UNIVERSITÉ CATHOLIQUE DE LOUVAIN

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES, ELECTRONICS AND APPLIED MATHEMATICS

MACHINE LEARNING GROUP

Mode estimation based filtering in digital imaging and application to medical image denoising

Arnaud De Decker

Thesis presented for the Ph.D. degree
in Engineering Sciences

Thesis Jury:
Prof. **M. Verleysen**, Advisor (UCL - ICTEAM Institute, Machine Learning Group)
Dr. **J. Lee**, Advisor (UCL - Center for Molecular Imagery and Experimental Radiotherapy)
Dr. **A. Weismuller** (University of Rochester, New York, Dept. of Radiology and Dept. of Biomedical Engineering)
Prof. **J.P. Peters** (FUNDP - Departement de Physique)
Prof. **B. Gosselin** (UMONS - TCTS lab)
Prof. **L. Vandendorpe**, President (UCL - ICTEAM Institute)

August 23, 2011

Contents

Remerciements

Arrivé au terme de ce travail, je voudrais remercier tous ceux qui ont contribué à sa réalisation d'une manière ou d'une autre.

Premièrement, je voudrais remercier les professeurs Michel Verleysen et John Lee, mes promoteurs, sans qui cette thèse n'aurait jamais pu voir le jour. Tout au long de l'élaboration de ce travail, j'ai profité de leurs conseils et de leurs connaissances. Je les remercie de m'avoir donné la chance de faire cette thèse avec eux. En plus d'être des collaborateurs de qualité, ils ont aussi été des personnes agréables et attentionnées.

Je voudrais aussi remercier Gaël Delannoy pour m'avoir présenté à Michel, ainsi que pour les longues journées passées ensemble dans le même bureau. Il m'a supporté depuis les humanités jusqu'à la fin du doctorat, peu de personnes auraient pu en faire autant (malgré une calvitie naissante, due, selon lui, au fait de devoir me voir tous les jours au bureau pendant 5 ans...)

Je remercie de tout coeur mes parents pour l'éducation qu'ils m'ont donnée qui m'a permis d'entreprendre des études et ce doctorat tout en accordant une place à d'autres activités comme le sport, les amis, ou la musique. J'ai bénéficié de leur soutien durant toutes mes années d'études et j'en suis extrêmement reconnaissant.

Enfin, je remercie Christel pour sa patience, sa compréhension et sa présence à mes côtés tous les jours. Que cela soit durant la thèse ou après, je sais que partager des projets et ma vie avec elle me permet de grandir dans tous les domaines qui m'inspirent et me réjouissent.

Chapter 1

Foreword

Today, digital images are massively produced in all kind of fields: entertainment, graphical design, military, multimedia, meteorology, climatology, astrology, geography, medical applications, computer vision ... The diversity of the acquisition devices and the situations in which the image is taken are important factors affecting the quality of these images: they can be over or under exposed, blurred, noisy, or contain artefacts. The combination of all of these types of pollutions sometimes causes major problems for the application: misclassification, bad diagnosis or lack of precision.

One of the most common types of pollution on images is the noise (see Figure 1.1). Noise arises during the acquisition of the image and depends on the quality of the components used in the acquisition device, on the type of signal detected, the exposure time, the detector sensitivity, and many other factors.

This thesis focuses on algorithms designed to remove noise from digital images, commonly named filtering algorithms and in particular, on filtering methods based on statistics drawn from the images. The major drawbacks of filtering methods are the computational time required to apply them, and the unwanted removal or smoothing of some features of the images during the filtering process that can result in blurry, or unnatural images.

The rest of this chapter is organized as follows. Section 1.1 defines the objectives of this thesis and briefly presents the methods proposed to tackle them. Section 1.2 gives an overview of the types of images used in this work and their characteristics.

(a) Lena image

(b) Noisy Lena image with additional noise $N(0,40)$

Figure 1.1: Examples of clean and noisy images.

1.1 Objectives

The aim of this thesis is to develop and test new algorithms for image filtering. These algorithms should provide better and more robust filtering results than the existing methods at a lower or equivalent computational cost. The number of parameters of the methods should as low as possible.

The following solutions are proposed in this manuscript:

- In Chapter 2, the basic image model is presented, then state-of-the-art denoising algorithms are discussed and the statistical filters which are at the core of the new filtering algorithms developed in this thesis are detailed. The derivation of each type of filter is detailed.

- Chapter 3 proposes new types of filtering algorithms in high-dimensional spaces based on the statistical filters. They are based on the observation that mode separation is usually better in high-dimensional spaces than in 1-dimensional spaces. The filters are modified to use blocks of images, called *patches* in the filtering process. Two types of high-dimensional filters are created: patch-based filters, and patchwise filters. Experiments are made to assess the filtering performances of these new algorithms in comparison with traditional statistical filtering algorithms.

