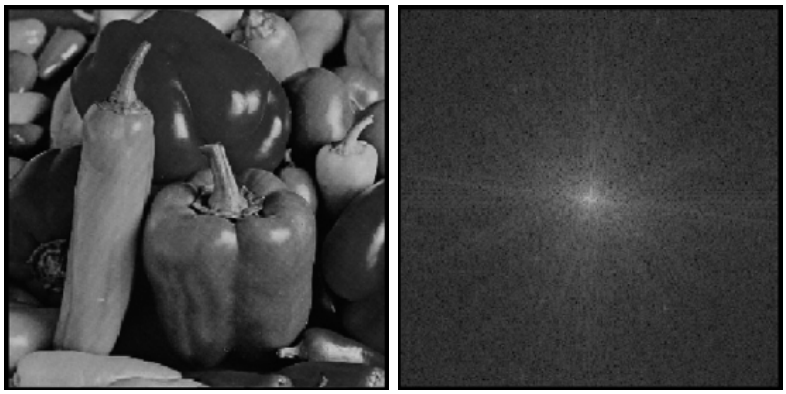
Given the transform $F(u, v)$, we can obtain $f(x, y)$ by using the *inverse discrete Fourier Transform (IDFT)*:

$$f(x,y) = \frac{1}{NM} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u,v) \cdot e^{j2\pi(ux/N+vy/M)}, \tag{2.4}$$

for $x = 0, 1, \ldots, N-1$ and $y = 0, 1, \ldots M-1$. Eq. (2.3) and Eq. (2.4) constitute the 2-D *discrete Fourier transform pair*.

As occurs in the time domain, rapid variations in the amplitude is equal to high values in the high frequency components, and on the contrary, slow variations in the amplitude correspond to low frequency components. In the 2-D case it is easy to associate spatial frequencies with the variation patterns of the image intensity.

For instance, low frequencies correspond to homogenous areas of the image, in other words values that vary slowly, as it is shown in Figure 2.3, that shows the *peppers* image and its *Fourier spectrum*. This image mainly has homogeneous regions separated by abrupt intensity changes. Smooth areas contribute to the low frequency component values, which causes high values in the central area of the DFT. Specifically, the central point with $u = v = 0$ is what is known as the DC component of the image, and corresponds to the average grey value.

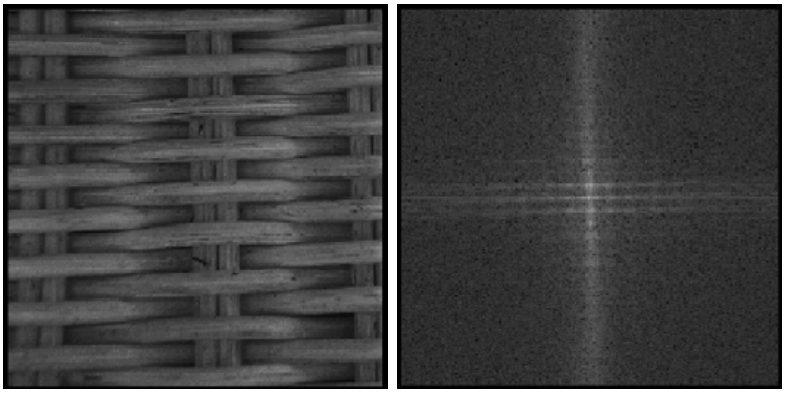


(a) *Peppers* image (b) Fourier spectrum of (a)

Figure 2.3: *Fourier spectrum* of an homogeneous image.

On the other hand, fast variations of the intensity due to the borders and noise correspond to the values of the high frequencies. Figure 2.4 shows a braided wicker image that presents a repetitive pattern in the DFT spectrum. Clearly

this series of peaks are because of the harmonics of the image, present in the periodic structure of the signal.



(a) *Wicker* image (b) Fourier spectrum of (a)

Figure 2.4: *Fourier spectrum* of a textured image.

2.3 Image restoration

Digital images captured by photo and video cameras, including high quality ones, are usually degraded by a certain amount of noise and blur.

Noise occurs on images for multiple reasons, meaning that noise is due to stochastic variations in contrast to deterministic variations caused by, among other reasons, blur or lack of contrast. This noise is mainly introduced during the image capturing (sensors, amplifiers), the transmission or the recording [45, 104], although in some modalities can be also introduced by their reconstruction algorithm or by the subject to be imaged, as for instance magnetic resonance imaging (MRI) or computer tomography (CT). This noise can e.g. be caused by dust sitting on the lens, by a dissipation in the electronic components or by electromagnetic distortions during transmission.

Blurring is a form of bandwidth reduction of an ideal image owing to the imperfect image formation process. The most common causes of blur [18, 45] are due to an optical system that is out of focus, to the atmospheric turbulence and to the relative motion between the camera and the original scene. These blurs are not limited to optical images, for instance, the scattered X-rays radiation produces blurring and loss of contrast in radiographs; or the electron micrographs may be damaged by the spherical aberrations of electron lenses.

In addition to these blurring effects, digital imaging techniques need to deal with the noise present in the images, which occasionally can lead to a bad performance since they affect image quality, what negatively influences in an adequate interpretation and analysis of the data, as well as other post-processing computations (e.g. image registration, segmentation). There are two solutions for this problem. A first approach is to make the methods *more robust* against noise; a second solution is to apply *noise suppression* (colloquially known as *denoising*) as a pre-processing step. In both solutions, an accurate noise model is necessary: the more pre-knowledge about the noise is gotten, a better technique will be built, leading to a better performance [51].

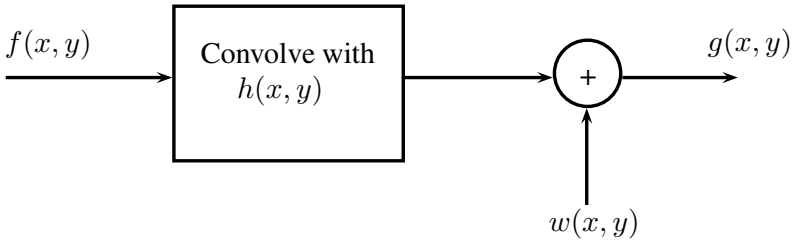
Then, restoration aims to estimate the uncorrupted image from a blurred and noisy one. This is achieved by modeling the system that degrades the image, and the subsequent application of a reverse procedure. Unlike image enhancement, which are based on human subjective preferences, producing more pleasing results to an observer. Image restoration is objective, in the sense that tends to be based on mathematical or probabilistic models of image degradation.

In the field of image restoration, sometimes referred to as *image deblurring* or *image deconvolution*, it is assumed that the characteristics of the degradation system are known *a priori*; however, in real situations may not be so, i.e., the required information to model the system can not be obtained directly from the image formation process. So in these cases, *blur identification* is necessary to estimate the properties of the system from the observed degraded image itself, prior to the *restoration process*. The combination of the *blur identification* and the *restoration process* is often referred to as *blind image deconvolution* [66, 87].

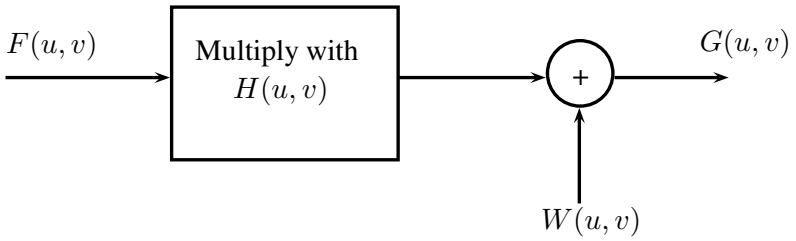
Assuming image restoration methods are linear and spatially invariant systems [68]. The restoration process can be carried out by means of a linear filter of which the convolution kernel or the *point-spread function (PSF)* is spatially invariant, i.e., is constant throughout the image. These modeling assumptions can be mathematically formulated as follows. If we denote by $f(x, y)$ the desired ideal spatially discrete image that does not contain any blur or noise, then the recorded image $g(x, y)$ is modeled as in the Eq. (2.5) (see also Figure 2.5(a)):

$$
\begin{aligned}
g(x,y) &= h(x,y) * f(x,y) + w(x,y) \\
&= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} h(k_1,k_2) f(x-k_1, y-k_2) + w(x,y).
\end{aligned}
\tag{2.5}
$$

Here $w(x,y)$ is the noise that corrupts the blurred image. Clearly the objective of image restoration is to make an estimate $f(x,y)$ of the ideal image, given only the degraded image $g(x,y)$, the blurring function $h(x,y)$, and some information about the statistical properties of the ideal image and the noise.



(a)



(b)

Figure 2.5: Image formation: (a) in the spatial domain; (b) in the frequency domain.

An alternative way of describing Eq.(2.5) is through its spectral equivalence. By applying discrete Fourier transforms to Eq. (2.5), we obtain the following representation (see also Figure 2.5(b)):

$$
G(u,v) = H(u,v) \cdot F(u,v) + W(u,v), \tag{2.6}
$$

where (u,v) are the spatial frequency coordinates and capitals represent Fourier transforms. Either Eq. (2.5) or Eq. (2.6) can be used for developing restoration algorithms. In practice the spectral representation is more often used since it leads to efficient implementations of restoration filters in the (discrete) Fourier

domain.

In Eqs. (2.5) and (2.6), the noise $w(x, y)$ is modeled as an additive term. Typically the noise is considered to have a zero-mean and to be white, i.e., spatially uncorrelated. In statistical terms this can be expressed as in Eq. (2.7) and Eq. (2.8):

$$E\{w(x,y)\} \approx \frac{1}{NM} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} w(x,y) = 0, \tag{2.7}$$

$$\begin{aligned} R_w(k_1,k_2) &= E\{w(x,y)w(x-k_1,y-k_2)\} \\ &\approx \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} w(x,y)w(x-k_1,y-k_2) = \begin{cases} \sigma_w^2, & \text{if } k_1 = k_2 = 0, \\ 0, & \text{else.} \end{cases} \end{aligned} \tag{2.8}$$

Here σ_w^2 is the variance or power of the noise, $E\{\cdot\}$ refers to the expected value operator, and R_w the autocorrelation. Sometimes the noise is assumed to have a Gaussian probability density function, but not only. As there exist several noise sources and probability density distributions that affect images.

In general the noise $w(x, y)$ may not be independent of the ideal image $f(x, y)$. This may happen for instance if the image formation process contains nonlinear components, or if the noise is multiplicative instead of additive. Unfortunately, this dependency is often difficult to model or to estimate. Therefore, noise and ideal image are usually assumed to be orthogonal, which is (in this case) equivalent to being uncorrelated because the noise has zero-mean. Expressed in statistical terms, the following condition holds:

$$\begin{aligned} R_w(k_1,k_2) &= E\{w(x,y)w(x-k_1,y-k_2)\} \\ &\approx \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} w(x,y)w(x-k_1,y-k_2) = 0. \end{aligned} \tag{2.9}$$

The models from Eqs. (2.5) and (2.6) form the foundations for the class of linear spatially invariant image restoration and accompanying blur identification algorithms. In particular these models apply to monochromatic images. For colour images, one approach is to extend Eqs. (2.5) and (2.6) to incorporate multiple colour components. In many practical cases of interest this is indeed the proper way of modeling the problem of colour image restoration since the degradations of the different colour components (such as the tri-stimulus signals red-green-blue,

luminance-hue-saturation, or luminance-chrominance) are not independent. This leads to a class of algorithms known as "multichannel filters" [9, 29, 42].

Below, in Section 2.3.1 several restoration methods are introduced. These can be grouped in: *inverse filter*, *least square filter, as the Wiener*, and *iterative restoration filters*. Followed by Section 2.3.2 that introduces different noise models.

2.3.1 Image restoration algorithms

As already mentioned above, image restoration methods pretend to estimate the ideal image without imperfections $\hat{f}(x, y)$ from the degraded image $g(x, y)$ [68]. For it is assumed that the *point-spread function (PSF)* is known *a priori*, and therefore once the restoration filter, denoted by $h(x, y)$, has been designed, the restored image is given by the Eq. (2.10):

$$\begin{aligned} \hat{f}(x, y) &= h(x, y) * g(x, y) \\ &= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} h(k_1, k_2) g(x - k_1, y - k_2), \end{aligned} \tag{2.10}$$

or in the spectral domain as in Eq. (2.11):

$$\hat{F}(u, v) = H(u, v) \cdot G(u, v). \tag{2.11}$$

Furthermore, there exist techniques that use information from multiple blurred images to reconstruct the original image [2, 38, 81, 103], however this section is focused on algorithms that need only one degraded image and some prior knowledge that characterizes the PSF. Therefore, the objective of the following algorithms is to design the filter $h(x, y)$ in the spatial domain, or $H(u, v)$ in its spectral version.

Inverse filter

An inverse filter is a linear filter whose PSF h_{inv} is the inverse of the blurring function h, and therefore it satisfies:

$$h_{\text{inv}}(x, y) * h(x, y) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} h_{\text{inv}}(k_1, k_2) h(x - k_1, y - k_2) = \delta(x, y). \tag{2.12}$$

When formulated as in Eq. (2.12), inverse filters seem difficult to design. However, the spectral counterpart of Eq. (2.13) immediately shows the solution to this design problem:

$$H_{\text{inv}}(u,v) \cdot H(u,v) = 1 \;\Rightarrow H_{\text{inv}}(u,v) = \frac{1}{H(u,v)}. \tag{2.13}$$

So the reconstruction has an advantage in the spectral domain, being sufficient to replace (2.13) into (2.11).

$$\begin{aligned} \hat{F}_{\text{inv}}(u,v) &= H_{\text{inv}}(u,v) \cdot G(u,v) = \frac{1}{H(u,v)}(H(u,v) \cdot F(u,v) + W(u,v)) \\ &= F(u,v) + \frac{W(u,v)}{H(u,v)}. \end{aligned} \tag{2.14}$$

If the noise is absent, the second term $\left(\frac{W(u,v)}{H(u,v)}\right)$ disappears in Eq. (2.14), so that the restored image is identical to the ideal image. The main advantage of this filter is its simplicity because it just needs *a priori* knowledge of the PSF. Although there also exist drawbacks [45], namely:

- The inverse filter may not exist because $H(u,v)$ is zero at selected frequencies (u,v), as happens with the blur models *linear motion* and *out-of-focus*. This drawback can be minimized to some extent by removing points that cause instability in the system using the filter known as *pseudo-inverse*. This is possible because the zeros are located at few points in the plane (u,v), and therefore can be easily removed from the calculation in Eq. (2.14) without affecting the end result.

- As noise exists, $H(u,v)$ would not be zero but it would have small values. So the term $\frac{W(u,v)}{H(u,v)}$ may dominate the estimation of $\hat{F}_{\text{inv}}(u,v)$, amplifying the noise and spoiling the end result. One possible solution is to limit the filter to a frequency range closeby the origin, reducing the possibility to find values close to zero.[2]

[2] In literature, this effect is commonly referred to as the *ill-conditionedness* or *ill-posedness* of the restoration problem.

Least-square filters

To overcome the noise sensitivity of the inverse filter, a number of restoration filters have been developed that are collectively called *least-squares filters* [45, 68]. The best known are the *Wiener filter* and the *constrained least-squares filter*, although both reach a similar solution, the theoretical basis that lies behind is very different.

Wiener filter assumes that the degradation function and the noise are random variables, with the aim to find an estimation of $\hat{f}$ that minimizes the mean square error between the ideal and the restored image.

$$\text{error} = E\{(f^2 - \hat{f}^2)\}. \tag{2.15}$$

Whereas the noise and the image are uncorrelated, the minimum of the error function given in Eq. (2.15), is obtained with the (2.16) presented in the frequency domain [45, 68, 69].

$$\begin{aligned} \hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right] G(u,v) \\ &= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v) \\ &= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v), \end{aligned} \tag{2.16}$$

where:

- $H(u,v) \equiv$ PSF.
- $H^*(u,v) \equiv$ Complex conjugate of $H(u,v)$.
- $|H(u,v)|^2 \equiv H^*(u,v)H(u,v)$.
- $S_f(u,v) \equiv$ Power spectrum of the ideal image ($|F(u,v)|^2$).
- $S_\eta(u,v) \equiv$ Power spectrum of the noise ($|W(u,v)|^2$).

The output of Eq. (2.16) is known as the *Wiener filter* in honor to N. Wiener, who was the first to propose this approach in 1942. In [45] can be found a more detailed explanation of this filter, also known as *minimum mean square error filter*.

The problem of spectral zeros in the PSF is avoided for the Wiener filter, as it is unlikely that the denominator is zero for a given value of the frequencies u and v. According to the term $S_\eta(u,v)/S_f(u,v)$, the filter is equivalent to the *pseudo-inverse filter* when the power spectrum of the noise is low, i.e., $H_{\text{wiener}} = \dfrac{1}{H}$ if $H \neq 0$, and 0 otherwise. On the other hand, if this is a high value, it behaves as a lowpass filter that eliminates noise.

On the downside, there is the need to estimate the amount of noise and to previously know the power spectrum of the ideal image. While the noise can be modeled as white Gaussian noise, and thus the estimation of $|W(u,v)|^2$ can be simplified to a constant. It still remains the problem that ideal image, in most practical cases, is unknown. This is why the approximation shown in Eq. (2.17) is usually used, where K is an adjustable parameter which is used to avoid having to know $S_f(u,v)$.

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v). \tag{2.17}$$

Another possibility is to replace $S_f(u,v)$ by an estimation of the power spectrum of the blurred image and compensate for the variance of the noise, as follows in Eq. (2.18).

$$S_f(u,v) \approx S_g(u,v) - \sigma_w^2 \approx \frac{1}{NM} G^*(u,v)G(u,v) - \sigma_w^2. \tag{2.18}$$

Finally, a statistical model can be also used to the ideal image. Often, these models have parameters that can be tuned to the actual image being used. A widely used model (not only popular in image restoration but also in image compression) is the following 2-D causal autoregressive model [15, 16, 101]:

$$f(x,y) = a_{0,1}f(x,y-1) + a_{1,1}f(x-1,y-1) + a_{1,0}f(x-1,y) + \sigma_w^2. \tag{2.19}$$

In this model, the intensities at the spatial location (x,y) are described as the sum of weighted intensities at the neighbouring spatial locations and a small unpredictable component, which is often modeled as white noise with variance σ_w^2. Once the model parameters are chosen, the power spectrum can be calculated as:

$$S_f(u,v) = \frac{\sigma_w^2}{|a_{0,1}e^{-ju} + a_{1,1}e^{-ju-jv} + a_{1,0}e^{-jv}|^2}. \tag{2.20}$$

Constrained least-squares filter is another approach for overcoming some of the difficulties of the inverse filter (excessive noise amplification) and of the Wiener filter (estimation of the power spectrum of the ideal image), while still retaining the simplicity of a spatially invariant linear filter.

It remains essential to know the degradation function PSF, in addition to the mean and variance of the noise. Parameters that can be easily estimated from the degraded image [6], what presents a great advantage for this technique.

If the restoration is a good one, the blurred version of the restored image should be approximately equal to the recorded distorted image, except for a certain amount of noise (2.21):

$$\begin{aligned} g(x,y) &= f(x,y) * h_{\text{PSF}}(x,y) + w(x,y), \\ \hat{f}(x,y) &= g(x,y) * h(x,y), \\ g(x,y) &\approx \hat{f}(x,y) * h_{\text{PSF}}(x,y). \end{aligned} \tag{2.21}$$

Therefore, it stands to reason that equality of Eq. (2.22) is satisfied.

$$\|g(x,y) - \hat{f}(x,y) * h_{\text{PSF}}(x,y)\|^2 \approx \sigma_w^2. \tag{2.22}$$

The aim of this technique is to find the optimal solution that gives the smallest possible value of σ_w^2, thereby minimizing Eq. (2.22). Considering the fact that the inverse filter tends to amplify the noise, a criterion is to select the solution that is as "smooth" as possible. It is therefore often used a high pass filter $C(x,y)$, such as the Laplacian, so it obtains information about the high frequency content of the restored image, i.e., the noise.

So what we pretend is to find the minimum of the function C [45] defined as:

$$C = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [\nabla^2 f(x,y)]^2, \tag{2.23}$$

subject to the constraint:

$$\|G - H\hat{F}\|^2 \approx \|\eta\|^2. \tag{2.24}$$

The solution in the spectral domain to this problem is given by Eq. (2.25)

[45, 68].

$$\hat{F}(u,v) = \left[\frac{|H(u,v)|^*}{|H(u,v)|^2 + \alpha|C(u,v)|^2}\right] G(u,v), \tag{2.25}$$

where α is the parameter that must be tuned so that the constraint in Eq. (2.24) is satisfied, and $C(u,v)$ is the Fourier transform of the Laplacian mask operator in Eq. (2.26).

$$c(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{2.26}$$

As was previously introduced, the great similarity between the Eqs. (2.25) and (2.17) is obvious.[3]

Iterative restoration filters

The filters formulated in the previous sections are usually implemented in the frequency domain using Eq. (2.11), since that way the direct convolution with the 2-D PSF $h(x,y)$ can be avoided. However, there are situations in which *spatial domain convolutions* are preferred over the spectral domain implementation, such as when the size of the images is excessively large.

It has been shown in numerous articles [1, 16, 27, 40, 61, 70, 86, 109, 113] that iterative processes can be particularly effective in several situations, such as those in which prior knowledge of the image is available in the form of restrictions; the blur function is roughly estimated; and the degree of noise removal depends on local image information or noise functions spatially variants.

Unlike methods discussed so far, *iterative methods* previously assume an initial ideal image and a known model of PSF, and at each iteration, the image and blur parameters are reestimated for its use in the next iteration.

The basic form of iterative restoration filters is the one that iteratively approaches the solution of the *inverse filter*, and is given by the following spatial domain iteration:

$$\hat{f}_{i+1}(x,y) = \hat{f}_i(x,y) - \beta(g(x,y) - h(x,y) * \hat{f}_i(x,y)). \tag{2.27}$$

Here $\hat{f}_i(x,y)$ is the restoration result after i iterations. Moreover, it is noticed

[3] In chapter 5 from R.C. González and R.E. Woods [45] can be found a more detailed description of this filter.

that if the number of iterations becomes very large, the $\hat{f}_i(x, y)$ approaches the solution of the inverse filter:

$$\lim_{i \to \infty} \hat{f}_i(x, y) = h_{\text{inv}}(x, y) * g(x, y). \tag{2.28}$$

Among the many advantages that these methods have, can be highlighted the followings:

- No requirement of the convolution of images with 2-D PSFs containing many coefficients. The only convolution is that of the restored image with the PSF of the blur, which has relatively few coefficients.
- No Fourier transform is required, making applicable to images of arbitrary size.
- No excess noise amplified as happens with the inverse filter, because the iteration can be terminated whenever an acceptable restoration result has been achieved.
- The basic form can be extended to include all type of *a priori* knowledge, in such a way that all the knowledge is formulated in the form of projective operations on the image.
- Easily extended for spatially variant restoration, i.e., restoration where either PSF of the blur or the model of the ideal image vary locally.

On the negative side, a very significant disadvantage of iterative schemes is a slow convergence. Per iteration, the restored image $\hat{f}_{i+1}(x, y)$ changes only a little. Therefore, many iteration steps are required before an acceptable point of termination of the iteration is reached. The reason is that the above iteration is essentially a *steepest descent* optimization algorithm, which is known to be slow in convergence. However, it is possible to reformulate the iterations in the form of, for instance, a conjugate gradient algorithm, which exhibits a much higher convergence rate; or into a primal or dual formulation based on an alternating direction method of multipliers (ADMM), that appear to be more efficient because of the dual formulation.[4]

[4] A more detailed explanation of these algorithms can be viewed in [1, 16, 27, 40, 61, 70, 113].

2.3.2 Image noise models

Images are known to suffer from a wide range of degradations and artifacts due to acquisition, processing or transmission, such as noise and interferences. That is, each element involved in the pipeline used to obtain the final (reconstructed) image (sensors, lens, A/D converter, enhancement algorithm, reconstruction algorithm...), influences the noise characteristics. For example, CCD sensors of digital photo and video cameras, are photon count devices, i.e. count photons to produce electrons, commonly known as photoelectrons. As the number of counted photons is a random amount, images often tend to suffer "*photon counting noise*", especially in low light conditions. So this noise is often modeled as a Poisson distribution, although under certain conditions can be modeled as Gaussian noise. Other many images are corrupted by impulse noise, also known as *salt-and-pepper* noise, because they are transmited by noisy digital channels. Its effect is similar to sprinkle black and white dots in the image. Although only few pixels are noisy, these are very harmful visually. Other known degradations are the one found in the grain of photographic films, that sometimes is modeled as Gaussian, and others as Poisson noise; and the quantization noise that is inherent in the amplitude quantization process that occurs in the analog-digital converter.

Different classifications are found in literature. One of these classifies the noise models in three main categories, namely *impulse noise*, *additive noise* and *multiplicative noise*. In addition to other specific noise probability density functions that commonly affect images, such as *Gaussian* (a type of additive noise), *Poisson* and *Rice*. Moreover, noise models can also present other characteristics, as the stationarity in the noise process. I.e, if the noise is stationary, the noise statistics (such as the variance) are invariant to the position in the image. While for the non-stationary noise, the noise characteristics depend on the position in the image.

In the remainder of this section, the previously mentioned noise models and their characteristics are introduced. As well as the common situations that produce such degradations. Finally, it explains what the non-stationary noise consists.

Impulse noise

Impulse noise is found in situations where quick transients, such as faulty switching, take place during imaging. It is characterized by the fact that only part of the image pixels are affected, while the others remain unchanged. Furthermore, a changed grey value of a noisy pixel, is not related to the original noise free value. Two types of impulse noise can be found in literature:

- *Fixed (valued) impulse noise*: the grey level of a corrupted pixel is always replaced by one of k fixed grey values $n_1, \ldots, n_k$:

$$g(x,y) = \begin{cases} n_1, & \text{with probability } pr_1, \\ n_2, & \text{with probability } pr_2, \\ & \cdots \\ n_k, & \text{with probability } pr_k, \\ f(x,y), & \text{with probability } 1 - \sum_{i=1}^{k} pr_i, \end{cases}$$

  where g is the resultant noisy image and f is the noise free image.

  A well-known example of this type of noise is *salt-and-pepper* noise, where there are only two noise values n_1 and n_2, given by the minimum and maximum allowed grey level (i.e., $n_1 = 0$ (black) and $n_2 = 2^m - 1$ (white) if we work with integer grey values stored by m bits). As an example, in Figure 2.6(a) is given the popular picture from Lena adding salt-and-pepper noise with the probability $pr_1 = pr_2 = 10\%$.

- *Random (valued) impulse noise*: contrary to the fixed valued impulse noise case, the grey level of an affected pixel is now replaced by a random grey value instead of one of a few fixed values:

$$g(x,y) = \begin{cases} f(x,y), & \text{with probability } 1 - pr, \\ \eta(x,y), & \text{with probability } pr, \end{cases}$$

  where g is the resultant noisy image, f is the noise free image; and $pr \in [0,1]$ denotes the probability that a grey value is corrupted and replaced by a random grey value $\eta(x,y)$ coming from a given distribution.

(a) Salt-and-pepper noise ($pr = 0.1$) (b) Gaussian noise ($\sigma = 0.1$)

Figure 2.6: Lena picture with impulse and additive noise.

Additive noise

In the case of additive noise, a random noise value is added to the grey value of each pixel:

$$g(x, y) = f(x, y) + \eta(x, y),$$

where g is the resultant noisy image, f is the noise free image; and $\eta(x, y)$ is a random noise value coming from a given distribution. A case of additive noise extensively studied is the Gaussian noise [20, 90, 114].

Gaussian distribution or normal distribution arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and for high temperature. It is considered the most prominent and most studied probability distribution in statistics. Let $g(x; \mu, \sigma)$ denote a pixel intensity of an observed image, and $x \in f$ indicates the corresponding pixel intensity of the noise free image, where μ and σ respectively denote the mean and standard deviation of the noise.

$$g(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2}(\frac{x - \mu}{\sigma})^2.$$

Additive white Gaussian noise (AWGN) has generally been found to be a reasonable model for noise originating from electronic amplifiers. As an example, in Figure 2.6(b) is shown the *Lena* image with Gaussian noise, where $\mu = 0$ and $\sigma = 0.1$.

Multiplicative noise

If an image is corrupted with multiplicative noise, then to each grey value, a noise value is added that is a random multiple of the original grey value:

$$g(x, y) = f(x, y) + \eta(x, y) \cdot f(x, y),$$

where $\eta(x, y)$ is a random noise value coming from a given distribution. For example speckle noise, that e.g. occurs in satellite images (SAR images), medical images (ultrasound images) and in television environments, is usually modeled this way with $q_U(x; \sigma)$ coming from a uniform distribution, given by:

$$q_U(x; \sigma) = \begin{cases} \frac{1}{2\sigma\sqrt{3}}, & |x| \leq \sqrt{3}\sigma, \\ 0, & \text{else}, \end{cases}$$

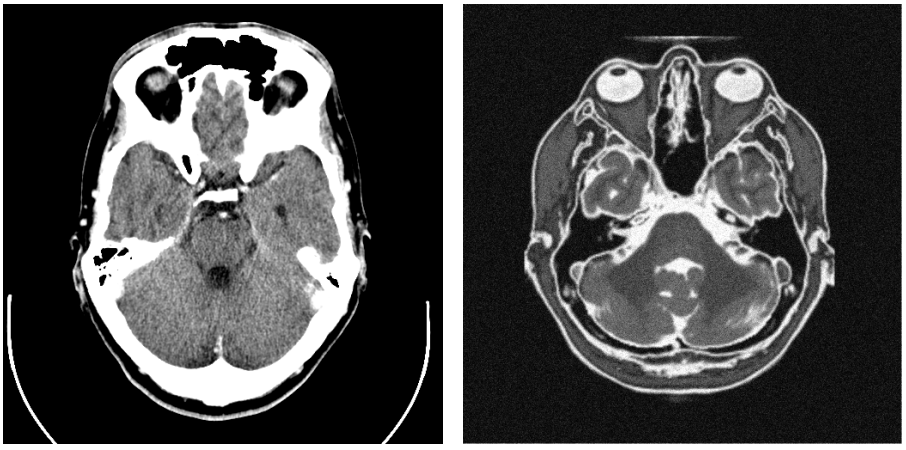
for $x \in \mathbb{R}$ where σ denotes the standard deviation of the noise. The higher this standard deviation, the higher the noise level. As an example, in Figure 2.7 is shown a Synthetic Aperture Radar, known as SAR images, from the city of Munich in 1994, courtesy of NASA.

Poisson noise density function

Poisson density function is used in any field related to counting. Can be e.g. in astronomy, photons arriving at a telescope; in biology, the number of mutations on a given strand of DNA; in computed tomography, photons arriving to the X-ray detector. According to the quantum mechanics, the measurement of light intensity can be interpreted as a spatio-temporal integration, for which the total number of photons emitted by the source in the considered spatio-temporal interval is often assumed to be a Poisson distribution.

Let $x \in f$, the corresponding pixel intensity of the noise free image, λ is the expected number of occurrences in a given interval, and g is the resultant noisy image:

Figure 2.7: SAR image from Munich, Germany (April 18, 1994).



(a) CT brain (b) MRI brain ($\sigma = 0.15$)

Figure 2.8: Poisson and Rice noise image examples.

$$g(x, \lambda) = e^{-\lambda} \frac{\lambda^x}{x!}.$$

Poisson noise is unbiased: it does not alter the intensity mean. Also, it has a variance that increases linearly with the original intensity x. As an example, in Figure 2.8(a) is shown a CT brain image provided by the Uppsala University Hospital. It can be seen how it is affected by Poisson noise.

Rice noise density function

Rice, or Rician distribution, is the probability density function formed by taking the absolute value of a complex Gaussian with nonzero mean. This distribution, known as Rice distribution, is defined as:

$$g(x|\nu,\sigma) = \frac{x}{\sigma^2}\exp\left(\frac{-(x^2+\nu^2)}{2\sigma^2}\right) I_0\left(\frac{x\nu}{\sigma^2}\right), \tag{2.29}$$

where σ is the standard deviation of Gaussian distribution in the complex domain (considered equal in real and imaginary image), ν is the amplitude of the signal without noise, x is the value in the image and I_0 is the modified Bessel function of order zero.

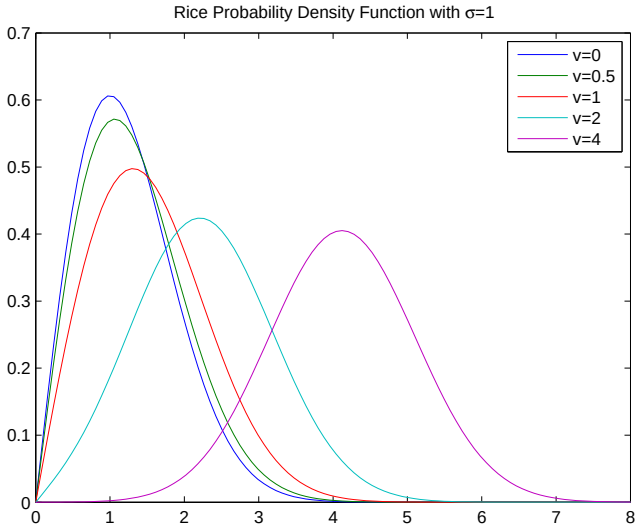


Figure 2.9: Probability density function in a Rice distribution, depending on the amplitude ν.

In contrast to Poisson noise processes, Rice noise processes do alter the intensity mean. Furthermore, it is dependant on the amplitude of the signal ν. When the signal is zero ($\nu = 0$), the Rice distribution becomes a Rayleigh distribution:

$$g(x|\sigma) = \frac{x}{\sigma^2}\exp\left(\frac{-x^2}{2\sigma^2}\right), \quad x \geq 0. \tag{2.30}$$

While if ν is big enough (i.e. $\nu \geq 3$) the distribution can be approximated with a Gaussian distribution (Eq. (2.31)).

$$g(x|\nu,\sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x^2 - \sqrt{\nu^2+\sigma^2})^2}{2\sigma^2}\right). \tag{2.31}$$

A graphical interpretation of the probability density function is shown in Figure 2.9 for a given $\sigma = 1$, where we can observe for this case the distribution behaviours we just mentioned.

Rice noise is one of the probability density functions that characterizes magnetic resonance imaging (MRI) [55, 57]. Concretely, MRIs that consist of an array of complex numbers in the *K-space* domain. This data is transformed with a 2D Fourier transform, where two resulting images can be displayed, specifically real and imaginary images. Additionally a *magnitude image* can be formed, called magnetic resonance magnitude image (MR magnitude image), by taking the square root of the sum of the squares of the real and imaginary images on a pixel-by-pixel basis [13]. The noise in each signal is assumed to have a Gaussian distribution with zero mean, and each channel of the complex image is assumed to be contaminated with white noise. The real and the imaginary images are reconstructed from the acquired data by the *complex Fourier transform*, which preserves the Gaussian characteristics of the noise. Furthermore, the variance of the noise is uniform over the whole field of view and, due to the Fourier transform, the noise in the corresponding real and imaginary voxels can be assumed uncorrelated. As an example, a T2 MRI image of a synthetic brain, generated from *BrainWeb data set* [31] with Rician noise of $\sigma = 0.15$ can be seen in Figure 2.8(b).

From stationary noise to non-stationary noise

In practice, there are many situations where the noise distribution depends on the position in the image (non-stationary noise). This can be both due to the acquisition device itself (e.g. interference from other devices), or by various post-processing steps (e.g. locally adaptive filtering, reconstruction algorithm). An example of an image with artifically generated non-stationary noise is given in Figure 2.10. In this example, the noise variance varies with the position in the image, but does not depend on the underlying image. It means that σ^2 becomes dependent on the position $\mathbf{x}$: $\sigma^2(\mathbf{x})$.

In general, the estimation of non-stationary noise is an issue that presents serious difficulties because it usually requires too many parameters. Moreover, it also needs some prior information that is not always available. Therefore in

Figure 2.10: Example of an image with artificially generated non-stationary noise.

cases we must deal with variant noise, we will introduce a model that allows us to simplify the whole process without loss of generality.

2.4 Image quality assessment measures

Images can suffer distortion due to several sources, from the acquisition process itself to compression, transmission through noisy channels and others. On the other hand, images can also undergo quality improvement processes, like enhancement or restoration techniques. In each case it is useful to quantify the *quality* of the resulting image.

An ideal quality assessment method should be able to cope and to quantify any kind of distortion. However this may be quite a hard task and probably application dependent, since the importance of a specific type of degradation is different depending on the purpose of the image, i.e., a particular noise level may be acceptable in home pictures but may lead to poor results in a segmentation application; some blur of the edges may lead to a critical information lost in MRI whereas the same process may even be able to ease the interpretation of an ultrasound image. Then, it follows that different indexes take into account different features of the image to assess its quality. It is important to know the behaviour of a particular index with different distortions in order to properly understand the results. So the different assessment measures must be seen as

companions rather than *competitors*.

Full-reference methods for quality assessment are those in which a signal is compared to a ground truth image, i.e. a golden standard. Many are the quality measures presented along the years [7, 32, 34, 39, 94]. However, within these methods, the most frequently used are those error based methods, as the *mean square error (MSE)*, and variations as the *peak signal to noise ratio (PSNR)*. However, these measures are not bounded; a higher MSE represents worst quality, although a higher PSNR means better quality. Although they give a measure of pixelwise similarity between the images, they do not explicitly take into account any structural information in the images or any sort of subjective measure. Despite they are simple to calculate and have clear physical meaning, their limitations have been widely reported in the literature [43, 94, 108].

On the other side, there are methods for quality assessment that rely on the analysis of the structural information in the image. These methods have proved being of great interest for very different kinds of images, ranging from natural scenes to medical scenarios. A full-reference quality assessment method based on the structural similarity of two images is the so-called *structural similarity (SSIM)* index. As of today, it has proved to be versatile and robust in many different environments [91]. However, it has a bias considering some degradation more important than others [3, 7]. For instance, blur is minimally taken as degradation, although for medical images it may constitute an important structural change. Unlike the SSIM index that hardly interprets blurring like a distortion, *quality index based on local variance (QILV)* gives a high penalty to it. QILV is based on the assumption that a great amount of the structural information of an image is encoded in the distribution of its local variance. Both indexes are bounded; the closer to one, the better the image.

2.4.1 Mean square error (MSE)

The mean square error (MSE) [39, 100] is the most used criterion to evaluate the performance of an estimator. It gives a measure of how pixelwise similar two images are. It is also useful to relay the concepts of bias, precision, and accuracy in the statistical estimation. Though, it does not take into account any structural information of the image.

The MSE of two images f and g is defined as

$$\mathrm{MSE}(f,g) = \frac{1}{NM}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}(f(x,y)-g(x,y))^2. \tag{2.32}$$

Taking N and M as the height and width of the image, respectively; and $f(x,y)$ the pixel value in the position (x,y) in the image f.

2.4.2 Peak signal-to-noise ratio (PSNR)

The peak signal-to-noise ratio (PSNR) [54], is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. Although it is easy to calculate, it does not always match ideally with the visual assessment, as in some specific situations it can give a wrong impression of the image quality [43].

The PSNR of two images f and g is calculated as:

$$\mathrm{PSNR}(f,g) = 20\cdot\log_{10}\left(\frac{\max_{x,y}\{f(x,y)\}}{\sqrt{\frac{1}{NM}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}(f(x,y)-g(x,y))^2}}\right), \tag{2.33}$$

where N and M the height and width of the image, $\max_{x,y}\{f(x,y)\}$ is the highest possible intensity value (e.g., in the case of an 8 bit image is 255), and $f(x,y)$ the pixel value in the position (x,y) in the image f.

It should be noted that PSNR can be expressed in terms of the MSE as follows:

$$\begin{aligned}\mathrm{PSNR}(f,g) &= 20\cdot\log_{10}\left(\frac{\max_{x,y}\{f(x,y)\}}{\sqrt{MSE}}\right)\\ &= 10\cdot\log_{10}\left(\frac{\max_{x,y}\{f^2(x,y)\}}{MSE}\right).\end{aligned} \tag{2.34}$$

2.4.3 Mean structural similarity index (SSIM)

Based on the assumption that human visual perception is highly adapted for extracting structural information, Z. Wang *et al.* [108] proposed an approach fo-

cused in the structural information of an image as those attributes that represent the structure of objects in the scene, independent of the average luminance and contrast. The system separates the task of similarity measurement into three comparisons: luminance, contrast and structure.

1. *Luminance comparison*:

$$l(f_x, g_x) = \frac{2\mu_{f_x}\mu_{g_x} + C_1}{\mu_{f_x}^2 + \mu_{g_x}^2 + C_1}, \tag{2.35}$$

with μ_{f_x} and μ_{g_x} the local means of the x-th local window at image f and g, respectively, and C_1 a constant.

2. *Contrast comparison*:

$$c(f_x, g_x) = \frac{2\sigma_{f_x}\sigma_{g_x} + C_2}{\sigma_{f_x}^2 + \sigma_{g_x}^2 + C_2}, \tag{2.36}$$

with σ_{f_x} and σ_{g_x} the local standard deviations of the x-th local window at image f and g, respectively, and C_2 a constant.

3. *Structure comparison*:

$$s(f_x, g_x) = \frac{\sigma_{fg_x} + C_3}{\sigma_{f_x}\sigma_{g_x} + C_3}, \tag{2.37}$$

with σ_{fg_x} the local covariance of the x-th local window between the images f and g, and C_3 a constant.

The *local ssim-index* then is defined as

$$\text{ssim}(f_x, g_x) = [l(f_x, g_x)]^\alpha \cdot [c(f_x, g_x)]^\beta \cdot [s(f_x, g_x)]^\gamma, \tag{2.38}$$

with α, β and γ weights in the interval $[0, 1]$. The overall value is obtained using the mean structural similarity index (SSIM):[5]

$$\text{SSIM}(f, g) = \frac{1}{M}\sum_{x=1}^{M} \text{ssim}(f_x, g_x), \tag{2.39}$$

[5]Also found in the literature in its abbreviation as MSSIM.

where f and g are the reference and the distorted image, respectively; f_x and g_x are the image contents at the x-th local window; and M is the number of local windows in the image.

2.4.4 Quality index based on local variance (QILV)

Quality index based on local variance (QILV) [3] is based on the assumption that a great amount of the structural information of an image is coded in its local variance distribution. Although the local variance itself has been taken into account in SSIM, its statistics, however, have been widely ignored. On the other hand, QILV is not so affected by noise, and should be decreasing with respect to an increasing noise level.

The QILV between two images f and g is defined as:

$$\mathrm{QILV}(f,g) = \frac{2\mu_{V_f}\mu_{V_g}}{\mu_{V_f}^2 + \mu_{V_g}^2} \cdot \frac{2\sigma_{V_f}\sigma_{V_g}}{\sigma_{V_f}^2 + \sigma_{V_g}^2} \cdot \frac{\sigma_{V_f V_g}}{\sigma_{V_f}\sigma_{V_g}}, \tag{2.40}$$

where

$$\mu_{V_f} = \frac{1}{NM}\sum_{x=1}^{N}\sum_{y=1}^{M}\mathrm{Var}(f(x,y)), \tag{2.41}$$

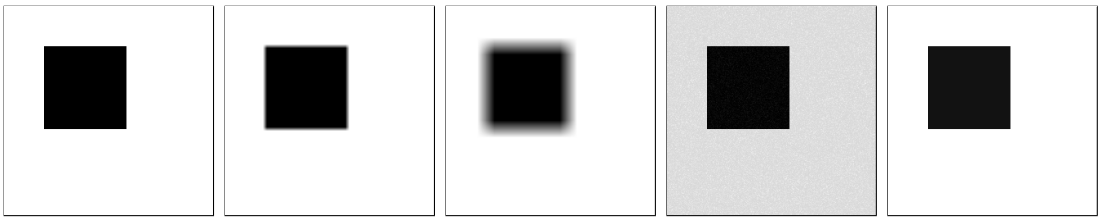
$$\sigma_{V_f} = \left(\frac{1}{NM-1}\sum_{x=1}^{N}\sum_{y=1}^{M}(\mathrm{Var}(f(x,y)) - \mu_{V_f})^2\right)^{1/2}, \tag{2.42}$$

$$\sigma_{V_f V_g} = \frac{1}{NM-1}\sum_{x=1}^{N}\sum_{y=1}^{M}(\mathrm{Var}(f(x,y)) - \mu_{V_f})(\mathrm{Var}(g(x,y)) - \mu_{V_g}), \tag{2.43}$$

and N and M are the height and width of the images. $\mathrm{Var}(\cdot)$ is the variance.

The first term in Eq. (2.40) carries out a comparison between the means of the local variances of both images. The second one, compares the standard deviation of the local variances. This term is related with the blur and the sharpness of the image. The third term is the one to introduce cross information in the two images. To avoid computational problems with small values, some constants may be added to every term in Eq. (2.40).

Note that although there is a purposedly great similarity between Eq. (2.40) and the SSIM index in Eq. (2.38), the latter relies on the means of the local statistics of the images, the former deals with the (global) statistics of the local



(a) (b) (c) (d) (e)

Figure 2.11: Synthetic experiment: Black square (256 grey levels). (a) Original Image; (b) Blurred image using a square 5×5 window; (c) Blurred Image using a square 21×21 window; (d) Image with additive Gaussian noise with 0 mean and $\sigma = 5$; (e) Image plus constant 10.

variances of the images.

In order to make the index more sensible to certain kind of degradations, each of the three components may be weighted by a different positive exponent σ, β and γ:

$$\mathrm{QILV}(f,g) = \left[\frac{2\mu_{V_f}\mu_{V_g}}{\mu_{V_f}^2 + \mu_{V_g}^2}\right]^{\alpha} \cdot \left[\frac{2\sigma_{V_f}\sigma_{V_g}}{\sigma_{V_f}^2 + \sigma_{V_g}^2}\right]^{\beta} \cdot \left[\frac{\sigma_{V_f V_g}}{\sigma_{V_f}\sigma_{V_g}}\right]^{\gamma}. \tag{2.44}$$

2.4.5 Measuring similarity across distortion types

Due to the great variety of possible degradations one may think of situations in which the information provided by a specific measure does not match a subjective quality judgement. Methods may have a bias towards the *image statistics* of the structural measure on which the design is grounded. Therefore, it is important to know the behaviour of a particular measure with different distortions in order to properly understand the results.

Many studies exist where the behaviour of quality measures with respect to various sources of degradation is analized [3, 7, 94, 108]. As an example, consider the synthetic image in Fig. 2.11(a), degraded from different sources:

- The image is blurred via convolution with a 5×5 averaging kernel, Fig. 2.11(b).
- The image is blurred via convolution with a 21×21 averaging kernel, Fig. 2.11(c).
- The image is corrupted by Gaussian noise with 0 mean and $\sigma = 5$, Fig. 2.11(d).
- Finally, a constant (10) is added to the image, Fig. 2.11(e).

Distorsion	MSE	PSNR	SSIM	QILV
Blur 5×5	160.04	26.09	**0.96**	0.42
Blur 21×21	692.49	19.73	0.87	0.01
White Noise	943.09	18.39	0.63	0.92
Constant	**100**	**28.13**	0.86	**1.00**

Table 2.1: Quality assessment using different measures for the black square experiment in Fig. 2.11.

The quality of the degraded images is assessed using different methods: MSE, PSNR, SSIM and QILV. Results are on Table 2.1.

From this particular example we can observe that Error-based methods, MSE and PSNR, give measures that are not totally consistent with respect to the different degradations (e.g., blur), although they give a general quality assessment. On the other hand, SSIM is bounded and takes into account structural information. However, this index considers some sources of degradation more important than others, i.e., there exists a bias towards some features of the image. For instance, blur is taken as a minimal degradation, although for many applications it may constitute an important structural loss; on the other hand, white noise is seen as a substantial degrading effect although, as a matter of fact, noisy structures may be *clearer* to the human eye than the blurred ones (when, for instance, identifying organs in ultrasound images). Finally, QILV seems to be a good bounded measure that reduces the bias introduced by blurring and noise, where blur is highly penalized.

In conclusion, to properly test restoration or enhancement methods, several quality indexes should be used complementarily, in such a way that they consider different sources of degradation, such as structural information, blur and noise. For instance, the goodness of a filtering method lays not only in its ability to remove the noise, but also in the preservation of edges within the image.

3

Introduction to fuzzy logic and fuzzy set theory

Set theory, along with mathematical logic, is one of the axiomatic foundations of classical mathematics. We are used to dealing with so-called "crisp" sets, whose membership only can be true or false in the sense of bi-valued logic, with 1 typically indicating true and 0 indicating false. However, in real life situations, an object often satisfies a property to some degree, i.e., it does not completely satisfy the property, but also does not completely not satisfy the property. For instance, a "tall" person may be someone over 1.90m high. A person 2m high is considered tall. This person is taller than a person 1.89m, but we cannot say the person 1.89m is not tall at all. As it is the case for crisp sets. Therefore, fuzzy sets allow to define membership functions between 0 and 1. They provide a framework for incorporating human knowledge in the solution of problems whose formulation is based on imprecise concepts. Based on this principle, fuzzy sets have been also applied to image processing. Intensity transformations and spatial filtering are the two most frequent areas in which fuzzy techniques for image processing are applied.

3.1 Fuzzy sets

As noted, the origin of fuzzy logic is the notion of fuzzy set. Before addressing the study of the theory of fuzzy sets, some of the basic concepts of ordinary (crisp) set theory, are reviewed in order to achieve a better understanding of both.