- In Chapter 4, the Gaussian kernel used in the filtering process is proven to be suboptimal in high-dimensional spaces. In high-dimensional spaces, the Eculidean norm used in the filters is subject to the norm concentration phenomenon. As the dimensionality increases, the mean of the norm between data points increases, while the variance does not change. As a result, the distances between all data points look the same. In this case, the Gaussian kernel traditionally used in the statistical filters is not an efficient similarity measure anymore. New kernels are proposed in order to deal efficiently with the curse of dimensionality. Experiments show that the new kernels outperform the traditional Gaussian kernel in high-dimensional spaces.

- The choice of the parameters of the filtering algorithms is of crucial importance on the filtering results. However, it is often impossible to estimate these parameters in a real situation. Furthermore, these parameters are fixed as constant values for the whole image while the noise encountered in the images is not necessarily constant on the image. Chapter 5 details a procedure to estimate the value of the filter parameters for each pixel of the image based on statistical models of the standard deviation of the noise in the images. The models are valid for a particular acquisition device and could be used instead of the set of default parameters usually associated with this device. Experiments show that the results obtained with these models are much more robust to the choice of parameters than the classical filters.

- In some medical applications, the noise approximately follows a Poisson law, while the hypothesis behind most statistical filters is that the noise is Gaussian. In Chapter 6, variance stabilizing transforms are used to transform Poisson noise into Gaussian noise. We demonstrate the use of two types of variance stabilizing transforms on images polluted with Poissonian noise. The filtering algorithms are modified to integrate one of these variance stabilizing transforms in the filtering process. Experiment show that the stabilized images are better filtered than the images filtered without the variance stabilizing transforms.

1.2 Digital images used in this manuscript

In this work, three types of images are used.

Artificial images are created in order to gather most types of difficulties that can be encountered in one single image. Artificial images are usually created in order to test

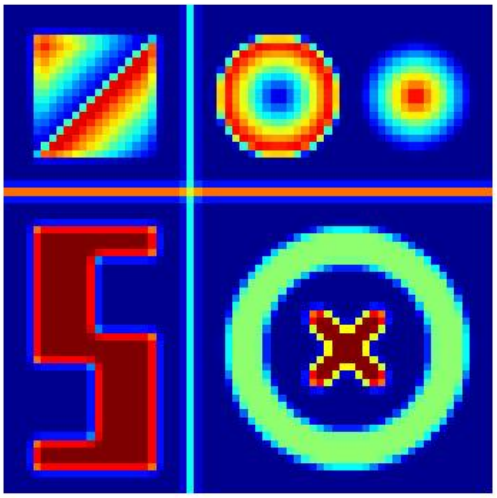


Figure 1.2: Example of artificial image used in this thesis.

the behaviour of a filter in a particular case, such as constant areas, gradients, edges, or textures. These images are noiseless when created; noise is generated from a statistical model and added to the image. The ground truth image is always known and can be used to evaluate the filter performances. Figure 1.2 presents the artificial image used in this thesis. This image is made of specific areas that present specific difficulties to the filtering algorithms: plateaux, edges and gradients.

Digital pictures are images taken from a traditional digital camera. Although they naturally contain some noise, artificial noise is usually added to these pictures in order to have a better view of the effect of the filters in difficult situations. This method, however, has a serious drawback: the original image, without added noise, is considered as the ground truth image. As this image already contains some noise, if a filter removes this noise, the filtered image will be considered different from the ground truth, while, in fact, the filtered image is cleaner than the ground truth one. This effect can limit the precision level of the algorithms performance evaluation, and of the parameters estimation.

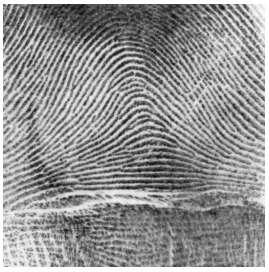
Figure 1.3 display the digital images used in this thesis. These images are widely used in the image processing community. The interested reader can download them

(a) Lena

(b) Barbara

(c) Couple

(d) Hill

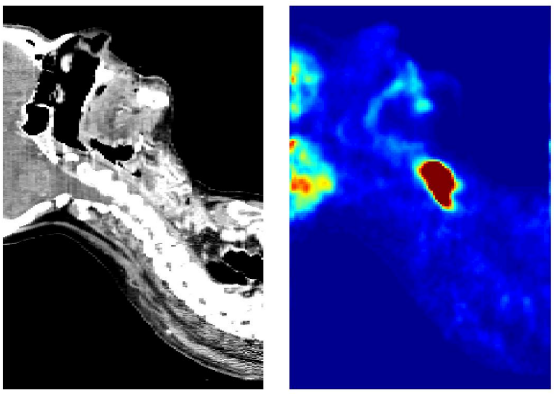


(e) Fingerprint

(f) Boat

(g) House

Figure 1.3: Digital pictures traditionally used in the image processing community



(a) CT scan image. (b) PET scan image.

Figure 1.4: Example of CT and PET images

on the site of the BM3D algorithm: `http://www.cs.tut.fi/~foi/GCF-BM3D/`.

Medical images are images acquired with a medical scanner. Scanners can have a lot of different specifications and the images vary greatly depending on them, and the acquisition parameters used for it.

In this thesis, the problem of the denoising of Computed tomography (or CT scan) images or Positron emission tomography (or PET scan) images (see Figure 1.4a) is tackled too.

During the CT images acquisition, X-rays are emitted around the patient and detected after they went through him/her. These detections are used to construct a precise 3-dimensional vision of the anatomy of a patient. While this imagery method provides an image of the patient's anatomy, it can be very difficult to spot the tumoral tissues as their density is similar to the density of sane tissues. The details about the process of image acquisition can be found in Chapter 5.

Chapter 2

Statistical filtering for digital image denoising

Digital image denoising is a widely studied problem and many methods have been developed to tackle it.

No matter how good cameras are, digital images are contaminated noise, mainly due to the acquisition settings (light conditions, time of exposure, sensibility of the sensors, ...) and sensors. This noise can follow a simple statistical law (salt and pepper noise, Gaussian noise, Poisson noise, ...) or a more complex pattern (multiplicative noise). Noise is a significant problem in some imaging applications in which the acquisition process cannot be controlled or optimized (minimizing a patient's exposure to radiations, kind of signal to be detected (high energy photons), ...).

Digital images are stored as a set of pixels located on a regular grid. Let I denote the set of grid coordinates that are associated with each pixel of the image. Each pixel can then be uniquely identified on the grid by its coordinate vector $\mathbf{i} \in I$ and we (somewhat abusively) call it the $\mathbf{i}$th pixel. The observed intensity of each pixel then is given by $x_{\mathbf{i}}$; This intensity is a real number. The lower the intensity value, the darker the pixel.

For an image I, The observed intensity of each pixel is given by

$$x_{\mathbf{i}} = y_{\mathbf{i}} + \varepsilon_{\mathbf{i}} \quad , \tag{2.1}$$

where $y_{\mathbf{i}}$ is the noisefree value and $\varepsilon_{\mathbf{i}}$ is the noise component. The latter is independent and identically distributed for each pixel. More specifically, for most denoising meth-

ods, it is assumed that the noise is Gaussian with zero mean and standard deviation ν, that is, $\varepsilon_{\mathbf{i}} \sim G(0,\nu^2)$.

A model with Gaussian i.i.d. noise is too simple to reflect many real life situations. Noise can indeed result from a combination of several physical phenomena and the assumptions of independence and normality are often invalidated in practice. However, the knowledge of an appropriate noise model can lead to a data transformation whose purpose is to make noise (nearly) normal. For example, Fisz's and Anscombe's variance stabilizing transforms convert Poissonian noise into Gaussian noise [39], making classical filters applicable. The use and integration of these variance stabilizing transforms in the statistical filters is developed in Chapter 6. Consequently, Gaussian noise provides a nice, and well understood background from which it is possible to develop and refine the filtering algorithms efficiently.

2.1 Digital Image denoising

Digital image denoising refers to the problem of finding the best possible approximation $\hat{y}_{\mathbf{i}}$ of the true pixel $y_{\mathbf{i}}$ from the observed pixel $x_{\mathbf{i}}$. This definition implies that the quality of the denoised images has to be assessed by a quality criterion. In the image denoising community, the commonly used criteria are:

- Root Mean Square Error (RMSE): defined as

$$\text{RMSE} = \sqrt{\sum_{i\in I} (\hat{y}_{\mathbf{i}} - y_{\mathbf{i}})^2} \ . \tag{2.2}$$

  This is a commonly used quantitative measure of the squared difference between the real image and the denoised image. If many images are used for an experiment, the RMSE is calculated for each image, and the mean RMSE (MRMSE) and standard deviation of the RMSE (stdRMSE) are often used:

$$MRMSE = \frac{1}{M}\sum_{m=1}^{M} RMSE_m, \tag{2.3}$$

  where M is the total number of images used in the experiment, $RMSE_m$ the RMSE of the mth image, $1 \leq m \leq M$.

$$stdRMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M} (RMSE_m - MRMSE)^2} \ . \tag{2.4}$$

- Peak signal to noise ratio (PSNR): another quantitative measure derived from the RMSE. The advantage of the PSNR over the RMSE is that it is independent of the dynamic range of the image. The PSNR is defined as

$$\text{PSNR} = 10\log_{10}\left(\frac{d^2}{RMSE^2}\right) \quad , \tag{2.5}$$

  where d stands for the dynamic range of the image (i.e. for an image where the pixel intensities vary between 0 and 255, $d = 256$).