3.1.1 Review of ordinary (crisp) set theory

Definitions, terminology and notation

The starting point of set theory are the concepts of *element* and *set*. A set is usually defined as a collection of elements. Typically the elements that conform a set have some property in common that makes them capable of belonging to the set, but this requirement is merely anecdotic. The set is usually represented by a capital letter, type A, B, C, ...; and the elements thereof are represented by a lowercase letter (a, b, c , ...).

A membership relation is defined on the sets, which is denoted by the symbol $\in$. So, if the element a belongs to the set A, this fact is formalized by the expression

$$a \in A.$$

In case b does not belong to A it is expressed as

$$b \notin A.$$

Regarding the way of description of the set, this can be done in enumerative way, $A = \{a_1, a_2, \ldots, a_n\}$, or by the formation rule to which makes reference, for instance, $A =$ *'the ten first natural numbers'*. Such definition, as it can be expected, is equivalent to write it in enumerative way $A = \{1, 2, 3, \ldots, 10\}$.

The *cardinal* of a set is defined as the number of elements that are part of the set. If that is a finite cardinal number, the set is called finite. Otherwise, the set is infinite. Within the latter, must be distinguished the set of countable cardinal, which will be those whose elements can be within a 1:1 correspondence with integer numbers (for example, the set of even numbers); on the other hand, we find the set of uncountable cardinal, like for instance, the set of real numbers between two numbers a and b.

The *inclusion* or *containment* is extracted from the membership relationship; a set B is said to be "contained" in a set A when all elements from B are in A. If such is the case, we can express in abbreviated form $B \subseteq A$, or either that $A \supseteq B$. If it is verified that $B \subseteq A$ and $A \subseteq B$ simultaneously, then the two sets are equal.

Two sets are said to be *disjoint* if they have no elements in common. These sets are called also mutually exclusive.

Given a problem, the universal set, denoted by U, is the set of all the elements of the problem. In a complementary way, the empty set, denoted by $\emptyset$, is a set without any element. Naturally, the sets U and $\emptyset$ are mutually exclusive.

Let U be a universe which any set A is a subset, i.e.:

$$A \subseteq U,\ \forall A.$$

In ordinary (crisp) set theory any element x belonging to U belongs or does not belong to the subset A, in a clear and unequivocal manner, without other options apart from these two.

The membership or not of an arbitrary element x in a subset A is given in most cases by verifying whether or not a predicate that characterizes A and leads to a bipartition of the universe of discourse U.

For instance, let U be the universe that consists of all the rivers in the world. The set A is defined as that which consists of all elements of U that verified the predicate "x flows in Europe". To cite a few examples:

"The Rhine"	$\in A$
"The Nile"	$\notin A$
"The Ebro"	$\in A$
"The Guadalquivir"	$\in A$

It should be noted that it has been possible to provide a definition of the set A because its corresponding predicate allows the bipartitioning of the universe U.

Characteristic function

The concept of belonging or not of an element in a set A can be expressed numerically by a *characteristic function*.[1] This function assigns to each element x of the universe a binary value (1 or 0) depending if x belongs or not to the set A [60, 65].

$$\varphi_A : U \longrightarrow \{0,1\},\ \varphi_A(x) = \begin{cases} 1, & \text{if } x \in A, \\ 0, & \text{if } x \notin A. \end{cases} \tag{3.1}$$

[1] Some authors also called *membership function*. In this text we have chosen to use *characteristic function* for ordinary set theory and *membership function* for fuzzy set theory.

Any set $A \subseteq U$ can be defined by the pairs forming each element x of the universe and its characteristic function, expressed as follows:

$$A = \{(x, \varphi_A(x)) \mid x \in U\}. \tag{3.2}$$

For instance, the set $A = \{3, 4, 5, 6, 7, 8, 9, 10\}$ is represented by its characteristic function:

$$\varphi_A(x) = \begin{cases} 1, & \text{if } x \in \{3, 4, 5, 6, 7, 8, 9, 10\}, \\ 0, & \text{in other case.} \end{cases}$$

Basic operations on sets

Given any two sets A and B included in U it is possible to define a set of basic operations between them, as the *complement*, *intersection* and *union* [65, 107].

Definition 3.1.1 (Complement)**.** The complement of A is denoted by $\overline{A}$, and consists of all the elements of U that do not belong to A (unary operator).

$$x \in \overline{A} \text{ if } x \notin A.$$

Its characteristic function is:

$$\varphi_{\overline{A}}(x) = 1 - \varphi_A(x). \tag{3.3}$$

Definition 3.1.2 (Intersection)**.** The intersection is denoted by $A \cap B$ and it is defined as the set formed by those elements of U that belong to A and B simultaneously:

$$x \in A \cap B \text{ if } x \in A \text{ and } x \in B.$$

The corresponding characteristic function is

$$\varphi_{A \cap B}(x) = \min(\varphi_A(x), \varphi_B(x)). \tag{3.4}$$

Definition 3.1.3 (Union)**.** The union is the set formed by those elements that belong to A, or belong to B, or both simultaneously. It is denoted by $A \cup B$ and its characteristic function is:

$$\varphi_{A \cup B}(x) = \max(\varphi_A(x), \varphi_B(x)). \tag{3.5}$$

Fundamental properties of ordinary sets operations

The operations between crisp sets have certain laws and properties:

1. Commutativity

$$\begin{aligned} A \cup B &= B \cup A \\ A \cap B &= B \cap A \end{aligned} \tag{3.6}$$

2. Associativity

$$\begin{aligned} (A \cup B) \cup C &= A \cup (B \cup C) \\ (A \cap B) \cap C &= A \cap (B \cap C) \end{aligned} \tag{3.7}$$

3. Idempotence

$$\begin{aligned} A \cup A &= A \\ A \cap A &= A \end{aligned} \tag{3.8}$$

4. Absorption

$$\begin{aligned} (A \cup B) \cap A &= A \\ (A \cap B) \cup A &= A \end{aligned} \tag{3.9}$$

5. Distributivity

$$\begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap (A \cup C) \\ A \cap (B \cup C) &= (A \cap B) \cup (A \cap C) \end{aligned} \tag{3.10}$$

6. Absorption by U and $\emptyset$

$$\begin{aligned} A \cup U &= U \\ A \cap \emptyset &= \emptyset \end{aligned} \tag{3.11}$$

7. Identity

$$\begin{aligned} A \cup \emptyset &= A \\ A \cap U &= A \end{aligned} \tag{3.12}$$

8. Involution

$$\overline{\overline{A}} = A \tag{3.13}$$

9. De Morgan's laws

$$\begin{aligned} \overline{A \cup B} &= \overline{A} \cap \overline{B} \\ \overline{A \cap B} &= \overline{A} \cup \overline{B} \end{aligned} \tag{3.14}$$

10. Law of contradiction

$$A \cap \overline{A} = \emptyset \tag{3.15}$$

11. Law of excluded middle

$$A \cup \overline{A} = U \tag{3.16}$$

3.1.2 Extension to fuzzy sets

L.A. Zadeh introduced the concept of a fuzzy set [115] by extending the *characteristic functions* to *membership functions* and in this way allowing a gradual transition between satisfying a property (belonging to a set) or not. Giving the possibility that an element may have a membership degree between 0 and 1. The more an object belongs to a set (e.g., the taller a person), the higher its membership degree.

Then, while in ordinary (crisp) set theory a set A in a universe U divides the universe into two parts: the elements that belong to A (and thus satisfy a given defining property), and the elements that do not belong to A. On the other hand, in fuzzy set theory, a fuzzy set A in U is characterized by a membership function, μ_A, that associates with each element of U a real number in the interval $[0, 1]$ [62, 65, 107]. The value of $\mu_A(x)$ represents the grade of membership of x in A. The nearer the value of $\mu_A(x)$ is to unity, the higher the membership grade of x in A, and conversely when the value of $\mu_A(x)$ is closer to zero. The concept "belongs to", so familiar in ordinary sets, does not have the same meaning in fuzzy set theory. With ordinary sets, we say that an element either belongs or does not belong to a set. With fuzzy sets, we say that all x for which $\mu_A(x) = 1$ are *full members* of the set, all x for which $\mu_A(x) = 0$ are *not members* of the set, and all x for which $\mu_A(x)$ is between 0 and 1 have *partial membership* in the set.

Using mathematical notation a fuzzy set A is defined as:

$$A = \{(x, \mu_A(x)) | x \in U\}. \tag{3.17}$$

Membership function

The characteristic function is replaced by a *membership function* that is defined as:

$$\begin{aligned} \mu_A : \quad & U \to [0, 1], \\ & x \to \mu_A(x), \end{aligned} \tag{3.18}$$

in such a way that $\mu_A(x)$ is the membership degree of an element $x \in U$ into the fuzzy set A. If $\mu_A(x) = 0$ the element x does not belong to the set, and if

$\mu_A(x) = 1$, x is a full member of the set.

We observe that crisp sets are a special case of fuzzy sets. This situation arises when $\mu_A(x)$ can have only two values, say 0 and 1, the membership function reduces to the characteristic function of a crisp set A.

The shape of the membership function has a certain *subjective* component, opposite to the rigid (objective) form of the characteristic functions from ordinary set theory. Depending on the application of the set or the concepts represented by them, these functions can acquire different shapes, and often can be selected with a large degree of freedom by the "designer", which in practise can be understood as the possibility of including some expertise.

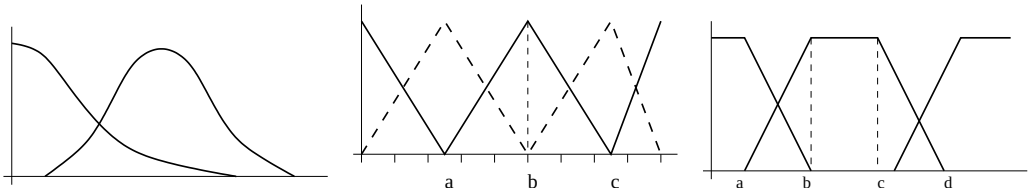
Although functions may have any shape, the literature tends to work with standard membership shapes [97] (Figure 3.1):

1. *Gaussian functions or with S-shape* (Figure 3.1(a)). They use the formula

$$\mu(x) = \frac{1}{d\sqrt{2\pi}} \exp\left(\frac{(x-m)^2}{2d^2}\right), \tag{3.19}$$

   with parameters m and d that allow to model "modifications", as for instance "very small". Although these are complex to calculate.

2. *Triangular and trapezoidal functions* [56]. They are defined according to the number of vertices. $\Delta(a, b, c)$ for the triangular functions (Figure 3.1(b)) and $T(a, b, c, d)$ for the trapezoidal functions (Figure 3.1(c)). These are simple to handle in numerical algorithms.



(a) Gaussian (b) Triangular (c) Trapezoidal

Figure 3.1: Typical membership shapes.

X-J. Zeng and G. Singh Madan [117] defines a model of membership function which represents the main classes, the *pseudo trapezoid-shaped* function.

Definition 3.1.4 (Pseudo trapezoid-shaped (PTS))**.** z The *pseudo trapezoid-shaped* function is a continuous function given by

$$A(x;a,b,c,d,h) = \begin{cases} I(x), & x \in [a,b), \\ h, & x \in [b,c], \\ D(x), & x \in (c,d], \\ 0, & x \in U - [a,d] = \{x | x \in U, x \notin [a,d]\}, \end{cases} \tag{3.20}$$

where $a \leq b \leq c \leq d$, $a < d$, $I(x) \geq 0$ is a strictly increasing monotonic function on $[a,b)$ and $D(x) \geq 0$ is a strictly monotonically decreasing function on $(c,d]$ (Figure 3.2). When the membership function of a fuzzy set A is a PTS function, it is called PTS membership function and is denoted as $A(x) = A(x;a,b,c,d,h)$. When the fuzzy set is normalized (i.e., $h = 1$), its membership function is simply denoted by $A(x) = A(x;a,b,c,d)$.

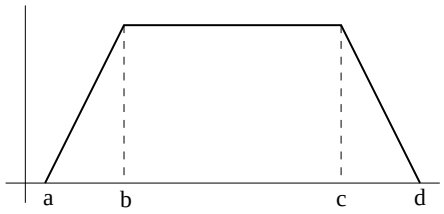


Figure 3.2: Pseudo trapezoid-shaped function (PTS).

According to the Definition 3.1.4, trapezoidal functions are a special case of PTS functions when $b < c$ and

$$I(x) = \frac{x-a}{b-a},\ D(x) = \frac{x-d}{c-d}, \tag{3.21}$$

and the triangular functions are the special case when $b = c$ and

$$I(x) = \frac{x-a}{b-a} = \frac{x-a}{c-a},\ D(x) = \frac{x-d}{c-d} = \frac{x-d}{b-d}. \tag{3.22}$$

As an example, in Figure 3.3 is shown some fuzzy sets defined in the universe *age*. The fuzzy set "young" represents the membership degree regarding the youth parameter that would have each age. In other words, the set expresses the degree to which each element would be characterized as "young". A fuzzy set could be considered as a possibility distribution, that it is different than a probability distribution [36].

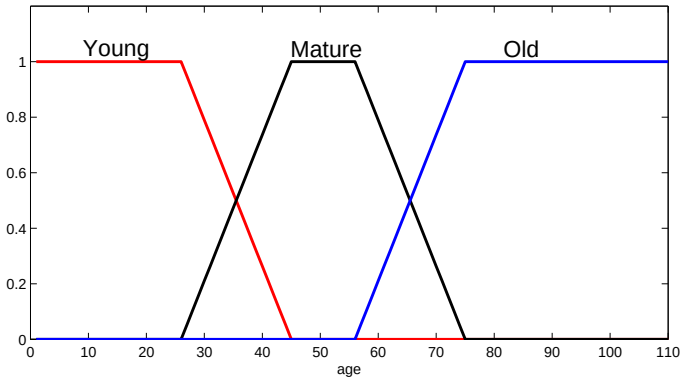


Figure 3.3: Example of fuzzy sets.

It can be seen in Figure 3.3 that fuzzy sets overlap, then an element x, as for instance $x = 29$, may have different membership degree in two fuzzy sets: "young" and "mature", indicating that it owns qualities associated with both fuzzy sets.

For practical reasons, it is often assumed that the unverse U is finite, i.e. $S = \{x_1, \cdots, x_n\}$, and the pair $\{(\mu_A(x), x)\}$ is denoted as $\mu_A(x)/x$, and each pair $\mu_A(x)/x$ is called *fuzzy singleton* [60, 65, 107]. Then, the fuzzy set A can be rewritten as

$$\begin{aligned} A &= \{(\mu_A(x), x)\} = \{\mu_A(x)/x\} = \\ &= \mu_A(x_1)/x_1 + \cdots + \mu_A(x_n)/x_n = \textstyle\sum_{i=1}^{n} \mu_A(x_i)/x_i, \end{aligned} \tag{3.23}$$

where $+$ and $\sum$ must be understood in the sense union. By convention, the pairs $\mu_A(x)/x$ with $\mu_A(x) = 0$ are omitted.

3.1.3 Basic definitions on fuzzy sets

Here are some basic definitions useful to handle fuzzy sets [60, 62, 65, 107]. As for instance, the *support* of the fuzzy set, that are the elements of the universe whose membership is larger than zero. As well the *kernel* is determined by all the elements that have the membership degree equal to 1. Here, we define some of the most relevant.[2]

Definition 3.1.5 (Empty set)**.** A fuzzy set A is empty, and written $A = \emptyset$, if

[2]It is considered that $U \subset \mathbb{R}$ is the universe of all fuzzy sets.

and only if

$$\mu_A(x) = 0, \ \forall x \in U.$$

Definition 3.1.6 (Equality)**.** Two fuzzy sets A and B defined on the same universe U are equal, and written $A = B$, if and only if

$$\mu_A(x) = \mu_B(x), \ \forall x \in U.$$

Definition 3.1.7 (Support)**.** The support $\text{supp}(A)$ is defined as

$$\text{supp}(A) = \{x \in U \mid \mu_A(x) > 0\}.$$

Definition 3.1.8 (Kernel)**.** The kernel or core $\ker(A)$ is defined as

$$\ker(A) = \{x \in U \mid \mu_A(x) = 1\}.$$

Definition 3.1.9 (Normalized)**.** A is called normalized if

$$(\exists x \in U)(\mu_A(x) = 1).$$

Definition 3.1.10 (Pseudo-normalized)**.** A is called pseudo-normalized if

$$\sup_{x \in U} \mu_A(x) = 1.$$

Definition 3.1.11 (Height)**.** The height $\text{h}(A)$ is defined as

$$\text{h}(A) = \sup_{x \in U} \mu_A(x).$$

Definition 3.1.12 (Plinth)**.** The plinth $\text{plt}(A)$ is defined as

$$\text{plt}(A) = \inf_{x \in U} \mu_A(x).$$

Definition 3.1.13 (Convexity)**.** A fuzzy set A is convex if and only if for any $x_1, x_2 \in U$ and any $\lambda \in [0, 1]$,

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\}.$$

As in the case of ordinary sets, the notion of *containment* plays a central role

in the case of fuzzy sets [115]. This notion is defined as follows in Definition 3.1.14

Definition 3.1.14 (Containment)**.** A is contained in B (or, equivalently, A is a subset of B, or A is smaller than or equal to B) on the universe U if and only if

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x),\ \forall x \in U.$$

3.1.4 Basic operations on fuzzy sets

In order to manipulate fuzzy sets, several operations can be defined. The usual set operations defined on crisp sets (*union*, *intersection* and *complement*) can been extended to deal with fuzzy sets in different ways [65, 107, 115]. We first present these definitions which are obvious extensions of the corresponding definitions for ordinary sets (Figure 3.4).

Definition 3.1.15 (Union)**.** A union of two fuzzy sets A, B with respective membership functions μ_A and μ_B is characterized by the membership function:

$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \vee \mu_B(x),\ \forall x \in U.$$

Definition 3.1.16 (Intersection)**.** An intersection of two fuzzy sets A, B with respective membership functions μ_A and μ_B is characterized by the membership function:

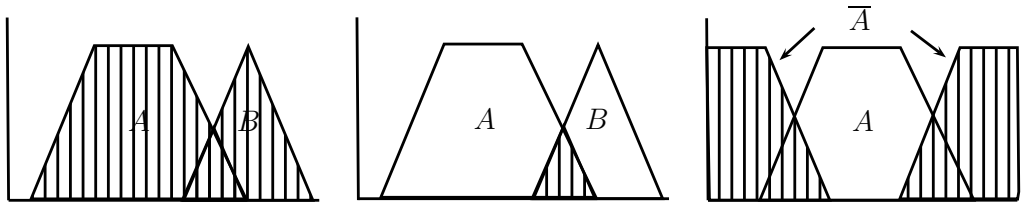
$$\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} = \mu_A(x) \wedge \mu_B(x),\ \forall x \in U.$$

Note that $\cup$ and $\cap$ have the associativity property that is, $A \cup (B \cup C) = (A \cup B) \cup C$, and $A \cap (B \cap C) = (A \cap B) \cap C$, respectively.

Definition 3.1.17 (Complement)**.** A complement of a fuzzy set A is defined as

$$\mu_{\neg A}(x) = 1 - \mu_A(x).$$

However, unlike what happens with crisp sets, this is not the only possible way to define these operations; different functions may be suitable to represent them in different contexts. Therefore, not only the membership functions of fuzzy sets will be dependent on the context but also the operations on these sets [65].



(a) Union (b) Intersection (c) Complement

Figure 3.4: Basic operations on fuzzy sets. Standard definition.

Fuzzy complement

Given a fuzzy set $A \subset U$, its complement is defined as the fuzzy set $\overline{A}$ whose membership function is given by the expression:

$$\mu_{\overline{A}}(x) = C(\mu_A(x)),\ \forall x \in U, \tag{3.24}$$

where C is a function from $[0,1]$ to $[0,1]$ that must fulfill the following properties:

1. *Boundary conditions*: $C(0) = 1$, $C(1) = 0$.
2. *Monotonicity*: for all $a, b \in [0,1]$, if $a \leq b$, then $C(a) \geq C(b)$.

In most cases, it is desirable to consider some additional requirements for these functions:

3. C is a *continuous* function.
4. C is *involutive*, what means that $C(C(a)) = a$, $\forall a \in [0,1]$.

There are many functions that fulfill the above described properties, and therefore they can be used to represent blur complement. Some of them are:

$C(x) = 1 - x$,	Standard negation.
$C(x) = \frac{1-x}{1-\lambda x}\ \lambda \in (0, \infty)$,	Sugeno negation.
$C(x) = (1 - x^w)^{1/w}\ w \in (0, \infty)$,	Yager negation.

Fuzzy intersection: t-norm

Given two fuzzy sets A and B, defined on the same universe U, their intersection is defined as a fuzzy set $A \cap B$ whose membership function is given by the expression:

$$\mu_{A \cap B}(x) = T(\mu_A(x), \mu_B(x)),\ \forall x \in U, \tag{3.25}$$

where the function T is a *triangular norm* or *t-norm* [65]. A t-norm is a mapping $T : [0,1] \times [0,1] \to [0,1]$ that verifies the following properties:

1. *Commutativity*: $T(x,y) = T(y,x),\ \forall x,y \in [0,1]$.
2. *Associativity*: $T(T(x,y),z) = T(x,T(y,z)),\ \forall x,y,z \in [0,1]$.
3. *Monotonicity*:
   if $(x \leq y)$ and $(w \leq z)$ then $T(x,w) \leq T(y,z)$, $\forall x,y,w,z \in [0,1]$.
4. *Absorption*: $T(x,0) = 0$, $\forall x \in [0,1]$.
5. *Identity*: $T(x,1) = x$, $\forall x \in [0,1]$.

There are many functions that meet these properties and thus can be used to represent the intersection of fuzzy sets. Some of them are:

$$
\begin{array}{ll}
T(x,y) = \min(x,y), & \text{Minimum t-norm (or Gödel t-norm).} \\
T(x,y) = \max(0, x+y-1), & \text{Łukasiewicz t-norm.} \\
T(x,y) = x \cdot y, & \text{Product t-norm.} \\
T(x,y) = \begin{cases} b, & \text{if } a = 1, \\ a, & \text{if } b = 1, \\ 0, & \text{otherwise.} \end{cases} & \text{Drastic t-norm.}
\end{array}
$$

Sometimes it is necessary to restrict the possible t-norms considering three additional requirements [65]:

1. *Continuity*: T is a continuous function.
2. *Subidempotency*: $T(x,x) < x,\ \forall x \in]0,1[$.
3. *Strict monotonicity*: $a_1 < a_2$ and $b_1 < b_2$ implies $T(a_1,b_1) < T(a_2,b_2)$.

The axiom of continuity prevents a situation in which a small change in the degree of membership in the fuzzy set A or B produces a large change (discontinuous) in the degree of membership in $A \cap B$. The subidempotency is taken into account when the degrees of membership in A and B for some x have the same value. This axiom expresses the requirement that the degree of membership in $A \cap B$ in this case does not exceed this value. The third requirement is a stronger condition on monotonicity.

Fuzzy union: t-conorm

Given two fuzzy sets A and B defined on the same universe U, their union is defined as a fuzzy set $A \cup B$ whose membership function is given by the expression:

$$\mu_{A \cup B}(x) = S(\mu_A(x), \mu_B(x)), \ \forall x \in U, \tag{3.26}$$

where the function S is a *triangular conorm*, also called *t-conorm* or *s-norm*. This is a mapping $S : [0,1] \times [0,1] \longrightarrow [0,1]$ that satisfies the following requirements:

1. *Conmutativity*: $S(x,y) = S(y,x)$, $\forall x, y \in [0,1]$.
2. *Associativity*: $S(x, S(y,z)) = S(S(x,y), z)$, $\forall x, y, z \in [0,1]$.
3. *Monotonicity*: if $(x \leq y)$ and $(w \leq z)$ then $S(x,w) \leq S(y,z)$, $\forall x, y, w, z \in [0,1]$.
4. *Absorption*: $S(x,1) = 1$, $\forall x \in [0,1]$.
5. *Identity*: $S(x,0) = x$, $\forall x \in [0,1]$.

As in the previous case for the *t-norm*, there exist a large number of functions that satisfy these properties and can be used to represent the union. Some examples are:

$$\begin{aligned}
S(x,y) &= \max(x,y), && \text{Maximum t-conorm.} \\
S(x,y) &= \min(1, x+y), && \text{Bounded sum.} \\
S(x,y) &= x + y - x \cdot y, && \text{Probabilistic sum.} \\
S(x,y) &= \begin{cases} y, & \text{if } x = 0, \\ x, & \text{if } y = 0, \\ 1, & \text{otherwise.} \end{cases} && \text{Drastic t-conorm.}
\end{aligned}$$

Sometimes it is necessary to restrict the possible t-conorms considering three additional requirements [65], which take into account special cases, as was done for the t-norm:

1. *Continuity*: S is a continuous function.
2. *Superidempotency*: $S(x,x) > x, \ \forall x \in]0,1[$.
3. *Strict monotonicity*: $a_1 < a_2$ and $b_1 < b_2$ implies $S(a_1, b_1) < S(a_2, b_2)$.

Relationship between fuzzy operations

From the general properties of t-norms and t-conorms it is easy to deduce that these functions are bounded by the minimum and maximum functions:

$$T(x, y) \leq \min(x, y) \ \forall x, y \in [0, 1],$$

$$S(x, y) \geq \max(x, y) \ \forall x, y \in [0, 1].$$

A particular choice of the operators union, intersection and complement can verify the generalized Morgan's laws:

$$C(T(x, y)) = S(C(x), C(y)),$$

$$C(S(x, y)) = T(C(x), C(y)).$$

In this case we say that the t-norm and t-conorm are dual with respect to the fuzzy complement. In general, given a complement function, you can associate a t-norm for each s-norm (and vice versa). For example, using the negation as a complement function, the pairs minimum-maximum and product-probabilistic sum verify the generalized Morgan's law.

Not all t-norms and t-conorms will be dual, nor will be mutually distributive functions. According to [65]:

1. The operators min and max are dual with respect to the fuzzy complement C.

2. The minimum and maximum operators also verify the distributivity property:

$$\min(x, \max(y, z)) = \max(\min(x, y), \min(x, z)),$$

$$\max(x, \min(y, z)) = \min(\max(x, y), \max(x, z)).$$

3. Given a t-norm T and an involutive fuzzy complement C, the binary operation S on $[0, 1]$ defined by

$$S(a, b) = C(T(C(a), C(b)),$$

$\forall a, b \in [0, 1]$, is a dual t-conorm w.r.t. C.

4. Given a t-conorm S and an involutive fuzzy complement C, the binary

operation T on $[0,1]$ defined by

$$T(a,b) = C(S(C(a), C(b)),$$

$\forall a, b \in [0,1]$, is a dual t-norm w.r.t. C.

5. Let T and S be dual operators that satisfy the complementary laws w.r.t. C, i.e.

$$S(x, C(x)) = U,$$

$$T(x, C(x)) = \emptyset.$$

Then T and S do not satisfy the distributivity property.

3.1.5 Properties of fuzzy sets

The laws and properties, as has been seen, that are fulfilled by crisp sets, are not always met in the case of fuzzy sets. Then, we analyze what laws and properties are verified by fuzzy sets, and which are not:

1. *Commutativity*: is always verified, because the t-norms and t-conorms are commutative by definition.

2. *Associativity*: is also verified since the t-norms and t-conorms are also associative.

3. *Idempotency*: is met if the minimum and maximum are chosen as operators for the intersection and union respectively. But if for example the product t-norm and the probabilistic sum are chosen then it is not met.

4. *Absorption*: is also met if the pair minimum-maximum is chosen. With other norms, it does not necessarily happens the same.

5. *Distributivity*: also holds for the minimum and maximum, but not necessarily for other norms.

6. *Absorption and identity*: is always met because the last property of t-norms and t-conorms.

7. *Involution of the complement*: is verified if it is defined $\mu_{\overline{A}}(x) = 1 - \mu_A(x)$, and then:

$$\mu_{\overline{\overline{A}}}(x) = 1 - \mu_{\overline{A}}(x) = 1 - (1 - \mu_A(x)) = \mu_A(x). \tag{3.27}$$

8. *De Morgan's laws*: is effectively enforced if the t-norm and s-norm chosen are derived one from the other. I.e.: $T(x, y) = 1 - S(1 - x, 1 - y)$.

9. *Complementary laws*: generally is not met. It is perhaps the most obvious consequence of introducing the concept of fuzziness in the sets.

It can be easily verified that if the sets are crisp sets (membership function restricted to 0 or 1) the differences between the several norms disappear, becoming the classic intersection and union operators.

Some authors in fuzzy set theory attribute to it the fact that there is arbitrariness in the choice of the operators union and intersection. Despite this, that seems a drawback, it can be on the other hand an advantage, because it allows great flexibility to address different problems involving "vague" concepts. If certain properties of crisp set must be fulfilled, a t-norm and t-conorm must be chosen that allow it. This choice will result in one or another type of fuzzy logic.

3.1.6 The extension principle

Permits a generalization of conventional operators, providing a mechanism for calculating fuzzy sets obtained by a "crisp" transformation (not fuzzy) of a certain number (N) of fuzzy sets. The extension principle states that the image of fuzzy set X under the mapping f can be expressed as a fuzzy set Y [107]. Specifically, if X_1, X_2, ..., X_n are fuzzy sets with membership functions $\mu_1(x_1)$, $\mu_2(x_2)$, ..., $\mu_N(x_N)$, the new fuzzy set $Y = f(X_1, X_2, \ldots, X_N)$ is the function of belonging:

$$\mu(y) = \max_{\mathbf{x}=f^{(-1)}(y)} \left[\min_{i=1}^{N} \mu_i(x_i) \right] \tag{3.28}$$

3.1.7 α-cuts: from fuzzy sets to crisp sets

There exists a direct way to pass from fuzzy sets to crisp sets, through so-called α-cuts [65].

Definition 3.1.18 (weak α-cut)**.** Let the fuzzy set $A \in U$, and let $\alpha \in (0, 1]$. The weak α-cut A_α is defined as

$$A_\alpha = \{u \in U \mid \mu_A(u) \geq \alpha\}.$$

Remark that the choice $\alpha = 0$ would not yield new information (because it would result in the universe U). Further, in a lot of properties this special case would need to be excluded. Therefore, this case is usually excluded from the definition.

Definition 3.1.19 (strong α-cut)**.** Let the fuzzy set $A \in U$, and let $\alpha \in [0, 1[$. The strong α-cut $A_{\alpha+}$ is defined as

$$A_{\alpha+} = \{u \in U \mid \mu_A(u) > \alpha\}.$$

Remark that the choice $\alpha = 1$ would not yield new information (because it would result in the empty set $\emptyset$). Further, in a lot of properties this special case would need to be excluded. Therefore, this case is usually excluded from the definition.

Then, given a $\alpha \in]0, 1]$ and a fuzzy set A, it is defined the α-cut of A as the set A_α, whose characteristic function is defined as:

$$\varphi_{A_\alpha}(x) = \begin{cases} 1, & \text{if } \mu_A(x) \geq \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{3.29}$$

Definitely, the weak α-cut is formed by those elements whose membership functions equals or exceeds the threshold α. In the case of the strong α-cut, is formed by those elements whose membership functions exceeds the threshold α:

$$\varphi_{A_\alpha}(x) = \begin{cases} 1, & \text{if } \mu_A(x) > \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{3.30}$$

Any fuzzy set A can be represented by the union of its α-cuts as follows:

$$\mu_A(x) = \max_{\alpha \in]0,1]} [\alpha \cdot \varphi_{A_\alpha}(x)] \tag{3.31}$$

The α-cuts are particularly useful in the study of properties such as reflexivity, symmetry and transitivity.

3.2 Fuzzy logic

The fuzzy set theory can be used to represent linguistic expressions that are used to describe sets or algorithms. Fuzzy sets can express the vagueness of words and sentences commonly accepted as "*red flower*" or "*slight change*". The human ability to communicate by vague or uncertain definitions is an important attribute of intelligence.

3.2.1 Linguistic variables

A linguistic variable is a variable whose values are words or sentences that fall in a default language. Each of these words or terms is known as *linguistic label* and is represented by a fuzzy set defined on the universe of the variable. For instance, the body temperature can be classified as *low*, *normal*, *high* or *too high*. Each of these terms is a linguistic label that can be defined as a fuzzy set. The sets that conform the variable are shown in Figure 3.5.

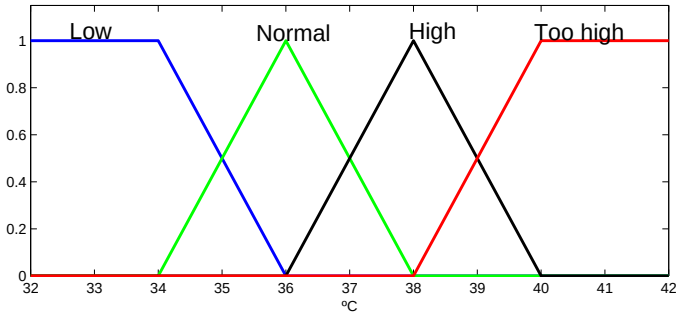


Figure 3.5: Definition of the linguistic variable *body temperature*.

3.2.2 Fuzzy relations: logical operators

As discussed in the previous Section 3.1.4, the operators union, intersection and complement are calculated all in the same universe. However, the cartesian product allows products from more than one universe.

Cartesian product

Let U and V be two universes. A fuzzy relation R between U and V is defined as a fuzzy set whose universe is the cartesian product $U \times V$. I.e.:

$$\begin{aligned} &R = \{((x,y), \mu_R(x,y)) \ / \ (x,y) \in U \times V\}, \\ &\mu_R : U \times V \longrightarrow [0,1]. \end{aligned} \tag{3.32}$$

If $A_1 \subseteq U$ and $A_2 \subseteq V$, and if the cartesian product of A_1 and A_2 is defined as:

$$\mu_{A_1 \times A_2}(x,y) = \min(\mu_{A_1}(x), \mu_{A_2}(y)), \tag{3.33}$$

then, this function would be represented as shown in Figure 3.6.

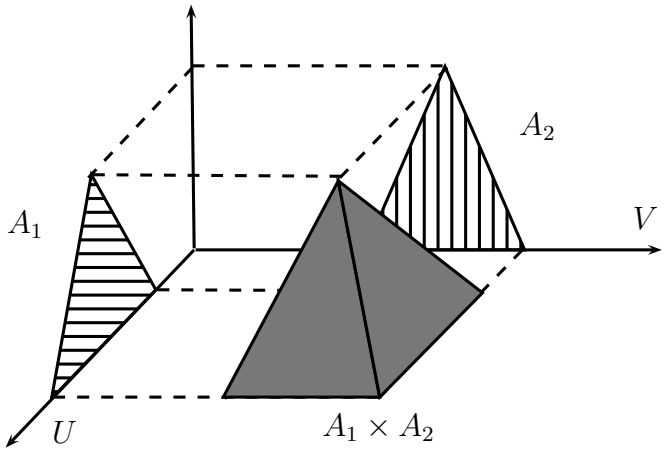


Figure 3.6: Cartesian product of two fuzzy sets A_1 and A_2.

For instance, if $U = V = \mathbb{R}$, the following fuzzy relation can be defined which expresses how similar are two real numbers x and y:

$$R = \{((x,y), \mu_R(x,y)) | (x,y) \in \mathbb{R}^2\}. \tag{3.34}$$

$$\mu_R(x,y) = \begin{cases} (1 + (x-y)^4)^{-1}, & \text{if } |x-y| \leq 5, \\ 0, & \text{otherwise.} \end{cases} \tag{3.35}$$

However, the main utility of fuzzy relations is the ability to act as logical connectives.

Logical connective "AND"

If we have two fuzzy sets $A \subseteq U$ and $B \subseteq V$, and a pair $(x, y) \in U \times V$, the *AND* connective which indicates the extent to $x \in A$ and $y \in B$ can be deployed by the following fuzzy relation:

$$\mu_{AND}(x, y) = \min(\mu_A(x), \mu_B(y)). \tag{3.36}$$

One can see that this concept is very similar to the intersection of fuzzy sets. Not exactly the same, since the intersection operation is defined for sets in the same universe. However, precisely because of this similarity, it is common to define "AND" by means of any t-norm, and not just by using the minimum.

Logical connective "OR"

The "OR" connective, that gives an idea of the extent that $x \in A$ or $y \in B$, is usually defined as the relation:

$$\mu_{OR}(x, y) = \max(\mu_A(x), \mu_B(y)), \tag{3.37}$$

or by any other t-conorm.

Logical implication "IF-THEN"

In fuzzy logic there are many ways in which implication can be defined; different implication functions can be used based on *t-norms* and *t-conorms*. Following are introduced the more common fuzzy implications, of which the Mandani implication (minimum) and Larsen (product) are the easier to implement.

$\mu_M(x,y) = \min(\mu_A(x), \mu_B(y)),$	Mamdani.
$\mu_P(x,y) = \mu_A(x) \cdot \mu_B(y),$	Larsen.
$\mu_R(x,y) = 1 - \mu_A(x) + \mu_A(x) \cdot \mu_B(y),$	Reichenbach.
$\mu_L(x,y) = \min(1 - \mu_A(x) + \mu_B(y), 1),$	Łukasiewicz.
$\mu_W(x,y) = \max(1 - \mu_A(x), \min(\mu_A(x), \mu_B(y))),$	Willmott.
$\mu_{KD}(x,y) = \max(1 - \mu_A(x), \mu_B(y)),$	Kleene-Dienes.
$\mu_{RG}(x,y) = \begin{cases} 1, & \forall(x,y)\ \mu_A(x) \leq \mu_B(y), \\ 0, & \forall(x,y)\ \mu_A(x) > \mu_B(y). \end{cases}$	Reschner-Gaines.
$\mu_G(x,y) = \begin{cases} 1, & \forall(x,y)\ \mu_A(x) \leq \mu_B(y), \\ \mu_B(y), & \forall(x,y)\ \mu_A(x) > \mu_B(y). \end{cases}$	Brouwer-Godel.
$\mu_{RG}(x,y) = \begin{cases} \min(\mu_A(x)/\mu_B(y), 1), & \forall(x,y)\ \mu_B(y) \neq 0, \\ 1, & \forall(x,y)\ \mu_A(x) = 0. \end{cases}$	Goguen.

When μ_A and μ_B only take the values 0 and 1, the above definitions are consistent with the implication of classical logic (where 0 the value of falsehood and 1 to truth), except the implications μ_M and μ_P, which therefore are not real extensions of the binary implication.

3.3 $\mathcal{L}$-fuzzy sets

In some cases, the unit interval does not suffice as an evaluation space. Therefore, J.A. Goguen generalized the fuzzy sets introduced by Zadeh to $\mathcal{L}$-fuzzy sets [37, 44] where it is assumed a set, denoted as $\mathcal{L}$, of degrees of membership. Then, in order to make sense to ask what the maximum and minimum values of an $\mathcal{L}$-fuzzy set are, it needs some kind of ordering. Therefore, $\mathcal{L}$ must be, in general, a lattice.

3.3.1 Lattice theory

A lattice is a partially ordered set in which every two elements have a supremum (also called a *least upper bound*) and an infimum (also called a *greatest lower bound*). An example is given by the natural numbers, partially ordered by divisibility, for which the supremum is the least common multiple and the infimum is the greatest common divisor. Therefore, to give the definition of a complete

lattice [17], we have to start from that of a partially ordered set.

Definition 3.3.1 (Poset)**.** A partially ordered set (poset) is couple $(P, \leq_P)$, where P is a non-empty set and $\leq_P$ is a binary relation on P that satisfies:

- $(\forall x \in P)(x \leq_P x)$ (reflexivity).
- $(\forall (x, y) \in P^2)(x \leq_P y \text{ and } y \leq_P x \Rightarrow x = y)$ (anti-symmetry).
- $(\forall (x, y, z) \in P^3)(x \leq_P y \text{ and } y \leq_P z \Rightarrow x \leq_P z)$ (transitivity).

If further also each two elements in the partially ordered set $(P, \leq_P)$ are comparable (i.e., $(\forall (x, y) \in P^2)(x \leq_P y \text{ or } y \leq_P x)$), then $(P, \leq_P)$ is called a totally ordered set or chain. The length of a chain is given by the cardinality of the chain minus one.

Some important concepts that are defined in a poset are the following:

Definition 3.3.2. Let $(P, \leq_P)$ be a poset, $A \subseteq P$ and $b \in P$.

- b is an upper bound of $A \Leftrightarrow (\forall a \in A)(a \leq_P b)$,
- b is a lower bound of $A \Leftrightarrow (\forall a \in A)(b \leq_P a)$,
- A is bounded above in $(P, \leq_P) \Leftrightarrow (\exists b \in P)$ (b is an upper bound of A),
- A is bounded below in $(P, \leq_P) \Leftrightarrow (\exists b \in P)$ (b is a lower bound of A),
- A is bounded in $(P, \leq_P) \Leftrightarrow A$ is bounded above and A is bounded below,
- b is the greatest element of $A \Leftrightarrow b \in A$ and b is an upper bound of A,
- b is the least element of $A \Leftrightarrow b \in A$ and b is a lower bound of A,
- b is the supremum of $A (b = \sup A) \Leftrightarrow b$ is the least upper bound of A,
- b is the infimum of $A (b = \inf A) \Leftrightarrow b$ is the greatest lower bound of A.

By the help of those concepts, the definition of a complete lattice can be given.

Definition 3.3.3 (Lattice)**.** A poset $(P, \leq_P)$ is called a lattice if every doubleton in P has a supremum and infimum.

Definition 3.3.4 (Complete lattice)**.** A lattice $(L, \leq_L)$ is called complete if every non-empty subset of L has a supremum and infimum.

Definition 3.3.5 (Bounded lattice)**.** A bounded lattice is a lattice $(L, \leq_L)$ that additionally has a greatest element 1 and a least element 0, which satisfy

$$0 \leq x \leq 1, \forall x \in L.$$

Remark that $([0,1], \leq)$ forms a complete lattice and that fuzzy sets as introduced by Zadeh are a special case of $\mathcal{L}$-fuzzy sets.

3.3.2 Membership function

For a complete lattice $\mathcal{L} = (L, \leq_L)$, an $\mathcal{L}$-fuzzy set A in a universe U is characterised by its membership function χ_A:

$$\begin{aligned} \chi_A : \quad & U \to L, \\ & x \to \chi_A(x), \end{aligned} \tag{3.38}$$

in such a way that $\chi_A(x)$ is the membership degree of an element $x \in U$ in the $\mathcal{L}$-fuzzy set A. The higher this degree (w.r.t. $\leq_L$), the more the element belongs to the set.

Remark that $([0,1], \leq)$ forms a complete lattice and the fuzzy sets as introduced by Zadeh are a special case of $\mathcal{L}$-fuzzy sets.

3.3.3 Cartesian product of lattices

Given a lattice $L = \{L, \leq, \vee, \wedge\}$, which is a poset with the partial ordering $\leq$ in L and operations $\vee$ and $\wedge$ which satisfy the properties of absorption, idempotency, commutativity, and associativity. That is, a poset such that any two elements a, b have an *unique minimal upper bound* $a \vee b$ and an *unique maximal lower bound* $a \wedge b$ in L. The cartesian product of lattices can be defined as follows:

Proposition 3.3.1. [24] Let $L_1 = \{L_1, \leq_1, \vee_1, \wedge_1\}$ and $L_2 = \{L_2, \leq_2, \vee_2, \wedge_2\}$ be two lattices. The cartesian product $L_1 \times L_2 = \{L_1 \times L_2, \leq, \vee, \wedge\}$ with $\leq$ defined componentwise by

$$(x_1, x_2) \leq (y_1, y_2) \text{ if and only if } x_1 \leq_1 y_1 \text{ and } x_2 \leq_2 y_2$$

and

$$\vee((x_1, x_2), (y_1, y_2)) = (\vee_1(x_1, y_1), \vee_2(x_2, y_2)),$$

$$\wedge((x_1, x_2), (y_1, y_2)) = (\wedge_1(x_1, y_1), \wedge_2(x_2, y_2))$$

is a lattice.

3.3.4 $\mathcal{L}$-Fuzzy logical operators

Definitions

The fuzzy logical operators on $[0, 1]$ can be extended to operators on $\mathcal{L} = (L, \leq_L)$ as follows.

Definition 3.3.6 (Negator)**.**

- A negator $\mathcal{N}$ on $\mathcal{L}$ is a decreasing $L - L$ mapping (w.r.t. $\leq_L$) that satisfies $\mathcal{N}(0_\mathcal{L}) = 1_\mathcal{L}$ and $\mathcal{N}(1_\mathcal{L}) = 0_\mathcal{L}$.
- A negator $\mathcal{N}$ is an involutive negator on $\mathcal{L}$ if $(\forall x \in L)(\mathcal{N}(\mathcal{N}(x)) = x)$.

Definition 3.3.7 (Conjunctor)**.**

- A conjunctor $\mathcal{C}$ on $\mathcal{L}$ is an increasing $L^2 - L$ mapping (w.r.t. $\leq_L$) that satisfies $\mathcal{C}(0_\mathcal{L}, 0_\mathcal{L}) = \mathcal{C}(0_\mathcal{L}, 1_\mathcal{L}) = \mathcal{C}(1_\mathcal{L}, 0_\mathcal{L}) = 0_\mathcal{L}$ and $\mathcal{C}(1_\mathcal{L}, 1_\mathcal{L}) = 1_\mathcal{L}$.
- A conjunctor $\mathcal{C}$ is a semi-norm on $\mathcal{L}$ if it satisfies $(\forall x \in L)(\mathcal{C}(1_\mathcal{L}, x) = \mathcal{C}(x, 1_\mathcal{L}) = x)$.
- A semi-norm $\mathcal{C}$ is a t-norm on $\mathcal{L}$ if it is commutative and associative.

Definition 3.3.8 (Disjunctor)**.**

- A disjunctor $\mathcal{D}$ on $\mathcal{L}$ is an inreasing $L^2 - L$ mapping (w.r.t. $\leq_L$) that satisfies $\mathcal{D}(1_\mathcal{L}, 1_\mathcal{L}) = \mathcal{D}(0_\mathcal{L}, 1_\mathcal{L}) = \mathcal{D}(1_\mathcal{L}, 0_\mathcal{L}) = 1_\mathcal{L}$ and $\mathcal{D}(0_\mathcal{L}, 0_\mathcal{L}) = 0_\mathcal{L}$.
- A disjunctor $\mathcal{D}$ is a semi-conorm on $\mathcal{L}$ if it satisfies $(\forall x \in L)(\mathcal{D}(0_\mathcal{L}, x) = \mathcal{D}(x, 0_\mathcal{L}) = x)$.
- A semi-conorm $\mathcal{D}$ is a t-conorm on $\mathcal{L}$ if it is commutative and associative.

Definition 3.3.9 (Implicator)**.**

- An implicator $\mathcal{I}$ on $\mathcal{L}$ is a hybrid monotonic $L^2 - L$ mapping (i.e., decreasing in the first argument (w.r.t. $\leq_L$) and increasing in the second argument (w.r.t. $\leq_L$)) that satisfies $\mathcal{I}(0_\mathcal{L}, 0_\mathcal{L}) = \mathcal{I}(0_\mathcal{L}, 1_\mathcal{L}) = \mathcal{I}(1_\mathcal{L}, 1_\mathcal{L}) = 1_\mathcal{L}$ and $\mathcal{I}(1_\mathcal{L}, 0_\mathcal{L}) = 0_\mathcal{L}$. Every implicator $\mathcal{I}$ induces a negator $\mathcal{N}_\mathcal{I}(x) = \mathcal{I}(x, 0_\mathcal{L}), \forall x \in L$.

- An implicator $\mathcal{I}$ is a border implicator on $\mathcal{L}$ if it satisfies $(\forall x \in L)(\mathcal{I}(1_{\mathcal{L}}, x) = x)$.

- A border implicator $\mathcal{I}$ is a model implicator on $\mathcal{L}$ if it is contrapositive w.r.t. its induced negator, i.e., $(\forall (x, y) \in L^2)(\mathcal{I}(x, y) = \mathcal{I}(\mathcal{N}_{\mathcal{I}}(y), \mathcal{N}_{\mathcal{I}}(x)))$, and if it fulfills the exchange principle, i.e., $(\forall (x, y, z) \in L^3)(\mathcal{I}(x, \mathcal{I}(y, z)) = \mathcal{I}(y, \mathcal{I}(x, z)))$.

In the above definition, it is already mentioned that every implicator $\mathcal{I}$ on $\mathcal{L}$ induces a negator $\mathcal{N}_{\mathcal{I}}$ on $\mathcal{L}$ given by $\mathcal{N}_{\mathcal{I}}(x) = \mathcal{I}(x, 0_{\mathcal{L}}), \forall x \in L$. Further, also conjunctors and implicators can be induced by other logical operators.

Let $\mathcal{N}$ and $\mathcal{C}$ be respectively a negator and a conjunctor on $\mathcal{L}$. Then the operator $\mathcal{D}_{\mathcal{CN}}$ given by

$$\mathcal{D}_{\mathcal{CN}}(x, y) = \mathcal{N}(\mathcal{C}(\mathcal{N}(x), \mathcal{N}(y))), \forall (x, y) \in L^2,$$

is a disjunctor on $\mathcal{L}$. Analogously, if $\mathcal{N}$ and $\mathcal{D}$ are respectively a negator and a disjunctor on $\mathcal{L}$, then the operator $\mathcal{C}_{\mathcal{DN}}$ given by

$$\mathcal{C}_{\mathcal{DN}}(x, y) = \mathcal{N}(\mathcal{D}(\mathcal{N}(x), \mathcal{N}(y))), \forall (x, y) \in L^2,$$

is a disjunctor on $\mathcal{L}$. If $\mathcal{N}$ is an involutive negator, then a conjunctor $\mathcal{C}$ and a disjunctor $\mathcal{D}$ are called dual with respect to $\mathcal{N}$ if and only if $\mathcal{C} = \mathcal{C}_{\mathcal{D},\mathcal{N}}$ and $\mathcal{D} = \mathcal{D}_{\mathcal{C},\mathcal{N}}$.