- Visual analysis of the residuals: the residuals are defined as the filtered image minus the noisy image: $(\hat{y}_\mathbf{i} - x_\mathbf{i})$. The observation of this quantity is a way to analyse the changes that happened on a specific area of the image during the filtering process. Since it does not provide a numerical estimation of the effect of the denoising algorithm, it is mainly used in order to estimate the edge-preserving properties of the denoising algorithm.

- Visual quality: this is the easiest and simplest way to assess an image quality. However it is impossible to quantify the effect of the denoising algorithm, and it is easy to overlook artefacts or residual noise by visual inspection. In this work, the original picture will often be presented cropped in order to give a better view of the details. However, the calculations and filtering performances are always done evaluated on the whole image.

The quantitative methods of evaluation are the best way to assess a filtering algorithm quality. However, since the noise arises during the acquisition, the true image is often not known, making it impossible to calculate the RMSE and PSNR. In Chapter 5, we propose to model the noise induced by a particular imaging machine as a way to overcome this problem. In other chapters, the acquired image will be considered as the true one, and artificial noise will be added to create a noisy picture.

Most denoising algorithms tend to smooth some of the features of the images (corners, edges, gradients, textures, ...). It is then of primary importance to evaluate the capacity of a filter to preserve these features, which can be done by visual inspection only. The filtering algorithms able to preserve the features of the image are called *Edge-preserving denoising algorithms*. They can be derived from different approaches and have been studied extensively in the literature.

2.2 Approaches to image denoising

Many paradigms have been used to remove or attenuate perturbations in order to recover the true image: an early and very popular approach is to achieve filtering in the frequency domain, just by trimming high-frequency components of the image spectrum. This computationally fast method has however a major drawback. It tends to smooth out the salient features of the signal, such as edges and textures.

Wavelets [18, 22, 33] and other transformations in a combined space-frequency domain nicely address this issue. However, the results depend of the choice of wavelet, and of scale used. This last parameter in particular can have a drastic effect on the filtering result.

Heat diffusion has inspired denoising techniques such as anisotropic diffusion [61, 9] and several other techniques based on partial differential equations [73, 74]. However, these methods can be computationally intensive. Recently, wavelets and collaborative filtering were combined to produce BM3D [24], a new filtering algorithm that provides extremely competitive denoising performances.

Total variation denoising (TV) [58, 71, 67] is a special case of image regularization methods that balances a smoothness measure and a fidelity term. Modelling images as random fields [64, 63] provides yet another strategy.

The statistical nature of noise can also be addressed by robust statistics and mode estimation. This is an elegant approach as most noises present on the images can be described by a statistical law (Poisson law for the PET images, Gaussian for the digital images, or other kinds of additive or multiplicative noise for more complicated cases ...). Furthermore, the filters derived from this approach are often quite easy to implement, interpret, and they are composed of simple blocks easy to adapt for a particular situation. For these reasons, the statistical filters are the heart of the work in this manuscript. The most known statistical filters are the mean-shift procedure [20, 23], the local M-smoothers [21, 74] and the bilateral filters [69, 35]. Statistical denoising shows its full power when it is applied on multidimensional data. As in classification tasks, adding dimensions is thought of as a mean to increase the gap between the modes, and therefore the probability to identify them correctly. In the case of image filtering, instead of single pixels, the filtering algorithms use blocks of images called *patches*, which are high-dimensional vectors. The non-local means (NLmeans) [13, 12]), unsupervised information-theoric adaptative filtering (UINTA) [2] and some Bayesian approaches [49] are the most popular of these patch-based filters. Adaptive patch sizes [46] and iterative updates [46, 32] have also been investigated. As statis-

tical filters are at the basis of further developments made in this work, they will be detailed further in the next section.

2.3 Statistical filters

To apply a statistical filtering algorithm on pixels basically consists in replacing each pixel intensity by an average of the surrounding pixel intensities. Doing so is motivated by statistical considerations: if several measurements were available for each pixel, computing their average would yield a good estimate of the noisefree intensity. As pixels unfortunately contain a single measurement, it is not possible to do so. However, provided they share the same noisefree intensity, the average of the neighbouring pixels should be a useful surrogate of the noisefree intensity. If this is not the case, for instance near an intensity jump, then there is a risk of introducing a bias in the filtered values. This justifies that only close neighbours of the pixel to be filtered bring a non-zero contribution in the average. The statistical filters are therefore divided in two families; local and non-local filters. While the local filtering algorithms only use pixels in a local neighbourhood to provide the surrogate, the non-local filters use the pixels of the whole image in the calculations.

Working locally in the image entails the definition of a distance function on the image grid. Let $\|\mathbf{i}-\mathbf{j}\|_p$ denote the distance derived from the L_p norm between the **i**th and **j**th pixels on the grid. This distance will be (somewhat abusively) called 'the distance between **i**th and the **j**th pixels'.

To avoid boundary effects, we consider that the image has a toroidal topology (in the two-dimensional case). In practice, the torus consists of four mirrored copies of the image. If the image depicts symbol `p`, then mirroring leads to $\begin{smallmatrix}\texttt{p}\,\texttt{q}\\ \texttt{b}\,\texttt{d}\end{smallmatrix}$. Next, the top and bottom borders of the resulting image are connected together, and so are its left and right borders. This produces a torus without any image discontinuity.

2.3.1 Isotropic Local filter

Given a distance on the grid, the neighbourhood of the **i**th pixel can be defined as

$$N_{\mathbf{i}} = \{\mathbf{j} \text{ s.t. } \|\mathbf{i}-\mathbf{j}\|_p \leq r\} \ , \tag{2.6}$$

where r is some radius. The toroidal topology ensures that $|N_{\mathbf{i}}| = |N_{\mathbf{k}}|$ for all $\mathbf{i}$ and $\mathbf{k}$. In other words, all pixels have the same number of neighbours.

Isotropic local filtering can then be obtained starting from the L_2 error

$$E_{\text{iso}}(\hat{\mathbf{y}}) = \frac{1}{2}\sum_{\mathbf{i}\in I}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}(\hat{y}_{\mathbf{i}} - x_{\mathbf{j}})^2 \ , \tag{2.7}$$

where $\hat{\mathbf{y}} = [\hat{y}_{\mathbf{i}}]_{\mathbf{i}\in I}$ denotes the denoised image and $w_{\mathbf{ij}}$ is a weight that depends on $\|\mathbf{i}-\mathbf{j}\|_p$. The global minimum of this error function is attained when its gradient vanishes. Therefore, if its partial derivative with respect to $\hat{y}_{\mathbf{i}}$ is equated to zero, then the closed-form solution

$$\hat{y}_{\mathbf{i}} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} x_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}} \tag{2.8}$$

comes out. The usual choice is a softly decaying window such as

$$w_{\mathbf{ij}} = \exp\left(-\frac{\|\mathbf{i}-\mathbf{j}\|_2^2}{2\rho^2}\right) \ = K_\rho(\mathbf{i},\mathbf{j}) \ . \tag{2.9}$$

$K_\rho(\mathbf{i},\mathbf{j})$ is called the *spatial kernel* as this function weights the spatial dependency of the neighbours in the filtering process. Other options may be used as well, such as a hard window, that is, $w_{\mathbf{ij}} = 1$.

In practice, the values of ρ need to be chosen in order to attain a good tradeoff between residual noise variance and bias. In other words, noise should be attenuated without smoothing the underlying signal. Anisotropic variants of local filtering, such as the local M-smoothers and bilateral filtering elegantly simplifies the search for this tradeoff.

2.3.2 Local M-smoother

The local M-smoothers [21, 74, 56] are inspired by robust statistics [44, 43], which aim at deriving reliable estimators even in the presence of outliers. In order to reduce the influence of the latter in the calculations, standard L_2 estimators are refined in such a way that the contribution of outliers saturates. For instance, one can modify (2.7) into

$$E_{\text{LMSs}}(\hat{\mathbf{y}}) = \sum_{\mathbf{i}\in I}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\Psi_\sigma((\hat{y}_{\mathbf{i}} - x_{\mathbf{j}})^2/2) \ , \tag{2.10}$$

where

$$\Psi_\sigma(z^2) = \sigma^2(1-\exp(-z^2/\sigma^2)) \tag{2.11}$$

is chosen as the Leclerc's function, as illustrated in Figure 2.1.

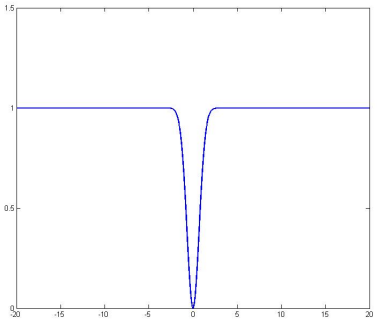


Figure 2.1: Representation of Leclerc's function written in (2.3.2), with $\sigma = 1$

Function E_{LMSs} cannot be minimized in closed-form, for it is no longer quadratic. Instead, a gradient descent approach [51] allows us to write

$$\hat{y}_{\mathbf{i}}^{t+1} = \hat{y}_{\mathbf{i}}^{t} - \alpha^{t} \left. \frac{\partial E_{\text{LMSs}}}{\partial \hat{y}_{\mathbf{i}}} \right|_{\hat{\mathbf{y}}^{t}} \tag{2.12}$$

$$= \hat{y}_{\mathbf{i}}^{t} - \alpha^{t} \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - x_{\mathbf{j}})^2/2)\,(\hat{y}_{\mathbf{i}}^{t} - x_{\mathbf{j}}) \ , \tag{2.13}$$

where t denotes the iteration index and α is the step size.