Let $\mathcal{N}$ and $\mathcal{I}$ be respectively a negator and an implicator on $\mathcal{L}$. Then the operator $\mathcal{C}_{\mathcal{IN}}$ given by

$$\mathcal{C}_{\mathcal{IN}}(x, y) = \mathcal{N}(\mathcal{I}(x, \mathcal{N}(y))), \forall (x, y) \in L^2,$$

is a conjunctor on $\mathcal{L}$ and it is called the conjunctor induced by $\mathcal{I}$ and $\mathcal{N}$.

Let $\mathcal{N}$ and $\mathcal{C}$ be respectively a negator and a conjunctor on $\mathcal{L}$. Then the operator $\mathcal{I}_{\mathcal{CN}}$ given by

$$\mathcal{I}_{\mathcal{CN}}(x, y) = \mathcal{N}(\mathcal{C}(x, \mathcal{N}(y))), \forall (x, y) \in L^2,$$

is an implicator on $\mathcal{L}$ and it is called the implicator induced by $\mathcal{C}$ and $\mathcal{N}$.

Let $\mathcal{N}$ and $\mathcal{D}$ be respectively a negator and a disjunctor on $\mathcal{L}$. Then the

operator $\mathcal{I}_{\mathcal{DN}}$ given by

$$\mathcal{I}_{\mathcal{DN}}(x,y) = \mathcal{N}(\mathcal{D}(\mathcal{N}(x),y)), \forall (x,y) \in L^2,$$

is an implicator on $\mathcal{L}$ and it is called the implicator induced by $\mathcal{D}$ and $\mathcal{N}$.

Let $\mathcal{C}$ be a conjunctor on $\mathcal{L}$ that satisfies $(\forall x \in L)(\mathcal{C}(1_{\mathcal{L}}, x) = 0_{\mathcal{L}} \Rightarrow x = 0_{\mathcal{L}})$. Then the operator $\mathcal{I}_{\mathcal{C}}$ given by

$$\mathcal{I}_{\mathcal{C}}(x,y) = \sup\{z \in L \mid \mathcal{C}(x,z) \leq_L y\},\ \forall (x,y) \in L^2,$$

is an implicator on $\mathcal{L}$ and it is called the residual implicator of $\mathcal{C}$ [37].

Further, the order relation $\leq_L$ on the lattice $\mathcal{L}$ can be extended to the logical operators as follows:

Definition 3.3.10 (Ordering of $\mathcal{L}$-fuzzy logical operators)**.**

- Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be two negators on $\mathcal{L}$, then

$$\mathcal{N}_1 \leq_L \mathcal{N}_2 \Leftrightarrow (\forall x \in L)(\mathcal{N}_1(x) \leq_L \mathcal{N}_2(x)).$$

- Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two conjunctors on $\mathcal{L}$, then

$$\mathcal{C}_1 \leq_L \mathcal{C}_2 \Leftrightarrow (\forall (x,y) \in L^2)(\mathcal{C}_1(x,y) \leq_L \mathcal{C}_2(x,y)).$$

- Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two disjunctors on $\mathcal{L}$, then

$$\mathcal{D}_1 \leq_L \mathcal{D}_2 \Leftrightarrow (\forall (x,y) \in L^2)(\mathcal{D}_1(x,y) \leq_L \mathcal{D}_2(x,y)).$$

- Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be two implicators on $\mathcal{L}$, then

$$\mathcal{I}_1 \leq_L \mathcal{I}_2 \Leftrightarrow (\forall (x,y) \in L^2)(\mathcal{I}_1(x,y) \leq_L \mathcal{I}_2(x,y)).$$

3.3.5 $\mathcal{L}$-Fuzzy set operations

Complement, intersection and union of $\mathcal{L}$-fuzzy sets

The $\mathcal{L}$-fuzzy logical operators can be used to define the complement, intersection and union of $\mathcal{L}$-fuzzy sets:

Definition 3.3.11 ($\mathcal{N}$-complement)**.** Let A be an $\mathcal{L}$-fuzzy set in the universe U. If $\mathcal{N}$ is a negator on $\mathcal{L}$, then the $\mathcal{N}$-complement $\mathrm{co}_{\mathcal{N}}(A)$ of A is defined as the $\mathcal{L}$-fuzzy set in U given by:

$$(\mathrm{co}_{\mathcal{N}}(A))(x) = \mathcal{N}(A(x)),\ \forall x \in U.$$

Definition 3.3.12 ($\mathcal{C}$-intersection)**.** Let A and B be two $\mathcal{L}$-fuzzy sets in the universe U. If $\mathcal{C}$ is a conjunctor on $\mathcal{L}$, then the $\mathcal{C}$-intersection $A \cap_{\mathcal{C}} B$ of A and B is defined as the $\mathcal{L}$-fuzzy set in U given by:

$$(A \cap_{\mathcal{C}} B)(x) = \mathcal{C}(A(x), B(x)),\ \forall x \in U.$$

Definition 3.3.13 ($\mathcal{D}$-union)**.** Let A and B be two $\mathcal{L}$-fuzzy sets in the universe U. If $\mathcal{D}$ is a disjunctor on $\mathcal{L}$, then the $\mathcal{D}$-union $A \cup_{\mathcal{D}} B$ of A and B is defined as the $\mathcal{L}$-fuzzy set in U given by:

$$(A \cup_{\mathcal{D}} B)(x) = \mathcal{D}(A(x), B(x)),\ \forall x \in U.$$

If $\mathcal{C}$ (respectively $\mathcal{D}$) is the infimum operator (respectively the supremum operator), then the intersection (respectively union) is called the *Zadeh-intersection* (respectively *Zadeh-union*) and the notation $\cap_{\mathcal{C}}$ is simplified to $\cap$ (respectively $\cup_{\mathcal{D}}$ is simplified to $\cup$).

If $\mathcal{C}$ and $\mathcal{D}$ are commutative and associative (in particular if they are a t-norm and a t-conorm), then the above definitions can be extended to the intersection and union of an arbitrary finite family of $\mathcal{L}$-fuzzy sets. If further the conjunctor $\mathcal{C}$ and the disjunctor $\mathcal{D}$ can also be extended to an infinite number of arguments, then also an extension to infinite families is possible. For the *Zadeh-intersection* and *Zadeh-union* and an arbitrary (infinite) family $(A_j)_{j \in J}$ of $\mathcal{L}$-fuzzy sets in U, this becomes:

$$(\cap_{j \in J} A_J)(x) = \inf_{j \in J} A_j(x),\ \forall x \in U,$$
$$(\cup_{j \in J} A_J)(x) = \sup_{j \in J} A_j(x),\ \forall x \in U.$$

Containment of $\mathcal{L}$-fuzzy sets

Containment of $\mathcal{L}$-fuzzy sets is defined as follows:

Definition 3.3.14 (Containment)**.** Let A and B be two $\mathcal{L}$-fuzzy sets in the universe U, then

$$A \subseteq B \Leftrightarrow (\forall x \in U)(A(x) \leq_L B(x)).$$

II

Proposed methodology

4

Consensus decision-making for image restoration

Aggregation of several input values into a single output value is an indispensable tool in many disciplines and applications such as decision making. Specifically, *penalty-based decision making* is a strategy mainly used when the best solution among the availables is not known in advance, in such a way that we choose the solution that produces less error among the available solutions. In other words, it is a *consensus methodology* that obtains a global solution that combines the single inputs, instead of using one of them as solution for the whole process [24–26]. This philosophy has no information about whether all the inputs are representative or just some of them. This is our motivation to use a set of aggregation functions that previously merges the input candidates. We start from a set of aggregation functions and select, by means of consensus done through penalty functions, the most suitable aggregation function in order to aggregate the individual preferences for each of the elements. The whole consensus strategy consists of two phases: an *aggregation phase* and an *exploitation phase*.

For the **aggregation phase** the input candidates are merged using *averaging aggregation functions*. Specifically, we choose the family of parameterized averaging aggregation functions formed by the *ordered weighted averaging (OWA)* operators since they offer more flexibility when combining weighted information. Moreover, the OWA aggregated value is often interpreted as some sort of representative, or consensus value of the inputs. Nonetheless, the possible operators to consider are unmanageable and we are not aware of the best candidate, so we can only assume a subset of operators based on our experience. In other words,

the choice of the different sets of q aggregation functions to be used will depend on the specific problem under consideration.

Then in the **exploitation phase**, we transform the set of aggregated outputs in only one that represents the largest number of inputs. For this purpose, a penalty function is used to select the aggregation value that minimizes the penalty with respect to the inputs and is given as a solution. We must also consider that using penalty functions, if we take all the q aggregation functions as equal then we recover the classical methods for the aggregation phase.

Hence the consensus methodology can be seen as a *framework* applied on different situations depending on the problem we are dealing with. In image restoration, we start from an input set consisting of a set of matrices (images) where their membership functions are usually an intensity degree or a confidence value. Then, we work with these images to finally get a single image that reaches the consensus. We focus on the study of *penalty-based decision making over cartesian product of lattices*, a methodology within fuzzy decision-making. The use of cartesian product of lattices allows to define penalty functions over a lattice or to smaller chains of this lattice, i.e., consensus methodology can be applied over the entire set of matrices (images) or over small sets of these matrices (pixel regions) without distinction. However, to be able to consider all possible alternatives for each element (pixel) using penalty functions by regions, an *intermediate phase* is needed to add into the consensus process. In such a way that penalty-based decision making over cartesian product of lattices is as follows: aggregation, combination, and exploitation phase. Where in the combination phase all possible alternatives for each element are generated. Note that if the chain size is simplified to one element, we recover the method applied element by element, so the combination phase is not required.

So far an overview of the methodology phases has been given. However, various definitions are necessary to define them. Thus the remaining of this chapter is organized as follows: Section 4.1 provides an introduction of the influence of *fuzzy decision-making* in image processing. Followed by Section 4.2 that introduces *multifuzzy sets*, a representation method of the set of images. Then, the *idempotent aggregation functions* are in Section 4.3, that presents the idempotent functions, a construction method and their properties. Section 4.4 introduces the *averaging functions*, a family of idempotent aggregation functions, and a specific case: the *OWA operators*. They are the functions we use in the aggregation

phase, in such way that they convert the multifuzzy set of images into a single image. Then, for the exploitation phase, Section 4.5 explains the *penalty functions*. Followed by an extension of the *penalty functions over cartesian product of lattices* in Section 4.6. To conclude, a detailed description of the *proposed method* is given in Section 4.7.

4.1 Background: fuzzy decision-making in image processing

Fuzzy set theory has already been widely used in image processing to model uncertainty, being used with success in segmentation [21, 22, 74], compression and decompression [73, 84], image clustering [14], and definitely noise reduction [28, 78, 98, 106], among other applications.

However, there are situations where the use only of a single fuzzy technique is not enough to model the uncertainty of the entire problem, so that the use of a set of fuzzy alternatives to build a solution can help to solve the problem, and thus better model the uncertainty. A simple solution is to use some fusion operator that aggregates the individual techniques. For instance, an *aggregation operator* [11, 65] can provide an appropriate solution as already shown with other techniques in image processing [82, 83]. Nonetheless, it is obvious that using different techniques we can accomplish different actions according to the desirability in the problem, that in many cases, cannot be done by using a single fusion criterion. Therefore, decision making can be used to address this issue.

The study of decision making is necessary and very important not only in decision theory but also in areas such as operational research, management, science, politics, social psychology, artificial intelligence, etc. By decision making in a fuzzy environment is meant a decision process in which the goals and/or the constraints constitute classes of alternatives that are fuzzy [30, 63, 64]. In other words, fuzzy goals and fuzzy constraints are defined precisely as fuzzy sets in the space of alternatives. Before achieving the decision, a *decision-making method* mainly follows two steps: aggregation and exploitation. The aggregation phase defines a collective relation between the alternatives indicating the preference between them. Besides the exploitation phase transforms the collective preferences into a global ranking. This can be done in different ways, the most common one

being the use of a ranking method to obtain a score function.

Due to the flexibility presented in the aggregation and exploitation phase definition, fuzzy decision-making can be easily adapted in accordance to the necessities of the problem. In this way, some methods are already presented in image segmentation [59, 79] and image reduction [12] using fuzzy decision-making. For instance, the colour segmentation method proposed in [79] uses a decision model based on the linguistic fuzzy representation of 2-tuples. It obtains a segmentation transforming the colour space for the original image into a new space, and taking into account the preference degree provided by a set of experts in the assignment of each pixel to one object or another in the image. In the aggregation phase, fuzzy linguistic quantifiers aggregate the preference values associated with a pixel and with its neighbouring ones. Then, the exploitation phase classifies each pixel creating the segmented image by assigning the object label that presents the highest membership value. They also consider the possibility to choose the set of experts, what adds a great value to their method. In this sense, the choice of experts which is best suited for a given application can improve the global results. The other segmentation method proposed in [59] obtains the threshold for segmenting the white and grey matter for magnetic resonance imaging (MRI) using grouping functions through a decision-making process. In the aggregation phase, it chooses a set of grouping functions that are aggregated by a convex combination of several of them, in such a way that avoids the selection of a suitable grouping function for each image. Then, in the exploitation phase, the threshold is obtained as the one that presents the maximum sum of grouping. On the other side, three algorithms for colour image reduction based on minimizing penalty functions has been presented in [12], showing that the obtained reduced image is even robust to impulse noise. For the aggregation phase, they use a set of aggregation functions that aggregates blocks of neighbour pixels. After, in the exploitation phase, the solution for each block is taken as the aggregation function that presents the minimum penalty with respect to the original pixels. Concretely, *fuzzy decision-making based on penalty functions* has been previously studied [23–26], although not yet been used in image restoration.

4.2 Multifuzzy sets

Multifuzzy sets, also known as fuzzy multisets, are a great tool to handle uncertainty by allowing several membership values. They are a natural generalization of the (crisp) multisets, that have been sometimes called *bags* [76]. Therefore, we will use multifuzzy sets to represent a set of elements in a cartesian product. In this thesis we use them to represent image elements where their membership function is usually an intensity or a confidence degree.

Definition 4.2.1. [10, 110] A multifuzzy set of dimension $n \geq 2$ over a finite universe U is defined by a mapping

$$A : U \to [0,1]^n,$$

given by

$$A(u) = (A_1(u), \ldots, A_n(u)),$$

where each of the A_j for $j = 1, \ldots, n$ is a fuzzy set $A_j : U \to [0,1]$.

Notice that the previous definition is equivalent to the following. Take a family of $n \geq 2$ fuzzy sets $Q_1, \ldots, Q_n$ on the same referential set U. Then an n-dimensional multifuzzy set on U is just the ordered combination of these n fuzzy sets as follows:

$$A = \{(u, A(u)) | u \in U\} \text{ given by } A(u) = (Q_1(u), \ldots, Q_n(u)).$$

In this sense, the space of all multifuzzy sets inherits the order from the usual fuzzy sets, which endows it with a partial, bounded order.

In this thesis, we deal with two finite referential sets $X = \{0, 1, \ldots, N-1\}$ and $Y = \{0, 1, \ldots, M-1\}$, where N and M are the number of rows and columns of the image, respectively. We consider multifuzzy sets defined on the cartesian product $X \times Y$.

Notice that an n-dimensional multifuzzy set can also be understood as a type n fuzzy set, as well as an $\mathcal{L}$-fuzzy set with $\mathcal{L} = [0,1]^n$.

Example 4.2.1. Being $U = X \times Y$, where $X = \{0, 1, 2\}$ and $Y = \{0, 1, 2, 3\}$. Then, the multifuzzy set A (Figure 4.1) is defined as:

0.9	0.8	0.7	0.8
0.9	0.6	0.7	0.8
0.5	0.4	0.3	0.3

(a) A_1

0.8	0.7	0.8	0.8
0.1	0.9	0.9	0.9
0.3	0.2	0.2	0.1

(b) A_2

1.0	1.0	0.9	0.8
0.8	0.9	0.9	0.7
0.6	0.6	0.4	0.3

(c) A_3

Figure 4.1: Multifuzzy set $A = (A_1, A_2, A_3)$.

0.3	0.2	0.6
0.1	1.0	0.2
0.2	0.4	0.5

(a) B_1

0.2	0.3	0.4
1.0	1.0	0.6
0.7	1.0	0.8

(b) B_2

0.2	0.2	0.3
0.6	0.4	0.5
0.7	0.8	0.9

(c) B_3

0.1	0.1	0.3
0.4	0.6	0.8
0.0	1.0	0.8

(d) B_4

Figure 4.2: Multifuzzy set $B = (B_1, B_2, B_3, B_4)$.

$$\begin{aligned} A = (&A(0,0), A(0,1), A(0,2), A(0,3),\\ &A(1,0), A(1,1), A(1,2), A(1,3),\\ &A(2,0), A(2,1), A(2,2), A(2,3)) \end{aligned}$$

where

$$\begin{aligned} A(0,0) &= (A_1(0,0), A_2(0,0), A_3(0,0)) = (0.9, 0.8, 1.0)\\ A(0,1) &= (A_1(0,1), A_2(0,1), A_3(0,1)) = (0.8, 0.7, 1.0)\\ A(0,2) &= (A_1(0,2), A_2(0,2), A_3(0,2)) = (0.7, 0.8, 0.9)\\ A(0,3) &= (A_1(0,3), A_2(0,3), A_3(0,3)) = (0.8, 0.8, 0.8) \end{aligned}$$

$$\begin{aligned} A(1,0) &= (A_1(1,0), A_2(1,0), A_3(1,0)) = (0.9, 0.1, 0.8)\\ A(1,1) &= (A_1(1,1), A_2(1,1), A_3(1,1)) = (0.6, 0.9, 0.9)\\ A(1,2) &= (A_1(1,2), A_2(1,2), A_3(1,2)) = (0.7, 0.9, 0.9)\\ A(1,3) &= (A_1(1,3), A_2(1,3), A_3(1,3)) = (0.8, 0.9, 0.7) \end{aligned}$$

$$A(2,0) = (A_1(2,0), A_2(2,0), A_3(2,0)) = (0.5, 0.3, 0.6)$$
$$A(2,1) = (A_1(2,1), A_2(2,1), A_3(2,1)) = (0.4, 0.2, 0.6)$$
$$A(2,2) = (A_1(2,2), A_2(2,2), A_3(2,2)) = (0.3, 0.2, 0.4)$$
$$A(2,3) = (A_1(2,3), A_2(2,3), A_3(2,3)) = (0.3, 0.1, 0.3)$$

Example 4.2.2. Being $U = X \times Y$, where $X = \{0, 1, 2\}$ and $Y = \{0, 1, 2\}$. Then, the multifuzzy set B (Figure 4.2) is defined as:

$$\begin{aligned} B = (&B(0,0), B(0,1), B(0,2),\\ &B(1,0), B(1,1), B(1,2),\\ &B(2,0), B(2,1), B(2,2)) \end{aligned}$$

where

$$B(0,0) = (B_1(0,0), B_2(0,0), B_3(0,0), B_4(0,0)) = (0.3, 0.2, 0.2, 0.1)$$
$$B(0,1) = (B_1(0,1), B_2(0,1), B_3(0,1), B_4(0,1)) = (0.2, 0.3, 0.2, 0.1)$$
$$B(0,2) = (B_1(0,2), B_2(0,2), B_3(0,2), B_4(0,2)) = (0.6, 0.4, 0.3, 0.3)$$

$$B(1,0) = (B_1(1,0), B_2(1,0), B_3(1,0), B_4(1,0)) = (0.1, 1.0, 0.6, 0.4)$$
$$B(1,1) = (B_1(1,1), B_2(1,1), B_3(1,1), B_4(1,1)) = (1.0, 1.0, 0.4, 0.6)$$
$$B(1,2) = (B_1(1,2), B_2(1,2), B_3(1,2), B_4(1,2)) = (0.2, 0.6, 0.5, 0.8)$$

$$B(2,0) = (B_1(2,0), B_2(2,0), B_3(2,0), B_4(2,0)) = (0.2, 0.7, 0.7, 0.0)$$
$$B(2,1) = (B_1(2,1), B_2(2,1), B_3(2,1), B_4(2,1)) = (0.4, 1.0, 0.8, 1.0)$$
$$B(2,2) = (B_1(2,2), B_2(2,2), B_3(2,2), B_4(2,2)) = (0.5, 0.8, 0.9, 0.8)$$

4.3 Idempotent functions

A crucial step in the aggregation phase of the consensus methodology is how to recover a single fuzzy set from a multifuzzy set. Therefore, we need functions that take a set of inputs and obtain one single value satisfying one condition: if all the input values are the same, the value remains the same. For this reason we decide to use *idempotent functions*.

Definition 4.3.1. An n-dimensional idempotent function is a mapping $\gamma : [0,1]^n \to [0,1]$ such that

$$\gamma(x, \ldots, x) = x,$$

for every $x \in [0,1]$.

Example 4.3.1. Some of the idempotent functions are:

1. The mode, that is the value that occurs most frequently in a data set or a probability distribution.
2. Smallest idempotent function

$$\gamma_{\text{smallest}}(x_1, \ldots, x_n) = \begin{cases} 0, & \text{if } \exists i, j \in \{1, \ldots, n\} \text{ such that } x_i \neq x_j, \\ x_1, & \text{in other case.} \end{cases}$$

3. Largest idempotent function

$$\gamma_{\text{largest}}(x_1, \ldots, x_n) = \begin{cases} 1, & \text{if } \exists i, j \in \{1, \ldots, n\} \text{ such that } x_i \neq x_j, \\ x_1, & \text{in other case.} \end{cases}$$

Remark 4.3.1. Notice, that neither of the functions from Example (4.3.1) is monotone.

Example 4.3.2. The mode (γ_{mode}) is an example of an idempotent function which is not monotone increasing. In Figure 4.4(a) can be seen the results for $\gamma_{\text{mode}}(C)$ in case repeated elements exist, as in the matrices C (Figure 4.3). Otherwise, it could be applied, for instance, an adaptation of the mode as it is defined below:

0.2	0.4	0.5
0.8	0.1	0.7
0.1	0.4	0.2

(a) C_1

0.2	0.4	0.7
0.9	0.4	0.6
0.9	0.4	0.2

(b) C_2

0.3	0.4	0.5
0.9	0.4	0.7
0.9	0.5	0.2

(c) C_3

Figure 4.3: Multifuzzy set $C = (C_1, C_2, C_3)$.

0.2	0.4	0.5
0.9	0.4	0.7
0.9	0.4	0.2

(a) $\gamma_{\text{mode}}(C)$

0.8	0.7	0.7	0.8
0.1	0.9	0.9	0.7
0.3	0.2	0.2	0.3

(b) $\gamma_{\text{ad\_mode}}(A)$

Figure 4.4: Results mode examples.

$$\gamma_{\text{ad\_mode}}(x_1, \ldots, x_n) = \begin{cases} \text{mode}(x_1, \ldots, x_n), & \text{if } \exists i, j \in \{1, \ldots, n\} \text{ such that } x_i = x_j, \\ \min(x_1, \ldots, x_n), & \text{in other case.} \end{cases} \tag{4.1}$$

Figure 4.4(b) shows an example of the calculus with the adapted mode (Ex. (4.1)) with the matrices A from the previous section (Figure 4.1).

4.3.1 Construction of idempotent functions

In Proposition 4.3.1 we present a method for *constructing idempotent functions*.

Proposition 4.3.1. The mapping $\gamma : [0,1]^n \to [0,1]$ is an n-dimensional idempotent function if and only if there exist $f, g : [0,1]^n \to [0,1]$ such that

(i) $g(x, \ldots, x) \neq 0$ for every $x \in [0, 1[$;

(ii) $f(x, \ldots, x) = \frac{x}{1-x} g(x, \ldots, x)$ for $x \in [0, 1[$, $f(1, \ldots, 1) = 1$ and $g(1, \ldots, 1) = 0$;

(iii) $\gamma(x_1, \ldots, x_n) = \frac{f(x_1, \ldots, x_n)}{f(x_1, \ldots, x_n) + g(x_1, \ldots, x_n)}$.

Proof. Assume that γ is an n-dimensional idempotent function. Take $f = \gamma$ and $g = 1 - \gamma$. Then

(i) $g(x, \ldots, x) = 1 - \gamma(x, \ldots, x) = 1 - x \neq 0$ for every $x \in [0, 1[$;

(ii) $\frac{x}{1-x}g(x,\ldots,x) = \frac{x}{1-x}(1-x) = x = \gamma(x,\ldots,x) = f(x,\ldots,x)$ and $f(1,\ldots,1) = \gamma(1,\ldots,1) = 1$ and $g(1,\ldots,1) = 0$;

(iii) $\frac{f(x_1,\ldots,x_n)}{f(x_1,\ldots,x_n)+g(x_1,\ldots,x_n)} = \gamma(x_1,\ldots,x_n)$.

To see the converse, we only need to check the idempotency. But if γ is defined as in the statement of the proposition, we have that $\gamma(x,\ldots,x) = \frac{f(x,\ldots,x)}{f(x,\ldots,x)+g(x,\ldots,x)} = \frac{\frac{x}{1-x}g(x,\ldots,x)}{\frac{x}{1-x}g(x,\ldots,x)+g(x,\ldots,x)}$ which is equal to x for every $x \in [0,1[$. Finally, if $x = 1$ then clearly $\gamma(1,\ldots,1) = 1$. □

Remark 4.3.2. The following is an example of an idempotent function built by means of Proposition 4.3.1.

$$\gamma(x_1,\ldots,x_p) = \frac{f(x_1,\ldots,x_p)}{f(x_1,\ldots,x_p) + \max(1-x_1,\ldots,1-x_p)}$$

Here we have that $f(x,\ldots,x) = x$ and $g(x,\ldots,x) = 1-x$.

Example 4.3.3.

- Taking

$$f(x_1,\ldots,x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i \text{ and } g(x_1,\ldots,x_n) = \frac{1}{n}\sum_{i=1}^{n}(1-x_i),$$

we obtain as idempotent function the arithmetic mean ($\gamma_{\text{mean}}(\mathbf{x})$) shown in Eq. (4.2):

$$\gamma_{\text{mean}}(x_1,\ldots,x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i. \tag{4.2}$$

- Taking

$$f(x_1,\ldots,x_n) = \sqrt[n]{x_1\cdot x_2\cdot\ldots\cdot x_n} \text{ and } g(x_1,\ldots,x_n) = \max(1-x_1,\ldots,1-x_n),$$

we get as idempotent function the $\gamma_{\text{root}}(\mathbf{x})$ shown in Eq. (4.3):

$$\gamma_{\text{root}}(x_1,\ldots,x_n) = \frac{\sqrt[n]{x_1\cdot x_2\cdot\ldots\cdot x_n}}{\sqrt[n]{x_1\cdot x_2\cdot\ldots\cdot x_n} + \max(1-x_1,\ldots,1-x_n)} \tag{4.3}$$

Example 4.3.4. Using the matrices A from Figure 4.1, and the aggregation function $\gamma_{\text{root}}(\mathbf{x})$ from Eq. (4.3) , we can calculate $\gamma_{\text{root}}(A)$ as

$$\gamma_{\text{root}}\left(A_1(0,0), A_2(0,0), A_3(0,0)\right) = 0.8176$$
$$\gamma_{\text{root}}\left(A_1(0,1), A_2(0,1), A_3(0,1)\right) = 0.7332$$
$$\gamma_{\text{root}}\left(A_1(0,2), A_2(0,2), A_3(0,2)\right) = 0.7262$$
$$\gamma_{\text{root}}\left(A_1(0,3), A_2(0,3), A_3(0,3)\right) = 0.8000$$

$$\gamma_{\text{root}}\left(A_1(1,0), A_2(1,0), A_3(1,0)\right) = 0.3161$$
$$\gamma_{\text{root}}\left(A_1(1,1), A_2(1,1), A_3(1,1)\right) = 0.6628$$
$$\gamma_{\text{root}}\left(A_1(1,2), A_2(1,2), A_3(1,2)\right) = 0.7340$$
$$\gamma_{\text{root}}\left(A_1(1,3), A_2(1,3), A_3(1,3)\right) = 0.7262$$

$$\gamma_{\text{root}}\left(A_1(2,0), A_2(2,0), A_3(2,0)\right) = 0.3903$$
$$\gamma_{\text{root}}\left(A_1(2,1), A_2(2,1), A_3(2,1)\right) = 0.3124$$
$$\gamma_{\text{root}}\left(A_1(2,2), A_2(2,2), A_3(2,2)\right) = 0.2650$$
$$\gamma_{\text{root}}\left(A_1(2,3), A_2(2,3), A_3(2,3)\right) = 0.1877$$

For the case of B from Figure 4.2, we can calculate $\gamma_{\text{root}}(B)$ as

$$\gamma_{\text{root}}\left(B_1(0,0), B_2(0,0), B_3(0,0), B_4(0,0)\right) = 0.1714$$
$$\gamma_{\text{root}}\left(B_1(0,1), B_2(0,1), B_3(0,1), B_4(0,1)\right) = 0.1714$$
$$\gamma_{\text{root}}\left(B_1(0,2), B_2(0,2), B_3(0,2), B_4(0,2)\right) = 0.3539$$

$$\gamma_{\text{root}}\left(B_1(1,0), B_2(1,0), B_3(1,0), B_4(1,0)\right) = 0.3043$$
$$\gamma_{\text{root}}\left(B_1(1,1), B_2(1,1), B_3(1,1), B_4(1,1)\right) = 0.5384$$
$$\gamma_{\text{root}}\left(B_1(1,2), B_2(1,2), B_3(1,2), B_4(1,2)\right) = 0.3691$$

$$\gamma_{\text{root}}\left(B_1(2,0), B_2(2,0), B_3(2,0), B_4(2,0)\right) = 0.0000$$
$$\gamma_{\text{root}}\left(B_1(2,1), B_2(2,1), B_3(2,1), B_4(2,1)\right) = 0.5563$$
$$\gamma_{\text{root}}\left(B_1(2,2), B_2(2,2), B_3(2,2), B_4(2,2)\right) = 0.5943$$

Notice that the definition of an idempotent function is a very general one. In fact, we have directly the following construction result, that allows us to obtain a kind of inductive process.

Proposition 4.3.2. Let $f : [0,1]^{n-1} \to [0,1]$ be an $(n-1)$-dimensional idempotent function. Then the mapping:

$$\gamma(x_1, \ldots, x_n) = \sqrt{x_n \cdot f(x_1, \ldots, x_{n-1})},$$

is an n-dimensional idempotent function.

Proof. Straightforward. □

Regarding the structure of the space of n-dimensional idempotent functions, we also have the following.

Proposition 4.3.3. Let $\gamma_1, \gamma_2 : [0,1]^n \to [0,1]$ be two n-dimensional idempotent functions. Then:

1. $\frac{1}{2}(\gamma_1 + \gamma_2)$ is also an n-dimensional idempotent function;
2. $\sqrt{\gamma_1\gamma_2}$ is also an n-dimensional idempotent function.

Proof. Straightforward. □

It is known that a $[0,1]$-automorphism φ is a continuous and strictly increasing function, such that $\varphi(0) = 0$ and $\varphi(1) = 1$. We can use this concept to build n-dimensional idempotent functions.

Theorem 4.3.1. [80] Let $\gamma_1, \gamma_2 : [0,1]^n \to [0,1]$ be two n-dimensional idempotent functions. Let $F : [0,1]^2 \to [0,1]$ be a mapping such that $d(x) = F(x,x)$ is an automorphism of $[0,1]$. Then the mapping:

$$\gamma = d^{-1}(F(\gamma_1, \gamma_2)),$$

is also an n-dimensional idempotent function.

Proof.

$$\gamma(x,\ldots,x) = d^{-1}(F(\gamma_1(x,\ldots,x), \gamma_2(x,\ldots,x))) = d^{-1}(F(x,x)) = x. \square$$

4.3.2 Some interesting properties

Homogeneity, *shift-invariance* and *migrativity* are properties quite used in image processing. In fact, it is desirable that these properties are satisfied.

Definition 4.3.2. A mapping $f : [0,1]^n \to [0,1]$ is called homogeneous of order $k \geq 0$ if for every $x_1, \ldots, x_n, \alpha \in [0,1]$ the identity

$$f(\alpha x_1, \ldots, \alpha x_n) = \alpha^k f(x_1, \ldots, x_n)$$

holds.

Example 4.3.5. The mapping $f : [0,1]^2 \to [0,1]$, defined by $f(x_1, x_2) = \max(x_1, x_2)$ is homogeneous of order 1.

$$f(\alpha x_1, \alpha x_2) = \max(\alpha x_1, \alpha x_2) = \alpha \cdot \max(x_1, x_2).$$

Proposition 4.3.4. Every homogeneous idempotent operator is homogeneous of order 1.

Proof. Straightforward. $\square$

Corollary 4.3.1. There are no idempotent operators homogeneous of order $k \neq 1$.

Definition 4.3.3. A mapping $f : [0,1]^n \to [0,1]$ is called shift-invariant if for every $x_1, \ldots, x_n \in [0,1]$ and for every $\lambda \in [-1,1]$ such that $x_1 + \lambda, \ldots, x_n + \lambda \in [0,1]$ the identity

$$f(x_1 + \lambda, \ldots, x_n + \lambda) = f(x_1, \ldots, x_n) + \lambda$$

holds.

Proposition 4.3.5. An idempotent operator γ is shift-invariant over the diagonal; that is, $\gamma(x + \lambda, \ldots, x + \lambda) = \gamma(x, \ldots, x) + \lambda$, for every $\lambda \in [0,1]$.

Proof. Straightforward. □

Definition 4.3.4. A mapping $f : [0,1]^n \to [0,1]$ is called migrative if for every $x_1, \ldots, x_n, \alpha \in [0,1]$ the identity

$$f(x_1, \ldots, \alpha x_i, \ldots, x_j, \ldots, x_n) = f(x_1, \ldots, x_i, \ldots, \alpha x_j, \ldots, x_n)$$

holds for every $i, j \in \{1, \ldots, n\}$.

Example 4.3.6. The mapping $f : [0,1]^2 \to [0,1]$, defined by $f(x_1, x_2) = x_1 \cdot x_2$, is migrative.

$$f(\alpha x_1, x_2) = \alpha x_1 \cdot x_2 = x_1 \cdot \alpha x_2 = f(x_1, \alpha x_2).$$

An idempotent operator is not necessarily migrative, homogeneous and shift-invariant. We have the following result.

Proposition 4.3.6. Let γ be an idempotent migrative operator. Then $\gamma(x, 0, \ldots, 0) = 0$ for every $x \in [0,1]$.

Proof. Just observe that $\gamma(x, 0, \ldots, 0) = \gamma(x, 0 \cdot 0, \ldots, 0) = \gamma(0, 0, \ldots, 0) = 0$ from the migrativity with $\alpha = 0$ and the homogeneity. □

Proposition 4.3.7. Let γ be an idempotent migrative operator. Then $\gamma(x^n, 1, \ldots, 1) = x$ for every $x \in [0,1]$.

Proof.

$$\gamma(x^n, 1, \ldots, 1) = \gamma(x \cdot x^{n-1}, \underbrace{1, \ldots, 1}_{n-1}) =$$
$$\gamma(x^{n-1}, \underbrace{x \cdot 1}_{x}, 1, \ldots, 1) = \gamma(x \cdot x^{n-2}, x, \underbrace{1, \ldots, 1}_{n-2}) =$$
$$\gamma(x^{n-2}, x, \underbrace{x \cdot 1}_{x}, 1, \ldots, 1) = \gamma(x^{n-2}, x, x, \underbrace{1 \ldots, 1}_{n-3}) =$$
$$\gamma(x, x, \ldots, x) = x. \square$$

4.4 Idempotent aggregation functions: averaging functions

We have studied the use of idempotent functions in order to transform a multi-fuzzy set into a fuzzy set in Section 4.3. Now we study *monotonic non-decreasing idempotent functions*, that are a special case of aggregation functions called *averaging functions*. With these functions we have not only idempotence, but also the value of the function will be bounded by the minimum and maximum of the input arguments.

Definition 4.4.1. An aggregation function of dimension n (n-ary aggregation function) is a non-decreasing mapping $f : [0, 1]^n \to [0, 1]$ such that
$f(0, \ldots, 0) = 0$ and $f(1, \ldots, 1) = 1$.

Remark 4.4.1. Consider any averaging aggregation function M. Then, since an averaging aggregation function is idempotent, it follows that M is an idempotent function.

Remark 4.4.2. The mode, the smallest and largest idempotent functions are not an aggregation function because they are not monotone. However, they are idempotent.

Remark 4.4.3. An idempotent aggregation function, is known as an idempotent averaging aggregation function.

Definition 4.4.2. An aggregation function $f : [0, 1]^n \to [0, 1]$ is called averaging

or a mean aggregation function if

$$\min(x_1, \ldots, x_n) \leq f(x_1, \ldots, x_n) \leq \max(x_1, \ldots, x_n).$$

Proposition 4.4.1. [41] Idempotent monotonic non-decreasing functions and idempotent averaging functions are the same.

Example 4.4.1. Some examples of the averaging aggregation functions are:

1. The arithmetic mean (introduced in Eq. (4.2)).
2. The median operator.

$$\gamma_{\text{med}}(x_1, \ldots, x_n) = \begin{cases} \frac{1}{2}(x_k + x_{k+1}), & \text{if } n = 2k, \\ x_k, & \text{if } n = 2k - 1. \end{cases} \tag{4.4}$$

3. The min operator.

$$\gamma_{\min}(x_1, \ldots, x_n) = \min(x_1, ..., x_n). \tag{4.5}$$

4. The max operator.

$$\gamma_{\max}(x_1, \ldots, x_n) = \max(x_1, ..., x_n). \tag{4.6}$$

Example 4.4.2. Using the matrices B from Figure 4.2, and the aggregation function $\gamma_{\text{mean}}(\mathbf{x})$ from Eq. (4.2), we can calculate $\gamma_{\text{mean}}(B)$ as

$$\begin{aligned} \gamma_{\text{mean}}\left(B_1(0,0), B_2(0,0), B_3(0,0), B_4(0,0)\right) &= 0.2 \\ \gamma_{\text{mean}}\left(B_1(0,1), B_2(0,1), B_3(0,1), B_4(0,1)\right) &= 0.2 \\ \gamma_{\text{mean}}\left(B_1(0,2), B_2(0,2), B_3(0,2), B_4(0,2)\right) &= 0.4 \end{aligned}$$

$$\begin{aligned} \gamma_{\text{mean}}\left(B_1(1,0), B_2(1,0), B_3(1,0), B_4(1,0)\right) &= 0.525 \\ \gamma_{\text{mean}}\left(B_1(1,1), B_2(1,1), B_3(1,1), B_4(1,1)\right) &= 0.750 \\ \gamma_{\text{mean}}\left(B_1(1,2), B_2(1,2), B_3(1,2), B_4(1,2)\right) &= 0.525 \end{aligned}$$

$$\gamma_{\text{mean}}\left(B_1(2,0), B_2(2,0), B_3(2,0), B_4(2,0)\right) = 0.40$$
$$\gamma_{\text{mean}}\left(B_1(2,1), B_2(2,1), B_3(2,1), B_4(2,1)\right) = 0.80$$
$$\gamma_{\text{mean}}\left(B_1(2,2), B_2(2,2), B_3(2,2), B_4(2,2)\right) = 0.75$$

For the case of $\gamma_{\text{mean}}(A)$ from Figure 4.1, results are shown in Figure 4.5(a).

Based on the same matrices A, and the aggregation function $\gamma_{\text{med}}(\mathbf{x})$ from Eq. (4.4). The results for *aggregation function* $\gamma_{\text{med}}(A)$ are

$$\gamma_{\text{med}}\left(A_1(0,0), A_2(0,0), A_3(0,0)\right) = 0.9$$
$$\gamma_{\text{med}}\left(A_1(0,1), A_2(0,1), A_3(0,1)\right) = 0.8$$
$$\gamma_{\text{med}}\left(A_1(0,2), A_2(0,2), A_3(0,2)\right) = 0.8$$
$$\gamma_{\text{med}}\left(A_1(0,3), A_2(0,3), A_3(0,3)\right) = 0.8$$

$$\gamma_{\text{med}}\left(A_1(1,0), A_2(1,0), A_3(1,0)\right) = 0.8$$
$$\gamma_{\text{med}}\left(A_1(1,1), A_2(1,1), A_3(1,1)\right) = 0.9$$
$$\gamma_{\text{med}}\left(A_1(1,2), A_2(1,2), A_3(1,2)\right) = 0.9$$
$$\gamma_{\text{med}}\left(A_1(1,3), A_2(1,3), A_3(1,3)\right) = 0.8$$

$$\gamma_{\text{med}}\left(A_1(2,0), A_2(2,0), A_3(2,0)\right) = 0.5$$
$$\gamma_{\text{med}}\left(A_1(2,1), A_2(2,1), A_3(2,1)\right) = 0.4$$
$$\gamma_{\text{med}}\left(A_1(2,2), A_2(2,2), A_3(2,2)\right) = 0.3$$
$$\gamma_{\text{med}}\left(A_1(2,3), A_2(2,3), A_3(2,3)\right) = 0.3$$

In the case of matrices B (Figure 4.2). The results for the aggregation function $\gamma_{\text{med}}(B)$ are

0.9	0.83	0.8	0.8
0.6	0.8	0.83	0.8
0.46	0.4	0.3	0.23

(a) $\gamma_{\text{mean}}(A)$

0.8	0.7	0.7	0.8
0.1	0.6	0.7	0.7
0.3	0.2	0.2	0.1

(b) $\gamma_{\text{min}}(A)$

1.0	1.0	0.9	0.8
0.9	0.9	0.9	0.9
0.6	0.6	0.4	0.3

(c) $\gamma_{\text{max}}(A)$

Figure 4.5: Mapping matrices of the aggregation functions.

$$\gamma_{\text{med}}\left(B_1(0,0), B_2(0,0), B_3(0,0), B_4(0,0)\right) = 0.20$$
$$\gamma_{\text{med}}\left(B_1(0,1), B_2(0,1), B_3(0,1), B_4(0,1)\right) = 0.20$$
$$\gamma_{\text{med}}\left(B_1(0,2), B_2(0,2), B_3(0,2), B_4(0,2)\right) = 0.35$$

$$\gamma_{\text{med}}\left(B_1(1,0), B_2(1,0), B_3(1,0), B_4(1,0)\right) = 0.50$$
$$\gamma_{\text{med}}\left(B_1(1,1), B_2(1,1), B_3(1,1), B_4(1,1)\right) = 0.80$$
$$\gamma_{\text{med}}\left(B_1(1,2), B_2(1,2), B_3(1,2), B_4(1,2)\right) = 0.55$$

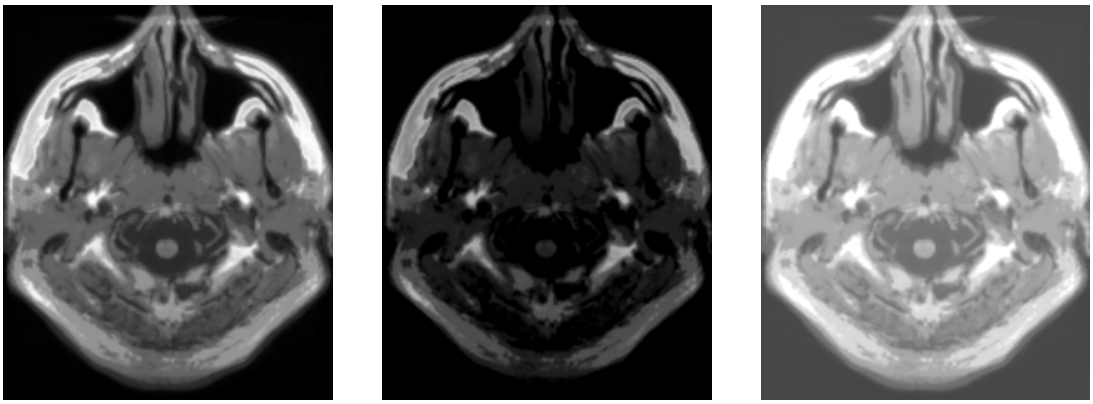
$$\gamma_{\text{med}}\left(B_1(2,0), B_2(2,0), B_3(2,0), B_4(2,0)\right) = 0.45$$
$$\gamma_{\text{med}}\left(B_1(2,1), B_2(2,1), B_3(2,1), B_4(2,1)\right) = 0.90$$
$$\gamma_{\text{med}}\left(B_1(2,2), B_2(2,2), B_3(2,2), B_4(2,2)\right) = 0.80$$

Using the aggregation function from Eq. (4.5). We obtain $\gamma_{\text{min}}(A)$ as shown in Figure 4.5(b).

The results for $\gamma_{\text{max}}(A)$ using the Eq. (4.6) are shown in Figure 4.5(c).

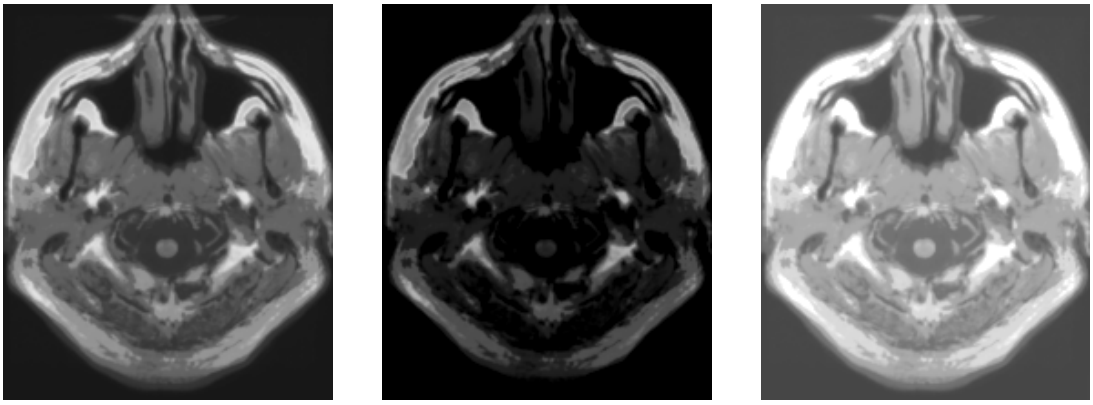
Example 4.4.3. Aggregation functions can be also applied on images. Figure 4.7 shows the result for the aggregation function $\gamma_{\text{mean}}(\mathbf{x})$, $\gamma_{\text{min}}(\mathbf{x})$ and $\gamma_{\text{max}}(\mathbf{x})$ applied to the images from Figure 4.6. These images form a multifuzzy set. From an image which is taken as a referential image (Figure 4.6(a)), with normal brightness, and two other images with altered brightness: one sums up 70 points of intensity (Figure 4.6(c)); and the other subtracting 70 points of intensity (Figure 4.6(b)). Bounded by the image range $[0, 255]$. Interpreting the

results, $\gamma_{\text{min}}(\mathbf{x})$ (Figure 4.7(b)) matches with the darker image, as the aggregation takes the minimum intensity values from the multifuzzy set. In the case of $\gamma_{\text{max}}(\mathbf{x})$ (Figure 4.7(c)) the brighter image is taken, because it takes the maximum intensity values from the multifuzzy set. For $\gamma_{\text{mean}}(\mathbf{x})$ (Figure 4.7(a)) it is easy to deduce that for any pixel the result will be the mean, due to $A(x_i) = \frac{x_i+(x_i+70)+(x_i-70)}{3} = x_i$.



(a) Original image (b) Original image -70 (c) Original image +70

Figure 4.6: Images taking part of the fuzzy sets. Used as example images for the aggregation functions.



(a) γ_{mean} (b) γ_{min} (c) γ_{max}

Figure 4.7: Aggregation results for the images of Figure 4.6.

4.4.1 Specific case: OWA operators and fuzzy quantifiers

The ordered weighted averaging operators [111, 112], commonly called OWA operators, are a parameterized family of idempotent averaging aggregation functions. These are different from the classical weighted average in that coefficients are not associated directly with a particular attribute but rather to an ordered position. Moreover, OWA operators fill the gap between the operators min and max, where the min, max, arithmetic mean or median are particular cases of this family. Furthermore, they can capture aggregations which emulate things like 'most', 'many', etc. Thus we see that these OWA operators provide an interesting class of operators.

Definition 4.4.3. [111] A mapping $F : [0,1]^n \to [0,1]$ is called an OWA operator of dimension n if there exists a weighting vector W, $W = (w_1, \ldots, w_n) \in [0,1]^n$ with $\sum_{i=1}^{n} w_i = 1$ and such that $F(a_1, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j$ with b_j the j-th largest of the a_i.

A fundamental aspect of this operation is the *re-ordering* step, in particular an aggregate a_i is not associated with a particular weight w_i but rather a weight is associated with a particular ordered position of aggregate. The use of an "ordered" weighted average allows to satisfy the condition of symmetry (generalized commutativity).

Example 4.4.4. Assume F is an OWA operator of dimension $n = 4$ with weighting vector $W = (0.2, 0.1, 0.6, 0.1)^T$ and $A = [0, 1, 0.1, 0.2]$. Then to calculate $F(A)$ we have that the corresponding ordered argument vector $B = [1, 0.2, 0.1, 0]$. Therefore, $F(A) = F(B) = (0.2)(1) + (0.1)(0.2) + (0.6)(0.1) + (0.1)(0) = 0.28$.