Imposing

$$\alpha^{t} = 1 \Big/ \sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - x_{\mathbf{j}})^2/2) \tag{2.14}$$

leads to update rule

$$\hat{y}_{\mathbf{i}}^{t+1} = \frac{\sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - x_{\mathbf{j}})^2/2)\, x_{\mathbf{j}}}{\sum_{\mathbf{j} \in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - x_{\mathbf{j}})^2/2)} \ . \tag{2.15}$$

The initialization of (2.15) is given by $\hat{y}_{\mathbf{i}}^{0} = x_i$.

The same update rule can alternatively emerge from a fixed-point approach, wherein the partial derivatives are equated to zero. Function Ψ' in the update rule is called the radiometric or tonal kernel because it modulates the contribution of the neighboring pixels according to their intensity. This dependence on the surrounding pixel intensities makes the filter locally adaptive and anisotropic. Width σ proves to be an additional parameter, compared to isotropic filtering. Its value is typically adjusted

with respect to both the noise standard deviation and the height of the intensity jumps one wants to preserve.

This particular value of the step size (2.14) ensures that the total photometry of the image is approximately preserved. We have

$$\lim_{\sigma\to\infty}\sum_{\mathbf{i}\in I}\hat{y}^t_{\mathbf{i}} = \sum_{\mathbf{i}\in I}x_{\mathbf{i}} \approx \sum_{\mathbf{i}\in I}y_{\mathbf{i}} \tag{2.16}$$

for all t (reminder: $\mathbf{I}$ is the total number of pixels of the image). The left equality is demonstrated in [51]: the effective weights of the neighbours are defined as

$$W_{\mathbf{ij}} = \begin{cases} \frac{w_{\mathbf{ij}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}} \text{ if } \mathbf{i}\in N_{\mathbf{i}} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases}, \tag{2.17}$$

and equation (2.8) can be written as $\hat{y}_{\mathbf{i}} = \sum_{\mathbf{j}=1}^{\mathbf{I}} W_{\mathbf{ij}} x_{\mathbf{i}}$. The left equality in (2.17) holds because

- There are no border effects in the image (due to the toroidal topology of the image).
- The weights and neighbourhoods are symetric: $\mathbf{j}\in N_{\mathbf{i}} \iff \mathbf{i}\in N_{\mathbf{j}}$ and $W_{\mathbf{ij}} = W_{\mathbf{ji}}$.
- The weights are normalized : $\sum_{\mathbf{j}=1}^{\mathbf{I}} w_{\mathbf{ij}} = 1$.

We can now write

$$\sum_{\mathbf{i}=1}^{\mathbf{I}}\hat{y}_{\mathbf{i}} = \sum_{\mathbf{i}=1}^{\mathbf{I}}\sum_{\mathbf{j}=1}^{\mathbf{I}} W_{\mathbf{ij}} x_{\mathbf{j}} \tag{2.18}$$

$$= \sum_{\mathbf{i}=1}^{\mathbf{I}}\sum_{\mathbf{j}=1}^{\mathbf{I}} W_{\mathbf{ji}} x_{\mathbf{j}} \tag{2.19}$$

$$= \sum_{\mathbf{j}=1}^{\mathbf{I}} x_{\mathbf{j}} \sum_{\mathbf{i}=1}^{\mathbf{I}} W_{\mathbf{ij}} \tag{2.20}$$

$$= x_{\mathbf{j}}\,. \tag{2.21}$$

This proves the left equality in 2.17.

The decaying influence of dissimilar pixel intensities makes the local M-smoother robust against outliers as well as an efficient mode estimator. This can be formally demonstrated by observing that the same update rule comes out of a hill-climbing procedure on a Parzen-window density estimator [70, 56]. For this reason, the local M-smoother is particularly good at denoising piecewise constant signals.

2.3.3 Bilateral filter

Bilateral filtering [69, 35, 56] happens to be a slightly different variant of the local M-smoother. More precisely, the bilateral filter compares the intensity of the pixel to be filtered with surrounding filtered intensities instead of the noisy ones. These changes are summarized in update rule

$$\hat{y}_{\mathbf{i}}^{t+1} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\, \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - \hat{y}_{\mathbf{j}}^{t})^2/2)\, \hat{y}_{\mathbf{j}}^{t}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - \hat{y}_{\mathbf{j}}^{t})^2/2)} \quad . \tag{2.22}$$

The initialization remains the same as for the local M-smoother and both techniques yield a different result starting from the second iteration.