A natural question in the definition of the OWA operators is how to obtain the associated weighting vector. It is noted that different OWA operators are distinguished by their weighting function. Furthermore, in [111] R.R. Yager showed three important special cases of OWA operators:

- F^* : In this case $W = W^* = (1, 0, \ldots, 0)^T$.
- F_* : In this case $W = W_* = (0, 0, \ldots, 1)^T$.

- F_{ave} : In this case $W = W_{\text{ave}} = (1/n,\, 1/n,\, \ldots,\, 1/n)^T$.

It can easily seen that

- $F^*(a_1, \ldots, a_n) = \max_i(a_i)$.
- $F_*(a_1, \ldots, a_n) = \min_i(a_i)$.
- $F_{\text{ave}}(a_1, \ldots, a_n) = \frac{1}{n} \sum_i a_i$.

There exist at least two ways that can be used to obtain the value of the w_i's. The first approach consists on using some kind of learning mechanism in such a way that the weights are learnt from observations. In this approach we use some sample data, arguments and associated aggregated values and try to fit the weights to this collection of sample data. The process usually involves the use of some kind of regression model. On the other side, a second approach tries to provide some semantics or meaning to the w_i's. Then based upon these semantics we can get directly the values of the w_i's. This approach also will provide some further insight into the meaning of the OWA operators [111, 112].

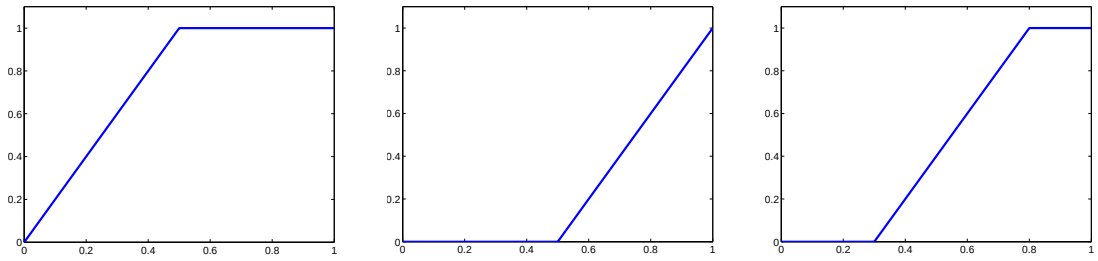
Our idea is to be able to calculate the weights for the aggregation operators using linguistic quantifiers, e.g., *about 5*, *almost all*, *a few*, *many*, *most*, *as many as possible*, *nearly half*, *least half*. The concept of a fuzzy quantifier was introduced by L.A. Zadeh [116], offering a more flexible tool for knowledge representation.

Zadeh also suggested that the semantics of a fuzzy quantifier can be captured by using fuzzy sets for its representation. Two types of fuzzy quantifiers are distinguished, *absolute* and *proportional or relative*. Absolute quantifiers are those used to represent amounts that are absolute in nature, such as *about 2* or *more than 5*. While proportional quantifiers, such as *most*, *at least half*, can be represented by fuzzy sets of the unit interval. For any $r \in [0, 1]$, $Q(r)$ indicates the degree to which the proportion r is compatible with the meaning of the quantifier it represents. In other words, an absolute quantifier can be represented by a fuzzy subset $Q : \mathbb{R}^+ \to [0, 1]$ that satisfies $Q(0) = 0$, and $\exists k \in \mathbb{R}^+$ such that $Q(k) = 1$. While a proportional quantifier $Q : [0, 1] \to [0, 1]$, satisfies $Q(0) = 0$, and $\exists r \in [0, 1]$ such that $Q(r) = 1$.

Proportional fuzzy quantifiers are usually of one of three types: *increasing*, *decreasing* and *unimodal*. For instance, some increasing quantifiers are 'at least half', 'as many as possible' and 'most of them'. Decreasing quantifiers are 'few of them' and 'as much k'. R.R. Yager [111] suggested an interesting way to compute

the weights of the OWA aggregation operator using fuzzy quantifiers, which, in the case of an increasing quantifier Q, is given by the expression:

$$Q(r) = \begin{cases} 0, & \text{if } r < a, \\ \frac{r-a}{b-a}, & \text{if } a \leq r \leq b, \\ 1, & \text{if } r > b. \end{cases} \qquad w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right). \tag{4.7}$$



(a) at least half (b) as many as possible (c) most of them

Figure 4.8: Proportional increasing quantifiers used to construct OWA operators.

Example 4.4.5. In Figure 4.8 are shown examples of the proportional increasing quantifiers, 'at least half', 'as many as possible' and 'most of them', where the parameters (a, b) are $(0, 0.5)$, $(0.5, 1)$ and $(0.3, 0.8)$, respectively.

Then, we assume an OWA operator of dimension $n = 4$, where the weights are computed using the increasing quantifier Q from Eq. (4.7) for the different proportional quantifiers introduced in Figure 4.8:

- at least the half
  $w_1 = Q\left(\frac{1}{4}\right) - Q(0) = \frac{0.25}{0.5} - 0 = 0.5,$
  $w_2 = Q\left(\frac{1}{2}\right) - Q\left(\frac{1}{4}\right) = \frac{0.5}{0.5} - 0.5 = 0.5,$
  $w_3 = Q\left(\frac{3}{4}\right) - Q\left(\frac{1}{2}\right) = 1 - 1 = 0,$
  $w_4 = Q(1) - Q\left(\frac{3}{4}\right) = 0,$
  Then, $W_{\text{least}} = (0.5,\ 0.5,\ 0,\ 0)^T$.

- as many as possible
  $w_1 = Q\left(\frac{1}{4}\right) - Q(0) = 0 - 0 = 0,$
  $w_2 = Q\left(\frac{1}{2}\right) - Q\left(\frac{1}{4}\right) = 0,$
  $w_3 = Q\left(\frac{3}{4}\right) - Q\left(\frac{1}{2}\right) = \frac{0.75-0.5}{0.5} - 0 = 0.5,$

$w_4 = Q(1) - Q\left(\frac{3}{4}\right) = 0.5,$
Then, $W_{\text{many}} = (0,\ 0,\ 0.5,\ 0.5)^T$.

- most of them
  $w_1 = Q\left(\frac{1}{4}\right) - Q(0) = 0 - 0 = 0,$
  $w_2 = Q\left(\frac{1}{2}\right) - Q\left(\frac{1}{4}\right) = \frac{0.5-0.3}{0.5} - 0 = 0.4,$
  $w_3 = Q\left(\frac{3}{4}\right) - Q\left(\frac{1}{2}\right) = \frac{0.75-0.3}{0.5} - 0.4 = 0.9 - 0.4 = 0.5,$
  $w_4 = Q(1) - Q\left(\frac{3}{4}\right) = 1 - 0.9 = 0.1,$
  Then, $W_{\text{most}} = (0,\ 0.4,\ 0.5,\ 0.1)^T$.

The OWA operators can also be studied by their properties. Thus Yager defined two important measures associated with an OWA operator: Disp(F) (Definition 4.4.4) and orness(F) (Definition 4.4.5). The dispersion (or entropy) measures the degree to which we use all the aggregates equally. While orness measures its behaviour or optimism. An OWA operator F with much of the weights near the top will be an 'orlike' operator, orness(F) ≥ 0.5. At the other extreme, when the weights are non-zero near the bottom the OWA operator F will be 'andlike', orness(F) $\leq$ 0.5.

Definition 4.4.4. [111] Let F be an OWA operator and W its weighting vector. The dispersion measure of F is defined as

$$\text{Disp}(F) = \sum_{i=1}^{n} w_i \log(w_i).$$

Definition 4.4.5. [111] Let F be an OWA operator and W its weighting vector. The orness measure of F is defined as

$$\text{orness}(F) = \frac{1}{(n-1)} \sum_{i=1}^{n} (n-i) w_i.$$

Example 4.4.6. We use the OWA operators represented in Figure 4.8 with dimension $n = 4$. The corresponding weights are $W_{\text{least}} = (0.5,\ 0.5,\ 0,\ 0)^T$, $W_{\text{many}} = (0,\ 0,\ 0.5,\ 0.5)^T$ and $W_{\text{most}} = (0,\ 0.4,\ 0.5,\ 0.1)^T$. It can be seen that the OWA 'at least half' is an 'orlike' operator with an orness(F_{least}) = 0.833 and Disp(F_{least}) = 0.6931. OWA 'most of them', presents an 'andlike' behaviour with an orness(F_{most}) = 0.1667 and Disp(F_{most}) = 0.693. It shares the same dispersion as OWA 'at least half' because they aggregate with the same degree, although as we can observe the weights are distributed differently. In our last case, for OWA

'as many as possible', the orness(F_{many}) = 0.433, treated as 'andlike' operator. Its dispersion is Disp(F_{many}) = 0.9433, considering almost all the weights.

4.5 Penalty functions

For the exploitation phase, the concept of a penalty function P, as shown schematically in Figure 4.9, allows us to measure the disagreement or dissimilarity between n candidates, $\{x_1, \ldots, x_n\}$, and an output y.

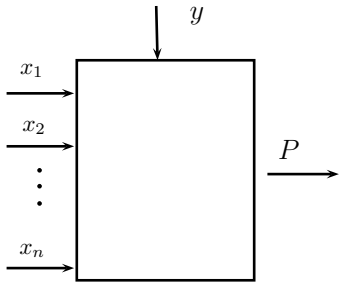


Figure 4.9: Schematic representation of a penalty function P.

Then, we rely on P to define the *penalty-based function* as is introduced in Definition 4.5.1 for a set of q functions in order to measure the disagreement with respect all the inputs, $\{x_1, \ldots, x_n\}$, that in our case is a multifuzzy set. In other words, we choose the function f using a consensus procedure based on testing several functions until we find the one providing the least dissimilar result with respect to the values of the inputs. The motivation to use this concept is because we do not know beforehand which function is better to use, so our aim is to find the one that minimizes the overall error with respect to the inputs. Obviously, the result provided by the penalty function depends on the set of q aggregation functions that we use in each case.

Definition 4.5.1. [24–26] A penalty function is a mapping $P : [0,1]^{n+1} \to [0, \infty)$ such that:

1. $P(x_1, \ldots, x_n; y) \geq 0$ for all $x_1, \ldots, x_n \in [0,1], y \in [0,1]$;
2. $P(x_1, \ldots, x_n; y) = 0$ if $x_i = y$ for all $i = \{1, \ldots, n\}$;
3. $P(x_1, \ldots, x_n; y)$ is quasi-convex in y for any $\mathbf{x}$; that is $P(\mathbf{x}, \lambda \cdot y_1 + (1-\lambda) \cdot y_2) \leq \max(P(\mathbf{x}, y_1), P(\mathbf{x}, y_2))$ for any λ, y_1, $y_2 \in [0,1]$.

Let P be a penalty function. We call *penalty-based function* the mapping:

$$f(\mathbf{x}) = \arg\min_{y} P(\mathbf{x}, y), \text{ where } y \in \{y_1, ..., y_q\}. \tag{4.8}$$

If there exists only one point y in which $P(\mathbf{x}, \cdot)$ has a minimum, then $f(\mathbf{x}) = y$. Whereas if $P(\mathbf{x}, \cdot)$ has more than one minimum point, then from the quasi-convexity it follows that $P(\mathbf{x}, \cdot)$ attains its minimum on the whole interval of $[a, b]$ and we define $f(x) = \frac{a+b}{2}$. The quasi-convexity enforces that one and only one of these two possibilities happens. So, the minimum always exists, and either it is at a single point or at a whole interval.

In other words, if some input $x_i \neq y$, then we impose a 'penalty' for this disagreement. The larger the disagreement, the larger (in general) is the imposed penalty (Definition 4.5.1.(1)). But, if all the inputs are the same $x_1 = \ldots = x_n$, then the output is y and the penalty is zero, where we have unanimous vote and no penalization (Definition 4.5.1.(2)). Moreover, quasi-convexity ensures that the set of minima is non-empty and in fact it is either a single point or an interval (Definition 4.5.1.(3)).

Example 4.5.1. For instance a penalty function is as follows

$$P(x_1, \ldots, x_n; y) = \left(\sum_{p=1}^{n} |x_p - y| \right)^2. \tag{4.9}$$

We prove that the function $P(x_1, \ldots, x_n; y)$ from Eq. (4.9) fulfills the three conditions previously exposed on Definition 4.5.1.

1) Straightforward. Because $x^2 \geq 0$.

2) Straightforward. Taking $P(y, \ldots, y; y) = 0$.

3) To prove the *quasi-convexity* we apply two properties for convex functions [19].

   (a) The sum of two convex functions is a convex function. Then we know that a function f is convex if, for any two points y_1, y_2 in its domain and for any $\lambda \in [0, 1]$, the following is true $f(\lambda y_1 + (1 - \lambda)y_2) \leq \lambda f(y_1) + (1 - \lambda) f(y_2)$. Therefore if f and h are both convex then we know $(f+h)(\lambda y_1 + (1-\lambda)y_2) = f(\lambda y_1 + (1-\lambda)y_2) + h(\lambda y_1 + (1-\lambda)y_2) \leq$

$\lambda f(y_1) + (1-\lambda)f(y_2) + \lambda h(y_1) + (1-\lambda)h(y_2) = \lambda(f+h)(y_1) + (1-\lambda)(f+h)(y_2)$. So, $f+h$ is also convex.

(b) If f is convex and non-decreasing and h is a convex, then the composition $h \circ f$ is also convex. Then, by convexity of h:

$$h(\lambda x + (1-\lambda)y) \leq \lambda h(x) + (1-\lambda)h(y).$$

So, using the fact that f is non-decreasing:

$$f(h(\lambda x + (1-\lambda)y)) \leq f(\lambda h(x) + (1-\lambda)h(y)).$$

Therefore, again by convexity:

$$f(h(\lambda x + (1-\lambda)y)) \leq \lambda f(h(x)) + (1-\lambda)f(h(y)).$$

This reasoning can be used inductively in order to prove the result that $f_n \circ f_{n-1} \cdots \circ f_1 \circ h$ is convex under the stated hypothesis. And the composition will be non-decreasing if h is non-decreasing.

The quasi-convexity is simple to prove using *(a)* and *(b)*. We take a fixed z, $f_z(y) = |z-y|$. Then using *(a)* for each fixed $\mathbf{x} = (x_1, \ldots, x_n)$ it follows that $g(x_1, \ldots, x_n, y) := f_{x_1}(y) + \cdots + f_{x_p}(y)$ is convex. Now, we apply *(b)* to the function $h(x) : \mathbb{R} \to \mathbb{R}$ given by $h(x) = x^2$ and to g, from what follows that $h \circ g$ is convex, but $h \circ g = P(x_1, \ldots, x_n; y)$, i.e., P is convex and hence also quasi-convex. □

Theorem 4.5.1. [25] Any averaging aggregation function can be represented as a penalty-based function in the sense of Definition 4.5.1.

Example 4.5.2. There are several well-known aggregation functions that can be presented as a penalty aggregation function. In [25] are presented some of the followings. For it, we take the weighted vector $\mathbf{w}$ where $\sum_i w_i = 1$.

1. The *weighted arithmetic mean*:

$$P(\mathbf{x}, y) = \sum_{i=1}^{n} w_i (x_i - y)^2. \tag{4.10}$$

2. The *weighted median*:

$$P(\mathbf{x}, y) = \sum_{i=1}^{n} w_i |x_i - y|. \tag{4.11}$$

3. The *weighted quasi-arithmetic mean* with the generator h:

$$P(\mathbf{x}, y) = \sum_{i=1}^{n} w_i \left(h(x_i) - h(y)\right)^2. \tag{4.12}$$

4. The *generalized OWA*:

$$P(\mathbf{x}, y) = \sum_{i=1}^{n} w_i (x_{(i)} - y), \tag{4.13}$$

where $x_{(i)}$ is the i-th largest component of $\mathbf{x}$.

These and other well-known penalty-based aggregation functions are discussed in further details in [25].

Example 4.5.3. Suppose we want to get the best aggregation function among 3 possible ones using a *penalty-based function* for the input $\mathbf{x} = \{0.7, 0.2, 0.5\}$. The selected aggregation functions are $\gamma_1(\mathbf{x}) = \frac{1}{n}\sum_{k=1}^{n} x_k$, $\gamma_2(\mathbf{x}) = \min(\mathbf{x})$ and $\gamma_3(\mathbf{x}) = \max(\mathbf{x})$. Then, the selected penalty function $P(\mathbf{x}; y)$ is the function introduced in Eq. (4.9).

First of all, we calculate the outputs for the different aggregation functions using the input set $\mathbf{x}$:

$$\begin{aligned}
\gamma_1(\mathbf{x}) &= \tfrac{0.7+0.2+0.5}{3} = 0.4667,\\
\gamma_2(\mathbf{x}) &= \min(0.7, 0.2, 0.5) = 0.2,\\
\gamma_3(\mathbf{x}) &= \max(0.7, 0.2, 0.5) = 0.7.
\end{aligned}$$

Then, we can compute the different penalties for the different aggregation functions as

$$\begin{aligned}
P(\mathbf{x}; \gamma_1(\mathbf{x})) &= (0.7 - 0.4667)^2 + (0.2 - 0.4667)^2 + (0.5 - 0.4667)^2 = 0.1267,\\
P(\mathbf{x}; \gamma_2(\mathbf{x})) &= (0.7 - 0.2)^2 + (0.2 - 0.2)^2 + (0.5 - 0.2)^2 = 0.34,\\
P(\mathbf{x}; \gamma_3(\mathbf{x})) &= (0.7 - 0.5)^2 + (0.2 - 0.5)^2 + (0.5 - 0.5)^2 = 0.13.
\end{aligned}$$

Finally, $f(\mathbf{x})$ can be obtained using Eq. (4.8) and $\gamma = \{\gamma_1(\mathbf{x}), \gamma_2(\mathbf{x}), \gamma_3(\mathbf{x})\}$ as

$$\begin{aligned} f(\mathbf{x}) &= \arg\min_{\gamma}(P(\mathbf{x}, \gamma)) = \arg\min\left(P(\mathbf{x}, \gamma_1(\mathbf{x})), P(\mathbf{x}, \gamma_2(\mathbf{x})), P(\mathbf{x}, \gamma_3(\mathbf{x}))\right) \\ &= 0.4667. \end{aligned}$$

The solution aggregation function has been γ_1 because it presents the minimum penalty.

We have shown that penalty functions can be applied with an input set $\mathbf{x}$ and an output y. However, $\mathbf{y}$ can be also a vector of elements instead of an element, namely $y = \{y_1, \ldots, y_q\}$. Therefore, the penalty-based function receives a set such as $\gamma = \{\gamma_1(\mathbf{x}), \ldots, \gamma_q(\mathbf{x})\}$, given by $\gamma_j(\mathbf{x}) \in \{y_{j1}, \ldots, y_{jq}\}$.

Example 4.5.4. We take the best aggregation function among 3 possible ones using a *penalty-based function* for the input $A = (A_1, A_2, A_3)$ shown in Figure 4.1. The selected aggregation functions are $\gamma_1(\mathbf{x}) = \frac{1}{n}\sum_{k=1}^{n} x_k$, $\gamma_2(\mathbf{x}) = \min(\mathbf{x})$ and $\gamma_3(\mathbf{x}) = \max(\mathbf{x})$. The results for the different aggregation functions are shown in Figure 4.5 for the input multifuzzy set A. Then, the penalty function $P(\mathbf{x}; y)$ is the function introduced in Eq. (4.9).

We calculate the different penalties for the different aggregation functions as

$$
\begin{aligned}
P(\mathbf{x};\gamma_1(\mathbf{x})) = &(|0.9-0.9|+|0.8-0.9|+|1.0-0.9|)^2 \\
&+(|0.8-0.83|+|0.7-0.83|+|1.0-0.83|)^2 \\
&+(|0.7-0.8|+|0.8-0.8|+|0.9-0.8|)^2 \\
&+(|0.8-0.8|+|0.8-0.8|+|0.8-0.8|)^2 \\
&\\
&+(|0.9-0.6|+|0.1-0.6|+|0.8-0.6|)^2 \\
&+(|0.6-0.8|+|0.9-0.8|+|0.9-0.8|)^2 \\
&+(|0.7-0.83|+|0.9-0.83|+|0.9-0.83|)^2 \\
&+(|0.8-0.8|+|0.9-0.8|+|0.7-0.8|)^2 \\
&\\
&+(|0.5-0.46|+|0.3-0.46|+|0.6-0.46|)^2 \\
&+(|0.4-0.4|+|0.2-0.4|+|0.6-0.4|)^2 \\
&+(|0.3-0.3|+|0.2-0.3|+|0.4-0.3|)^2 \\
&+(|0.3-0.23|+|0.1-0.23|+|0.3-0.23|)^2 \\
= &\ 1.8503, \\
P(\mathbf{x};\gamma_2(\mathbf{x})) = &\ 4.06, \\
P(\mathbf{x};\gamma_3(\mathbf{x})) = &\ 2.11.
\end{aligned}
$$

Finally, $f(\mathbf{x})$ can be obtained using $\gamma = \{\gamma_1(\mathbf{x}), \gamma_2(\mathbf{x}), \gamma_3(\mathbf{x})\}$ as

$$
f(\mathbf{x}) = \arg\min_{\gamma}(P(\mathbf{x},\gamma)) = \arg\min\left(P(\mathbf{x},\gamma_1(\mathbf{x})), P(\mathbf{x},\gamma_2(\mathbf{x})), P(\mathbf{x},\gamma_3(\mathbf{x}))\right) = \\
= \begin{pmatrix} 0.9 & 0.83 & 0.8 & 0.8 \\ 0.6 & 0.8 & 0.83 & 0.8 \\ 0.46 & 0.4 & 0.3 & 0.23 \end{pmatrix}
$$

The taken solution is γ_1 because it presents the minimum penalty. It corresponds to the arithmetic mean of the input set A.

4.6 Penalty functions over a cartesian product of lattices

Cartesian product of lattices, introduced in Section 3.3, allows us to extend penalty-based functions, introduced in Section 4.5, over a lattice or to smaller chains of this lattice. Therefore, as we use multifuzzy sets, it can also be understood as a $\mathcal{L}$-Fuzzy set, that is a lattice. In this way, we can apply penalty functions with all guarantee over subsets. In other words, when working with images, the use of cartesian product of lattices gives the possibility to use penalty functions over subsets of pixel regions. Moreover, this property introduces flexibility from the computational point of view. As we already mentioned, the input set $\mathbf{x}$ and the output set $\mathbf{y}$ can contain as many elements as we need, increasing the computational cost. However, nowadays computers could present computational time constraints or lack of memory to deal with these vast sets. Thereby the theory behind a cartesian product of lattices allows us to deal with these sets in a smarter way. The goal of this section is to define the penalty functions over a product of lattices and to propose some construction method such that always the convexity property holds. We propose a construction method of penalty functions over cartesian product of lattices.

4.6.1 Building method of penalty functions

Using the definition of a cartesian product of lattices in Theorem 4.6.1, we present a building method of penalty functions over it, such that it assures the accomplishment of the previous exposed properties in Definition 4.5.1, specially the convexity property.

Theorem 4.6.1. [24] Let $\mathcal{F}(U)$ be the set of all fuzzy sets defined on the finite referential and non-empty set $U = \{u_1, \ldots, u_n\}$ $(\#U = n)$. Let denote by B_{y_q} the fuzzy set over U such that all its membership values are equal to $y_q \in [0, 1]$; that is, $B_{y_q}(u_i) = y_q$ for all $u_i \in U$. Let $Y = (y_1, \ldots, y_m) \in [0, 1]^m$ and $B_Y = (B_{y1}, \ldots, B_{ym}) \in \mathcal{F}(U)^m$. Let C^* be a chain, that is a linear lattice, whose elements belong to $[0, 1]$ and let L^*_m be the cartesian product $L^*_m = C^* \times \cdots \times C^*$. Let $K_i : \mathbb{R} \to \mathbb{R}^+$ be convex functions with a unique minimum at $K_i(0) = 0$,

$(i = 1, \ldots, m)$, and take the taxi-distance between fuzzy sets in U defined as

$$D(A, B) = \sum_{i=1}^{n} |A(u_i) - B(u_i)|. \tag{4.14}$$

Then the mapping $P_\nabla : \mathcal{F}(U)^m \times L_m^* \to \mathbb{R}^+$ given by

$$P_\nabla(\mathbf{A}, Y) = \sum_{q=1}^{m} K_q\left(D(A_q, B_{y_q})\right) = \sum_{q=1}^{m} K_q\left(\sum_{p=1}^{n} |A_q(u_p) - y_q|\right), \tag{4.15}$$

satisfies

1. $P_\nabla(\mathbf{A}, Y) = 0$ if and only if $A_q = y_q$ for every $q = 1, \ldots, m$;
2. $P_\nabla(\mathbf{A}, Y)$ is convex in y_q for every $q = 1, \ldots, m$.

Example 4.6.1.

- From the hypothesis in Theorem 4.6.1 we take as convex functions with unique minimum at zero the following: $K_q(x) = x^2$ for all $q \in \{1, \ldots, m\}$, then
$$P_\nabla(\mathbf{A}, Y) = \sum_{q=1}^{m} \left(\sum_{p=1}^{n} |A_q(u_p) - y_q|\right)^2. \tag{4.16}$$
- If $K_q(x) = x$ for all $q \in \{1, \ldots, m\}$, then
$$P_\nabla(\mathbf{A}, Y) = \sum_{q=1}^{m} \sum_{p=1}^{n} |A_q(u_p) - y_q|. \tag{4.17}$$

Theorem 4.6.2. [24] In the setting of Theorem 4.6.1, the mapping

$$F(\mathbf{A}) = \mu(\arg\min_{Y} P_\nabla(\mathbf{A}, Y)) \text{ where } Y \in \{y_1, ..., y_m\}, \tag{4.18}$$

where μ is the rounding to the smallest closest element, is an averaging aggregation function.

Proof. Just observe that

$$\arg\min_{(y_1, \ldots y_m)} P_\nabla(\mathbf{A}, (y_1, \ldots, y_m)) = \arg\min_{(y_1, \ldots y_m)} \sum_{q=1}^{m} K_q\left(\sum_{p=1}^{n} |A_q(u_p) - y_q|\right)$$
$$= \sum_{q=1}^{m} \arg\min_{y} K_q\left(\sum_{p=1}^{n} |A_q(u_p) - y_q|\right),$$

so it is enough to consider each of the quantities

$$\arg\min_{y} K_q\left(\sum_{p=1}^{n} |A_q(u_p) - y_q|\right),$$

but each of these functions is an aggregation function and since K_q is convex, the result follows. □

Example 4.6.2. We want to get the best aggregation function among 3 possible ones using a *penalty-based function* over a cartesian product of lattices for the input $\mathbf{A} = (A_1, A_2, A_3)$ shown in Figure 4.1. The selected aggregation functions are $\gamma_1(\mathbf{x}) = \frac{1}{n}\sum_{k=1}^{n} x_k$, $\gamma_2(\mathbf{x}) = \min(\mathbf{x})$ and $\gamma_3(\mathbf{x}) = \max(\mathbf{x})$. The results for the different aggregation functions are shown in Figure 4.5 for the input multifuzzy set $\mathbf{A}$. Then, the selected penalty function $P_\nabla(\mathbf{A}; Y)$ is the function introduced in Ex. (4.16).

The multifuzzy set $\mathbf{A}$ can be understood as a lattice because each element of the matrix occupies a position in the cartesian product and intrinsically, it is an ordered set. Besides, we build three chains with 4 elements each in such a way that $\mathbf{U} = \{U_1, U_2, U_3\}$, where $U_1 = \{(0,0),(0,1),(0,2),(0,3)\}$, $U_2 = \{(1,0),(1,1),(1,2),(1,3)\}$, and $U_3 = \{(2,0),(2,1),(2,2),(2,3)\}$. Then, we can compute the different penalties for the different aggregation functions with the different chains $\mathbf{A}(\mathbf{U})$:

$$\begin{aligned}
P_\nabla(\mathbf{A}(U_1); \gamma_1(\mathbf{A}(U_1))) &= (|0.9 - 0.9| + |0.8 - 0.9| + |1.0 - 0.9|)^2 \\
&\quad + (|0.8 - 0.83| + |0.7 - 0.83| + |1.0 - 0.83|)^2 \\
&\quad + (|0.7 - 0.8| + |0.8 - 0.8| + |0.9 - 0.8|)^2 \\
&\quad + (|0.8 - 0.8| + |0.8 - 0.8| + |0.8 - 0.8|)^2 \\
&= 0.1889, \\
P_\nabla(\mathbf{A}(U_2); \gamma_1(\mathbf{A}(U_2))) &= 1.2729, \\
P_\nabla(\mathbf{A}(U_3); \gamma_1(\mathbf{A}(U_3))) &= 0.3885.
\end{aligned}$$

$$
\begin{aligned}
P_{\nabla}(\mathbf{A}(U_1);\gamma_2(\mathbf{A}(U_1))) &= (|0.9-0.8|+|0.8-0.8|+|1.0-0.8|)^2 \\
&\quad +(|0.8-0.7|+|0.7-0.7|+|1.0-0.7|)^2 \\
&\quad +(|0.7-0.7|+|0.8-0.7|+|0.9-0.7|)^2 \\
&\quad +(|0.8-0.8|+|0.8-0.8|+|0.8-0.8|)^2 \\
&= 0.34, \\
P_{\nabla}(\mathbf{A}(U_2);\gamma_2(\mathbf{A}(U_2))) &= 2.86, \\
P_{\nabla}(\mathbf{A}(U_3);\gamma_2(\mathbf{A}(U_3))) &= 0.86.
\end{aligned}
$$

$$
\begin{aligned}
P_{\nabla}(\mathbf{A}(U_1);\gamma_3(\mathbf{A}(U_1))) &= (|0.9-1.0|+|0.8-1.0|+|1.0-1.0|)^2 \\
&\quad +(|0.8-1.0|+|0.7-1.0|+|1.0-1.0|)^2 \\
&\quad +(|0.7-0.9|+|0.8-0.9|+|0.9-0.9|)^2 \\
&\quad +(|0.8-0.8|+|0.8-0.8|+|0.8-0.8|)^2 \\
&= 0.43, \\
P_{\nabla}(\mathbf{A}(U_2);\gamma_3(\mathbf{A}(U_2))) &= 1.03, \\
P_{\nabla}(\mathbf{A}(U_3);\gamma_3(\mathbf{A}(U_3))) &= 0.65.
\end{aligned}
$$

Finally, $F(\mathbf{A})$ can be obtained using Ex. (4.18) and $\gamma = \{\gamma_1(\mathbf{A}), \gamma_2(\mathbf{A}), \gamma_3(\mathbf{A})\}$ as

$$
\begin{aligned}
F(\mathbf{A}) &= (\mu(\arg\min_{\gamma} P_{\nabla}(\mathbf{A}(U_1),\gamma(U_1))), \\
&\quad \mu(\arg\min_{\gamma} P_{\nabla}(\mathbf{A}(U_2),\gamma(U_2))), \\
&\quad \mu(\arg\min_{\gamma} P_{\nabla}(\mathbf{A}(U_3),\gamma(U_3)))) \\
&= \begin{pmatrix} 0.9 & 0.83 & 0.8 & 0.8 \\ 0.9 & 0.9 & 0.9 & 0.9 \\ 0.46 & 0.4 & 0.3 & 0.23 \end{pmatrix}
\end{aligned}
$$

$\gamma_1(U_1)$, $\gamma_3(U_2)$ and $\gamma_1(U_3)$ have been selected to get the solution $F(\mathbf{A})$.

We notice that using cartesian product of lattices the results may differ from the solution to the entire lattice, or also between different chain lengths. It is shown in Example 4.6.2, how the second row of the matrix differs with respect to the solution exposed in Example 4.5.4 for the same input data. Thereby

cartesian product of lattices shows a more flexible paradigm and gives the base of consensus. Our idea is to optimize the set to an element level, although it is not the only way. In other words, it allows us to obtain better solutions locally that results in an improvement of the global results. This concept is further detailed in Section 4.7.

4.7 Proposed method: Consensus methodology based on penalty functions

We have introduced all the necessary tools to build a *penalty-based decision making method*. However, we have not given yet a full view of it. In this section we explain the goodness and drawbacks of this strategy, as well as we explain the proposed method step by step.

The idea of consensus is based on obtaining the best solution with respect to an input set [24, 26]. However as we have previously defined the input set (a set of images), the set of possible solutions in the *aggregation phase* is limited to the number of selected aggregation functions, and one is given as the solution for the entire set in the *exploitation phase*. This alternative limits the result to a global minimum, when we could select the best aggregation function for each element. Even, in case we use penalty functions over a cartesian product of lattices (see Section 4.6), we limit the solution to the chosen chain size. Thus, we consider a new set that takes all possible permutations of the possible solutions, allowing to obtain better results to a local level and increasing consequently the global quality of the final result. This is our motivation to use consensus, and define the strategy in three phases as follows:

1. An *aggregation phase*, that puts in value all the input set to obtain a set of possible outputs.

2. A *combination phase*, that generates all the possible solutions from the set of aggregated outputs.

3. An *exploitation phase*, that selects the combined set that presents the minimum penalty among all the possibilities.

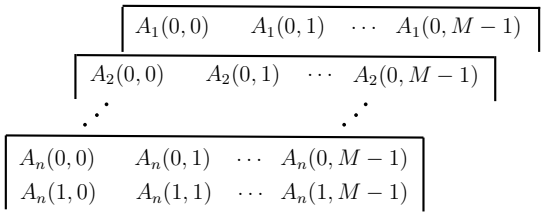


Figure 4.10: Representation of a stack of n images of size $N \times M$.

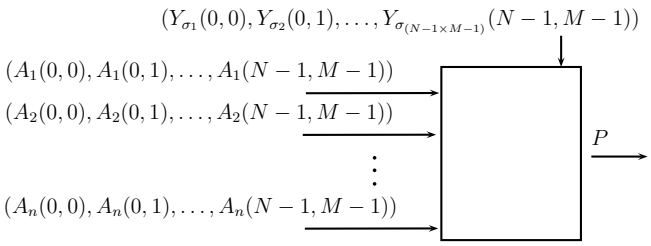


Figure 4.11: Schematic representation of a penalty function that receives a multifuzzy set as input. $\mathbf{A}$ represents the set of original inputs (images), and Y is one of the possible combination of the aggregation functions, where σ is the set of arrangements with repetition of q aggregation functions taken in groups of $N \times M$.

We can observe that the results are directly affected by the selected aggregation function set and the penalty function. Thus we base the election of them on our experience in the current problem. Moreover, the size set of the aggregation functions also increases the calculus cost with the increment on the number of aggregation functions, that increases the operations in the *combination phase* and the number of comparisons in the *exploitation phase*. From the computational point of view, this issue has to be taken into account to find a compromise.

A complete overview of the algorithm is given to reach a consensus from a collection of matrices (images). It consists of 5 main steps: (1) and (2) step correspond to the aggregation phase; (3) step is the combination phase; and (4) and (5) step that are part of the exploitation phase.

1. *Building the multifuzzy set A from the input matrices.*
   We start from a set like the one shown in Figure 4.10. Then, we build a multifuzzy set $\mathbf{A}$ as follows $A(i,j) = \{A_1(i,j), \ldots, A_n(i,j)\}$ where $i \in \{0, \ldots, N-1\}$, $j \in \{0, \ldots, M-1\}$ and n is the number of matrices. (Multifuzzy sets are introduced in Section 4.2).

2. *Selection of q aggregation functions* $(\gamma_1, \ldots, \gamma_q)$ *and calculus of the aggre-*

gated values with the inputs.
We calculate the set $\gamma(\mathbf{A}) = \{\gamma_1(\mathbf{A}), \ldots, \gamma_q(\mathbf{A})\}$, where $\gamma_k(\mathbf{A}(i,j)) = \big(\gamma_k(A_1(i,j)), \ldots, \gamma_k(A_n(i,j))\big)$. (Averaging aggregation functions, and specifically OWA operators are introduced in Section 4.4, and Section 4.4.1 respectively).

3. *Combination of the aggregated elements* $\gamma(\mathbf{A})$.
We build a new set Y conformed by all the possible variations, i.e. $Y = \{Y_1, Y_2, \ldots, Y_Q\}$, where $Q = q^{N\times M}$, $Y_k = \big(Y_{k\sigma_1}, \ldots, Y_{k\sigma_{(N-1\times M-1)}}\}$. So, $Y_{\sigma_k}(i,j)$ is the value for the element (i,j) in the matrix $\sigma_k \in \{\gamma_1(A), \ldots, \gamma_q(A)\}$, selected for the position $k \in \{0, 1, \ldots, N-1\times M-1\}$ in the candidate matrix. For instance, Y is composed of elements like the following:

$$Y_1 = \{Y_{1\gamma_1}(0,0), \ldots, Y_{1\gamma_1}(N-1, M-2), Y_{1\gamma_1}(N-1, M-1)\},$$

$$Y_2 = \{Y_{2\gamma_1}(0,0), \ldots, Y_{2\gamma_1}(N-1, M-2), Y_{2\gamma_2}(N-1, M-1)\},$$

$$\ldots$$

$$Y_Q = \{Y_{Q\gamma_q}(0,0), \ldots, Y_{Q\gamma_q}(N-1, M-2), Y_{Q\gamma_q}(N-1, M-1)\}.$$

4. *Estimation of the penalties for each of the permutations in* Y.
We select a penalty function P that fulfills the Definition 4.5.1 to apply to the collection of permutations. In a similar way as in the Figure 4.11. Then, we obtain a collection of penalties as follows $P\big((A_1, \ldots, A_{(N-1\times M-1)}); Y_1\big)$, $P\big((A_1, \ldots, A_{(N-1\times M-1)}); Y_2\big)$, $\ldots$, $P\big((A_1, \ldots, A_{(N-1\times M-1)}); Y_Q\big)$.

5. *Recovering the best candidate through the penalty-based function.*
We are ready to apply $f(\mathbf{A})$ from Eq. (4.8) to the penalties to get the solution. It is the consensus set $(Y_{\sigma_1}(0,0), Y_{\sigma_2}(0,1), \ldots, Y_{\sigma_{(N-1\times M-1)}}(N-1, M-1))$ that is minimum. (Penalty functions and penalty-based functions are explained in Section 4.5).

We introduce a global vision of *consensus decision-making based on penalty functions* for a multifuzzy set $\mathbf{A}$. We know from Theorem 4.6.1 that we can apply penalty functions over small chains of the multifuzzy set, and therefore *consensus decision-making* can also be used. It is translated to the possibility to work with regions of matrices (images). In this way, we have the possibility to give a different treatment to each region. In our case, as we use all the variations

of the output set, the use of different chains length is interchangeably. Note that using a chain size of one element, we recover the consensus approach element by element.

Therefore, the proposed framework is a useful methodology where a possible choice of an operator is needed and its choice can influence the result. For instance, it could also be applied in the context of relational calculus, preference structures and reliability theory.

Example 4.7.1. We want to get the best aggregation function among 3 possible ones using the proposed consensus decision-making methodology. For it, we use a *penalty-based function* over a cartesian product of lattices for the input $\mathbf{A} = (A_1, A_2, A_3)$ shown in Figure 4.1. The selected aggregation functions are $\gamma_1(\mathbf{x}) = \frac{1}{n}\sum_{k=1}^{n} x_k$, $\gamma_2(\mathbf{x}) = \min(\mathbf{x})$ and $\gamma_3(\mathbf{x}) = \max(\mathbf{x})$. The results for the different aggregation functions are shown in Figure 4.5 for the input multifuzzy set $\mathbf{A}$. Then, the selected penalty function $P_\nabla(\mathbf{A}; Y)$ is the function introduced in Ex. (4.16).

Before we compute the different penalties for the different aggregation functions, we need to create a new set $\mathbf{Y}$ that calculates all the variations of the aggregated values, as we introduced in the combination phase of the algorithm. Then, this set $\mathbf{Y}$ may look as follows

$$
\begin{aligned}
Y_1 &= \{Y_{1_{\gamma_1}}(0,0), Y_{1_{\gamma_1}}(0,1), \ldots, Y_{1_{\gamma_1}}(3,3), Y_{1_{\gamma_1}}(3,4)\},\\
Y_2 &= \{Y_{2_{\gamma_1}}(0,0), Y_{2_{\gamma_1}}(0,1), \ldots, Y_{2_{\gamma_1}}(3,3), Y_{2_{\gamma_2}}(3,4)\},\\
Y_3 &= \{Y_{3_{\gamma_1}}(0,0), Y_{3_{\gamma_1}}(0,1), \ldots, Y_{3_{\gamma_1}}(3,3), Y_{3_{\gamma_3}}(3,4)\},\\
Y_4 &= \{Y_{4_{\gamma_1}}(0,0), Y_{4_{\gamma_1}}(0,1), \ldots, Y_{4_{\gamma_2}}(3,3), Y_{4_{\gamma_1}}(3,4)\},\\
Y_5 &= \{Y_{5_{\gamma_1}}(0,0), Y_{5_{\gamma_1}}(0,1), \ldots, Y_{5_{\gamma_2}}(3,3), Y_{5_{\gamma_2}}(3,4)\},\\
&\ldots\\
Y_{3^{12}} &= \{Y_{3^{12}{}_{\gamma_3}}(0,0), Y_{3^{12}{}_{\gamma_3}}(0,1), \ldots, Y_{3^{12}{}_{\gamma_3}}(3,3), Y_{3^{12}{}_{\gamma_3}}(3,4)\}.
\end{aligned}
$$

As we are using a multifuzzy set that is a lattice, we can use different chain sizes over it, what can reduce the number of operations. Although as we already noticed, in this case the chain length is irrelevant for the final result. Then, we calculate the penalty for the elements in Y in similar way as in the example 4.6.2 with the chosen chain size. To finally reach the consensus matrix by the penalty-based function $F(\mathbf{A})$ with

$$F(\mathbf{A}) = \begin{pmatrix} 0.9 & 0.83 & 0.8 & 0.8 \\ 0.9 & 0.9 & 0.9 & 0.8 \\ 0.46 & 0.4 & 0.3 & 0.3 \end{pmatrix} \quad ,$$

where the aggregation function chosen for each position is as follows

$$\begin{pmatrix} \gamma_1 & \gamma_1 & \gamma_1 & \gamma_1 \\ \gamma_3 & \gamma_3 & \gamma_3 & \gamma_2 \\ \gamma_1 & \gamma_1 & \gamma_1 & \gamma_3 \end{pmatrix} .$$

Proving that introducing the combination phase in the consensus algorithm we can obtain better results locally and regardless of the chain size.

III

Applications

5

A consensus approach for image restoration with unknown noise model

In order to find an ideal image noise reduction algorithm, as introduced in Section 2.3, researchers have proposed hundreds of them. The most popular noise assumption is the additive Gaussian noise [20, 90, 114]. However a Gaussian noise assumption is too simplistic for most applications, specifically for medical and astronomical images [77]. In the particular case of medical images, in computer tomography (CT) the decay of the signal is better modeled with a Poisson distribution [58, 67, 99]. Other medical images, as single-photon emission computed tomography (SPECT) or positron emission tomography (PET), can also be well modeled with a Poisson distribution [89, 95]. In the case of magnetic resonance imaging (MRI), a Rice distribution better models the abnormalities in the image for a single-coil acquisition [4, 13].

Despite different approaches exist in order to reduce noise, all of them suffer a significant degradation in their performance with images owning a noise distribution for which these algorithms are not optimal; or when exist small deviations of the assumed noise model. It would be desirable to have a robust noise reduction algorithm being able to deal with different noise distributions, as well as combinations of them (various noise distributions are the introduced in Section 2.3.2). However this is a complex issue due that we rarely know the noise distribution exactly, at least some of its parameters cannot be always estimated correctly, or there is mixture of various noise types, and their relative contributions may not

always be clear or may be changing in some unpredictable way. Therefore, we propose to transform this blind noise reduction problem into a fuzzy decision-making process. For it, this approach is focused on the fusion of a set of filtered images, through a *multifuzzy set*, previously filtered from a noisy image with unknown noise distribution. We select methods existing in the literature that are optimal for a concrete noise. In particular, methods for impulse, Poisson, Gaussian and Rician noise are applied. Then, the fusion is carried out using the proposed consensus methodology via penalty functions on a cartesian product of lattices, where the penalty function chooses the value that minimizes the error for each pixel in accordance to different possibilities. This set of possibilities is formed by different *OWA operators* built from fuzzy linguistic quantifiers, since we can use language expressions as 'at least half', 'most of them' to define the weights. Moreover, fuzzy quantifiers provide a more flexible knowledge representation than classical logic, that it is restricted to the use of only two quantifiers, *there exists* and *for all* [30]. Our aim is to obtain consistent and stable results, regardless of the image nature (e.g. CT, MRI, digital image) and the noise characteristics under the image.

In the remainder of this chapter we explain how to transform this blind noise reduction problem into our consensus decision-making methodology as introduced in Chapter 4. First, we need to define the input set of our decision-making problem. For it we select different methods for different noise models that are introduced in Section 5.1. Once the input set is defined, we can present the proposed approach based on *penalty-based decision making* in Section 5.2. Followed by the results exposed in Section 5.3 where we compare our method with the noise reduction methods used to generate the input set, and the conclusions in Section 5.4.

5.1 Background: noise reduction methods

We select different estimators with the aim to generate the input set of our consensus methodology, in addition to using them to compare their performance with our approach. They cover different approaches to the image noise reduction problem, as well as they perform better for a specific noise distribution, in concrete for impulse, Gaussian, Rician and Poisson noise. Let's give an overview of the methods characteristics.

5.1.1 Impulse noise reduction method

The *DBAIN method* proposed by K.S. Srinivasan and D. Ebenezer [96] tackles the problem of impulse noise. The algorithm, in a first step detects if a processed pixel is noisy or noise-free depending on its occurrence in a corresponding window. If the pixel is determinated as corrupted, then the pixel is replaced by the median value of the window. Although, in case the median is considered corrupted, instead of the median, it is replaced by the value of neighbourhood pixels. This method does not require any parameter for its performance.

5.1.2 Gaussian noise reduction method

Additive white Gaussian noise (AWGN) has generally been found to be a reasonable model for noise originating from electronic amplifiers. The considered method to deal with white Gaussian noise has been the approach proposed by B. Goossens *et al.* [52]. This estimator is based on the *non-local mean (NLM)* estimator proposed by A. Buades *et al.* [20]. This version of NLM improves the original version, dealing with noise in non-repetitive areas with a post-processing step and presenting a new acceleration technique that computes the Euclidean distance by a recursive moving average filter. Moreover, they introduce an extension that can deal with correlated noise. However, its performance depends on a previous configuration. The standard deviation estimation, the searching window or the block size need to be defined previously. We use the configuration from the original paper for our experiments, specifically a block size of 11×11, and a searching window of 7×7.

5.1.3 Rician noise reduction method

The approach used to estimate Rician noise, the probability density function that mainly characterizes MRI in single-coil systems [5, 13, 55, 57], is proposed by S. Aja-Fernández *et al.* [4]. This estimator adapts the *linear minimum mean square error (LMMSE)* to Rician contaminated images. Moreover, noise estimation can be automatically calculated based on local statistics. Although the version used in our experiments is the approach in which the standard deviation is given as an input. The size of the square window used for the local estimation was 5×5.

5.1.4 Poisson noise reduction method

An *extension of the NLM* is proposed for images damaged by Poisson noise. C.A. Deledalle *et al.* [33] proposed to adapt the similarity criteria of NLM algorithm to Poisson distribution data. For this method, a previous configuration is required. For our experiments, the used parameters are those suggested in the original article, as the algorithm is tuned to obtain good results. Namely, a block size of 7×7, and a searching window of 21×21.

5.2 Proposed method: consensus for unknown noise reduction

We propose an approach based on penalty-based decision making that allows to remove noise from an image without any previous knowledge of the noise model. For it, we make use of the consensus methodology introduced in Chapter 4. This approach consists of four phases: preliminary, aggregation, combination and exploitation phase.

For a better understanding of the proposed approach a schema is shown in Figure 5.1. In the preliminary phase we start from a noisy image I_N. Then we apply four different methods optimized for a specific noise distribution to obtain several filtered images. With these new images, that in the case of our schema are (FI_1, FI_2, FI_3, FI_4), we build a multifuzzy set (introduced in Section 4.2). So each pixel $(i,\ j)$ is represented by four values (each value corresponds to pixel $(i,\ j)$ of each filtered image). In the aggregation phase we use three different OWA operators to build three fuzzy sets from the multifuzzy set. In particular, we use the OWA operators '*at least half*', '*most of them*' and '*as many as possible*', constructed from fuzzy quantifiers. (These operators are explained in more detail in Section 4.4.1). Then we apply these OWA operators to obtain $(I_{\mathrm{OWA}_{\mathrm{least}}}, I_{\mathrm{OWA}_{\mathrm{most}}}, I_{\mathrm{OWA}_{\mathrm{many}}})$. In the combination phase, we obtain a new set with all the possibilities for each pixel. Finally, in the exploitation phase, we take the best aggregated value for each pixel among the three availables. For that we use a penalty function that takes the value that minimizes the error with respect to the filtered images (FI_1, FI_2, FI_3, FI_4), and thus, the best fused image is obtained, I_{result}. (Penalty functions and penalty functions over cartesian product of

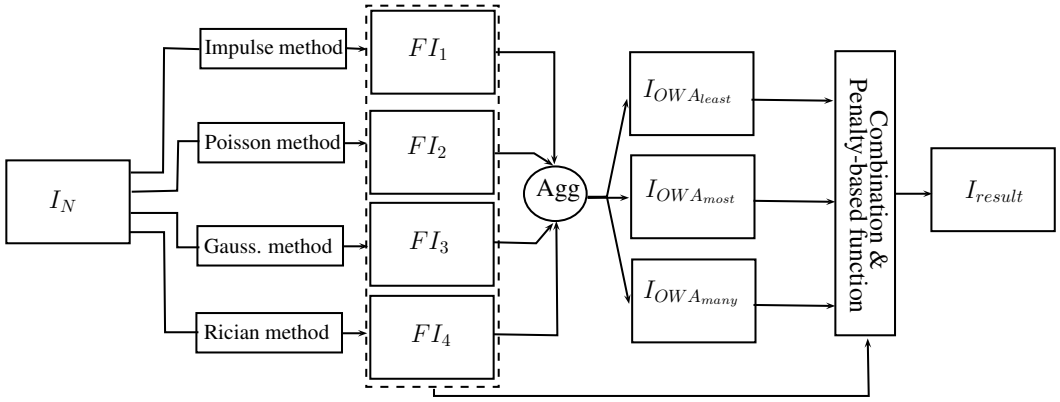


Figure 5.1: Schema consensus algorithm for unknown noise reduction.

lattices are explained in more detail in Section 4.5 and 4.6 respectively.)

Although we have exposed an approach that consists of the use of four methods and three aggregation functions. This approach can be extended to any number of methods and aggregation functions. In this manner, the consensus algorithm is defined as follows for p filters and q aggregation functions:

1. In the *preliminary phase*, for each pixel (i, j) a multifuzzy set $A(i, j) = \{A_1(i, j), \ldots, A_p(i, j)\}$ is obtained from the p filtering methods.

2. In the *aggregation phase*, we apply q different aggregation functions to $A(i, j)$, i.e. we get $\gamma_1(A(i, j)), \ldots, \gamma_q(A(i, j))$ where γ_k is any idempotent aggregation function, $\gamma_k(A(i, j)) = \gamma_k(A_1(i, j), \ldots, A_n(i, j))$.

3. In the *combination phase*, with the obtained images $\gamma_1(A(i, j)), \ldots, \gamma_q(A(i, j))$, we build a new set conformed by all the possible variations, Y.

4. In the *exploitation phase*, we take the element from Y that minimizes the penalty with respect to the multifuzzy set A using the penalty-based function $f(A)$. So, $Y_{\sigma_k}(i, j)$ is the value for the pixel (i, j) in the image $\sigma_k \in \{\gamma_1(A(i, j)), \ldots, \gamma_q(A(i, j))\}$, selected for the position $k \in \{0, 1, \ldots, N \times M\}$ in the candidate image. The consensus image is the $Y_{\sigma_1}(0, 0), Y_{\sigma_2}(0, 1), \ldots,$ $Y_{\sigma_{(N\times M)}}(N, M)$ that is minimum.

It can be deduced that the performance of this approach depends on the considered penalty function, as well as the selected aggregation functions.

5.3 Experiments and discussion

Different experiments are carried out to illustrate the behaviour of the consensus approach proposed in Section 5.2 facing the blind noise reduction task. That is why in this section we first introduce the used databases and the similarity measures. Followed by different experiments where the databases have been contaminated with different noise models, and combinations of them, to prove the effectiveness and robustness of consensus.

5.3.1 Materials and methods

To be able to compare the results to a ground truth, we work with images with 256 grey levels artificially corrupted with noise. Two databases are used: Live Image Quality Assessment Database (Live) [92] and a magnitude MR T1 volumen originally noise-free from the BrainWeb data set (brainWeb) [31]. The first database is corrupted with different noise distributions, as Gaussian and Poisson noise, while the second one is corrupted with Rician noise. (Noise density functions previously introduced in Section 2.3.2). In both cases, the noisy images are processed using the different noise reduction methods introduced in Section 5.1. For consensus methodology, the same parameters are used in all experiments. The aggregation functions used to reach a consensus are the *OWA operators*: 'at least half', 'as many as possible' and 'most of them'. (These operators are explained in more detail in Section 4.4.1). Namely, the OWA weights calculated for 4 elements are $w_{\text{least}} = \{0.5, 0.5, 0, 0\}$, $w_{\text{many}} = \{0, 0, 0.5, 0.5\}$, and $w_{\text{most}} = \{0, 0.4, 0.5, 0.1\}$, respectively. Finally, the *based-penalty function* (Eq. (4.18)) is applied over a cartesian product of lattices in groups of four neighbour pixels using the penalty function from Eq. (4.16). This equation is attached below for simplicity:

$$P_{\nabla}(\mathbf{A}, Y) = \sum_{q=1}^{m} \left(\sum_{p=1}^{n} |A_q(u_p) - y_q| \right)^2 .$$

To quantify the restoration performance of different methods, we use different similarity measures. Specifically the ones presented in Section 2.4. The PSNR is calculated. This is not bounded, a higher PSNR means better quality. However it is not very well matched to perceived structural information. This is our motivation to use also other quality indexes. In addition, the SSIM and the QILV

are used. Both give a measure of the structural similarity between the ground truth and the estimated images. Nonetheless, the former is more sensitive to the level of noise in the image and the latter to any possible blurring of the edges. This way we are able to assess the noise cleaning and border preserving capability of the different schemes. Both indexes are bounded; the closer to one, the better the image.

5.3.2 Experiments with images contaminated with Gaussian noise

A first experiment was accomplished with the 18 images from Live database corrupted with Gaussian noise. Table 5.1 contains the averages and the standard deviations achieved for this experiment. The Gaussian noise reduction method obtains the best results in average as expected, except for the QILV measure. However the difference between the best QILV result and the one obtained for the Gaussian method is negligible. Furthermore, the reached consensus is just behind the best, as the second better approach. We should note that the noise distribution is supposed unknown. We can also observe that if we would have not applied consensus, we could have obtained worse results. For instance, in case we would have just decided to use a single aggregation function and we chose it wrongly, as we do not know beforehand what aggregation function is better to use. However, for this experiment the results for the operator 'most of them' are similar to the proposed approach, due that this operator is chosen for consensus the 84% of the times. While 'at least half' is chosen the 9%, and 'as many as possible' the 7%.

In a second experiment we want to analyze how the noise level affects the consensus performance. For it, we executed several times the same database (Live) contaminated with various sigma values for Gaussian noise, from low to a high noise level. Figure 5.2 presents the graphs for the different quality measures. The image quality is affected as the noise level increases, although not all the measures and methods are affected in the same way. For example, the Rician method preserves its performance for SSIM as the noise amount increases due to its conservative behaviour around the borders; or the Poisson method worsens its achievement with the noise increasing. However, the consensus performance keeps as one of the three better approaches. The noise level affects its performance

Gaussian noise ($\sigma = 20$)						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	22.309	0.289	0.427	0.115	0.703	0.163
Impulse	22.218	0.263	0.432	0.114	0.711	0.165
Poisson	25.459	0.926	0.584	0.100	0.903	0.085
Gaussian	**29.818**	2.323	**0.840**	0.054	0.939	0.044
Rician	27.464	1.434	0.744	0.034	0.952	0.032
OWA_{least}	25.730	0.708	0.640	0.084	0.947	0.044
OWA_{many}	25.842	0.751	0.647	0.065	0.953	0.037
OWA_{most}	27.542	1.075	0.700	0.059	**0.965**	0.021
Consensus	27.548	1.081	0.700	0.061	**0.965**	0.022

Table 5.1: Results for the Live database [92], that contains 18 images 512x512 contaminated with Gaussian noise with $\sigma = 20$.

because it is affected by the errors of the used filtered and aggregated images that are also altered by noise. However, the proposed approach still shows a good compromise.

5.3.3 Experiments with images contaminated with Poisson noise

Consensus is an approach that can be used under different noise models without previous information. Then, in a new experiment we contaminated the same database (Live) with Poisson noise. We can recognize a coherent and consistent behaviour in Table 5.2, also in line with the previous experiments. The reached consensus obtains almost the same results as for the Poisson method, a PSNR and a SSIM in average close to the best achievement by the Poisson method. Furthermore, we can see that consensus obtains better results than any of the single aggregation functions. This is due to the cooperation of the different aggregation functions in the solution, where 'at least half' is chosen 18%, 'as many as possible' the 8% and 'most of them' the 74%. As we already mentioned in Section 4.7, thanks to the reached consensus locally, the global result also improves.

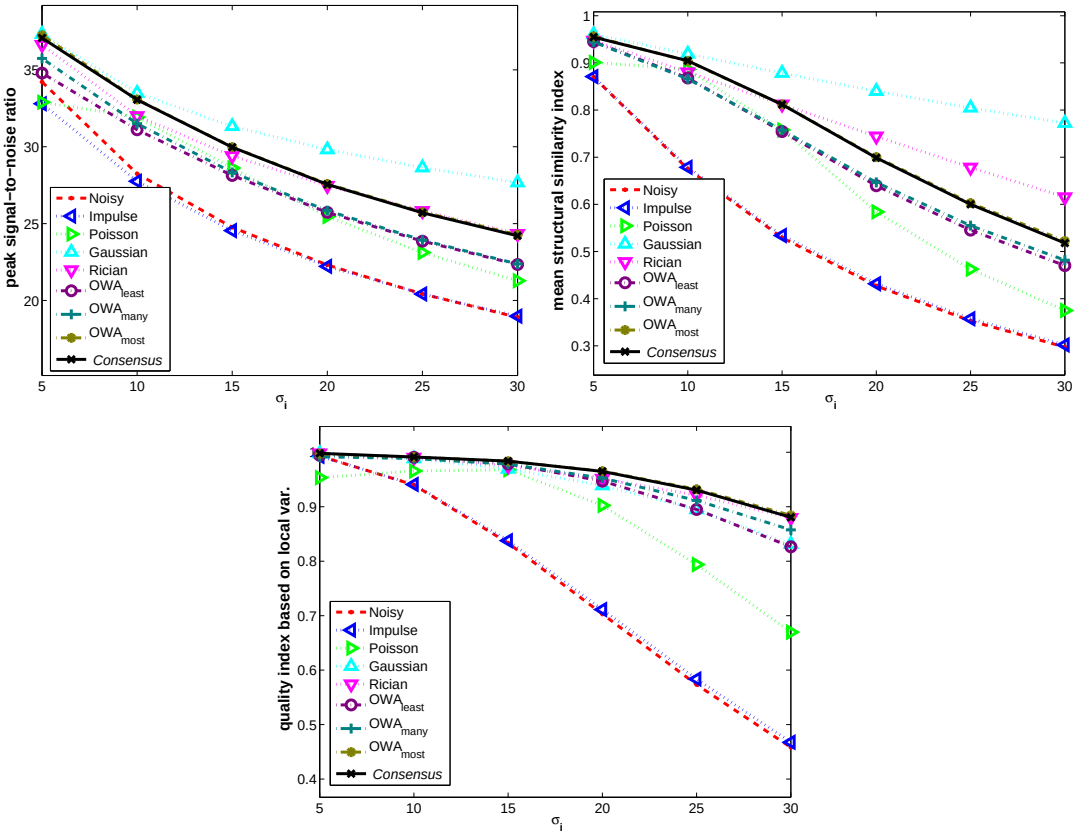


Figure 5.2: Performance of the different used noise reduction methods, the OWA operators and the reached consensus for different executions with various sigmas ($\sigma_i = \{5, 10, 15, 20, 25, 30\}$). It represents the mean values for the 18 images contained in the Live database [92]. By order, from left to right, and from up to down: PSNR, SSIM and QILV.

5.3.4 Experiments with images contaminated with Rician noise

Another experiment was performed with a different noise model, the Rician distribution. For this experiment an MRI volumen from brainWeb data set has been used. The volumen contains 181 images free of noise that were contaminated with Rician noise. To avoid any bias in the results due to the background, the quality measures are only applied to those areas of the image that are relevant, in other words, inside of the skull. Tables 5.3 and 5.4 show once again that consensus is one of the better approaches, just behind the best ones for a specific noise distribution. We can also observe that the Rician method performs worse than the

Poisson noise						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	27.665	0.424	0.666	0.099	0.926	0.068
Impulse	27.246	0.830	0.669	0.098	0.928	0.066
Poisson	**32.007**	2.722	**0.896**	0.031	**0.965**	0.022
Gaussian	30.005	2.540	0.820	0.076	0.871	0.100
Rician	28.101	2.275	0.792	0.069	0.879	0.091
OWA_{least}	28.983	0.946	0.790	0.052	0.982	0.010
OWA_{many}	28.884	0.992	0.769	0.052	0.974	0.015
OWA_{most}	30.919	0.953	0.804	0.046	0.986	0.008
Consensus	31.911	2.180	0.878	0.032	0.937	0.046

Table 5.2: Results for the Live database [92], which contains 18 images 512x512 contaminated with Poisson noise.

Rician noise ($\sigma = 10$)						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	30.803	1.951	0.871	0.043	0.970	0.056
Impulse	30.803	1.959	0.872	0.043	0.972	0.053
Poisson	35.395	2.089	0.960	0.015	0.991	0.008
Gaussian	**36.966**	2.900	**0.970**	0.013	0.994	0.004
Rician	33.446	2.370	0.942	0.019	0.994	0.004
OWA_{least}	32.503	1.952	0.920	0.026	0.990	0.022
OWA_{many}	33.139	2.179	0.924	0.027	0.993	0.010
OWA_{most}	33.742	2.008	0.927	0.024	0.992	0.016
Consensus	34.980	2.342	0.957	0.014	**0.995**	0.003

Table 5.3: Results for the MRI volumen, which contains 181 MR images contaminated with Rician noise with $\sigma = 10$.

Gaussian method. This is due to the conservative nature of the Rician method, that in case the data does not fit the model the method prefers to preserve the original data. While the Gaussian method applies a non-local averaging, which is a powerful and computationally expensive concept that is translated in a higher noise removing. Moreover, when the noise level increases (Table 5.4) the consensus works similarly to the previous case, where achievement gets affected by noise. It is natural since consensus looks for a cooperation between images, that also get affected by noise. In this experiment, the different aggregations ('least', 'many', and 'most') have provided on average 12%, 15% and 73% respectively to the consensus solution.

Rician noise ($\sigma = 20$)						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	24.866	1.986	0.720	0.089	0.826	0.154
Impulse	24.905	2.005	0.720	0.100	0.835	0.150
Poisson	27.836	1.915	0.808	0.061	0.940	0.114
Gaussian	**32.629**	2.483	**0.927**	0.030	0.970	0.021
Rician	29.310	2.156	0.873	0.040	0.973	0.032
$\text{OWA}_{\text{least}}$	27.578	1.864	0.825	0.055	0.959	0.084
OWA_{many}	29.012	2.318	0.854	0.052	0.977	0.029
OWA_{most}	29.853	2.170	0.865	0.032	**0.978**	0.032
Consensus	29.764	2.151	0.863	0.045	0.977	0.035

Table 5.4: Results for the MRI volumen, which contains 181 MR images contaminated with Rician noise with $\sigma = 20$.

5.3.5 Experiments with images contaminated with folded normal noise

For this experiment we use the images from Live database contaminated with a noise distribution which is not considered as an input of the consensus method. We consider a *folded normal distribution* that is calculated as the absolute value of a Gaussian distribution, i.e., given a normally distributed random variable X with mean μ and variance σ^2, the random variable $Y = |X|$ has a folded normal distribution. Tables 5.5 and 5.6 show that the proposed approach obtains a good performance in general, improving the performance of the single methods. In addition, it is interesting to observe how in this particular experiment the use of the single OWA operator 'as many as possible' obtains a better overall result than the proposed approach. This is due that for the consensus solution 'most of them' is chosen the 56%, 'at least half' the 29%, and 'as many as possible' the 15%.

5.3.6 Experiments with images contaminated with Poisson-Gaussian noise

So far we have assumed that noise comes from a single noise distribution. However, there are situations in which various distributions of noise can coexist in a single image. This is the case of many digital imaging devices that can be mod-

Folded normal noise ($\sigma = 10$)						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	28.340	0.579	0.832	0.066	0.980	0.014
Impulse	27.711	0.754	0.831	0.065	0.983	0.014
Poisson	28.121	1.115	0.892	0.031	0.950	0.029
Gaussian	28.810	0.518	0.913	0.025	0.977	0.018
Rician	29.404	0.783	0.907	0.025	0.978	0.021
OWA_{least}	26.992	0.631	0.909	0.016	**0.983**	0.011
OWA_{many}	**31.007**	0.771	**0.932**	0.014	0.975	0.019
OWA_{most}	29.678	0.589	0.928	0.018	0.980	0.015
Consensus	29.343	0.549	0.920	0.020	0.981	0.015

Table 5.5: Results for the Live database [92], which contains 18 images 512x512 contaminated with folded normal noise with $\sigma = 10$.

Folded normal noise ($\sigma = 25$)						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	20.501	0.677	0.544	0.133	0.834	0.128
Impulse	20.412	0.595	0.544	0.130	0.830	0.123
Poisson	21.553	0.679	0.762	0.044	**0.963**	0.027
Gaussian	21.433	0.731	0.763	0.092	0.780	0.138
Rician	22.998	1.010	0.756	0.071	0.794	0.130
OWA_{least}	19.886	0.663	0.691	0.062	0.927	0.050
OWA_{many}	**24.431**	0.737	**0.810**	0.029	0.906	0.064
OWA_{most}	22.653	0.698	0.806	0.032	0.907	0.064
Consensus	22.550	0.692	0.799	0.033	0.910	0.062

Table 5.6: Results for the Live database [92], which contains 18 images 512x512 contaminated with folded normal noise with $\sigma = 25$.

elled as Poisson-Gaussian noise, where the Poisson component accounts for the signal-dependent uncertainty inherent to photon accumulation, and the Gaussian component accounts for the other signal-indepedent noise sources, such as thermal noise. Tables 5.7 and 5.8 show the results for our consensus methodology using Live database contaminated with Poisson-Gaussian noise, where two different standard deviations are used for the Gaussian distribution. We can observe that the proposed method is robust to the mismatch with the assumed noise model, as it obtains a good overall performance among the better ones. However, Table 5.8 also shows that consensus performance gets affected as the noise increases, affecting further the structural information as the SSIM measure shows. Moreover, the impact of the aggregation functions is not so relevant in

Poisson-Gaussian noise ($\sigma = 15$)						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	22.974	0.157	0.457	0.116	0.743	0.154
Impulse	22.849	0.266	0.461	0.114	0.750	0.155
Poisson	26.300	0.992	0.625	0.096	0.930	0.054
Gaussian	**28.604**	1.486	**0.782**	0.033	**0.973**	0.012
Rician	27.064	1.047	0.697	0.051	0.962	0.024
OWA_{least}	25.999	0.648	0.638	0.089	0.935	0.055
OWA_{many}	26.002	0.642	0.634	0.073	0.939	0.052
OWA_{most}	27.083	0.826	0.674	0.072	0.954	0.037
Consensus	27.107	0.846	0.676	0.073	0.955	0.037

Table 5.7: Results for the Live database [92], which contains 18 images 512x512 contaminated with a Poisson-Gaussian noise, where $\sigma = 15$ for the Gaussian distribution.

Poisson-Gaussian noise ($\sigma = 20$)						
Method	PSNR		SSIM		QILV	
	mean	std	mean	std	mean	std
Noisy	21.207	0.175	0.384	0.111	0.630	0.171
Impulse	21.161	0.187	0.389	0.110	0.639	0.173
Poisson	24.117	0.661	0.509	0.105	0.849	0.120
Gaussian	**28.467**	2.038	**0.798**	0.041	**0.958**	0.022
Rician	26.100	1.145	0.667	0.047	0.948	0.026
OWA_{least}	24.529	0.594	0.572	0.096	0.904	0.080
OWA_{many}	24.495	0.609	0.571	0.077	0.912	0.074
OWA_{most}	25.885	0.809	0.614	0.078	0.933	0.055
Consensus	25.876	0.808	0.614	0.080	0.932	0.056

Table 5.8: Results for the Live database [92], which contains 18 images 512x512 contaminated with a Poisson-Gaussian noise, where $\sigma = 20$ for the Gaussian distribution.

the solution ('at least half' 11%, 'as many as possible' 9%, 'most of them' 80%). Therefore, it could be interesting to study a new framework configuration for consensus, where the input set and the aggregation functions may be tuned for this specific noise model.

5.3.7 Quality visual inspection

We carried out several experiments to verify the quantitative performance of the proposed consensus method, however visual quality is also important. Therefore, we proceed to inspect the images from Figures 5.3 to 5.7. We can see how the

reached consensus from the different images contaminated with different noise models (Gaussian, Poisson and Rician noise) exhibit better visual quality than any of the single methods. For instance, the Gaussian filtered images show over-filtering and loss of details in textured areas. In Figure 5.3(c) the area in the nose is over-filtered and the areas closeby have lost details in the texture. A similar situation occurs in Figure 5.4(c) where the image is in general over-filtered. For Figure 5.5(c) the general quality looks pleasant, although the face area is blurred, as well as some areas have lost the textures. The Gaussian filtered image in the MRI approach, Figure 5.6(c), presents a good quality. Nonetheless, the zoomed image, Figure 5.7(c), shows ringing artifacts close to the edges. A similar outcome is gotten for the Poisson method in the Rician case, Figure 5.6(e), that also shows ringing artifacts. On the contrary, the Poisson method achieves the best result for the Poisson approach, Figure 5.5(e), and comparable to the consensus image, Figure 5.5(f), as discussed for the results in Table 5.2. On the other hand, if we compare the remaining filtered images to the original ones, they are noisier than the reached consensus. The impulse filtered images, Figures 5.3(b), 5.4(b), 5.5(b) and 5.6(b), are not an alternative. Neither the Poisson method for the Gaussian problems, Figures 5.3(e) and 5.4(e). Finally, in the case of the Rician filtered images, they usually keep some noise. Although the effect could be visually pleasant for some approaches, as the Gaussian and Rician approach, Figures 5.3(d), 5.4(d) and 5.6(d).

On the other side, Figure 5.8 shows the results for an image contaminated with folded normal noise, that is a noise distribution not considered as an input of the consensus method. In this case, we can observe that the Gaussian method in Figure 5.8(c) over-filters some areas loosing details, such as in the grass-field. While the Rician method in Figure 5.8(d) presents a patch effect on the sky. However, consensus seems to overcome these results, as shown in Figure 5.8(f).

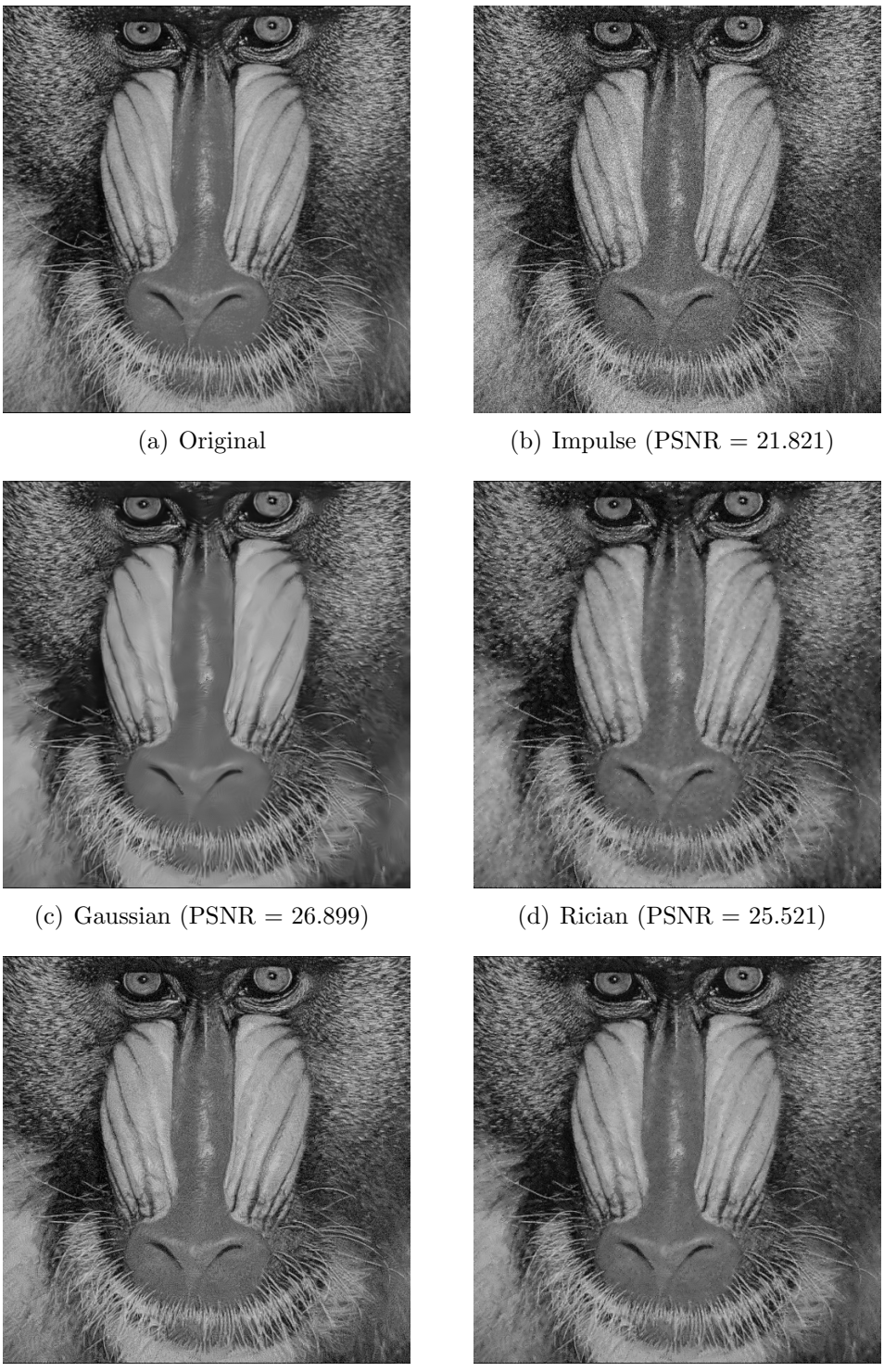
Finally, Figure 5.9 presents an image contaminated with Poisson-Gaussian noise. In the enlarged region we can see how the consensus method, Figure 5.9(f), keeps some details that the Gaussian method removes, such as the water texture (Figure 5.9(c)). The Poisson approach, in Figure 5.9(e), does not remove the Gaussian noise, while the Rician method in Figure 5.9(d) presents a block pattern in the water. Although consensus is still affected by noise, it seems to find a compromise with the visual quality.

5.3.8 Discussion

In summary, one can see that consensus methodology does not always get the best results, however it is a robust method that finds a cooperation between the considered methods. It generally obtains a good performance, and assures a result better than the worst of the individual solutions. Therefore, consensus is a good approach in situations where we do not know the noise distribution exactly; or when there is a mixture of various noise types. The main gains of this approach is the flexibility provided for an unknown noise model, where we can use several methods randomly, and the presented methodology finds a compromise respect to them. Quantitative and qualitative results already prove it.

5.4 Conclusions

In image noise reduction it is important to tune the method to the actual noise statistics, and the proposed consensus decision-making framework achieves this in an alternative way, by aggregating different filters. Results show that this methodology can be used for noise reduction with unknown noise distribution, because the noise is not known, the noise model does not follow the initial assumptions, or the image contains a mixture of different noise sources where their relative contributions may not be properly estimated. Therefore, consensus methodology is a good alternative in situations where we do not know beforehand the best filter to apply, or when a combination of different filters performs better than any single filter. Consensus is a robust and stable approach, although it must be contemplated that consensus computational time is dependent on the number of aggregation functions considered. For instance, if the number of used aggregation functions increases, the computational cost also increases. Additionally, the results tend to the mean when the number of idempotent functions increases. The challenge is to find a compromise on the number of aggregation functions to obtain a good performance. Furthermore, the used penalty function also affects the final results. Nonetheless, instead of considering it as a drawback, it can be seen as an advantage that brings flexibility to the system. Therefore, the system can be considered as a framework. Further research in the input set, aggregation functions and penalty function selection can be done.



(a) Original

(b) Impulse (PSNR = 21.821)

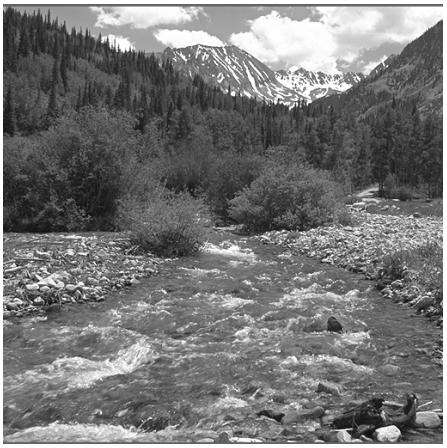
(c) Gaussian (PSNR = 26.899)

(d) Rician (PSNR = 25.521)

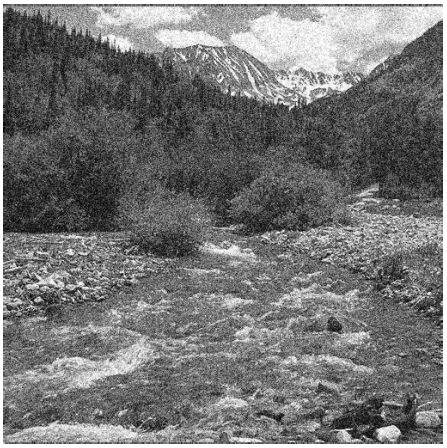
(e) Poisson (PSNR = 24.22)

(f) Consensus (PSNR = 25.924)

Figure 5.3: Noise reduction results for a Gaussian noisy image contaminated with $\sigma = 20$ using different noise reduction methods (Impulse, Gaussian, Rician and Poisson) and the reached consensus image.

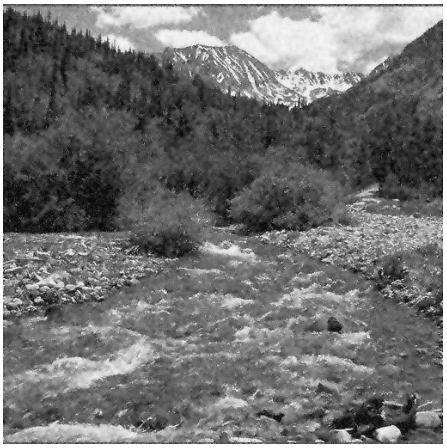


(a) Original

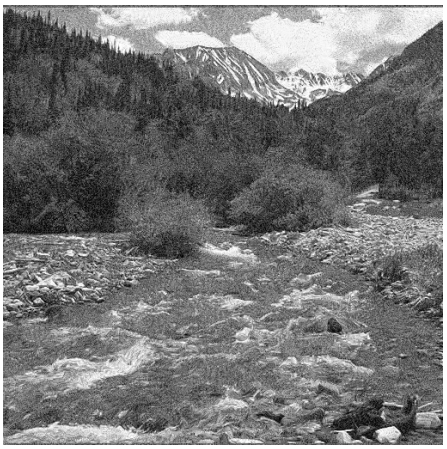


(b) Impulse (PSNR = 18.851)

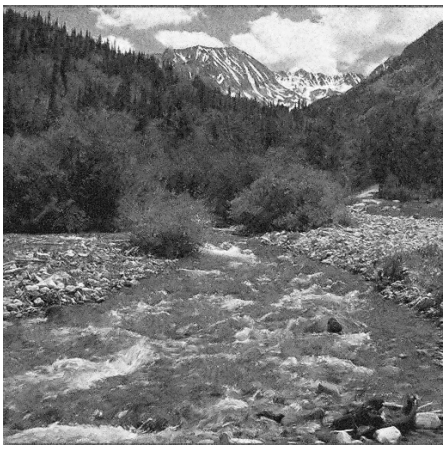
(c) Gaussian (PSNR = 23.62)



(d) Rician (PSNR = 22.591)



(e) Poisson (PSNR = 20.313)



(f) Consensus (PSNR = 22.612)

Figure 5.4: Noise reduction results for a Gaussian noisy image contaminated with $\sigma = 30$ using different noise reduction methods (Impulse, Gaussian, Rician and Poisson) and the reached consensus image.

(a) Original

(b) Impulse (PSNR = 27.555)

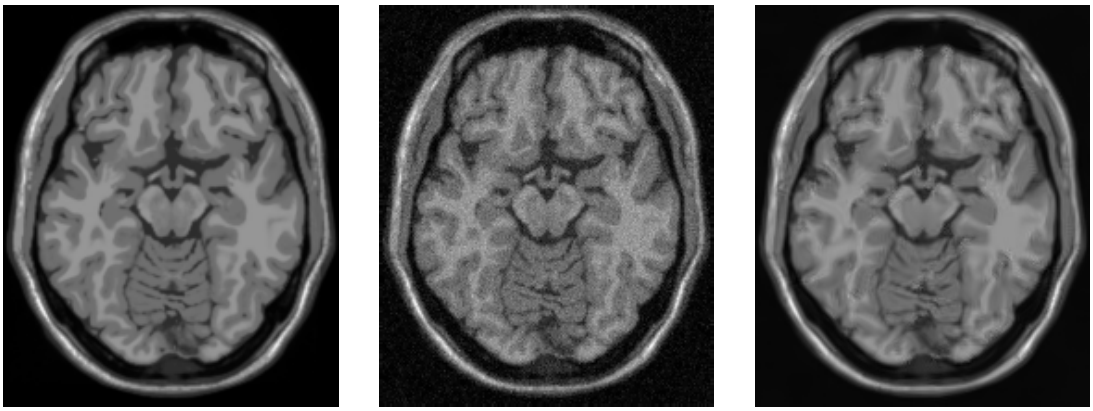
(c) Gaussian (PSNR = 31.799)

(d) Rician (PSNR = 27.821)

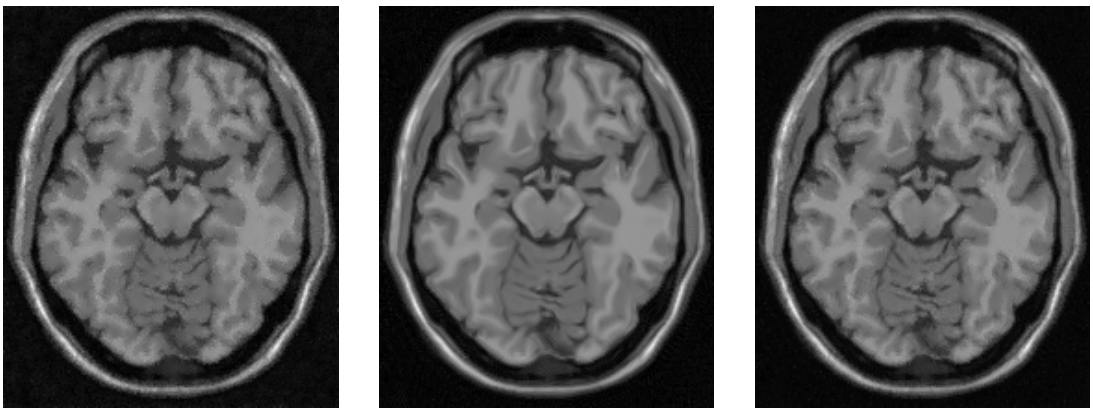
(e) Poisson (PSNR = 33.793)

(f) Consensus (PSNR = 32.699)

Figure 5.5: Noise reduction results for a Poisson noisy image using different noise reduction methods (Impulse, Gaussian, Rician and Poisson) and the reached consensus image.

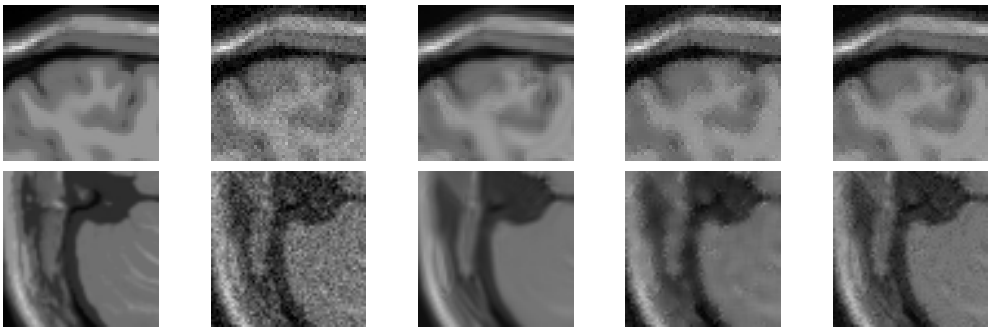


(a) Original (b) Impulse (PSNR = 33.15) (c) Gaussian (PSNR = 39.47)



(d) Rician (PSNR = 35.26) (e) Poisson (PSNR = 37.94) (f) Consensus (PSNR=37.09)

Figure 5.6: Noise reduction results for a Rician noisy image contaminated with $\sigma = 10$ using different noise reduction methods (Impulse, Gaussian, Rician and Poisson) and the reached consensus image.



(a) Original (b) Noisy (c) Gaussian (d) Rician (e) Consensus

Figure 5.7: Region extracted from two MR brain images contaminated with Rician noise. The first row is the image contaminated with $\sigma = 10$; the second row with $\sigma = 20$.

(a) Original

(b) Impulse (PSNR = 20.219)

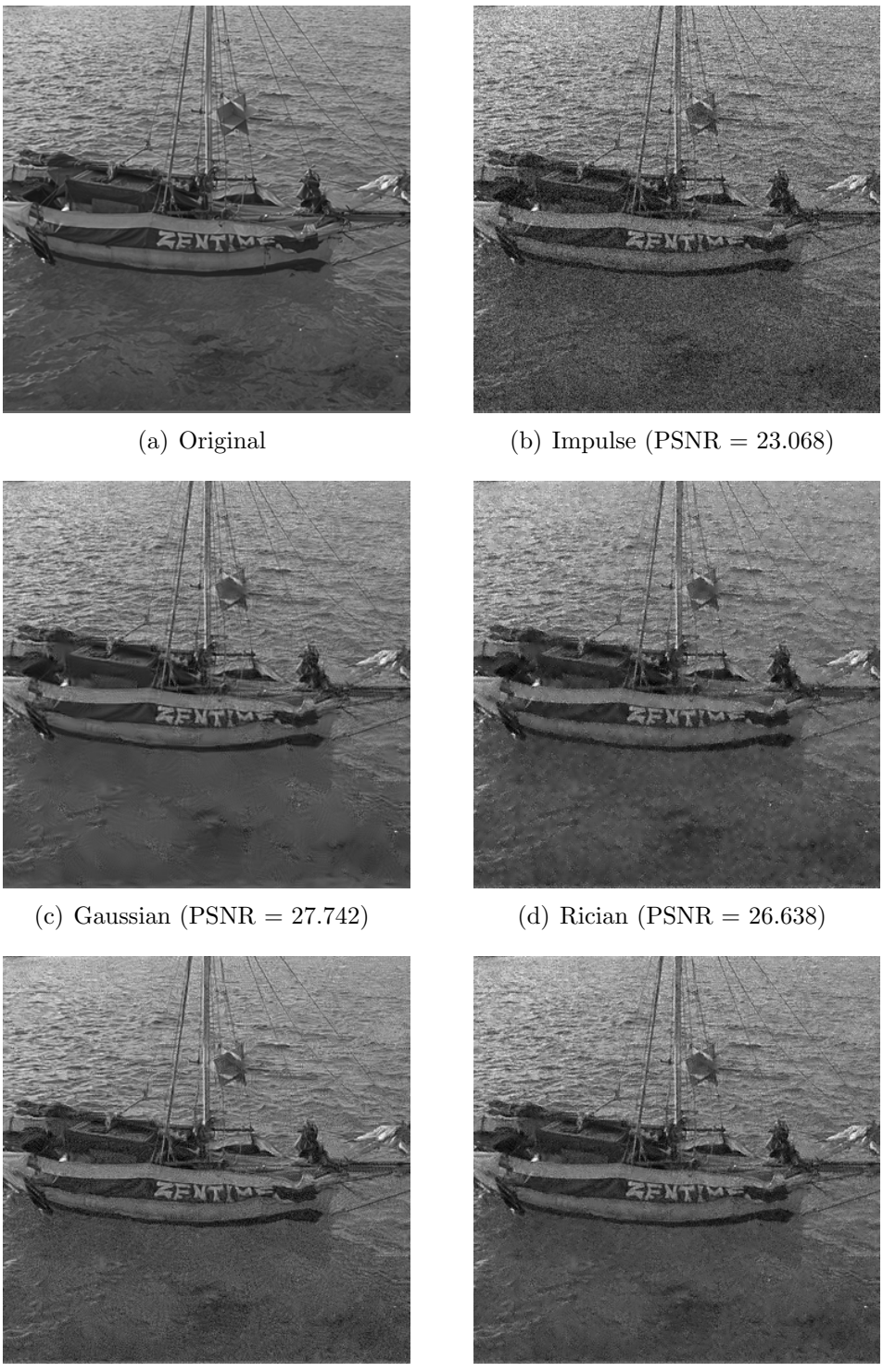
(c) Gaussian (PSNR = 21.596)

(d) Rician (PSNR = 23.377)

(e) Poisson (PSNR = 21.49)

(f) Consensus (PSNR = 22.565)

Figure 5.8: Noise reduction results for a folded normal noisy image contaminated with $\sigma = 25$ using different noise reduction methods (Impulse, Gaussian, Rician and Poisson) and the reached consensus image.



(a) Original (b) Impulse (PSNR = 23.068)

(c) Gaussian (PSNR = 27.742) (d) Rician (PSNR = 26.638)

(e) Poisson (PSNR = 25.811) (f) Consensus (PSNR = 26.671)

Figure 5.9: Noise reduction results for a Poisson-Gaussian noisy image, contaminated with $\sigma = 15$ for the Gaussian distribution, using different noise reduction methods (Impulse, Gaussian, Rician and Poisson) and the reached consensus image. Extracted region of the original image.

6

A consensus approach for non-stationary Gaussian noise filtering

Images are known to suffer from a wide range of degradations and artifacts due to acquisition, processing or transmission, such as noise, interferences, motion blur, misfocus, or lens distortions. Restoration techniques as introduced in Section 2.3, aim to estimate the original image by using a degradation model. Based on specific degradation models, many restoration algorithms have been proposed in the literature. Most of these techniques are parametric, i.e. they rely on the estimation of certain features of the degradation model such as the variance of noise or the direction of the motion blur.

Yet, there are situations in which either the needed information is not available, or it does not strictly follow the degradation model. Examples of this are the uncontrollable influence of different sources of degradation, noise with spatially-dependent variance, or the impossibility of estimating the direction of complex motion of the camera.

In this approach we propose a method for parametric restoration of images that copes with situations in which some information is missing, underlying parameters cannot be calculated or it does not exactly follow the initial assumptions. It is based on a consensus decision-making process among different realizations of the same algorithm. The method is designed to be able to cope with situations with uncertain input data. It allows us to introduce a range of input values and a reasoning strategy to produce a solution by consensus among different realiza-

tions.

The method is here applied in a simple but frequent situation where images are corrupted by additive Gaussian noise, but considering that the noise is non-stationary, i.e. the variance is spatially variant within the image. As a restoration algorithm, we considered the well-known Wiener filter [69] in its simplest version where no blurring is present. This filter needs an estimate of the variance of noise, σ^2. However, in our current case, the variance becomes a function of the position, $\sigma^2(\mathbf{x})$. We assume that we are not able to estimate the variability pattern that produces such a noise, so the Wiener filter is unable to estimate $\sigma^2(\mathbf{x})$ properly.

The agreed output of the restoration is obtained by combining the Wiener filter with a set of aggregation functions and a penalty function. The exploitation step selects an aggregation function that joins the information from different realizations of the filter with different input parameters (window size and noise variance). Then, the resulting agreed output takes the advantage of all the realizations to obtain a consistent spatially variant behaviour which benefits the restoration. The experiments showed better results than those obtained with the Wiener.

The remaining of the chapter is organised as follows. Section 6.1 explains the Wiener filter. The proposed method is presented in Section 6.2 where we make use of the consensus decision-making methodology introduced in Chapter 4. Followed by the experiments and results in Section 6.3. Finally, the Section 6.4 exposes the conclusion.

6.1 Background: the Wiener filter

The Wiener filter, as previously introduced in Section 2.3.1, is a parametric noise filter that performs uniform filtering of the image with no distinction for changes between textured and homogeneous regions, which sometimes results in an unacceptable blurring.

In the case under study, where no blur is present in the image and it just suffers degradations by noise. The Wiener filter can be simplified to its simplest version as the linear minimum mean square error (LMMSE) estimation of the original image $I_0(\mathbf{x})$, from a noisy one $I_N(\mathbf{x})$, when the former is corrupted with additive Gaussian noise:

$$I_F(\mathbf{x}) = \langle I_N(\mathbf{x})\rangle_{\mathbf{x}} + K(\mathbf{x}) \cdot (I_N(\mathbf{x}) - \langle I_N(\mathbf{x})\rangle_{\mathbf{x}}) \,, \tag{6.1}$$

where $I_F(\mathbf{x})$ is the estimate of $I_0(\mathbf{x})$; $\langle I_N(\mathbf{x})\rangle_{\mathbf{x}}$ is the local average of $I_N(\mathbf{x})$ in a neighbourhood W_s around $\mathbf{x}$ and

$$K(\mathbf{x}) = \frac{\mathrm{Var}\{I_N(\mathbf{x})\} - \sigma^2}{\mathrm{Var}\{I_N(\mathbf{x})\}}, \tag{6.2}$$

being $\mathrm{Var}\{I_N(\mathbf{x})\}$ the sample local variance of $I_N(\mathbf{x})$:

$$\mathrm{Var}\{I_N(\mathbf{x})\} = \langle I_N^2(\mathbf{x})\rangle_{\mathbf{x}} - \langle I_N(\mathbf{x})\rangle_{\mathbf{x}}^2 \,. \tag{6.3}$$

This function $K(\mathbf{x})$ can be seen as a confidence measure of how the data fit the proposed model. In those areas where $K(\mathbf{x}) \to 1$, the output of the filter $I_F(\mathbf{x}) \to I_N(\mathbf{x})$, i.e., there is no fitting and the output is the input data. This usually happens around the edges of the image. In homogeneous areas, $K(\mathbf{x}) \to 0$ and therefore the output is a smoothed version of the input, $I_F(\mathbf{x}) \to \langle I_N(\mathbf{x})\rangle_{\mathbf{x}}$.

6.2 Proposed method: a consensus Wiener

In this approach, we assume that an original image, $I_0(\mathbf{x})$, is corrupted with non-stationary Gaussian noise with zero mean and $\mathbf{x}$-dependent variance $\sigma^2(\mathbf{x})$:

$$I_N(\mathbf{x}) = I_0(\mathbf{x}) + N(\mathbf{x}; 0, \sigma^2(\mathbf{x})), \tag{6.4}$$

and we also assume our incapability to estimate the variation pattern of $\sigma^2(\mathbf{x})$ across the image. Thus, it is not possible to provide a proper estimate of $K(\mathbf{x})$ in Eq. (6.2).

The proposed method estimates $K(\mathbf{x})$ via a consensus procedure where multiple input choices are considered. It is based on the methodology exposed in Section 4.7. $K(\mathbf{x})$ is here interpreted as the confidence of a certain pixel to trust data or the model. Different input parameters will produce different $K(\mathbf{x})$ values for each pixel, and therefore a procedure to select a global final value is needed. We propose to reach a consensus from the combination of different realizations for different σ input values and different size of the neighbourhood as it is shown in

Figure 6.1. The consensus strategy is applied to a pixel level, in such a way that the combination phase is not required. Therefore, the consensus noise restoration method is achieved in four phases: the preliminary phase, that calculates the input set of the decision method; the aggregation phase, that transforms the input set with a set of aggregations functions, the exploitation phase, where a penalty function is used to obtain the consensus, and the estimation phase, where the final image is obtained with the output of the decision schema. The complete method to restore an image I_N works in the following way:

1. Previously, in a *preliminary phase*, a set of confidence matrices $\{K_i\}_{i=1}^{n}$, that conforms the input set of consensus, is calculated by using Eq. (6.2) for different configurations of the variance (σ_i^2) and neighbourhoods (W_{si}). The set of variances configuration can be obtained by sampling an inter-percentile interval on sample variance obtained from the noisy image, though other strategies can be adopted when some information on the underlying variance is known.

2. In the *aggregation phase*, a set of aggregated confidence matrices $\{Agg_j\}_{j=1}^{k}$ is generated by applying OWA operators with different weighting vectors over the input set $\{K_i\}_{i=1}^{n}$. (It can be read more about OWA operators in Section 4.4.1). A set of seven representative OWA operators was used, the weighting vectors are depicted in Figure 6.2. Note that the weights follow a trapezoidal shape which gives higher weights to the lower values of the sorted input. This way, the output of the OWA provides a higher confidence value when the majority of candidates agree.

3. In the *exploitation phase*, the selected $K_{\text{final}}(\mathbf{x})$, is calculated by minimizing the penalty function $K_{\text{final}}(\mathbf{x}) = \text{argmin}_{\text{Agg}_j} \sum_{i=1}^{n} |K_i(\mathbf{x}) - Agg_j(\mathbf{x})|$. (Penalty functions are explained further in Section 4.5).

4. Finally, in the *estimation phase*, the *Wiener filter* of Eq. (6.1) is applied with this final confidence estimation $K_{\text{final}}(\mathbf{x})$ to get the restored image $I_F(\mathbf{x})$.