While the local M-smoothers are strictly local, the bilateral filter is not: averaging intensities that are already filtered entails a diffusion process. This explains why the bilateral filtering is even better than the local M-smoother at denoising piecewise constant signals, provided width σ is small enough compared to the intensity jumps. If not, apply bilateral filtering for many iterations can eventually lead to a flat image; the diffusion process has to be stopped before this happens. Formally, one can observe that the bilateral filtering minimizes

$$E_{\mathrm{BF}}(\hat{\mathbf{y}}) = \sum_{\mathbf{i}\in I} \sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\, \Psi_{\sigma}((\hat{y}_{\mathbf{i}} - \hat{y}_{\mathbf{j}})^2/2) \quad , \tag{2.23}$$

which admits trivial solution $\hat{y}_{\mathbf{i}} = \hat{y}_{\mathbf{j}}$, for $\mathbf{i},\mathbf{j} \in I$.

2.3.4 Mean-shift

Choosing $N_{\mathbf{i}}$ as the whole image in (2.15) and discarding the spatial kernel denoted by $w_{\mathbf{ij}}$ leads to the mean-shift procedure [20, 23, 70, 4]:

$$\hat{y}_{\mathbf{i}}^{t+1} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - x_{\mathbf{j}})^2/2)\, x_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_{\sigma}((\hat{y}_{\mathbf{i}}^{t} - x_{\mathbf{j}})^2/2)} \quad . \tag{2.24}$$

The mean-shift minimizes the error function

$$E_{\mathrm{MS}}(\hat{\mathbf{y}}) = \sum_{\mathbf{i}\in I} \sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi_{\sigma}((\hat{y}_{\mathbf{i}} - x_{\mathbf{j}})^2/2) \quad . \tag{2.25}$$

The mean-shift is a non-local method as it uses pixels from the whole image to estimate the filtered image. Note that it can use the pixels from the last iteration as in (2.22). To prevent unwanted diffusion, the mean-shift used in this work will only be used in the (2.24) form.

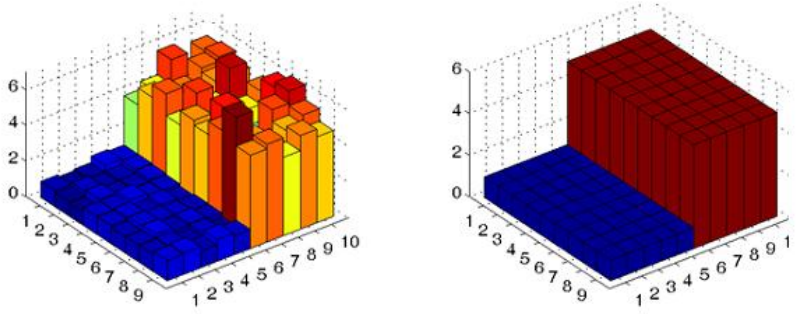


Figure 2.2: Left: Noisy image. Right: ideal, non noisy image.

2.3.5 Illustration: isotropic local filter and the local M-smoother

In practice, the isotropic local filter has a major drawback.: the edges of the images are smoothed by the filtering process. This filtering algorithm only takes the proximity of the pixels into account. Consequently, a pixel belonging to a very different area of the image, but close to the pixel to be filtered will have a big influence in the filtering process. The edges of the different areas of the images will then be smoothed out, which is to be avoided. This smoothing process is illustrated in the rest of this section. Let us imagine a part of image containing an abrupt change in intensity. This image is illustrated on the right of Figure 2.2. Noise is added to this ideal image, the resulting image is shown on the left of Figure 2.2.

When the isotropic local filter is applied to this image, a Gaussian kernel is centered on the selected pixel. This pixel is replaced by a weighted mean of the neighbouring pixels. The weights of the pixels depend of the Gaussian kernel uniquely (see Figure 2.3).

The resulting image after application of the isotropic local filter is show on Figure 2.4. The noise is removed, but the clear edge has been smoothed out.

On the contrary, if a radiometric kernel is used, as in the local M-smoother, bilateral filter or mean shift. The weights applied to the neighbouring pixels are not Gaussian anymore; they are the product of the spatial and radiometric kernel. The pixels with different intensities have a very low weight in the new pixel computation (see Figure 2.5). The resulting filtered image has a much lower noise level than the original noisy image, and the edges of the filtered image are preserved (see Figure 2.6).

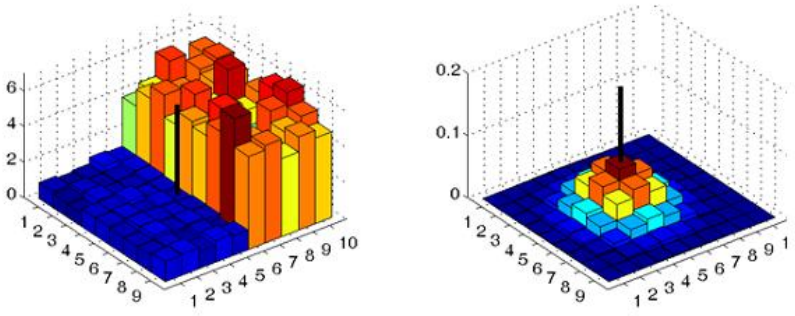


Figure 2.3: Left: selection of one pixel. Right: weights of the surrounding pixels with the isotropic local filter.