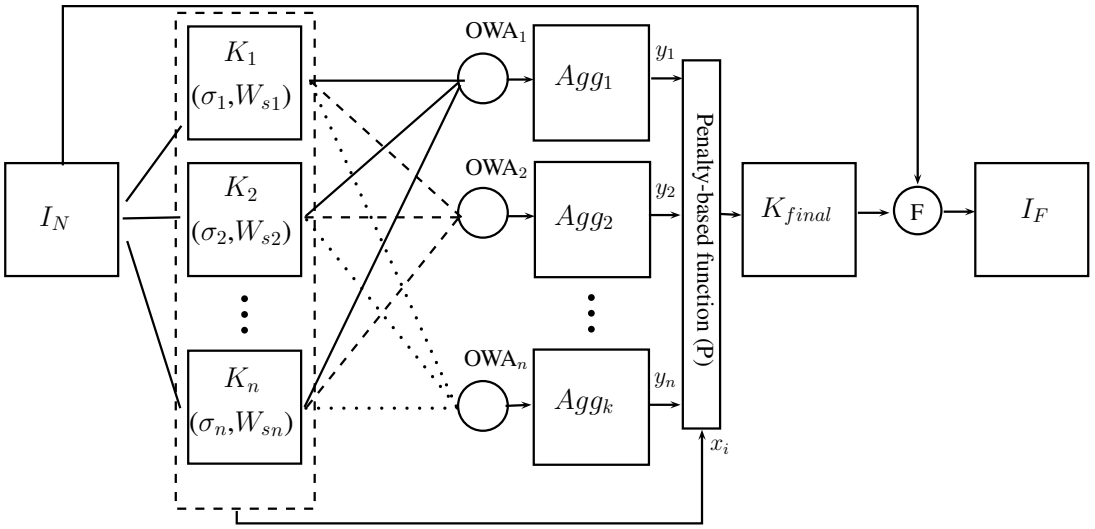


Figure 6.1: Proposed scheme for filtering of non-stationary noise using the parametric Wiener filter F (Eq. (6.1)) and the estimator K (Eq. (6.2)).

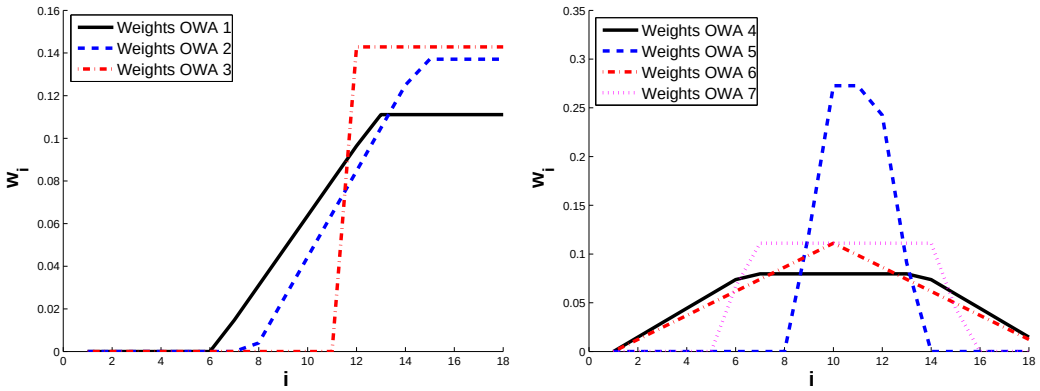


Figure 6.2: Weighting quantification for the used OWA operators.

6.3 Experiments and discussion

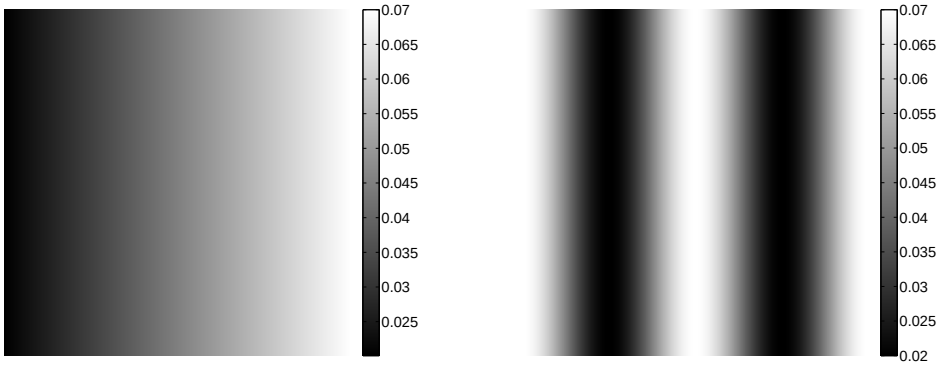
We carry out different experiments to illustrate the behaviour of the proposed approach in Section 6.2 facing the non-stationary Gaussian noise. In this section, we first introduce the images, the noise shapes, and the similarity measures used in the experiments. Followed by the experiments that show the advantages of the proposed approach.

6.3.1 Materials and methods

The proposed method was tested with the well-known images from Figure 6.3: *cameraman*, *barbara* and *mandrill* corrupted with additive non-stationary Gaussian noise. The images were normalized into the interval $[0, 1]$. An horizontal ramp and a cosine function were adopted for the spatial distribution of $\sigma^2(\mathbf{x})$ of noise as shown in Figure 6.4. The dynamic range of $\sigma^2(\mathbf{x})$ is $[0.02, 0.07]$ for *cameraman*, $[0.03, 0.08]$ for *barbara* and $[0.02, 1.0]$ for *mandrill*. The different realizations of σ_i were $\{0.02, 0.04, 0.06, 0.08, 1.0, 1.2\}$, which were combined with neighbourhoods of sizes 3, 5 and 7. Hence, 18 (3×6) different candidates were calculated for the aggregation phase.

(a) Cameraman (b) Barbara (c) Mandrill

Figure 6.3: Original images used in the experiments.



(a) Linear ramp (b) Cosine function

Figure 6.4: σ shapes used in the experiments.

The restoration performance was quantified by using the MSE, SSIM and QILV. (These similarity measures are explained further in Section 2.4). The

	Noisy	Ideal Ws [3x3]	Ideal Ws [5x5]	Ideal Ws [7x7]
MSE	142.6187	41.92737	37.88997	40.10132
SSIM	0.55594	0.83009	0.87973	0.88628
QILV	0.92905	0.99609	0.99519	0.99529

Table 6.1: Results using the ideal $\sigma^2(\mathbf{x})$ for the *cameraman* image with a dynamic range of $[0.02, 0.07]$ for the linear shape (Figure 6.4(a)).

	Noisy	Ideal Ws [3x3]	Ideal Ws [5x5]	Ideal Ws [7x7]
MSE	217.0559	84.29639	77.2783	78.99266
SSIM	0.5988	0.79997	0.84302	0.84767
QILV	0.8522	0.96353	0.95733	0.96327

Table 6.2: Results using the ideal $\sigma^2(\mathbf{x})$ for the *barbara* image with a dynamic range of $[0.03, 0.08]$ for the cosine shape (Figure 6.4(b)).

MSE is not bounded. A higher MSE represents worse quality. On the other side, the SSIM and the QILV give a measure of the structural similarity between the ground truth and the estimated images. Nonetheless, the former is more sensitive to the level of noise in the image and the latter to any possible blurring of the edges. This way we are able to assess the noise cleaning and border preserving capability of the different schemes. Both indexes are bounded; the closer to one, the better the image.

In order to compare with the ideal estimate of K, the Wiener filtered image was calculated from Eq. (6.1) with the original $\sigma^2(\mathbf{x})$ used to corrupt the images (see Figure 6.4). The neighbourhood size, that also affects the result, was set for each experiment depending on their performance. For instance, different executions were carried out for the neighbourhood size $[3 \times 3]$, $[5 \times 5]$, and $[7 \times 7]$. We took for each case the best solution to be compared with our approach. Such that $W_s = [5 \times 5]$ got the better results for almost all the cases, and in those that the results were similar to the $W_s = [7 \times 7]$, it was decided to use the previous one because the visual quality was more convincing. However, for the experiments done with the *mandrill* image the windows size was set to $[3 \times 3]$ as it obtained better results. Some of these experiment results are shown in Table 6.1, 6.2 and 6.3.

	Noisy	Ideal Ws [3x3]	Ideal Ws [5x5]	Ideal Ws [7x7]
MSE	264.608	118.3085	121.4869	126.7207
SSIM	0.69492	0.81014	0.80403	0.79756
QILV	0.8375	0.95495	0.94297	0.93773

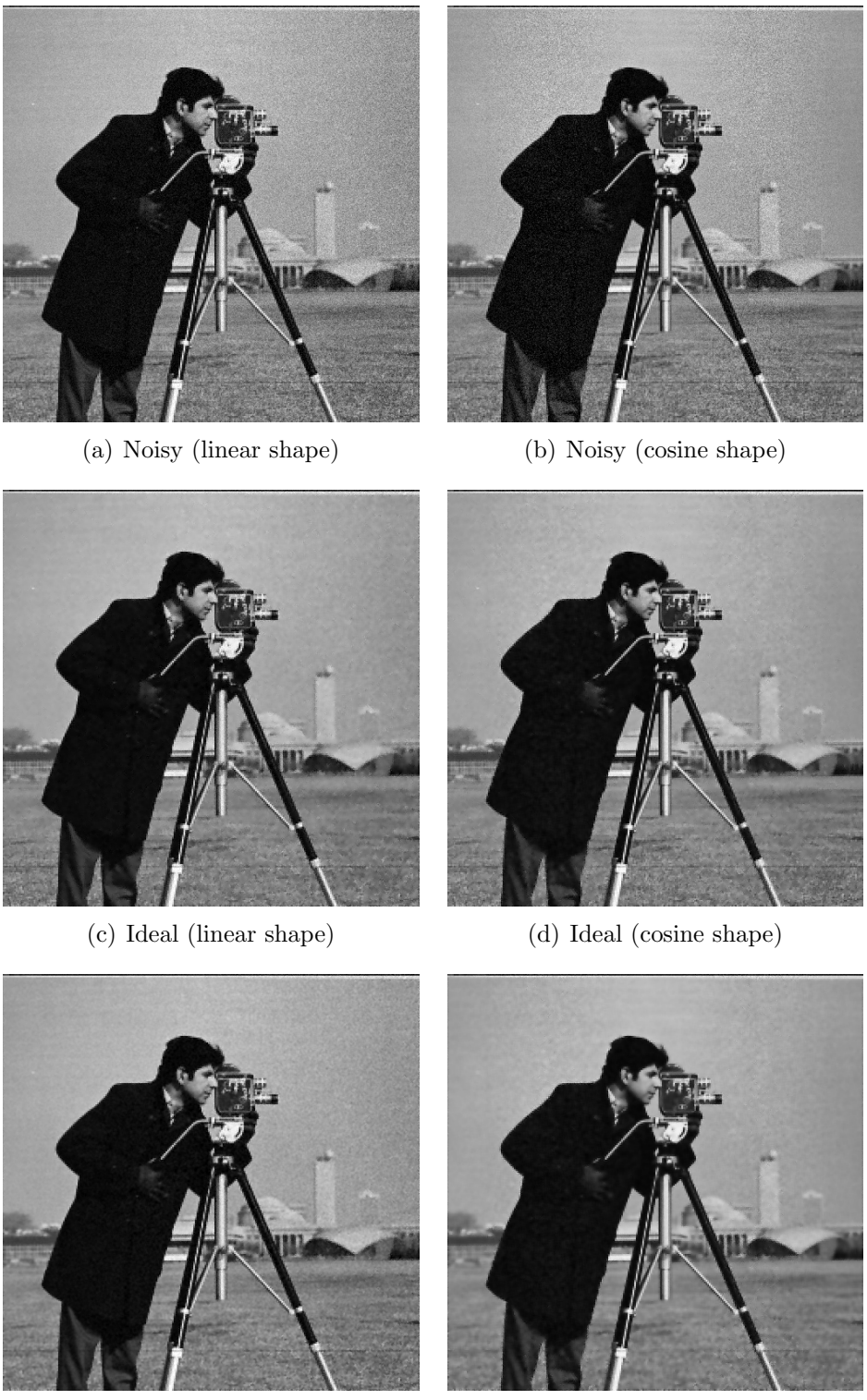
Table 6.3: Results using the ideal $\sigma^2(\mathbf{x})$ for the *mandrill* image with a dynamic range of $[0.02, 1.0]$ for the linear shape (Figure 6.4(a)).

	Noisy		Ideal [5x5]		Consensus [5x5]	
	mean	std	mean	std	mean	std
MSE	142.167	0.4444	37.615	0.1589	42.790	0.1776
SSIM	0.556	0.0006	0.880	0.0006	0.869	0.0005
QILV	0.930	0.0007	0.995	0.0002	0.961	0.0009

Table 6.4: Mean and standard deviation (std) from a hundred executions of the *cameraman* image contaminated with non-stationary Gaussian noise using a linear shape (Figure 6.4(a)).

6.3.2 Experiments with the *cameraman* image

The first experiment was carried out with the image *cameraman*, where the algorithm is tested in textured and homogeneous regions for the two different noise shapes: linear and cosine. Figure 6.5 shows that our approach is comparable to the ideal case in both experiments. As the MSE and SSIM also confirm (Figure 6.6(a) and 6.6(b), respectively). Although the QILV gets affected its performance by the noise (Figure 6.6(c)). It is not perceptible in the comparison between images, due to the fact that the magnitude order is negligible in variations in the decimal units. Furthermore, we can observe that a single realization of the algorithm also obtains good results, as for instance a configuration of $\sigma_i = 0.06$ and a $W_{si} = [5 \times 5]$ shows for the experiment with the horizontal ramp; and with $\sigma_i = 0.08$ and a $W_{si} = [5 \times 5]$ for the cosine function case. However, in both cases it is needed to know in advance the estimate, while we suppose it is unknown and our approach is robust to it. The results shown until now are from a single execution and therefore these are not significant statistically, that is our motivation to execute the algorithm a hundred times. The results shown in Table 6.4 and 6.5 present the expected behaviour for both cases.

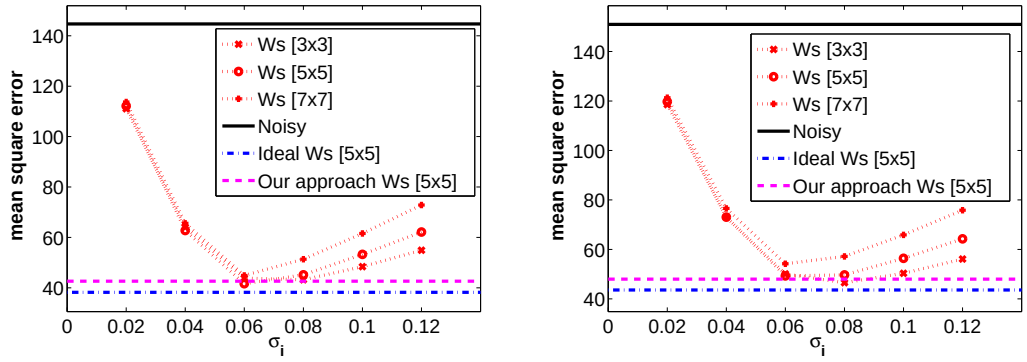


(a) Noisy (linear shape) (b) Noisy (cosine shape)

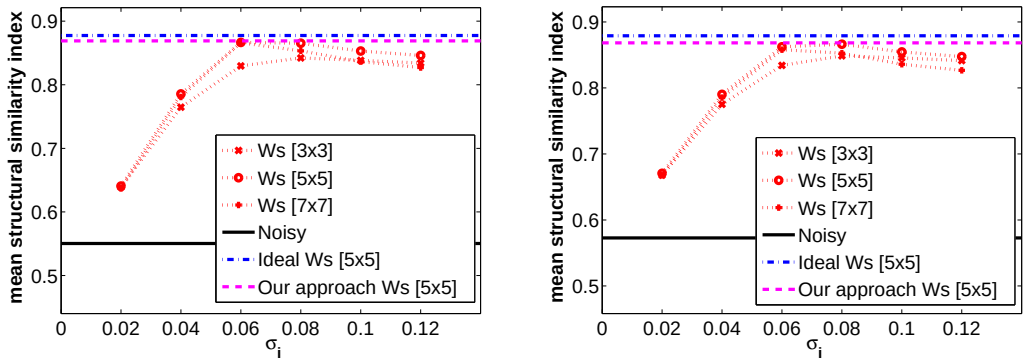
(c) Ideal (linear shape) (d) Ideal (cosine shape)

(e) Our approach (linear shape) (f) Our approach (cosine shape)

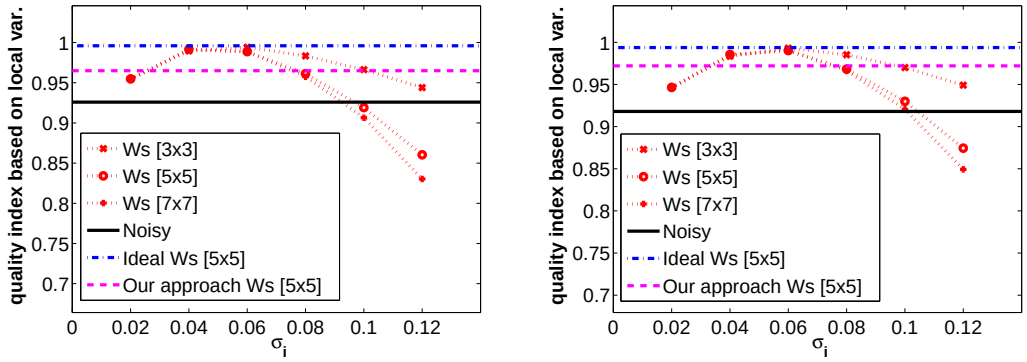
Figure 6.5: Results for *cameraman* contaminated with non-stationary Gaussian noise which oscillates in [0.02, 0.07]. Two different noise shapes were used, in concrete those shown in Figure 6.4.



(a) MSE



(b) SSIM



(c) QILV

Figure 6.6: Results using different quality measures for the *cameraman* image for non-stationary Gaussian noise. In the first column we used a linear shape (Figure 6.4(a)), while in the second column we used a cosine shape (Figure 6.4(b)).

	Noisy		Ideal [5x5]		Consensus [5x5]	
	mean	std	mean	std	mean	std
MSE	141.327	0.4690	32.290	0.2038	38.942	0.1877
SSIM	0.551	0.0005	0.886	0.0007	0.837	0.0007
QILV	0.929	0.0006	0.997	0.0001	0.988	0.0003

Table 6.5: Mean and standard deviation (std) from a hundred executions of the *cameraman* image contaminated with non-stationary Gaussian noise using a cosine shape (Figure 6.4(b)).

	Noisy		Ideal [5x5]		Consensus [3x3]	
	mean	std	mean	std	mean	std
MSE	209.599	0.6791	79.400	0.3478	87.427	0.3700
SSIM	0.599	0.0006	0.844	0.0008	0.803	0.0009
QILV	0.868	0.0010	0.958	0.0012	0.911	0.0018

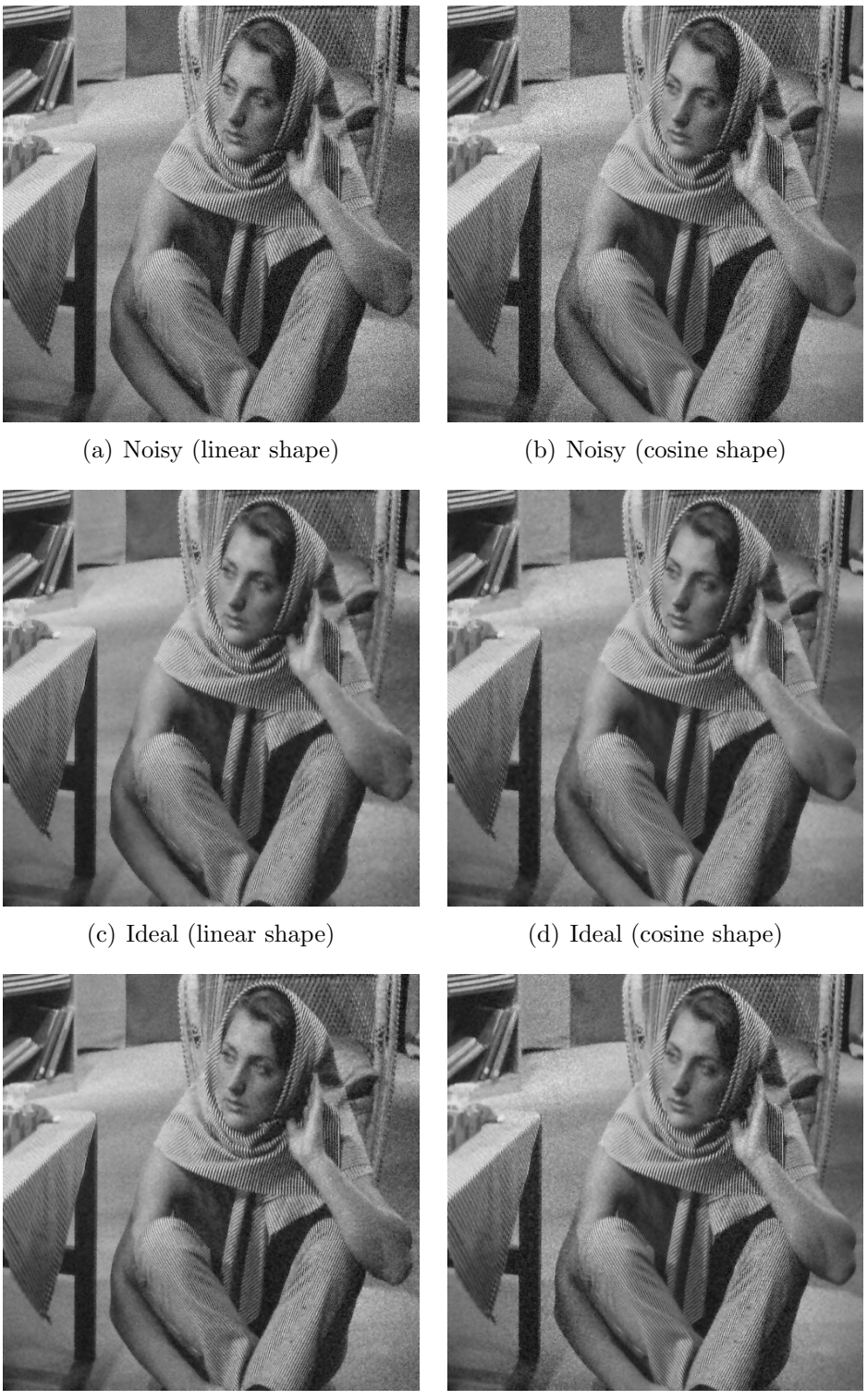
Table 6.6: Mean and standard deviation (std) from a hundred executions of the *barbara* image contaminated with non-stationary Gaussian noise using a linear shape (Figure 6.4(a)).

	Noisy		Ideal [5x5]		Consensus [3x3]	
	mean	std	mean	std	mean	std
MSE	216.613	0.6888	77.176	0.3728	90.333	0.4199
SSIM	0.598	0.0006	0.842	0.0008	0.790	0.0009
QILV	0.853	0.0013	0.959	0.0011	0.906	0.0016

Table 6.7: Mean and standard deviation (std) from a hundred executions of the *barbara* image contaminated with non-stationary Gaussian noise using a cosine shape (Figure 6.4(b)).

6.3.3 Experiments with the *barbara* image

Another experiment was done with a more textured image: the *barbara* image. This image is also contaminated with the linear and cosine shapes. The results shown in Figure 6.7 prove once more that our approach is comparable to the ideal case. Moreover, in the graphs for the MSE from Figure 6.8(a) we also presented the results from a single realization for a specific σ_i, where we can observe how our approach improves any of them. This is due to that with consensus, the image is improved locally, and consequently it improves globally. On the other hand, the SSIM and QILV from Figure 6.8(b) and 6.8(c) respectively, get affected their performance by the noise and worsens insignificantly with respect to the best individual cases. Moreover, the results for a hundred executions shown in Table 6.6 and 6.7 present the expected behaviour for both cases.

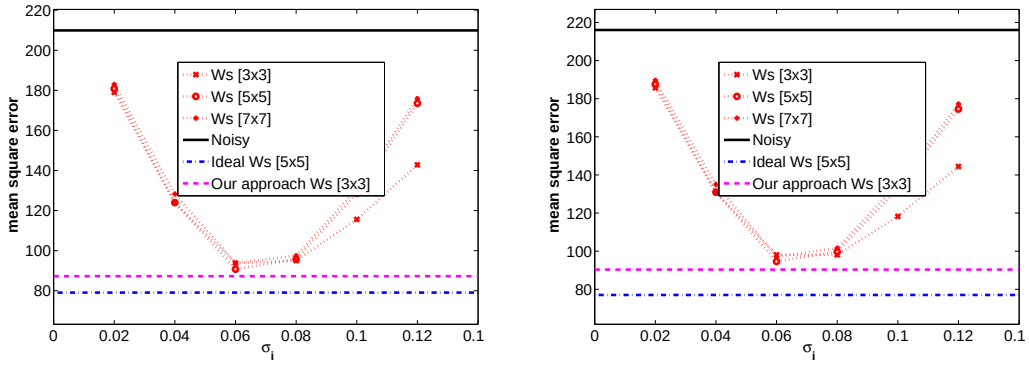


(a) Noisy (linear shape) (b) Noisy (cosine shape)

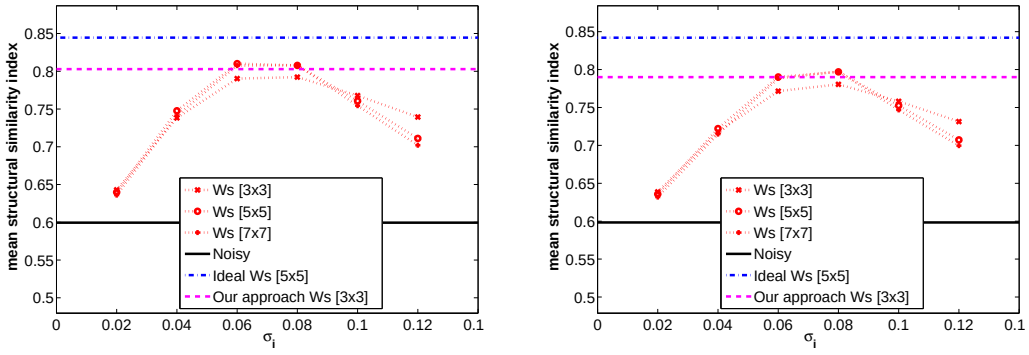
(c) Ideal (linear shape) (d) Ideal (cosine shape)

(e) Our approach (linear shape) (f) Our approach (cosine shape)

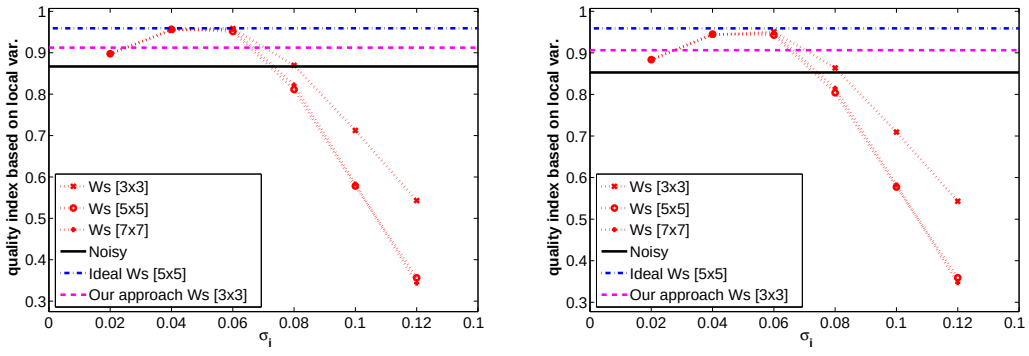
Figure 6.7: Results for *barbara* contaminated with non-stationary Gaussian noise which oscillates in [0.03, 0.08]. Two different noise shapes were used, in concrete those shown in Figure 6.4.



(a) MSE



(b) SSIM



(c) QILV

Figure 6.8: Results using different quality measures for the *barbara* image for non-stationary Gaussian noise. In the first column we used a linear shape (Figure 6.4(a)), while in the second column we used a cosine shape (Figure 6.4(b)).

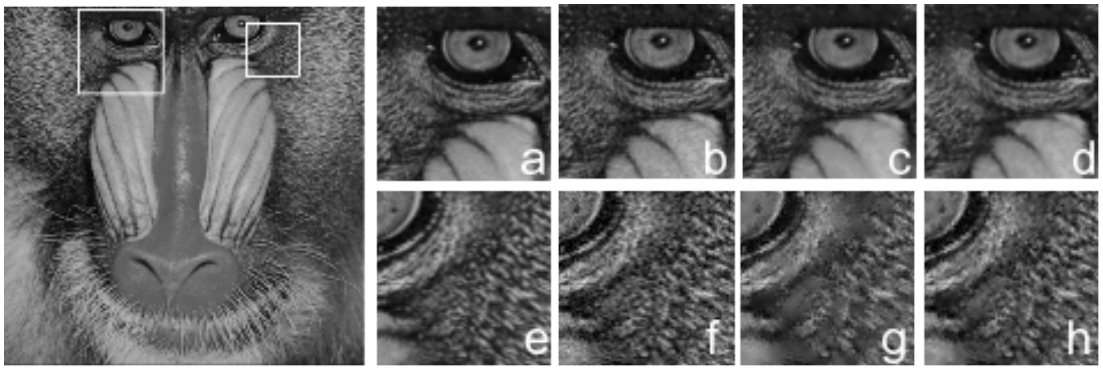


Figure 6.9: Detailed results for *mandrill* contaminated with non-stationary Gaussian noise using a linear shape which oscillates between 0.02 and 1.0. a,e: original; b,f: noisy; c,g: ideal; d,h: our approach.

	Noisy		Ideal [3x3]		Consensus [3x3]	
	mean	std	mean	std	mean	std
MSE	266.372	0.7949	118.931	0.4716	133.868	0.5175
SSIM	0.694	0.0006	0.810	0.0008	0.791	0.0007
QILV	0.835	0.0014	0.955	0.0010	0.920	0.0012

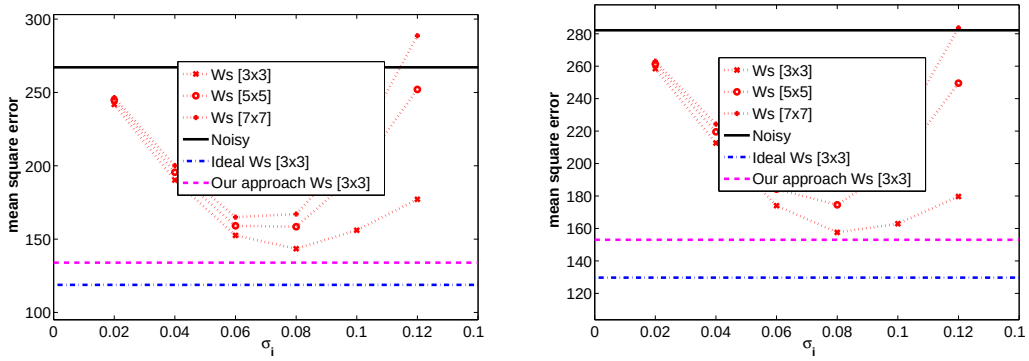
Table 6.8: Mean and standard deviation (std) from a hundred executions of the *mandrill* image contaminated with non-stationary Gaussian noise using a linear shape (Figure 6.4(a)).

	Noisy		Ideal [3x3]		Consensus [3x3]	
	mean	std	mean	std	mean	std
MSE	279.609	0.9271	128.395	0.5103	172.259	0.5762
SSIM	0.719	0.0006	0.819	0.0006	0.735	0.0007
QILV	0.813	0.0019	0.941	0.0015	0.853	0.0017

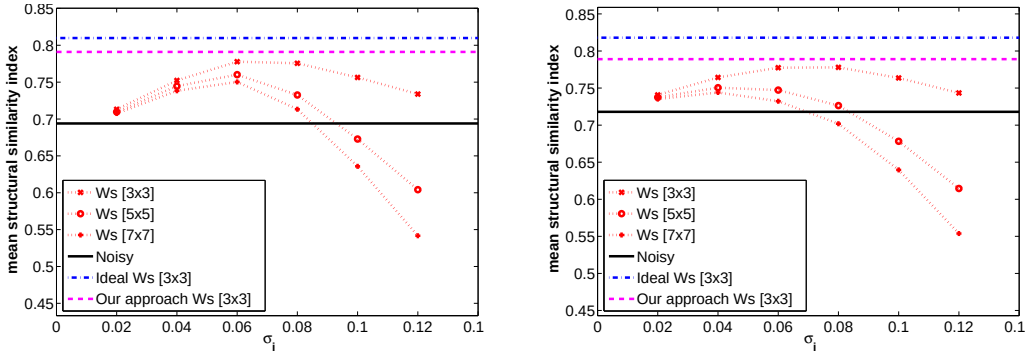
Table 6.9: Mean and standard deviation (std) from a hundred executions of the *mandrill* image contaminated with non-stationary Gaussian noise using a cosine shape (Figure 6.4(b)).

6.3.4 Experiments with the *mandrill* image

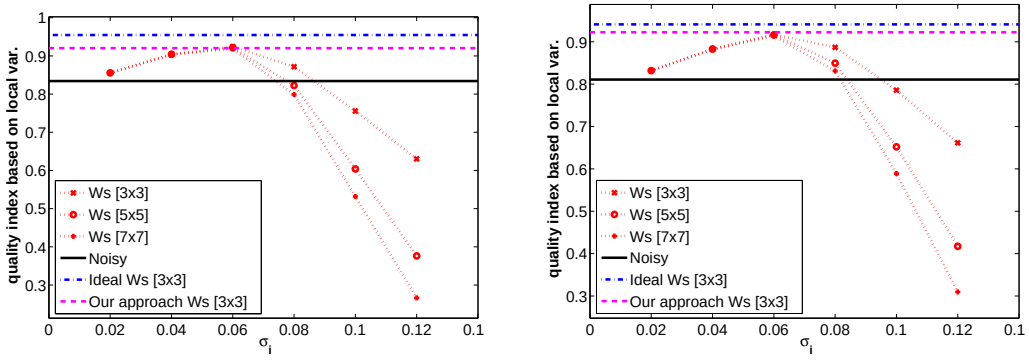
A new experiment was accomplished with the *mandrill* image, it is a very textured image and contaminated with an horizontal ramp and a cosine shape. Figure 6.9 shows that our approach gets better qualitative appearance than the ideal case. The edges are better defined and the textures are better preserved. Furthermore, the numerical results, MSE, SSIM and QILV shown in Figure 6.10(a), 6.10(b) and 6.10(c) respectively, support this behaviour. The results for a hundred executions for both shapes (linear and cosine) are shown in Table 6.8 and 6.9.



(a) MSE



(b) SSIM



(c) QILV

Figure 6.10: Results using different quality measures for the *mandrill* image for non-stationary Gaussian noise. In the first column we used a linear shape (Figure 6.4(a)), while in the second column we used a cosine shape (Figure 6.4(b)).

6.3.5 Discussion

In any case, we can figure out from the different experiments (Figure 6.6, 6.8 and 6.10) that if the σ_i estimation is lower or higher to the best particular case, the quality measures fall down quite fast. Nonetheless, the window size also affects the behaviour of the results. If the σ_i is lower than the best one, the behaviour is quite similar for any windows size. This is due to that $K(\mathbf{x}) \to 1$, and it is relying in the data, consequently, the Wiener filter is selecting the noisy pixels for the final image. On the other hand, when σ_i is over-estimated, it works differently: $K(\mathbf{x}) \to 0$, then the model is considered for that pixel. So the Wiener filter substitutes the pixels for their local average ($I_N(\mathbf{x})\rangle_{\mathbf{x}}$), affecting the window size to the local statistics and, therefore, the filter behaviour.

6.4 Conclusions

A new methodology based on consensus decision-making is presented, offering the possibility to use parametric restoration methods when any of the parameters cannot be properly estimated or the data do not strictly fit the underlying model. As an illustration we have applied this methodology to the case of images corrupted with non-stationary Gaussian noise, where there is a wide range of uncertainty on the value of the noise variance. To overcome this problem we have used the well-known Wiener filter (originally designed for stationary noise) and we obtained an output using a consensus procedure by selecting an aggregation function from a set of OWA operators by means of a penalty function. This allows us to use a wide range of input parameters at the same time, in this case the noise variance and the window size. The method presents a consistent and a conservative behaviour, preserving the borders. The experimental results show that the performance of the consensus method is similar to the case in which all the parameters are accurately known before.

Although the proposed method has been used in a simple scenario (signal plus noise), this has to be seen as a single illustration of the possibilities of this methodology in image restoration. The method can be extrapolated to other cases if (1) the restoration algorithm follows a parametric model; (2) there is an uncertainty on the input parameters or a slight misfit between data and model; (3) there is a way to measure the confidence of the model and the data.

7

A consensus approach for non-stationary Rician noise filtering

Noise is one source of degradation always present in magnetic resonance imaging (MRI) acquisitions. Thermal noise in MR scans is mainly originated by the subject or object to be imaged, followed by electronics noise during the acquisition of the signal in the receiver chain. Since noise is related to stochastic motion of free electrons, it is intrinsically imbricated with the acquisition process and therefore it is unavoidable. Some modern acquisition sequences are particularly affected by noise. For instance, techniques that demand large amounts of data, in order to reduce the acquisition time, also reduce the temporal averaging; as a consequence, the noise power is increased proportionally to the square root of the speedup. This is also true for those acquisitions in which the signal is attenuated, such as diffusion sequences with high b-values.

The degradation pattern introduced by noise affects the visual image quality and can negatively lead to an adequate interpretation and analysis of the data. Not only visual inspection is affected by noise, but also many common post-processing tasks (image registration, tissue segmentation, diffusion tensor estimation) and the obtaining of precise measures and quantitative imaging biomarkers.

The direct approach to minimize the influence of noise over the final image is the use of noise removal or signal estimation techniques. (In Chapter 2 can be read more about image restoration). Traditionally, noise filtering techniques in medical imaging are based on well-defined prior statistical models of data. The Gaussian model is the usual assumption in many algorithms. The definition of more evolved noise models for MRI have allowed the natural extension of

well-known image processing techniques to cope with features specific of MRI. (These density functions are presented in Section 2.3.2). Many examples can be found in the literature, such as the conventional approach (CA) [75], maximum likelihood (ML) [93], linear estimators [4], or adapted non-local mean (NLM) schemes [72, 105].

In the simplest case, when single-coil acquisitions are considered, the complex spatial MR data is typically assumed to be a complex Gaussian process, where real and imaginary parts of the original signal are corrupted with uncorrelated Gaussian noise with zero mean and equal variance σ_n^2. Thus, the magnitude signal calculated as the envelope of the complex signal is known to be Rician distributed [55, 57]. This Rician model has been the standard in MRI modeling for many years, and it has been the base for a myriad of filtering techniques as well as noise estimation algorithms [4, 71, 72, 105].

With the advent of multiple-coil systems to reduce acquisition time, parallel magnetic resonance imaging (pMRI) algorithms are used, dominant among them sensitivity encoding (SENSE) [85] and GeneRalized Autocalibrating Partially Parallel Acquisitions (GRAPPA) [53]. From a statistical point of view, the reconstruction process carried out by pMRI techniques is known to affect the spatial stationarity of the noise in the reconstructed data; i.e. the features of the noise become position dependent. Instead of assuming a single σ_n^2 value for each pixel within the image, the variance of noise varies with $\mathbf{x}$, i.e. $\sigma_n^2(\mathbf{x})$ [5, 8].

If SENSE is considered, the reconstruction process yields to the magnitude value of a complex Gaussian, and therefore, the final magnitude signal can still be considered Rician distributed, but with a different $\sigma_n^2(\mathbf{x})$ for each $\mathbf{x}$ [8, 35]. This way, many algorithms proposed for single coils systems can still be used if SENSE is considered, as long as the non-stationarity of the noise is taken into account. However, the estimation of the spatial pattern of $\sigma_n^2(\mathbf{x})$ is an issue that presents serious difficulties and some prior information is needed, such as the sensitivity maps in each coil. Unfortunately, this information is not always available.

Here we propose a novel approach to noise filtering in MRI assuming the spatially-variant patterns of noise created by SENSE. It assumes a Rician model but any additional information (such as the sensitivity of the coils or noise estimates) is not needed. The method is based on the consensus of different realizations of a given signal estimator for different σ_n^2 values. The idea is to generate a wide variety of candidates that are merged in a global solution that estimates

$\sigma_n^2(\mathbf{x})$. However, as the representative inputs are not known in advance, we use a set of aggregation functions to merge the realizations. Then, for each pixel, a penalty step will select the aggregated value that presents less dissimilarities with respect to the inputs [25, 26]. The final image is obtained with the information contained in the different candidates, showing a consistent spatially variant behaviour.

This approach extends a previous version of the method presented in Chapter 6, where Gaussian noise was considered, to the case of spatially-variant Rician noise in MRI. Although it is initially intended for SENSE acquisitions, it can really be applied to any other data where the noise follows a similar distribution.

As a restoration algorithm, we considered the *linear minimum mean square error (LMMSE)* estimator for Rician noise [4] due to its simplicity and robustness, which is the natural extension of the Wiener filter proposed in Section 6.1. However, the method can be applied to other signal estimators. An example of adaptation to the NLM algorithm will be provided as well.

This chapter is organized as follows: Section 7.1 introduces the statistical noise presented in SENSE reconstructed images and the LMMSE estimator. In Section 7.2 the new approach is presented. Then, in Section 7.3 different experiments are discussed for synthetic and real MR magnitude images using the proposed approach with LMMSE and adaptation for the NLM. Followed by the discussion in Section 7.4.

7.1 Background

In this chapter we propose an application of a consensus-based strategy between different realizations of the same filter with different parameters in order to obtain a final image independent to some parameter estimation. It is based on the consensus decision-making methodology introduced in Chapter 4. Although different noise filtering methods could have been selected, we will base our approach on the linear minimum mean square error (LMMSE) estimator for the Rician distribution proposed in [4] due to its simplicity and robustness. By means of a consensus methodology, we want to adapt this estimator for spatially variant noise scenarios where the original approach may fail. This methodology can easily be extended to other Rician filtering approaches.

7.1.1 Statistical noise model in SENSE reconstructed images

The starting point of this approach is the assumption that, under some acquisition circumstances and postprocessing procedures, the noise in the final magnitude image becomes non-stationary, i.e. the variance of noise σ_n^2 becomes dependent on the position $\mathbf{x}$: $\sigma_n^2(\mathbf{x})$. This is precisely the case in pMRI when SENSE is used. (Non-stationary noise is introduced in more detail in Section 2.3.2).

The formulation of the SENSE reconstruction is beyond the scope of this work. However, we follow prior studies [8, 85, 88, 102] where authors recall that the signal reconstructed with Cartesian SENSE in the $\mathbf{x}$-space, $S^{\mathcal{R}}(\mathbf{x})$, follows a Complex Gaussian distribution. If the magnitude is considered, i.e. $M(\mathbf{x}) = |S^{\mathcal{R}}(\mathbf{x})|$, the final magnitude image will follow a Rician distribution [8, 102], just like single-coil systems. However, in both cases, due to the reconstruction process, the resulting distributions are non-stationary. This means that the variance of noise σ_n^2 will vary from point to point across the image: $\sigma_n^2(\mathbf{x})$. The final value of the variance of noise at each point will depend on the covariance matrix between coils of the original data (prior to reconstruction) and on the sensitivity map of each coil, but not on the data themselves.

The *magnitude image* can be modeled as follows:

$$M(\mathbf{x}) = |S^{\mathcal{R}}(\mathbf{x})| \tag{7.1}$$

where

$$S^{\mathcal{R}}(\mathbf{x}) = I_0(\mathbf{x}) + N(\mathbf{x}; 0, \sigma_n^2(\mathbf{x})). \tag{7.2}$$

Being $M(\mathbf{x})$ the noisy magnitude image, $I_0(\mathbf{x})$ a noise-free SENSE reconstructed signal and $N(\mathbf{x}) = N_r(\mathbf{x}) + j \cdot N_i(\mathbf{x})$ some complex Gaussian noise with zero mean and $\mathbf{x}$-dependent variance $\sigma_n^2(\mathbf{x})$. Note that, noise is not really added to the reconstructed signal, as described in Eq. (7.1). However, the final model given by the reconstructed signal allows us to simplify the whole process and model it this way, without loss of generality. This way, we do not need the sensibility of the coils and correlation information, and the model is totally compatible with single-coil formulations.

7.1.2 LMMSE estimator

The selected noise filtering technique is the LMMSE signal estimator for the stationary Rician distribution, as proposed in [4], and based on the Wiener filter introduced in Section 2.3.1. It estimates the original signal $\widehat{I}_0(\mathbf{x})$ from the noise magnitude data, $M(\mathbf{x})$ as described in Eq. (7.1), using the local information and the original variance of noise σ_n^2. The estimator is defined as follows

$$\widehat{I}_0(\mathbf{x}) = \sqrt{\langle M^2(\mathbf{x})\rangle_{\mathbf{x}} - 2\sigma_n^2 + K(\mathbf{x}) \cdot (M^2(\mathbf{x}) - \langle M^2(\mathbf{x})\rangle_{\mathbf{x}})}, \tag{7.3}$$

with

$$K(\mathbf{x}) = 1 - \frac{4\sigma_n^2 \left(\langle M^2(\mathbf{x})\rangle_{\mathbf{x}} - \sigma_n^2\right)}{\langle M^4(\mathbf{x})\rangle_{\mathbf{x}} - \langle M^2(\mathbf{x})\rangle_{\mathbf{x}}^2}. \tag{7.4}$$

The operator $\langle M^n(\mathbf{x})\rangle_{\mathbf{x}}$ is the n-th local sample moment of $M(\mathbf{x})$ in a neighbourhood $\eta(\mathbf{x})$ around each pixel, defined as:

$$\langle M^n(\mathbf{x})\rangle_{\mathbf{x}} = \frac{1}{|\eta(\mathbf{x})|} \sum_{\mathbf{p}\in\eta(\mathbf{x})} M^n(\mathbf{p}). \tag{7.5}$$

When non-stationary noise is considered, the parameter σ_n^2 becomes $\mathbf{x}$-dependent, and it must be replaced in Eq. (7.3) and Eq. (7.4) by $\sigma_n^2(\mathbf{x})$.

The function $K(\mathbf{x})$ in Eq. (7.3) can be seen as a confidence measure of how data fits the considered model. In those pixels where $K(\mathbf{x}) \to 1$ (in the edges of the image, for instance, where the local variance is high), the data is far from the model, and therefore the final image $\widehat{I}_0(\mathbf{x}) \to M(\mathbf{x}) - 2\sigma_n^2$. Since the model is not trusted, the output is just the data (with some bias removed). On the other hand, in those areas where $K(\mathbf{x}) \to 0$ (homogeneous areas, for instance), the model totally fits the data, and the best possible output is given by an unbiased version of the averaged data, i.e., $\widehat{I}_0(\mathbf{x}) \to \langle M^2(\mathbf{x})\rangle_{\mathbf{x}} - 2\sigma_n^2$. This $K(\mathbf{x})$ function will be later used to control the consensus procedure.

7.2 Proposed method: a consensus LMMSE

The starting point of our approach are MR images corrupted with non-stationary Rician noise, as those generated after a SENSE acceleration and reconstruction.

Our aim is to estimate the noiseless signal. We work with MR magnitude images that are affected by Rician noise. This noise can be spatially $I_0(\mathbf{x})$ out of the noise data. We will assume a simplified corruption model as described in Eq. (7.1). As previously stated, we will use the LMMSE estimator in Eq. (7.3) as filtering technique. We assume our inability to properly assess a $\sigma_n^2(\mathbf{x})$ map. Thus, we cannot initially calculate a value for $K(\mathbf{x})$ in Eq. (7.4), since it depends on $\sigma_n^2(\mathbf{x})$.

The solution proposed to overcome this issue is based on a consensus strategy to a pixel level introduced in Section 4.7: from a set of different input values of $K_i(\mathbf{x})$, $i = 1, \ldots, n$ we try to reach a consensus for a unique $K(\mathbf{x})$ value:

$$\begin{bmatrix} K_1(\mathbf{x}) \\ K_2(\mathbf{x}) \\ \vdots \\ K_n(\mathbf{x}) \end{bmatrix} \xrightarrow{\text{Consensus}} K(\mathbf{x}).$$

These different $K_i(\mathbf{x})$ are calculated using different configurations of the input parameter set, namely a σ_n^2 value and the size of the neighbourhood where the local moments are calculated, $W_{si} = |\eta_i(\mathbf{x})|$, see Eq. (7.5). Here $K(\mathbf{x})$ will be used as a pixel confidence: it gives a measure of how the data fits the model. Since we cannot make an initial correct estimation of $\sigma_n^2(\mathbf{x})$, different candidates $K_i(\mathbf{x})$ calculated with different σ_i^2 values will contribute to the final decision.

A scheme is presented in Figure 7.1 where a complete overview of the method is given. The whole consensus-based algorithm is as follows:

1. In a *preliminary phase*, a set of confidence matrices $\{K_i\}_{i=1}^n$ is calculated by using Eq. (7.4) with different values for the noise variance (σ_i^2) and the neighbourhood size (W_{si}). A reference set $\{\sigma_i^2\}_{i=1}^n$ can be built from an initial reference variance. For instance, a reference variance can be estimated using any noise estimator already existing in the literature [4, 5]. This estimation is done assuming a single σ_n^2 value for the whole image, which will not be accurate for all pixels, but it gives a global reference value. A set of multiple $\{\sigma_i^2\}_{i=1}^n$ can be obtained by sampling an inter-percentile interval around the estimated value. Other strategies can be also adopted when some information on the underlying variance is known.

2. For a *aggregation phase*, a set of aggregation functions merges all the information from $\{K_i\}_{i=1}^n$. Then a set of aggregated confidence matrices

$\{Agg_j\}_{j=1}^{k}$ is generated by applying OWA operators with different weighting vectors. (These OWA operators are explained further in Section 4.4.1). A set of seven representative OWA operators was used, whose weighting vectors are depicted in Figure 7.2. Note that the weights distributions follow trapezoidal shapes with different tilt grades. They mainly give higher weights to the lower values of the sorted input. This way, the output of the OWA operator provides a higher confidence value when the majority of candidates agree. There are also null weights that correspond to the input omission.

3. In the *exploitation phase*, to build $K_{\text{final}}(\mathbf{x})$, we select the Agg_j that best suits and less disagrees with respect to the initial $\{K_i\}_{i=1}^{n}$. In order to help in this issue the Agg_j is calculated by minimizing the penalty-based function (introduced in Section 4.5):

$$K_{\text{final}}(\mathbf{x}) = \text{argmin}_{\text{Agg}_j} \sum_{i=1}^{n} |K_i(\mathbf{x}) - Agg_j(\mathbf{x})|.$$

4. Finally, in the *estimation phase*, to get the MR magnitude image, the LMMSE estimator from Eq. (7.3) has to be applied using the confidence estimation $K_{\text{final}}(\mathbf{x})$ and a spatial variance estimation $\widehat{\sigma_n^2}(\mathbf{x})$. The $\widehat{\sigma_n^2}(\mathbf{x})$ is calculated isolating the variable from Eq. (7.4) and using the input $K_{\text{final}}(\mathbf{x})$ as shown in Eq. (7.6).

$$\widehat{\sigma_n^2}(\mathbf{x}) = \frac{\langle M^2(\mathbf{x})\rangle_{\mathbf{x}} - \sqrt{\langle M^2(\mathbf{x})\rangle_{\mathbf{x}}^2 - (1 - K_{\text{final}}(\mathbf{x})) \cdot (\langle M^4(\mathbf{x})\rangle_{\mathbf{x}} - \langle M^2(\mathbf{x})\rangle_{\mathbf{x}}^2)}}{2}. \tag{7.6}$$

7.3 Experiments and discussion

Different experiments are carried out to illustrate the behaviour of the MRI noise reduction approach proposed in Section 7.2 to deal with the non-stationary noise created by SENSE. For it, we first introduce the used materials, to follow with the experiments. Two databases are used for the experiments, where one is contaminated by different non-stationary noise patterns. Moreover, our approach

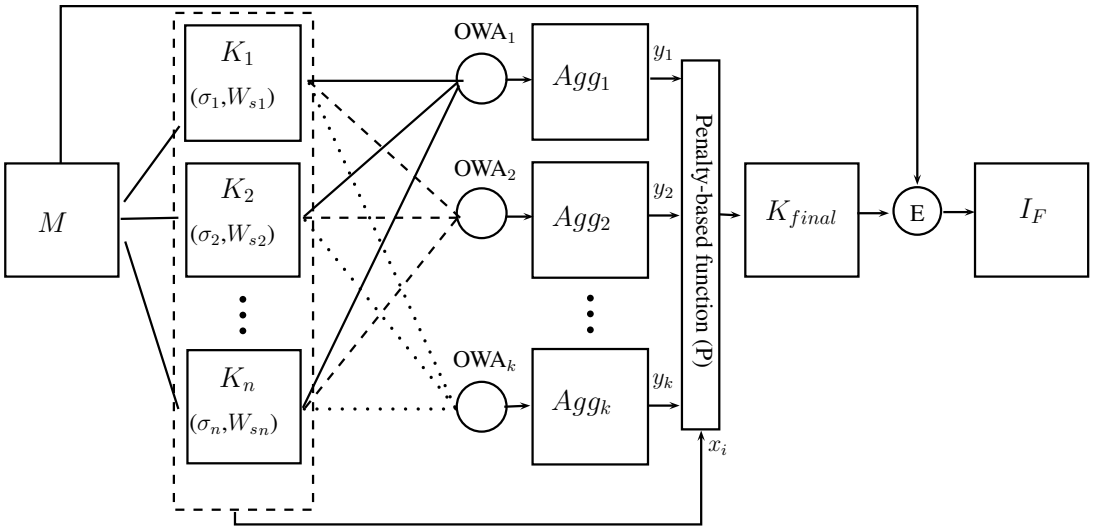


Figure 7.1: Proposed scheme for filtering of non-stationary noise using a LMMSE estimator for Rician noise. A consensus approach for multiple inputs as a function of $K(\mathbf{x})$ is considered.

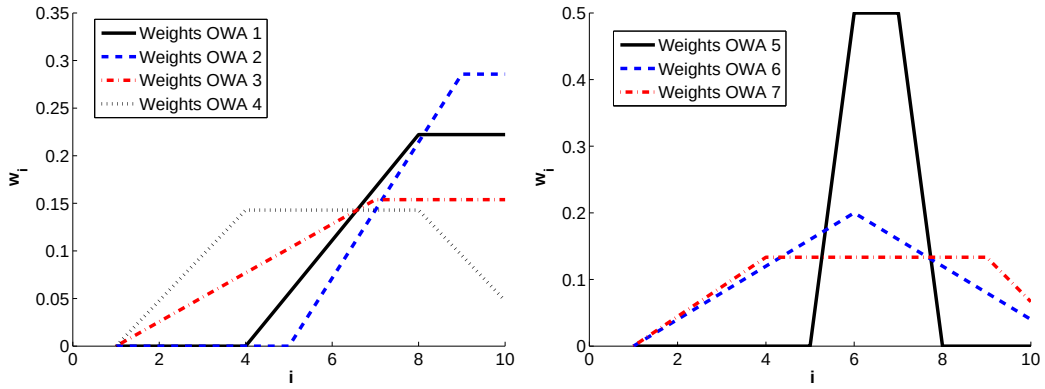
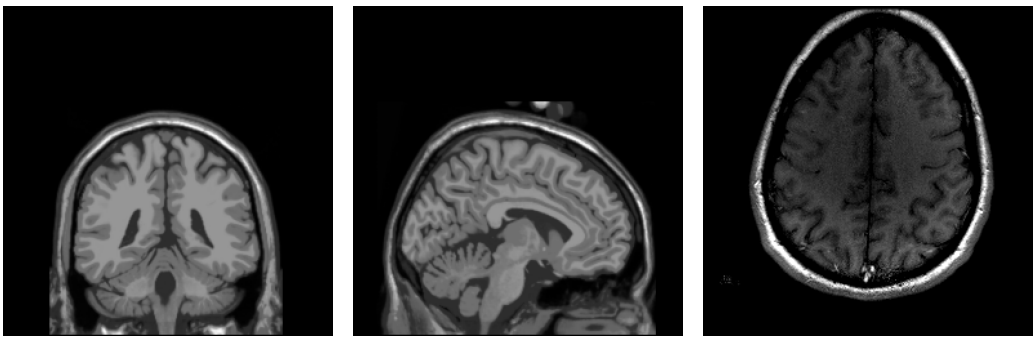


Figure 7.2: Weighting quantification for the 7 used OWA operators considering 10 elements.

is compared to other methods using several similarity measures that measures their performance. Then, in the different experiments we show the effectiveness of the proposed approach.

7.3.1 Materials and methods

We tested the proposed method with two different data sets as it is shown in Figure 7.3: (1) Synthetic noise-free MR slices from the BrainWeb data set [31];



(a) Coronal (512 × 512) (b) Sagital (512 × 512) (c) *In vivo* (256 × 256)

Figure 7.3: MRI slices used in the experiments. Images (a) and (b) come from the BrainWeb dataset; (c) is a real *in vivo* acquisition from a multi-coil GE Signa 1.5T EXCITE.

(2) one *in vivo*[1] T1 MR magnitude image acquired in a GE Signa 1.5T EXCITE, FSE pulse sequence, 8 coils, TR=500ms, TE=13.8ms, image size 256 × 256 and FOV: 20cm×20cm.

To simulate SENSE-like noise, the synthetic images were corrupted with non-stationary noise following the model in Eq. (7.1) with four different spatial patterns, $\mathcal{G}(\mathbf{x})$, shown in Figure 7.4. The noise variance is calculated from this pattern for different signal-to-noise ratio (SNR) simply by a linear scaling:

$$\sigma_n(\mathbf{x}) = \sigma_0 + \mathcal{G}(\mathbf{x}) \cdot \sigma_1.$$

The different patterns used are:

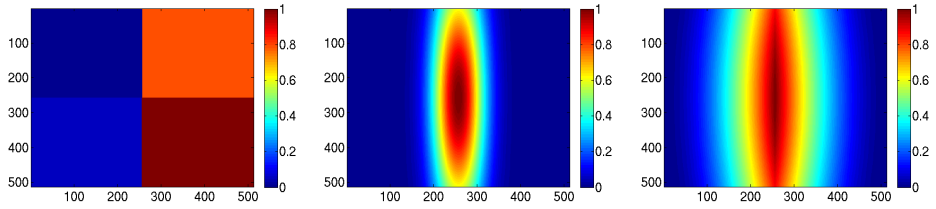
1. An unrealistic highly variant synthetic noise pattern, Figure 7.4(a). Although it is very unlikely that a pattern like this occurs in real acquisition, this 4-section scheme will give a very good insight of the behaviour of the filtering schemes.

2. A synthetic Gaussian-shaped noise pattern, Figure 7.4(b). This pattern follows the shape of some real patterns in SENSE acquisitions [8].

3. A noise shape generated with a SENSE simulator: Figure 7.4(c). This is the reconstruction from a sensitivity map belonging to 8-coils scheme as shown in Figure 7.5.

[1]Image provided by Doctor W. Scott Hoge from the LMI, Brigham and Womens Hospital, Boston.

4. In order to verify the behaviour of the method in presence of stationary noise, a fourth pattern is proposed, assuming $\mathcal{G}(\mathbf{x})$ to be a constant, i.e. $\sigma_n^2(\mathbf{x}) = \sigma_n^2$.



(a) Extreme (b) Slim Gaussian (c) SENSE

Figure 7.4: Non-stationary noise patterns used with the synthetic MR images. $\mathcal{G}(\mathbf{x})$ range is $[0, 1]$. It was scaled to obtain images with several SNRs.

Figure 7.5: Sensitivity map used for the experiments.

From the spatial pattern $\mathcal{G}(\mathbf{x})$, the variance $\sigma_n^2(\mathbf{x})$ is finally obtained for different SNRs. For the experiments, a neighbourhood size $W_s = [7 \times 7]$, and a range of 10 different central values for σ_n^2 are considered. They are calculated as elements of a range that varies around an initial variance $\widehat{\sigma_n^2}$. The minimum is between (0.5625-0.7225) times $\widehat{\sigma_n^2}$, as well as the maximum is between (1.44-1.69) times $\widehat{\sigma_n^2}$. In other words, the minimum is between (0.75-0.85) times $\widehat{\sigma_n}$, and the maximum is between (1.2-1.3) times $\widehat{\sigma_n}$. The initial variance is calculated as $\widehat{\sigma_n^2} = 4/9 \cdot \overline{\sigma_n^2(\mathbf{x})}$, when $\sigma_n^2(\mathbf{x})$ is available. Otherwise, $\widehat{\sigma_n^2}$ is estimated from the data. We also apply this initial variance as an input parameter of the filters used.

Our approach was compared with the following state-of-the-art Rician-based filtering schemes:

- The *Original LMMSE estimator (Original LMMSE)* as proposed in [4], assuming a single σ_n^2 value for the whole image. A 7×7 square window is used for the sample moments estimation.

- The *Non-local mean (NLM) algorithm without the Rician bias*, as proposed in [72] (Rice NLM). The essence of the NLM algorithm consists of a weighted

average that considers the distance and intensity between the target pixel and all observed pixels. The original idea was proposed by Buades *et al.* [20] for Gaussian noise. The required parameters for this approach are the radio search window ($R_{search} = 11$); the radio similarity window ($R_{sim} = 3$); the degree of filtering ($f = 1.2 \cdot \widehat{\sigma}_n$) and an estimation of the variance ($\widehat{\sigma_n^2}$).

- The *Chi-square unbiased risk estimator (CURE)*, as proposed in [71]. It considers the squared-magnitude magnetic resonance image data to derive an unbiased expression for the expected mean-squared error to remove noise, which are well modeled as independent non-central chi-square random variables on two degrees of freedom. The task is done in the wavelet-domain for its compromise between the execution speed and performance. It uses the *unnormalized Haar wavelet transform (Haar CURE)*, where each wavelet subband is treated independently. The other required parameter is the variance estimation, $\widehat{\sigma_n^2}$.

The restoration performance was quantified by using different similarity measures from the ones introduced in Section 2.4. Specifically, the mean square error (MSE), the structural similarity index (SSIM) and the quality index based on local variance (QILV) were used. The former one is simple to calculate and gives a measure of how pixelwise similar two images are. Though it does not take into account any structural information. It is not bounded; a higher MSE means worse quality. On the other hand, the SSIM index and the QILV give a measure of the structural similarity between the ground truth and the estimated image. However, the SSIM is more susceptible to the noise level in the image and the QILV to any blurring in the edge. Both measures are bounded in $[0, 1]$; the closer to one, the better the image. Moreover, all the measures are only applied on areas of interest in the image, this means that the background is excluded.

7.3.2 Experiments with synthetic data

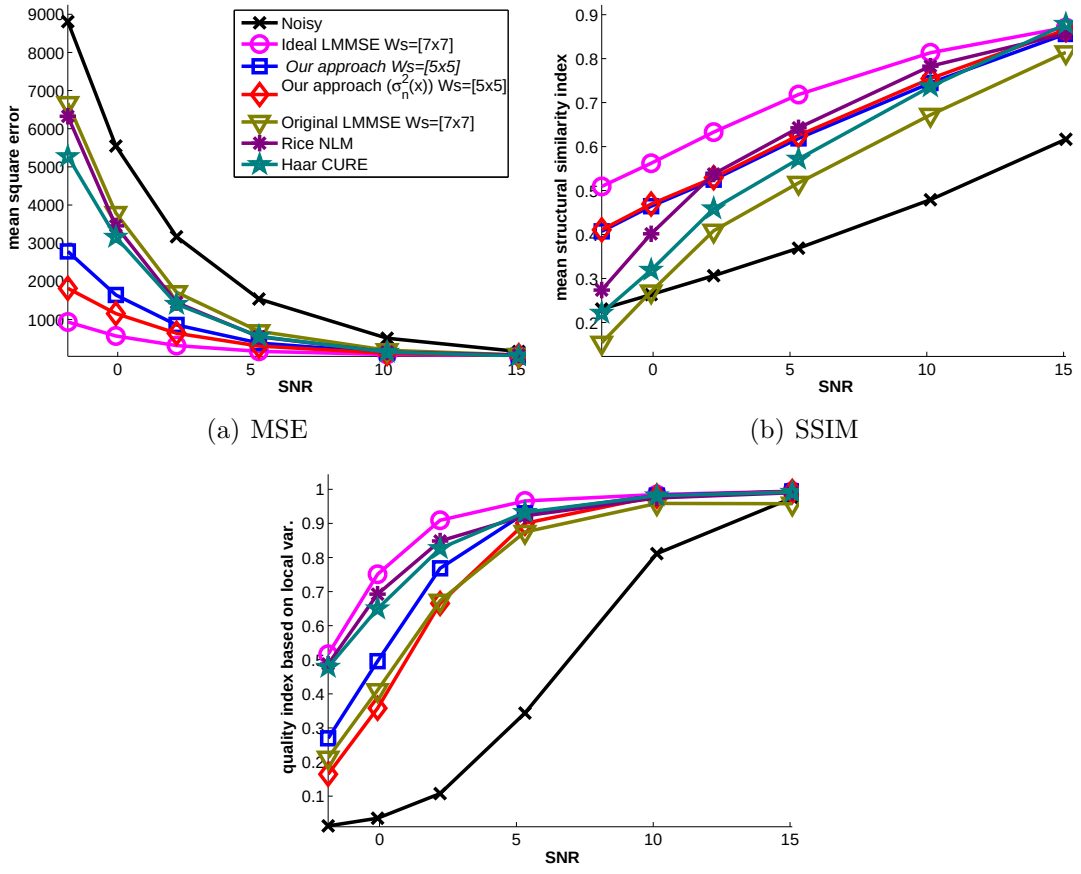
In order to show the relevance of our model, we compare our approach with three other methods: the original LMMSE, Rice NLM and Haar CURE; as well as with the ideal LMMSE estimation making use of the actual $\sigma_n^2(\mathbf{x})$ (Ideal LMMSE). Furthermore, we also compare our approach for the case that the LMMSE estimation (Eq. (7.3)) uses the actual $\sigma_n^2(\mathbf{x})$ instead of the estimated sigma $\widehat{\sigma_n^2}(\mathbf{x})$ from

Eq. (7.6) (Our approach $\sigma_n^2(\mathbf{x})$). For these two last estimations of the LMMSE, the windows size (W_s) chosen varies between $[5 \times 5]$ and $[7 \times 7]$. Each experiment was repeated 100 times to ensure a significant statistical analysis.

The first experiment evaluates the behaviour in an extreme case with an unreal noise shape. Figure 7.6 reveals how the SNR is highly affected with the increasing noise. Results show the lower performance of other methods in situations with very variant noise and low SNR, while our approach is able to overcome situations and highly improving the results of the original LMMSE. The results are comparable with the ideal case and with our approach using the actual $\sigma_n^2(\mathbf{x})$, where for the SSIM measures both cases are identical. Although our approach presents worse results for the QILV measure, we can appreciate that it complements the SSIM measure. Moreover, other approaches with better QILV and worse SSIM are over-filtered, as Figure 7.7 shows. The results for the three measures also manifest the convergence to similar results as the SNR increases.

In the second experiment, we selected a noise shape that approaches the SENSE shape contaminating the image. In this case, the SNR is less affected by the noise range, mainly due to the effect of a large image part where the noise is almost non-existent. In Figure 7.8 better results of our approach are observed when compared to the rest of the approaches and how even for this case, the different measures for our approach are again almost equivalent to the ideal case. Although the QILV is slightly worse than the ideal case for low SNRs. For this last measure, the original LMMSE curiously gets worse results as the SNR increases, what affects the sharpening of the edges. This behaviour can be due to the selected windows size and the remaining noise, although the magnitude is insignificant (less than two hundredths). Moreover, we can also appreciate how in this case the Rice NLM presents worse results than the original LMMSE. However, as the SNR increases, the Haar CURE becomes comparable to our approach (around 17.5 dB), while, in all the cases, our approach improves the original LMMSE. In Figure 7.9 we present the images for one of the executions for the different filters where we can observe how our approach better preserves the details.

The next experiment was performed on images contaminated by a SENSE reconstruction simulator that simulates a 8-coils acquisition scheme with equal correlation between coils ($\rho = 0.25$). The used sensitivity map belonging to the 8-coils system is shown in Figure 7.5. The results from Figure 7.10 exhibit



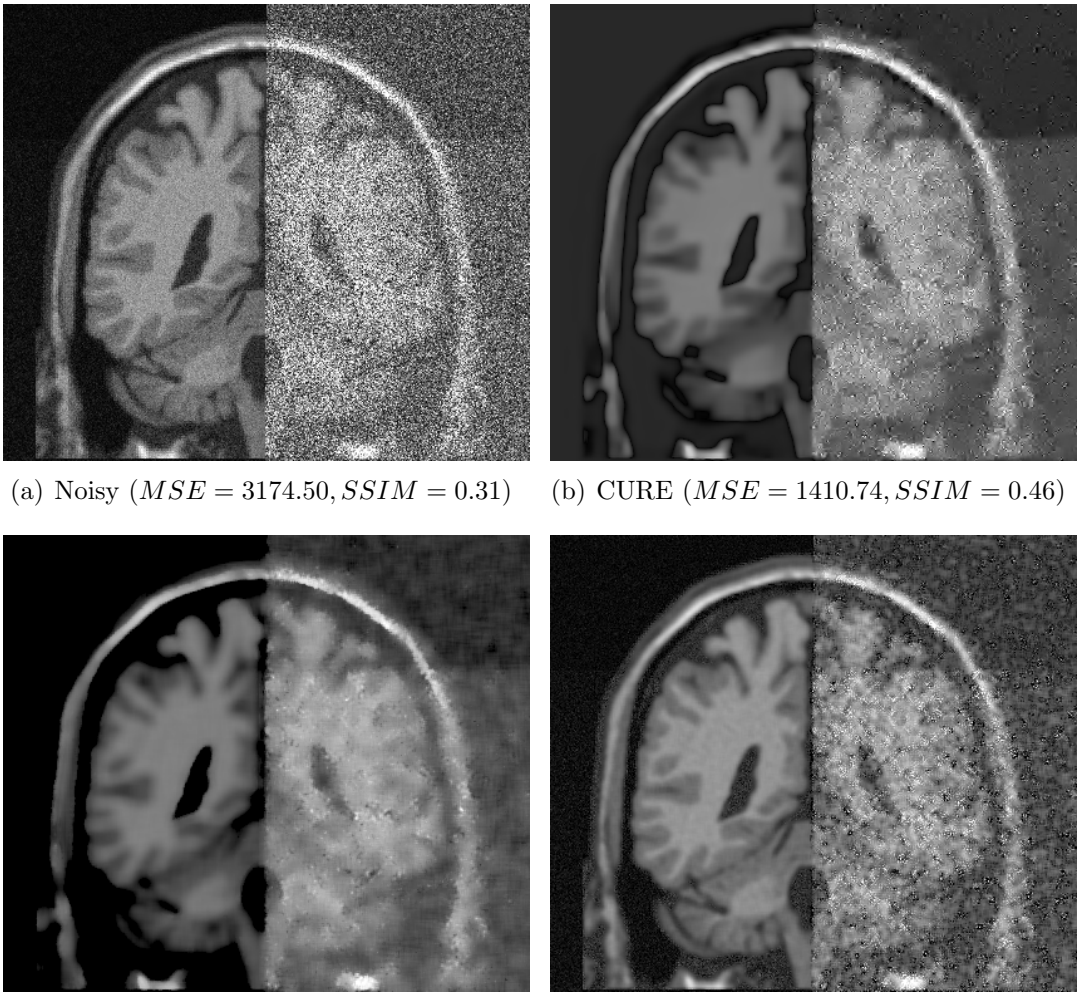
(a) MSE

(b) SSIM

(c) QILV

Figure 7.6: Results obtained from the synthetic MR magnitude image from Figure 7.3(a) with different SNRs. Each case was launched 100 times for the extreme noise shape (Figure 7.4(a)).

that there is no significant difference between the approaches for the MSE, although our approach obtains the worst performance for low SNRs. Nonetheless, the SSIM shows some differences among them. We can clearly appreciate that CURE outperforms the rest of the approaches. Though considering our maximal expectations with respect to the ideal case, we can conclude that our approach is equivalent to it. Moreover, the QILV measures neither show a significant difference, except for the original LMMSE, that as in the previous case, it curiously gets worse results as the SNR increases, although it is insignificant (less than two hundredths). In general, our approach shows a similar performance to the ideal case, or the original LMMSE except for the QILV measure where our approach improves. This behaviour may be because the noise range is not wide enough to show the benefits of our approach. Then the image is treated as an image with



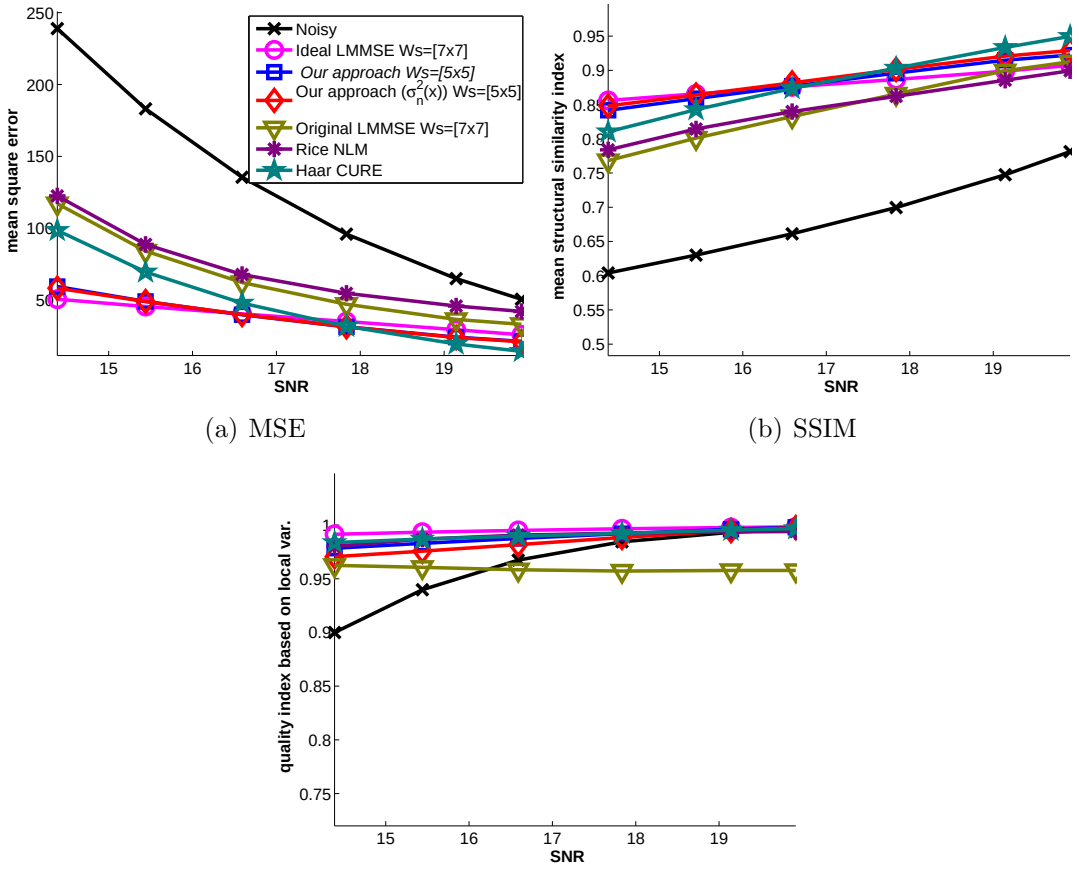
(a) Noisy ($MSE = 3174.50, SSIM = 0.31$) (b) CURE ($MSE = 1410.74, SSIM = 0.46$)

(c) Rice NLM ($MSE = 1440.85, SSIM = 0.54$) (d) Our appr. ($MSE = 874.30, SSIM = 0.52$)

Figure 7.7: Different image results obtained from the synthetic MR magnitude image from Figure 7.3(a) with the filters CURE, Rice NLM and our approach for the extreme noise shape from Figure 7.4(a) with a SNR = 2.21.

non-spatial distributed noise, since they behave as the original LMMSE. On the other side, it is important to note that, as the SNR increases, the performance differences decrease between the algorithms. In other words, we obtain the best result we can get using the selected estimator. In Figure 7.11 we may observe the equivalency of the methods from their image results.

The last experiment with synthetic data shows how our approach is also able to properly deal with images that contain stationary noise. To this end, we make use of different stationary noise shapes that obtain images with different



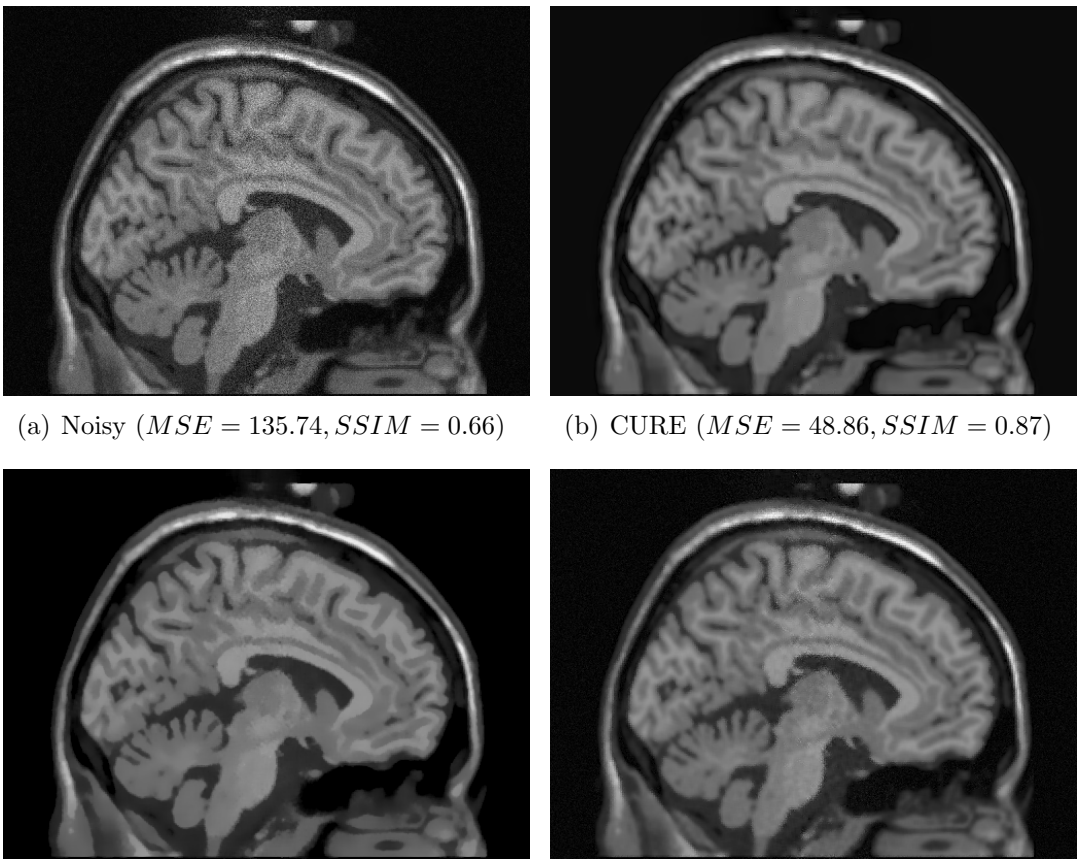
(a) MSE

(b) SSIM

(c) QILV

Figure 7.8: Results obtained from the synthetic MR magnitude image from Figure 7.3(b) with different SNRs. Each case was launched 100 times for the slim Gaussian noise shape (Figure 7.4(b)).

SNRs ($\sigma_n^2(\mathbf{x}) = \sigma_n^2$)). The results from Figure 7.12 manifest how all approaches, including our approach, obtain almost similar and coherent results. However, our approach is slightly affected by the noise introduced by our technique as we are using a range of $\sigma^2(\mathbf{x})$. But in any case, it shows a similar behaviour to the original LMMSE, that this time overlaps with the ideal case, as they are using the same $\sigma^2(\mathbf{x})$. We should take into account that the original LMMSE is the best result we can afford with our approach. Despite the SSIM performance is the worst for some cases, it also improves in some others. Possibly due to the selected windows size. For the QILV measure, it presents a similar behaviour to the two previous approaches, where for the original case it also decreases as the SNR increases. Besides that, the Haar CURE obtains the best performance. In Figure 7.13 we show the images for the different filters for one of the executions.

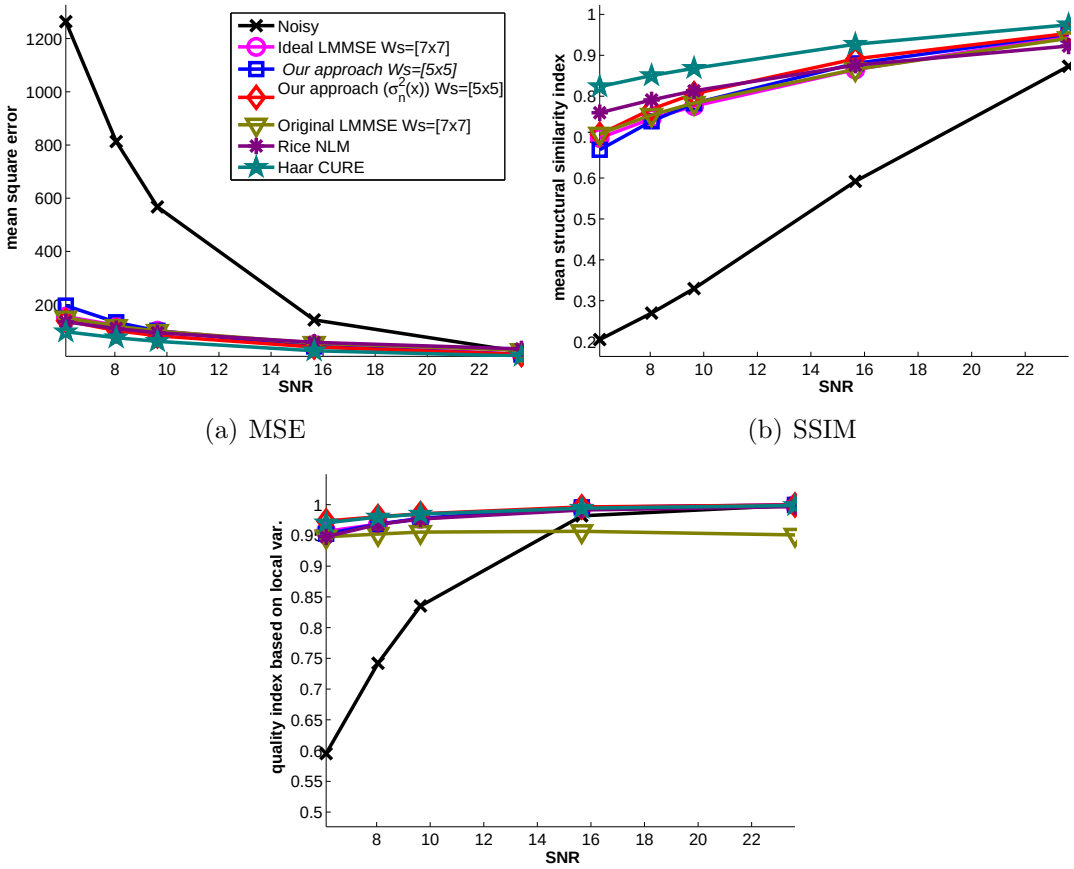


(a) Noisy ($MSE = 135.74, SSIM = 0.66$) (b) CURE ($MSE = 48.86, SSIM = 0.87$)

(c) Rice NLM ($MSE = 67.97, SSIM = 0.84$) (d) Our appr. ($MSE = 40.02, SSIM = 0.88$)

Figure 7.9: Different image results obtained from the synthetic MR magnitude image from Figure 7.3(b) with the filters CURE, Rice NLM and our approach for the slim Gaussian noise shape from Figure 7.4(b) with a SNR = 16.59.

Besides showing that our approach presents a good behaviour in situations with a variant non-stationary noise. It also exhibits good running times. In Table 7.1 we present the average time from the hundred executions for each algorithm. The original LMMSE has the best one, closely followed by our approach. Therefore, we can extract that our approach does not overload the running times despite making several runnings of the same filter. Although we should bear in mind that this running time will depend on the selected filter. On the other side, the Rice NLM obtains high running times with respect to the rest of the filters.



(a) MSE

(b) SSIM

(c) QILV

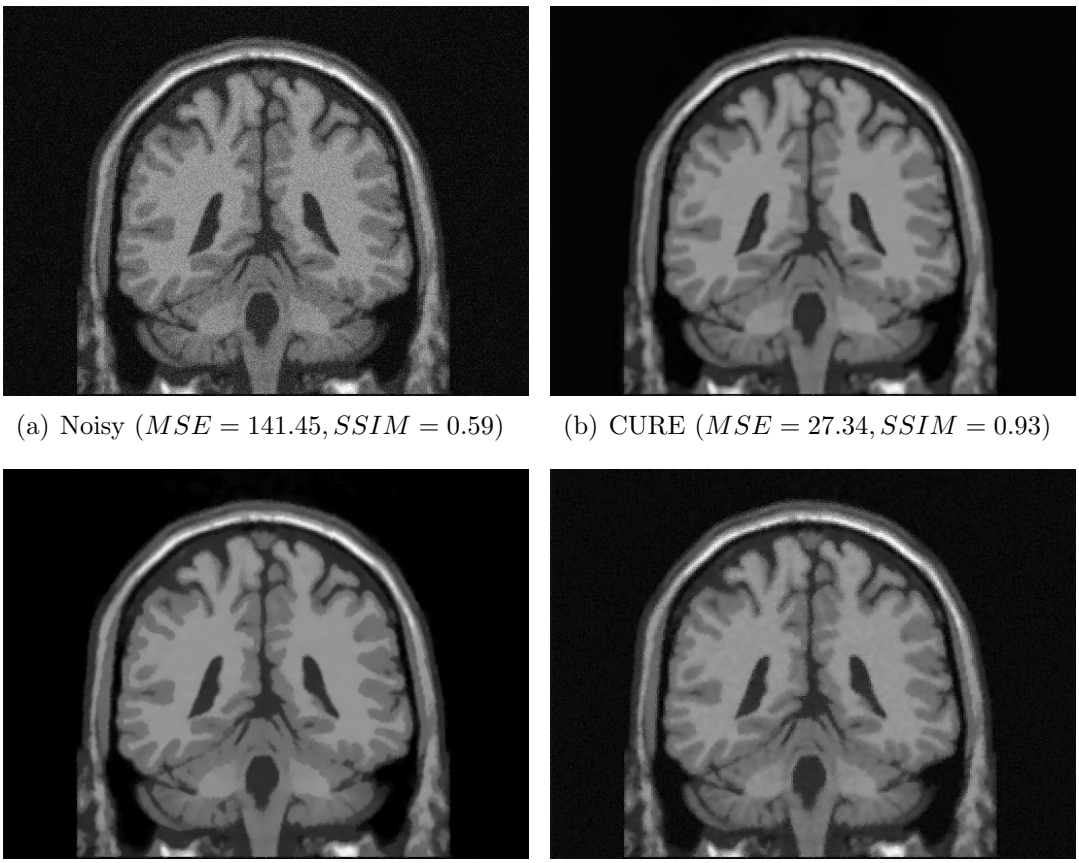
Figure 7.10: Results obtained from the synthetic MR magnitude image from Figure 7.3(a) with different SNRs. Each case was launched 100 times for the SENSE reconstruction simulator that generates the shape noise from Figure 7.4(c).

Experiment	Figure 7.6	Figure 7.8	Figure 7.10	Figure 7.12
Original LMMSE	0.061 ms	0.072 ms	0.060 ms	0.064 ms
Our approach	1.302 ms	1.405 ms	1.306 ms	1.327 ms
Haar CURE	5.828 ms	6.710 ms	5.727 ms	6.487 ms
Rice NLM	87.106 ms	96.602 ms	80.173 ms	85.123 ms

Table 7.1: Average running times for the hundred executions of the different algorithms and different experiments carried out.

7.3.3 Experiments with real data

In order to test the proposed method with real data, we use the real multicoil *in vivo* acquisition in Figure 7.14. For simplicity, the fully sampled **k**-space has been acquired, and the sensitivity map has been estimated for each of its 8 coils.



(a) Noisy ($MSE = 141.45, SSIM = 0.59$) (b) CURE ($MSE = 27.34, SSIM = 0.93$)

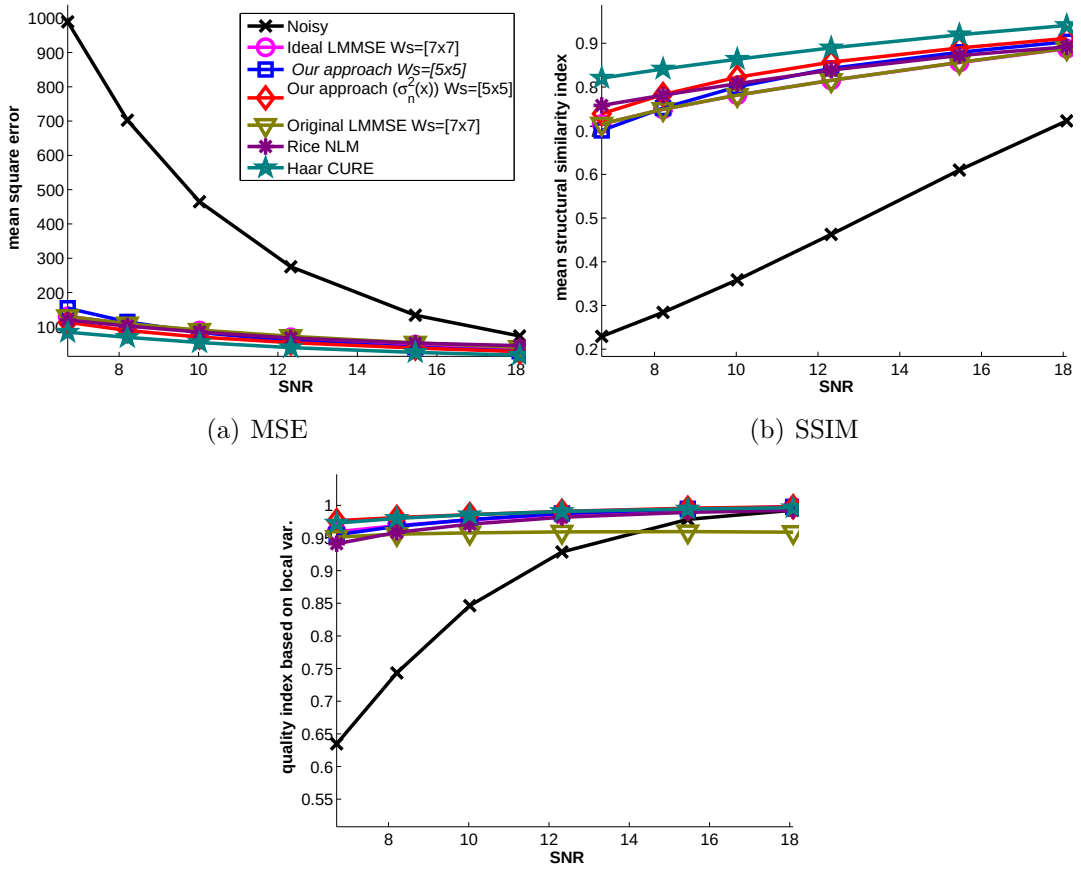
(c) Rice NLM ($MSE = 58.74, SSIM = 0.88$) (d) Our appr. ($MSE = 42.75, SSIM = 0.88$)

Figure 7.11: Different image results obtained from the synthetic MR magnitude image from Figure 7.3(a) with the filters CURE, Rice NLM and our approach for the SENSE reconstruction simulator that generates the shape noise from Figure 7.4(c) with a SNR = 15.66.

The data in each coil was subsampled to simulate a 2x acceleration, and the final magnitude image has been reconstructed using an offline SENSE algorithm. Since the initial $\sigma_n^2(\mathbf{x})$ is not available for this image, a prior estimation is done assuming stationary noise [4] as

$$\widehat{\sigma_n^2} = \text{mode}\{\langle M(\mathbf{x})^2 \rangle_{\mathbf{x}}\}.$$

The $\{\sigma_m^2\}_{m=1}^{10}$ elements are selected from the range $[0.0001 \cdot \widehat{\sigma_n^2}, 0,0009 \cdot \widehat{\sigma_n^2}]$ for its good performance, and in accordance with the range data that belongs to $[2.507 \cdot 10^{-4}, 3.0138]$. In Figure 7.14 it is shown how the consensus LMMSE and the Haar CURE obtain the best results among the restored images. Although consensus LMMSE slightly obtains better results in homogeneous areas than Haar



(a) MSE

(b) SSIM

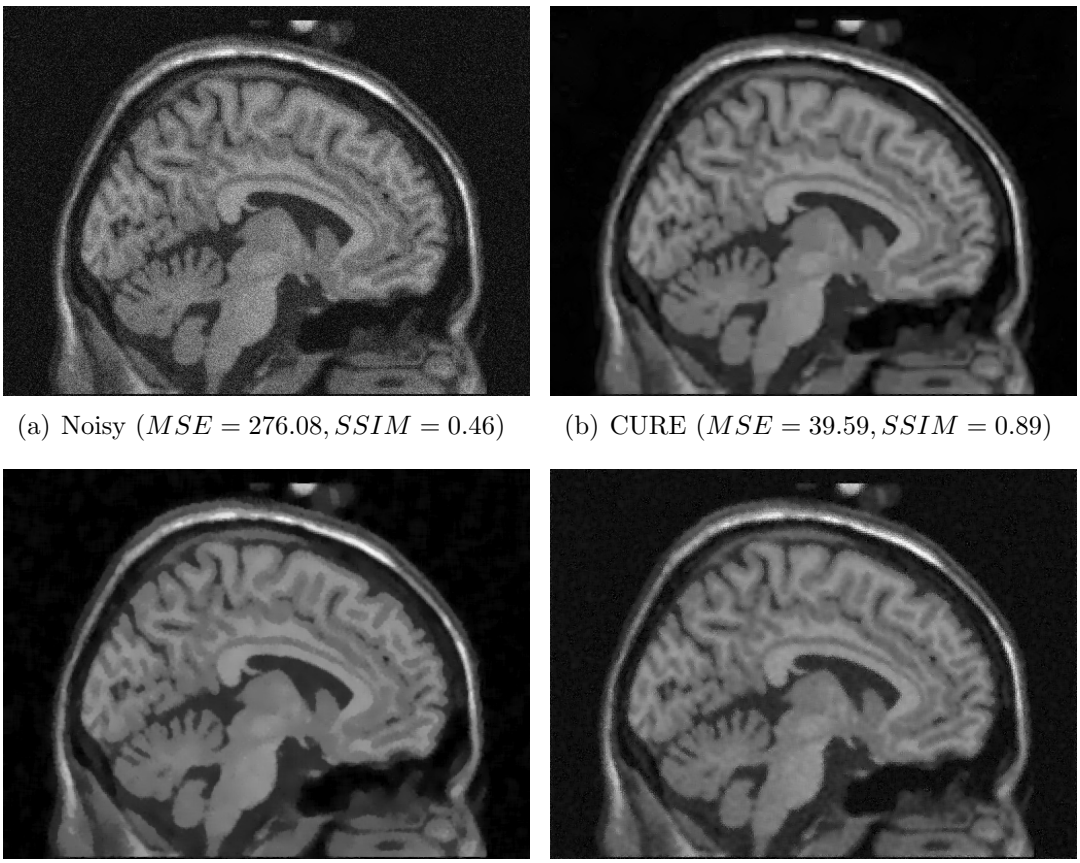
(c) QILV

Figure 7.12: Results obtained from the synthetic MR magnitude image from Figure 7.3(b) with different SNRs. Each case was launched 100 times for a stationary noise.

CURE. On the other hand, the restored Rice NLM image still keeps a lot of noise, while the original LMMSE removes more noise in exchange to blur the image (this effect is emphasized close to the borders). On the contrary, our consensus approach based on the original LMMSE does not exhibit this problem anymore.

7.3.4 Spatial variance distribution estimation

The proposed methodology not only improves the image quality. It also estimates the spatial variance distribution using Eq. (7.6) and the reached $K_{\text{final}}(\mathbf{x})$. We can analyze the behaviour of our approach using synthetic images, so we can compare the estimation with respect to the real introduced degradation. In Figure 7.15, different noise maps with their respective original noise degradations are



(a) Noisy ($MSE = 276.08, SSIM = 0.46$) (b) CURE ($MSE = 39.59, SSIM = 0.89$)

(c) Rice NLM ($MSE = 68.19, SSIM = 0.84$) (d) Our appr. ($MSE = 59.93, SSIM = 0.84$)

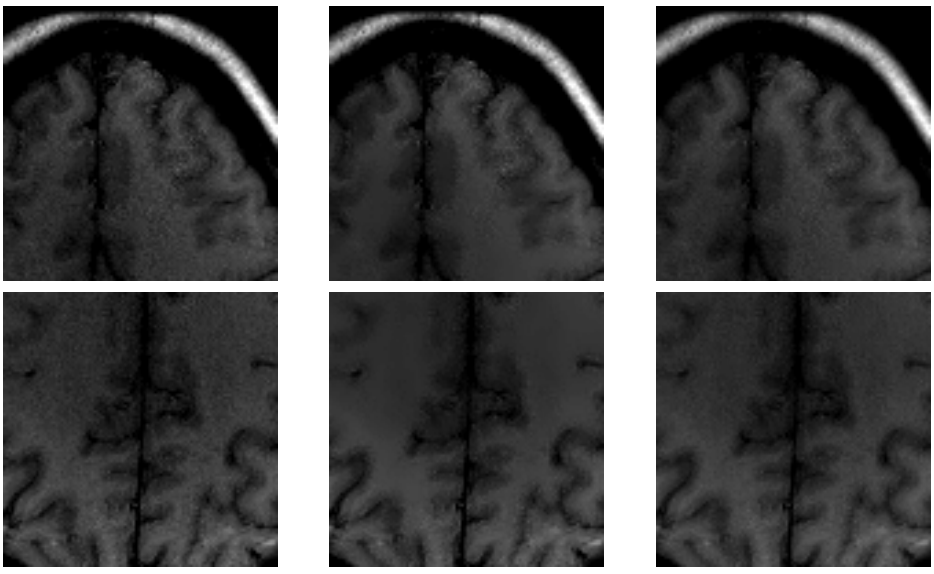
Figure 7.13: Different image results obtained from the synthetic MR magnitude image from Figure 7.3(b) with the filters CURE, Rice NLM and our approach for the case of stationary noise with a SNR = 10.02.

shown. These are obtained using the MR magnitude image from Figure 7.3(a) and Figure 7.3(b). We can observe that the estimation follows the noise pattern, as well as it detects the image borders.

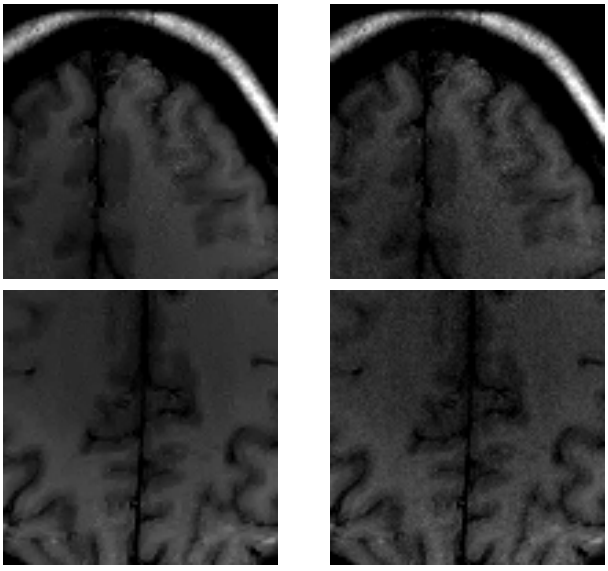
7.3.5 Experiments with NLM

The idea of using a consensus approach when input parameters are unknown is not just restricted to the LMMSE estimator. The methodology can easily be adapted to other filtering schemes. As an illustration, we will use it together with a Rician NLM. The original unbiased NLM scheme is defined as

$$\widehat{I}_0(\mathbf{x}) = \sqrt{\mathrm{NLM}(M^2(\mathbf{x})) - 2\sigma_n^2}.$$



(a) Original (b) Our approach (c) Original LMMSE



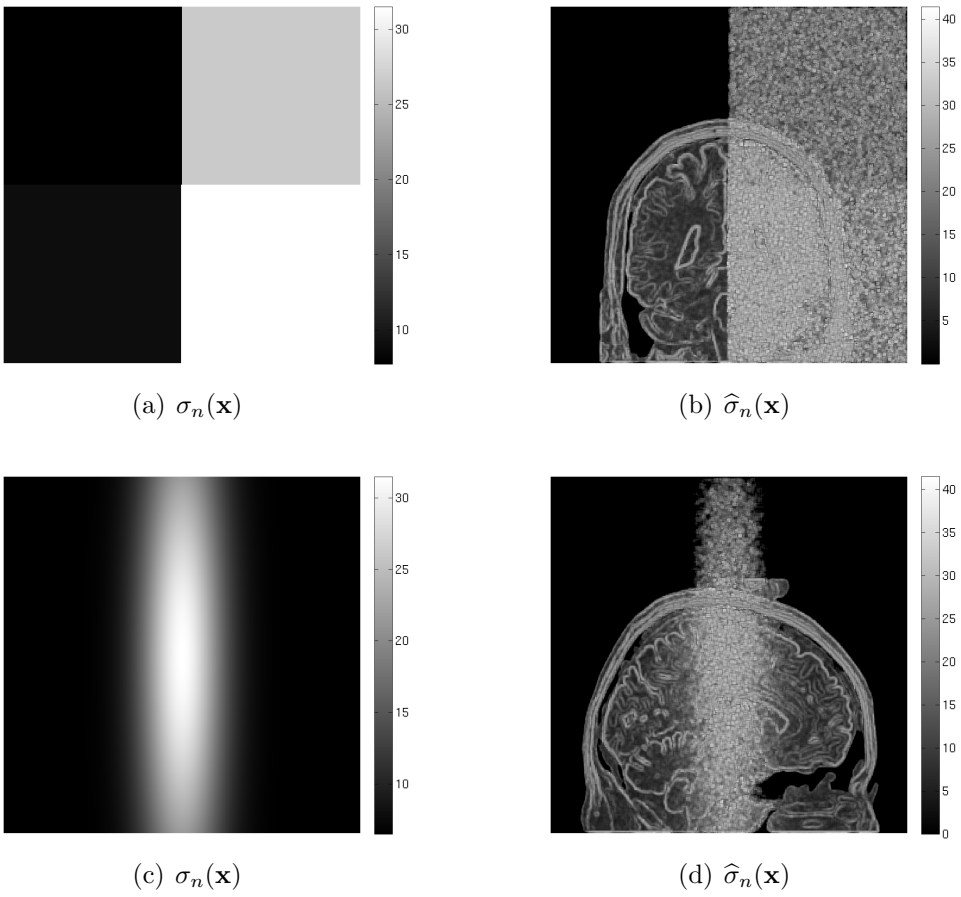
(d) Haar CURE (e) Rice NLM

Figure 7.14: Dynamic range regions extracted from the filtered images using the real data from Figure7.3(c).

If the single σ_n^2 value is replaced by the $\widehat{\sigma_n^2}(\mathbf{x})$ estimation given by Eq. (7.6), we can rewrite it as

$$\widehat{I_0}(\mathbf{x}) = \sqrt{\mathrm{NLM}(M^2(\mathbf{x})) - 2\widehat{\sigma_n^2}(\mathbf{x})}.$$

The synthetic experiment for the image in Figure 7.3(b) for different SNRs is repeated only for the different NLM schemes: Ideal NLM, that is the Rice NLM



(a) $\sigma_n(\mathbf{x})$ (b) $\widehat{\sigma}_n(\mathbf{x})$

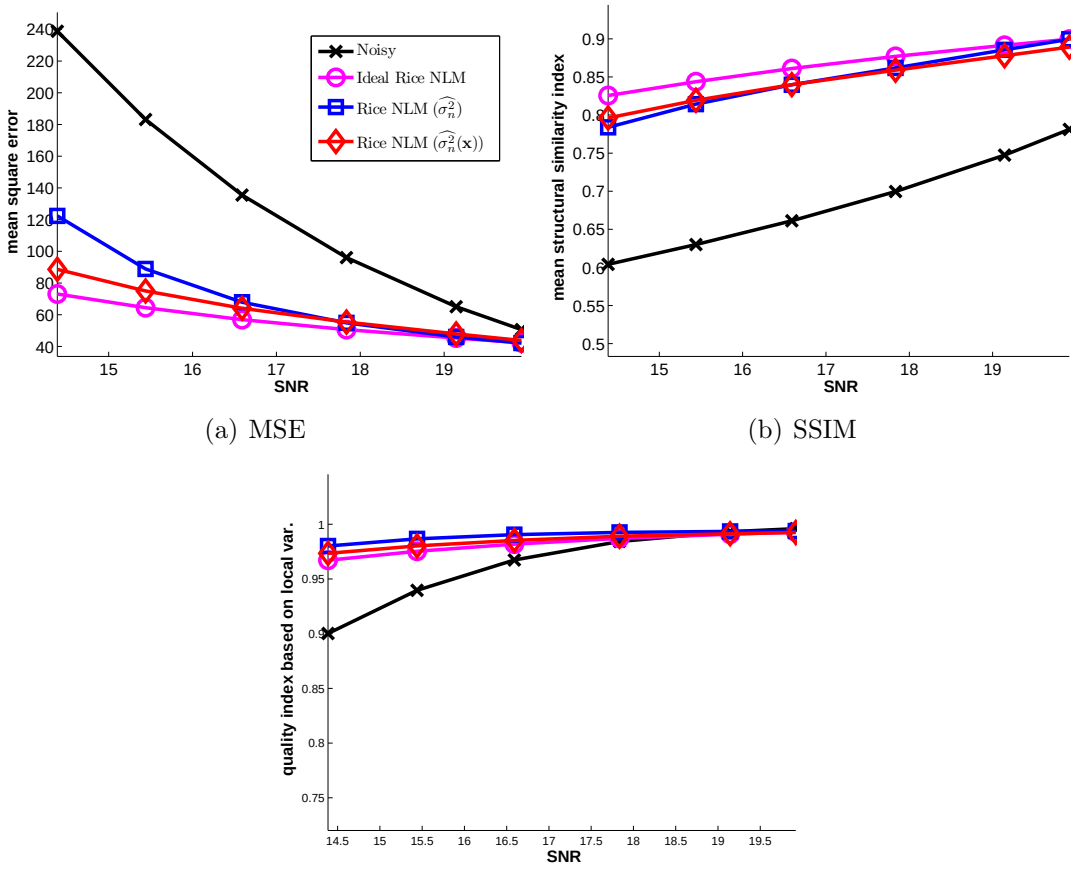
(c) $\sigma_n(\mathbf{x})$ (d) $\widehat{\sigma}_n(\mathbf{x})$

Figure 7.15: The noise maps $\sigma_n(\mathbf{x})$ compared with their respective estimation $\widehat{\sigma}_n(\mathbf{x})$ for the synthetic magnitude MRI from Figure 7.3(a) and Figure 7.3(b) using two different noise shapes and SNRs. Being the noise shapes from Figure 7.4(a), and Figure 7.4(b); and 10.13 and 14.39 the SNRs respectively.

using the actual $\sigma_n^2(\mathbf{x})$; Rice NLM using the estimated initial variance $\widehat{\sigma_n^2}$; and the proposed approach Rice NLM using our estimation $\widehat{\sigma_n^2}(\mathbf{x})$. Results from Figure 7.16 show that our adaptation improves significantly the MSE for low SNR images with respect to the original NLM, while it approaches the ideal case and preserves the behaviour for the SSIM. For the QILV the results slightly improve. Moreover, the different approaches tend to merge as the SNR increases.

7.4 Conclusions

A new methodology is presented as a solution to noise filtering when the input image shows a spatially variant noise pattern, and some of the input variables



(a) MSE

(b) SSIM

(c) QILV

Figure 7.16: Results obtained from the synthetic MR magnitude image from Figure 7.3(b) with different SNRs using the NLM algorithm. Each case was launched 100 times for the slim Gaussian noise shape (Figure 7.4(b)).

cannot be properly estimated. Spatial non-stationary noise is a kind of noise whose features (the variance in this case) depends on the position within the image. The clearer example of this kind of noise in MR data can be found in pMRI acquisitions that uses SENSE as a reconstruction process, but not only. The proposed method is applied together with some existing filtering method. In this approach, the LMMSE signal estimator for stationary Rician noise is considered. This filter on its own will fail when applied over a spatially variant $\sigma_n^2(\mathbf{x})$, since it is intended for a single σ_n^2. However, the combination of the LMMSE with the proposed consensus decision-making approach is able to take into account the non-stationarity of the data. The algorithm also assumes our incapability to proper estimate the input data, in this case the map of noise and the optimal size of the window in which the local moments are calculated.

Results of the experiments done using synthetic and real data show how the proposed method highly improves the behaviour of the stationary LMMSE, and its performance is very similar to the optimal case assuming a non-stationary LMMSE with $\sigma_n^2(\mathbf{x})$ perfectly known. In many cases, the new approach even outperforms Rician filters that in the past have shown even a better performance than the LMMSE itself. The method is particularly useful in those cases when the variability of $\sigma_n^2(\mathbf{x})$ is high and extreme. That will depend on the position and calibration of the acquisition coils.

As we have previously stated, this philosophy of work can be easily extended to other filtering techniques in MRI. This extension will allow other algorithms to better cope with non-stationary noise, but not only. They can also be adapted to automatically select the better set of input parameters, or to cope with deviation from the statistical model, to perform a different filtering around important structures and edges or even to combine the results of different kind of filters into a single output.

The main drawback of the method is that the number of operations increases, since the method carries out a filtering procedure for each input set. The more the input possibilities, the greater the number of times the filtering is repeated. The good news here is that each of the iterations is totally independent of the others, and therefore the method can easily be highly parallelized.

IV

Conclusions, & future work

8

Conclusions & future work

8.1 Conclusions

This thesis shows that *penalty-based decision making* can be used for image restoration as it has been previously used in other image processing areas, such as segmentation or image reduction. In this way, we transform our problem into making a choice between a set of possible solutions, e.g. a set of different techniques that accomplish different actions, where we take the solution, or technique, that best adapts to our constraints or goals. In short, penalty-based decision making gives the chance to work with scenarios where we do not know beforehand which alternative is better to use.

Among all fuzzy decision methodologies, we focus on decision-making based on penalty functions because it allows to reach a consensus through an evaluation of all the inputs. Moreover, we also use aggregation functions, what allows to put in value all the inputs. Thus, the aggregation functions generate the set of solutions for our consensus methodology and hence already obtaining an agreement between the inputs. We decide to work with OWA operators because they provide flexibility in the weights definition. We can even use fuzzy quantifiers to calculate the weights, what gives the possibility to use human expressions such as 'at least half', 'most' and 'many'. Then, through penalty-based functions we select the aggregation function that presents minimum penalty between the set of aggregation functions and the original inputs. Furthermore, we introduce the theory behind penalty functions over cartesian product of lattices, what gives the possibility to apply penalty functions on small groups of pixels. Therefore

this definition brings the possibility to work with penalty functions in each region independently, where these regions can be of any shape, and for instance, share some characteristics. Although in the proposed consensus methodology, as we want to get the best aggregation function for each position, a cartesian product of lattices allows to define the pixel regions independently of the result. Then, in case we work with pixel regions or the complete image, before applying penalty functions we build a new set that combines all possible variations of the aggregation functions.

Due to the flexibility that presents the introduced consensus methodology in the aggregation and exploitation phase definition, this methodology can be seen as a *framework*, where it can easily be adapted to different problems. Therefore, we develop three different approaches to show that consensus methodology can be used for restoration of noisy images.

A first approach has been developed for the task of blind noise reduction. In such way that this approach finds a cooperation between different noise reduction methods to deal with an image contaminated by an undetermined noise distribution. This proposed approach is a good alternative in situations where we do not know which filter to use, or when a combination of different filters perform better than any single one. For it, we define the aggregation phase as a set of three OWA operators where their weights are defined using fuzzy linguistic quantifiers. Specifically, 'at least half', 'most of them', 'as many as possible'. Followed by the exploitation phase, where a penalty function takes the OWA operator that presents the minimum penalty with respect to the original filtered images. The use of this methodology obtains always a good performance and assures a result better than the worst of the individual solutions.

A new approach based on consensus is presented, offering the chance to use parametric restoration methods when any of the parameters cannot be properly estimated or the data does not strictly fit the underlying model. As an illustration we apply this methodology to the case of images corrupted with non-stationary Gaussian noise, where there is a wide range of uncertainty on the value of the noise variance. To overcome this problem we adapt the Wiener filter to obtain an output using a consensus procedure by selecting an aggregation function from a set of OWA operators by means of a penalty function. The experimental results show that consensus performance is similar to the case in which all the parameters are accurately known before.

Finally, another approach is presented as a solution to noise filtering when the input image shows a spatially variant noise pattern, and some of the input variables cannot be properly estimated. For it, we adapt the parametric LMMSE signal estimator for stationary Rician noise to take into account the non-stationarity of the data. In such a way that we can deal with the noise present in MR data acquired with pMRI and reconstructed with SENSE, but not only. Inspired by the previous approach, we propose a combination of the LMMSE for stationary Rician noise with the consensus methodology by selecting an OWA operator by means of a penalty-based function. Results of the experiments done using synthetic and real data show how the proposed method highly improves the behaviour of the stationary LMMSE, and is particularly useful in those cases when the variability is high and extreme.

In summary, we prove that consensus methodology is an alternative for image noise reduction, showing that it can be easily adapted to deal with the uncertainty present in different problems. For instance, the previously introduced approaches can be easily extended to work with other filters, as well as to use a different set of aggregation functions and a penalty function. What allows us to see this methodology as a framework. The main drawback of the proposed methodology is that the number of operations increases with the number of inputs and aggregation functions. Although we should keep in mind that increasing the number of aggregation functions, does not always mean an improvement in the results. Then, the challenge is to find a compromise on the number of aggregation functions to obtain a good performance. The good news here is the method is highly parallelized.

8.2 Future work

We introduce some of the possible applications of consensus methodology. However, this is just the beginning, as new applications can be found for image restoration. For instance, a straightforward extension is the use of consensus methodology to deal with the non-stationary noise present in MR data acquired by pMRI and reconstructed with GRAPPA. Moreover, the study of new aggregation functions can be also carried out, what opens the possibility to fine-tune the results to specific problems. As well as the use of new penalty functions, what allows to use new criteria. On the other side, we can also study the use

of different penalty functions over pixel regions, in such a way that we can accomplish different actions according to different goals or constraints in regions that share some characteristics. In short, we present a *consensus framework* that offers many possibilities yet to be studied.

8.3 Publications related to this work

Below, all publications resulting from this work are listed. These were carried along the research, between 2011-2014, and corresponding to the proposed applications.

1. L. González-Jaime, G. Vegas-Sánchez-Ferrero, M. Nachtegael, E. E. Kerre, and S. Aja-Fernández. Spatially variant noise filtering in MRI. A Consensus approach. *Magnetic Resonance Imaging*, 2014

    - Status: **Submitted**.
    - Type: Journal Article.
    - Journal Title: *Magnetic Resonance Imaging*.
    - Impact Factor (JCR 2013): 2.022.
    - Subject Category: Radiology, Nuclear Medicine & Medical Imaging. Ranking 47 / 122 (Q2).
    - Correspondence with: Chapter 7.

    Abstract: In order to accelerate the acquisition process in multiple-coil MR scanners, parallel techniques were developed. These techniques reduce the acquisition time via a subsampling of the **k**-space and a reconstruction process. One of the most popular techniques among them is Sensitivity Encoding (SENSE). From a signal and noise perspective, the use of a SENSE will modify the structure of the noise within the image. The final magnitude image after the reconstruction is known to follow a Rician distribution for each pixel, just like single coil systems. However, the noise is spatially non-stationary, i.e. the variance of noise becomes **x**-dependent and its features change across the image. In this work we propose a method to adapt well-known noise filtering techniques initially designed to deal with stationary noise to the case of spatially variant noise, like the case of SENSE data.

The method copes with inaccurate estimates of variant noise patterns in the image, showing its robustness in realistic cases. The method employs a consensus strategy in conjunction with a set of aggregation functions and a penalty function. Multiple possible outputs are generated for each pixel assuming different unknown input parameters. The consensus approach merges them into a unique filtered image. As a filtering technique, we have selected the Linear Minimum Mean Square Error (LMMSE) estimator for Rician data, which has been used to test our methodology for its simplicity and robustness. Results with synthetic and *in vivo* data confirm the good behavior of our approach. Results also showed the flexibility of the proposed approach for other filtering methods, succeeding in roughly estimating the spatially variant noise variance.

2. L. González-Jaime, E. E. Kerre, M. Nachtegael, and H. Bustince. Consensus image method for unknown noise removal. *Knowledge-Based Systems*, 70: 64–77, 2014

   - Status: **Published**.
   - Type: Journal Article.
   - Journal Title: *Knowledge-Based Systems*.
   - Impact Factor (JCR 2013): 3.058.
   - Subject Category: Computer Science, Artificial Intelligence. Ranking 15 / 121 (Q1).
   - Correspondence with: Chapter 5.

   Abstract: Noise removal has been, and it is nowadays, an important task in computer vision. Usually, it is a previous task preceding other tasks, as segmentation or reconstruction. However, for most existing denoising algorithms the noise model has to be known in advance. In this paper, we introduce a new approach based on consensus to deal with unknown noise models. To do this, different filtered images are obtained, then combined using multifuzzy sets and averaging aggregation functions. The final decision is made by using a penalty function to deliver the compromised image. Results show that this approach is consistent and provides a good compromise between filters.

3. L. González-Jaime, G. Vegas-Sánchez-Ferrero, M. Nachtegael, E. E. Kerre, and S. Aja-Fernández. Applying a Parametric Approach for the Task of Nonstationary Noise Removal with Missing Information. In *Computational Cybernetics (ICCC), IEEE 9th International Conference on*, pages 23–28, 2013

   - Status: **Published**.
   - Type: International Conference paper.
   - Published in: IEEE 9^{th} International Conference on Computational Cybernetics (ICCC), 2013.
   - Correspondence with: Chapter 6.

   Abstract: The image capturing process still today introduces degradations that are unavoidable. The research community is compromised with this issue developing algorithms for the noise removal task. Most of the existing approaches in the literature are parametric, i.e. some information from the underlying model is required. However, there are situations in which this information cannot be captured accurately and the use of these approaches is dismissed. Therefore, we propose an approach where averaging functions are applied over different realizations of a parametric filter. Then, the required information for the parametric filter is extracted and combined from the different parameter configurations used. So, we give the possibility to use parametric approaches in situations where some information is missing. Results show that the averaging functions present promising outcomes for the nonstationary noise removal task.

4. L. González-Jaime, M. Nachtegael, E. E. Kerre, G. Vegas-Sánchez-Ferrero, and S. Aja-Fernández. Parametric Image Restoration Using Consensus: An Application to Nonstationary Noise Filtering. In J. M. Sanches, L. Micó, and J. S. Cardoso, editors, *Pattern Recognition and Image Analysis*, volume 7887 of *Lecture Notes in Computer Science*, pages 358–365. Springer Berlin Heidelberg, 2013

   - Status: **Published**.
   - Type: Lecture Notes in Computer Science.
   - Published in: Pattern Recognition and Image Analysis.

- Correspondence with: Chapter 6.

Abstract: Image quality gets affected by unavoidable degradations. Several techniques have been proposed based on a priori information of the degradation. However, these techniques fail when the underlying parameters cannot be estimated. We propose a method to deal with situations when the underlying parameters are not known. It is based on the consensus achieved by using a set of aggregation functions and a penalty function. The method is tested in the case of a nonstationary Gaussian noise, and the Wiener filter is used to prove this methodology. The results show that the approach is consistent and it achieves comparable results for known parameters.

5. L. González-Jaime, M. Nachtegael, E. E. Kerre, and H. Bustince. Use of Idempotent Functions in the Aggregation of Different Filters for Noise Removal. In *the seventh International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2012), Proceedings on*, volume 214 of *Advances in Intelligent Systems and Computing*, pages 495–507, 2012

   - Status: **Published**.
   - Type: International Conference paper.
   - Published in: Proceedings on the 7^{th} International Conference on Intelligent Systems and Knowledge Engineering, (ISKE 2012).
   - Correspondence with: Chapter 5.

   Abstract: The majority of existing denoising algorithms obtain good results for a specific noise model, and when it is known previously. Nonetheless, there is a lack in denoising algorithms that can deal with any unknown noisy images. Therefore, in this paper, we study the use of aggregation functions for denoising purposes, where the noise model is not necessary known in advance; and how these functions affect the visual and quantitative results of the resultant images.

V

Appendix

Appendix A

Probability distributions and moments

A.1 Gaussian distribution (Normal)

$$X \sim N(\mu, \sigma^2).$$

PDF:

$$p(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{A.1}$$

MGF:

$$M_X(t) = \exp\left(\mu t + \frac{\sigma^2 t^2}{2}\right).$$

Main parameters:

$$\begin{aligned} \text{Mean} &= \mu, \\ \text{Median} &= \mu, \\ \text{Mode} &= \mu, \\ \text{Variance} &= \sigma^2. \end{aligned}$$

Main moments:

$$\begin{aligned}\mu_1 &= \mu,\\ \mu_2 &= \mu^2+\sigma^2,\\ \mu_3 &= \mu^3+3\mu\sigma^2,\\ \mu_4 &= \mu^4+6\mu^2\sigma^2+3\sigma^4.\end{aligned}$$

A.2 Rayleigh distribution

$$R=\sqrt{X_1^2+X_2^2}\quad X_i\sim N(0,\sigma^2).$$

PDF:

$$p(x)=\frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{A.2}$$

MGF:

$$M_X(t)=1+\sigma te^{\sigma^2t^2/2}\sqrt{\frac{\pi}{2}}\left(\operatorname{erf}\left(\frac{\sigma t}{\sqrt{2}}\right)+1\right).$$

Raw moments:

$$\mu_k=\sigma^k 2^{k/2}\Gamma(1+k/2).$$

Main parameters:

$$\begin{aligned}\text{Mean} &= \sigma\sqrt{\frac{\pi}{2}},\\ \text{Median} &= \sigma\sqrt{\log 4},\\ \text{Mode} &= \sigma,\\ \text{Variance} &= \frac{4-\pi}{2}\sigma^2.\end{aligned}$$

Main moments:

$$\begin{aligned}\mu_1 &= \sqrt{\frac{\pi}{2}}\sigma,\\ \mu_2 &= 2\sigma^2,\\ \mu_3 &= 3\sqrt{\frac{\pi}{2}}\sigma^3,\\ \mu_4 &= 8\sigma^4.\end{aligned}$$

A.3 Rician distribution

$$R = \sqrt{X_1^2 + X_2^2}, \quad X_i \sim N(A_i, \sigma^2).$$
$$R = |X| \quad X = N(A_1, \sigma^2) + jN(A_2, \sigma^2).$$

PDF:

$$p_M(M|A,\sigma) = \frac{M}{\sigma^2}\, e^{-\frac{M^2+A^2}{2\sigma^2}}\, I_0\left(\frac{AM}{\sigma^2}\right)\, u(M), \tag{A.3}$$

where

$$A = \sqrt{A_1^2 + A_2^2}.$$

Raw moments:

$$\mu_k = \sigma^k 2^{k/2} \Gamma(1 + k/2) L_{k/2}\left(-\frac{A^2}{2\sigma^2}\right),$$

where

$$L_n(x) = M(-n, 1, x) = {}_1F_1(-n; 1; x).$$

Main parameters:

$$\begin{aligned} \text{Mean} &= \sigma\sqrt{\frac{\pi}{2}} L_{1/2}\left(-\frac{A^2}{2\sigma^2}\right), \\ \text{Variance} &= 2\sigma^2 + A^2 - \frac{\pi\sigma^2}{2} L_{1/2}\left(-\frac{A^2}{2\sigma^2}\right). \end{aligned}$$

Main moments:

$$\begin{aligned} \mu_1 &= \sqrt{\frac{\pi}{2}} L_{1/2}\left(-\frac{A^2}{2\sigma^2}\right)\sigma, \\ \mu_2 &= A^2 + 2\sigma^2, \\ \mu_3 &= 3\sqrt{\frac{\pi}{2}} L_{3/2}\left(-\frac{A^2}{2\sigma^2}\right)\sigma^3, \\ \mu_4 &= A^4 + 8\sigma^2 A^2 + 8\sigma^4. \end{aligned}$$

$$\begin{aligned} \text{Var} &= 2\sigma^2 + A^2 - \frac{\pi\sigma^2}{2} L_{1/2}^2\left(-\frac{A^2}{2\sigma^2}\right) \\ &\approx \sigma^2\left(1 - \frac{1}{4x} - \frac{1}{8x^2} + O(x^{-3})\right) \qquad \text{with } x = \frac{A^2}{2\sigma^2}. \end{aligned}$$

Series expansion of Hypergeometric Functions:

$$L_{1/2}(-x) = \frac{2\sqrt{x}}{\sqrt{\pi}} + \frac{1}{2\sqrt{\pi}\sqrt{x}} + \frac{1}{16\sqrt{\pi}x^{3/2}} + O\left(x^{-5/2}\right)$$

$$L_{3/2}(-x) = \frac{4x^{3/2}}{3\sqrt{\pi}} + \frac{3\sqrt{x}}{\sqrt{\pi}} + \frac{3}{8\sqrt{\pi}\sqrt{x}} + \frac{1}{32\sqrt{\pi}x^{3/2}} + O\left(x^{-5/2}\right).$$

Bibliography

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast Image Recovery Using Variable Splitting and Constrained Optimization. *Image Processing, IEEE Transactions on*, 19(9):2345–2356, 2010.

[2] A. Agrawal, Y. Xu, and R. Raskar. Invertible Motion Blur in Video. *ACM Transactions on Graphics*, 28(3):1–8, 2009.

[3] S. Aja-Fernández, R. S. J. Estepar, C. Alberola-López, and C. F. Westin. Image quality assessment based on local variance. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 4815–4818, 2006.

[4] S. Aja-Fernández, C. Alberola-López, and C-F. Westin. Noise and signal estimation in magnitude MRI and Rician distributed images: A LMMSE approach. *Image Processing, IEEE Transactions on*, 17(8):1383–1398, 2008.

[5] S. Aja-Fernández, A. Tristán-Vega, and C. Alberola-López. Noise estimation in single- and multiple-coil magnetic resonance data based on statistical models. *Magnetic Resonance Imaging*, 27(10):1397–1409, 2009.

[6] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, M. Martín-Fernández, and C. Alberola-López. Automatic noise estimation in images using local statistics. Additive and multiplicative cases. *Image and Vision Computing*, 27 (6):756–770, 2009.

[7] S. Aja-Fernández, R. S. Estépar, and C. Alberola-López. Full Reference Image Quality Assessment based on Local Statistics. Technical report, Universidad de Valladolid, 2014.

[8] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, and A. Tristán-Vega. Noise Estimation in Parallel MRI: GRAPPA and SENSE. *Magnetic Resonance Imaging*, 32(3):281–290, 2014.

[9] G. Angelopoulos and I. Pitas. Multichannel Wiener filters in color image restoration based on AR color image modelling. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, volume 4, pages 2517–2520, 1991.

[10] B. Bedregal, G. Beliakov, H. Bustince, T. Calvo, R. Mesiar, and D. Paternain. A class of fuzzy multisets with a fixed number of memberships. *Information Sciences*, 189:1–17, 2012.

[11] G. Beliakov, A. Pradera, and T. Calvo. *Aggregation functions: A guide for practitioners*, volume 221 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag Berlin Heidelberg, Berlin, 2007.

[12] G. Beliakov, H. Bustince, and D. Paternain. Image Reduction Using Means on Discrete Product Lattices. *Image Processing, IEEE Transactions on*, 21 (3):1070–1083, 2012.

[13] M. A. Bernstein, D. M. Thomasson, and W. H. Perman. Improved detectability in low signal-to-noise ratio magnetic-resonance images by means of a phase-corrected real reconstruction. *Medical Physics*, 16(5):813–817, 1989.

[14] J. C. Bezdek, J. Keller, R. Krisnapuram, and N. R. Pal. *Fuzzy models and algorithms for pattern recognition and image processing*, volume 4 of *The Handbooks of Fuzzy Sets Series*. Springer US, 1999.

[15] J. Biemond, F. G. van der Putten, and J. W. Woods. Identification and restoration of images with symmetric noncausal blurs. *Circuits and Systems, IEEE Transactions on*, 35(4):385–393, 1988.

[16] J. Biemond, R. L. Lagendijk, and R. M. Mersereau. Iterative methods for image deblurring. *Proceedings of the IEEE*, 78(5):856–883, 1990.

[17] G. Birkhoff. *Lattice theory*, volume 25. American Mathematical Soc., 1967.

[18] A. C. Bovik. *The essential guide to image processing.* Academic Press, Boston, second edition, 2009.

[19] S. P. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, Cambridge, 2004.

[20] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[21] H. Bustince, E. Barrenechea, and M. Pagola. Image thresholding using restricted equivalence functions and maximizing the measures of similarity. *Fuzzy Sets and Systems*, 158(5):496–516, 2007.

[22] H. Bustince, M. Pagola, E. Barrenechea, J. Fernández, P. Melo-Pinto, P. Couto, H. R. Tizhoosh, and J. Montero. Ignorance functions. An application to the calculation of the threshold in prostate ultrasound images. *Fuzzy Sets and Systems*, 161(1):20–36, 2010.

[23] H. Bustince, J. Fernández, R. Mesiar, G. Beliakov, and T. Calvo. Penalty functions over a Cartesian product of lattices. *AGOP 2011: Proceedings of 6th International Summer School on Aggregation Operators*, pages 59–64, 2011.

[24] H. Bustince, E. Barrenechea, T. Calvo, S. James, and G. Beliakov. Consensus in multi-expert decision making problems using penalty functions defined over a Cartesian product of lattices. *Information Fusion*, 17:56–64, 2014.

[25] T. Calvo and G. Beliakov. Aggregation functions based on penalties. *Fuzzy Sets and Systems*, 161(10):1420–1436, 2010.

[26] T. Calvo, R. Mesiar, and R. R. Yager. Quantitative weights and aggregation. *Fuzzy Systems, IEEE Transactions on*, 12(1):62–69, 2004.

[27] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems withApplications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[28] H-C. Chen and W-J. Wang. Efficient impulse noise reduction via local directional gradients and fuzzy logic. *Fuzzy Sets and Systems*, 160(13): 1841–1857, 2009.

[29] L. Chen, K-H. Yap, and Y. He. Efficient Recursive Multichannel Blind Image Restoration. *EURASIP Journal on Applied Signal Processing*, 2007 (1):8, 2007.

[30] F. Chiclana, F. Herrera, and E. Herrera-Viedma. Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations. *Fuzzy Sets and Systems*, 97(1):33–48, 1998.

[31] C. A. Cocosco, V. Kollokian, R. K-S. Kwan, G. B. Pike, and A. C. Evans. Brainweb: Online interface to a 3D MRI simulated brain database. In *NeuroImage*, volume 5, 1997.

[32] P. C. Cosman, R. M. Gray, and R. A. Olshen. Evaluating quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. *Proceedings of the IEEE*, 82(6):919–932, 1994.

[33] C. A. Deledalle, F. Tupin, and L. Denis. Poisson NL means: Unsupervised non local means for poisson noise. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 801–804, 2010.

[34] D. V. der Weken, M. Nachtegael, and E. E. Kerre. Using similarity measures and homogeneity for the comparison of images. *Image and Vision Computing*, 22(9):695–702, 2004.

[35] O. Dietrich, J. G. Raya, S. B. Reeder, M. Ingrisch, M. F. Reiser, and S. O. Schoenberg. Influence of multichannel combination, parallel imaging and other reconstruction techniques on MRI noise characteristics. *Magnetic Resonance Imaging*, 26(6):754–762, 2008.

[36] D. Dubois and H. Prade. Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions. *Fuzzy Sets and Systems*, 100:73–132, 1999.

[37] D. Dubois and H. Prade. *Fundamentals of fuzzy sets*, volume 7 of *The Handbooks of Fuzzy Sets*. Springer Science+Business Media New York, New York, 2000.

[38] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *Image Processing, IEEE Transactions on*, 6(12):1646–1658, 1997.

[39] A. M. Eskicioglu and P. S. Fisher. Image quality measures and their performance. *Communications, IEEE Transactions on*, 43(12):2959–2965, 1995.

[40] Q. Fan, D. Jiang, and Y. Jiao. A multi-parameter regularization model for image restoration. *Signal Processing*, 114(0):131–142, 2015.

[41] J. C. Fodor and M. R. Roubens. *Fuzzy preference modelling and multicriteria decision support*, volume 14 of *Theory and Decision Library D:*. Springer Netherlands, Dordrecht, 1994.

[42] N. P. Galatsanos and R. T. Chin. Digital restoration of multichannel images. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(3):415–421, 1989.

[43] B. Girod. What's Wrong with Mean-squared Error? In A. B. Watson, editor, *Digital Images and Human Vision*, pages 207–220. MIT Press, Cambridge, MA, USA, 1993.

[44] J. A. Goguen. L-fuzzy sets. *Journal of mathematical analysis and applications*, 18(1):145–174, 1967.

[45] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, third edition, 2006.

[46] L. González-Jaime, M. Nachtegael, E. E. Kerre, and H. Bustince. Use of Idempotent Functions in the Aggregation of Different Filters for Noise Removal. In *the seventh International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2012), Proceedings on*, volume 214 of *Advances in Intelligent Systems and Computing*, pages 495–507, 2012.

[47] L. González-Jaime, M. Nachtegael, E. E. Kerre, G. Vegas-Sánchez-Ferrero, and S. Aja-Fernández. Parametric Image Restoration Using Consensus: An Application to Nonstationary Noise Filtering. In J. M. Sanches, L. Micó, and J. S. Cardoso, editors, *Pattern Recognition and Image Analysis*, volume 7887 of *Lecture Notes in Computer Science*, pages 358–365. Springer Berlin Heidelberg, 2013.

[48] L. González-Jaime, G. Vegas-Sánchez-Ferrero, M. Nachtegael, E. E. Kerre, and S. Aja-Fernández. Applying a Parametric Approach for the Task of Nonstationary Noise Removal with Missing Information. In *Computational Cybernetics (ICCC), IEEE 9th International Conference on*, pages 23–28, 2013.

[49] L. González-Jaime, E. E. Kerre, M. Nachtegael, and H. Bustince. Consensus image method for unknown noise removal. *Knowledge-Based Systems*, 70: 64–77, 2014.

[50] L. González-Jaime, G. Vegas-Sánchez-Ferrero, M. Nachtegael, E. E. Kerre, and S. Aja-Fernández. Spatially variant noise filtering in MRI. A Consensus approach. *Magnetic Resonance Imaging*, 2014.

[51] B. Goossens. *Multiresolution image models and estimation techniques*. PhD thesis, Ghent University, Ghent, Belgium, 2010.

[52] B. Goossens, H. Luong, A. Pizurica, and W. Philips. An improved non-local denoising algorithm. In *Local and Non-Local Approximation in Image Processing, International Workshop, Proceedings*, pages 143–156, 2008.

[53] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. M. Wang, B. Kiefer, and A. Haase. Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA). *Magnetic Resonance in Medicine*, 47(6): 1202–1210, 2002.

[54] Video Quality Experts Group and Others. Final report from the video quality experts group on the validation of objective models of video quality assessment. Technical report, 2000.

[55] H. Gudbjartsson and S. Patz. The Rician distribution of noisy MRI data. *Magnetic Resonance in Medicine*, 34(6):910–914, 1995.

[56] R. J. Hathaway, J. C. Bezdek, and W. Pedrycz. A parametric model for fusing heterogeneous fuzzy data. *Fuzzy Systems, IEEE Transactions on*, 4 (3):270–281, 1996.

[57] R. M. Henkelman. Measurement of Signal Intensities in the Presence of Noise in MR images. *Medical Physics*, 12(2):232+, 1985.

[58] J. Hsieh. Adaptive streak artifact reduction in computed tomography resulting from excessive X-ray photon noise. *Medical Physics*, 25(11):2139–2147, 1998.

[59] A. Jurio, D. Paternain, M. Pagola, and H. Bustince. Image Thresholding by Grouping Functions: Application to MRI Images. In L. A. Zadeh, A. M. Abbasov, R. R. Yager, S. N. Shahbazova, and M. Z. Reformat, editors, *Recent Developments and New Directions in Soft Computing*, volume 317 of *Studies in Fuzziness and Soft Computing*, pages 195–208. 2014.

[60] J. Kacprzyk. *Fuzzy Sets and Fuzzy Systems: A Brief Introduction*, volume 41 of *Studies in Fuzziness and Soft Computing*, pages 3–30. Physica-Verlag HD, Heidelberg, 2000.

[61] A. K. Katsaggelos, S. D. Babacan, and T. Chun-Jen. *Iterative Image Restoration*, chapter 15, pages 349–383. Academic Press, Boston, second edition, 2009.

[62] E. E. Kerre. *Basic principles of fuzzy set theory for the representation and manipulation of imprecision and uncertainty*, pages 1–158. 1992.

[63] W. J. M. Kickert. *Fuzzy theories on decision making: A critical review*, volume 3 of *Frontiers in System Research*. Springer US, New York, 1978.

[64] L. Kitainik. *Fuzzy decision procedures with binary relations: towards a unified theory*, volume 13 of *Theory and Decision Library*. Springer Netherlands, Dordrecht, 1993.

[65] G. J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: theory and applications*. Prentice Hall New Jersey, Upper Saddle River, NJ, USA, 1995.

[66] D. Kundur and D. Hatzinakos. Blind image deconvolution. *Signal Processing Magazine, IEEE*, 13(3):43–64, 1996.

[67] P. J. La Riviere and D. M. Billmire. Reduction of noise-induced streak artifacts in X-ray computed tomography through spline-based penalized-likelihood sinogram smoothing. *Medical Imaging, IEEE Transactions on*, 24(1):105–111, 2005.

[68] R. L. Lagendijk and J. Biemond. *Basic Methods for Image Restoration and Identification*, chapter 14, pages 323–348. Academic Press, Boston, second edition, 2009.

[69] J. S. Lim. Two-dimensional signal and image processing. *Englewood Cliffs, NJ, Prentice Hall, 1990, 710 p.*, 1, 1990.

[70] V. Loyev and Y. Yitzhaky. Initialization of iterative parametric algorithms for blind deconvolution of motion-blurred images. *Applied Optics*, 45(11): 2444–2452, 2006.

[71] F. Luisier, T. Blu, and P. J. Wolfe. A CURE for noisy magnetic resonance images: Chi-square unbiased risk estimation. *Image Processing, IEEE Transactions on*, 21(8):3454–3466, 2012.

[72] J. V. Manjon, J. Carbonell-Caballero, J. J. Lull, G. Garcia-Marti, L. Marti-Bonmati, and M. Robles. MRI denoising using Non-Local Means. *Medical Image Analysis*, 12(4):514–523, 2008.

[73] F. D. Martino and S. Sessa. Compression and decompression of images with discrete fuzzy transforms. *Information Sciences*, 177(11):2349–2362, 2007.

[74] F. D. Martino, V. Loia, and S. Sessa. A segmentation method for images compressed by fuzzy transforms. *Fuzzy Sets and Systems*, 161(1):56–74, 2010.

[75] G. McGibney and M. R. Smith. An unbiased signal-to-noise ratio measure for magnetic-resonance images. *Medical Physics*, 20(4):1077–1078, 1993.

[76] S. Miyamoto. Multisets and Fuzzy Multisets. In Z. Q. Liu and S. Miyamoto, editors, *Soft Computing and Human-Centered Machines*, Computer Science Workbench, pages 9–33. Tokyo, 2000.

[77] R. Molina, J. Núñez, F. J. Cortijo, and J. Mateos. Image restoration in astronomy - A Bayesian perspective. *IEEE Signal Processing Magazine*, 18 (2):11–29, 2001.

[78] M. Nachtegael, Weken, D. Ville, and E. E. Kerre. *Fuzzy Filters for Image Processing*, volume 122 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag Berlin Heidelberg, Berlin, 2003.

[79] R. Orduna, A. Jurio, D. Paternain, H. Bustince, P. Melo-Pinto, and E. Barrenechea. Segmentation of color images using a linguistic 2-tuples model. *Information Sciences*, 258:339–352, 2014.

[80] P. J. Pahl and R. Damrath. *Mathematical foundations of computational engineering: a handbook*. Springer-Verlag Berlin Heidelberg, Berlin, 2001.

[81] K. Patanukhom. Image restoration based on a pair of noisy and motion blurred images. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*, pages 693–697, 2010.

[82] D. Paternain, H. Bustince, J. Fernández, G. Beliakov, and R. Mesiar. Some averaging functions in image reduction. In *Trends in applied intelligent systems*, volume 6098 of *Lecture Notes in Computer Science*, pages 399–408. 2010.

[83] D. Paternain, C. López-Molina, H. Bustince, R. Mesiar, and G. Beliakov. Image reduction using fuzzy quantifiers. In *Eurofuse 2011*, volume 107 of *Advances in Intelligent and Soft Computing*, pages 351–362. 2012.

[84] I. Perfilieva. Fuzzy Transforms and Their Applications to Image Compression. In I. Bloch, A. Petrosino, and A. G. B. Tettamanzi, editors, *Fuzzy Logic and Applications*, volume 3849 of *Lecture Notes in Computer Science*, pages 19–31. Springer Berlin Heidelberg, 2006.

[85] K. P. Pruessmann, M. Weiger, M. B. Scheidegger, and P. Boesiger. SENSE: Sensitivity encoding for fast MRI. *Magnetic Resonance in Medicine*, 42(5): 952–62, 1999.

[86] S. Qi, H. Wang, and L. Wei. An Iterative Blind Deconvolution Image Restoration Algorithm Based on Adaptive Selection of Regularization Parameter. In *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, volume 1, pages 112–115, 2009.

[87] S. Ramya and T. Mercy Christial. Restoration of blurred images using Blind Deconvolution Algorithm. In *Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on*, pages 496–499, 2011.

[88] P. M. Robson, A. K. Grant, A. J. Madhuranthakam, R. Lattanzi, D. K. Sodickson, and C. A. McKenzie. Comprehensive quantification of signal-to-noise ratio and g-factor for image-based and k-space-based parallel imaging reconstructions. *Magnetic Resonance in Medicine*, 60(4):895–907, 2008.

[89] M. S. Rosenthal, J. Cullom, W. Hawkins, S. C. Moore, B. M. W. Tsui, and M. Yester. Quantitative SPECT imaging - A review and recommendations by the focus committee of the society-of-nuclear-medicine computer and instrumentation council. *Journal of Nuclear Medicine*, 36(8):1489–1513, 1995.

[90] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[91] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms. *Image Processing, IEEE Transactions on*, 15(11):3440–3451, 2006.

[92] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. LIVE Image Quality Assessment Database Release 2, 2014.

[93] J. Sijbers, A. J. den Dekker, P. Scheunders, and D. Van Dyck. Maximum-likelihood estimation of Rician distribution parameters. *Medical Imaging, IEEE Transactions on*, 17(3):357–361, 1998.

[94] E. A. Silva, K. Panetta, and S. S. Agaian. Quantifying image similarity using measure of enhancement by entropy. volume 6579, page 65790U, 2007.

[95] S. Skiadopoulos, A. Karatrantou, P. Korfiatis, L. Costaridou, P. Vassilakos, D. Apostolopoulos, and G. Panayiotakis. Evaluating image denoising methods in myocardial perfusion single photon emission computed tomography (SPECT) imaging. *Measurement Science & Technology*, 20(10):104023+, 2009.

[96] K. S. Srinivasan and D. Ebenezer. A new fast and efficient decision-based algorithm for removal of high-density impulse noises. *IEEE Signal Processing Letters*, 14(3):189–192, 2007.

[97] E. Straszecka. *Defining membership functions*, volume 41 of *Studies in Fuzziness and Soft Computing*, pages 32–47. Physica-Verlag, Heidelberg, 2000.

[98] Z. Sun and G. Meng. An image filter for eliminating impulse noise based on type-2 fuzzy sets. In Wg Wan, editor, *2008 International Conference on Audio, Language and Image Processing, Vols 1 and 2, Proceedings*, pages 1278–1282, 2008.

[99] S. Suzuki. A comparative-study on pre-smoothing techniques for projection data with poisson noise in computed-tomography. *Optics Communications*, 55(4):253–258, 1985.

[100] H. Tang and L. W. Cahill. A new criterion for the evaluation of image restoration quality. In *TENCON'92."Technology Enabling Tomorrow: Computers, Communications and Automation towards the 21st Century.'1992 IEEE Region 10 International Conference.*, pages 573–577, 1992.

[101] A. Tekalp, H. Kaufman, and J. W. Woods. Identification of image and blur parameters for the restoration of noncausal blurs. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(4):963–972, 1986.

[102] P. Thunberg and P. Zetterberg. Noise distribution in SENSE- and GRAPPA-reconstructed images: a computer simulation study. *Magnetic Resonance Imaging*, 25(7):1089–1094, 2007.

[103] M. Tico and M. Vehvilainen. Estimation of motion blur point spread function from differently exposed image frames. *14th European Signal Processing Conference (EUSIPCO), 2006, Proceedings of*, 2006.

[104] M. Tim. *Computer vision and image processing.* Cornerstones of Computing. Palgrave Macmillan, London, 2003.

[105] A. Tristán-Vega, V. García-Pérez, S. Aja-Fernández, and C. F. Westin. Efficient and robust nonlocal means denoising of MR data based on salient features matching. *Computer Methods and Programs in Biomedicine*, 105 (2):131–144, 2012.

[106] S. T. Wang, F. L. Chung, Y. Y. Li, D. W. Hu, and X. S. Wu. A new gaussian noise filter based on interval type-2 fuzzy logic systems. *Soft Computing*, 9 (5):398–406, 2005.

[107] X. Wang, D. Ruan, and Kerre E. E. *Mathematics of fuzzinessbasic issues*, volume 245 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag Berlin Heidelberg, Berlin, 2009.

[108] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[109] Q. Wu, X. ce Wang, and P. Guo. Joint Blurred Image Restoration with Partially Known Information. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 3853–3858, 2006.

[110] R. R. Yager. On the theory of bags. *International Journal Of General System*, 13(1):23–37, 1986.

[111] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision-making. *Systems Man and Cybernetics, IEEE Transactions on*, 18(1):183–190, 1988.

[112] R. R. Yager. Families of OWA operators. *Fuzzy Sets and Systems*, 59(2): 125–148, 1993.

[113] Z-Z. Yang and Z. Yang. Noisy Image Reconstruction Via Fast Linearized Lagrangian Dual Alternating Direction Method of Multipliers. *Wireless Personal Communications*, 82(1):143–156, 2015.

[114] Y. L. You, W. Y. Xu, A. Tannenbaum, and M. Kaveh. Behavioral analysis of anisotropic diffusion in image processing. *Image Processing, IEEE Transactions on*, 5(11):1539–1553, 1996.

[115] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[116] L. A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with applications*, 9(1):149–184, 1983.

[117] X-J. Zeng and G. Singh Madan. Approximation Theory of fuzzy systems-SISO case. *Fuzzy Systems, IEEE Transactions on*, 2(2):162–176, 1994.