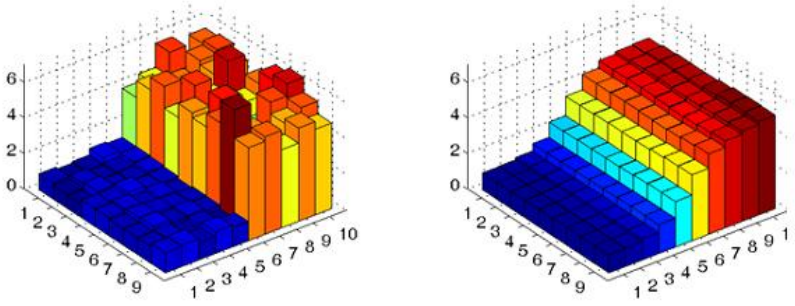


Figure 2.4: Left: noisy image. Right: smoothed filtered image after isotropic filter.

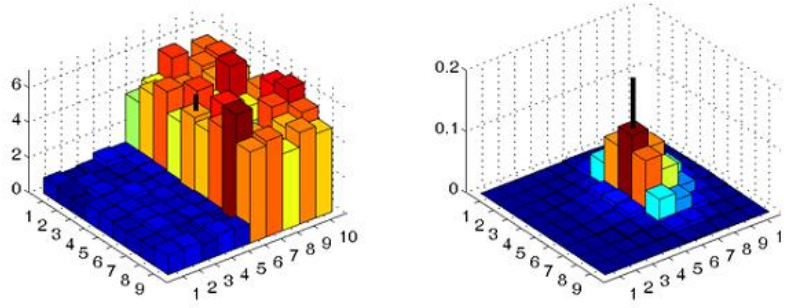


Figure 2.5: Left: noisy image. Right: weights of the surrounding pixels with the local M-smoother.

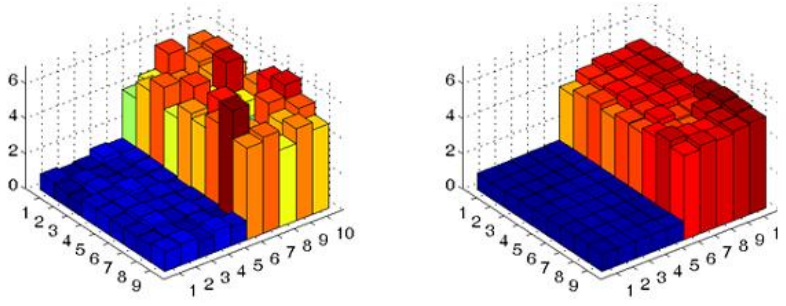


Figure 2.6: Left: noisy image. Right: smoothed filtered image after local-M smoother.

2.4 Conclusion

Image filtering has been intensely studied as it is a required step in a lot of different fields of application. The problem of image denoising can be summarized as trying to recover the true image from the noisy observation. Usually the filters are designed for a noise that is supposed to be Gaussian and additive. There are many paradigms for image filtering, each of them with advantages and drawbacks. In this manuscript, we focus on the statistical filtering methods for their simplicity, modularity, and efficiency. The statistical filters are well understood, simple, and yet powerful methods for image filtering. In practice, the bilateral filter proves to be the most efficient of these filters (this is proved by the experiments of Chapter 3), and in particular on piecewise constant images. The mean-shift has two major drawbacks. First, using the pixels from the whole image can provide biased results. Secondly, calculating the pixels on the whole image can in practice be computationally heavy. To avoid that, most people using non-local methods, end-up using a local constraint in order to limit the computational burden.

Chapter 3

Patch-based image denoising

The previous chapter derives some state-of-the-art statistical filters that minimize a cost function. In this chapter, it will be shown that it is also possible to derive these filters from a mode estimation framework. This framework is the starting point to introduce high-dimensional data into the filters. From a statistical point of view, is it easier to separate the modes in high-dimensional spaces. In images, the high-dimensional data are called *patches*. This chapter derives the state-of-the art patch-based filters and proposes a generalization of these filters. From now on, the non-patch filters presented in chapter 2 will be called *scalar filters*, in contrast to the patch-based filters. The experimental section compares these new filters with the state-of-the art statistical filters on digital pictures.

Figure 3.1 summarizes the content of this chapter: The scalar filter update rules are derived by minimizing an objective function with gradient descent. The patch-based filters are created by simply replacing the scalar intensity comparisons by a patch-to-patch comparison in the update rules; therefore, the link between the objective function and the patch-based update rule is broken. To restore this link, the patchwise filters start with the scalar objective function and the patches are introduced into this function, which is then minimized using gradient descent. The patchwise update rules then minimize the objective function with patches.

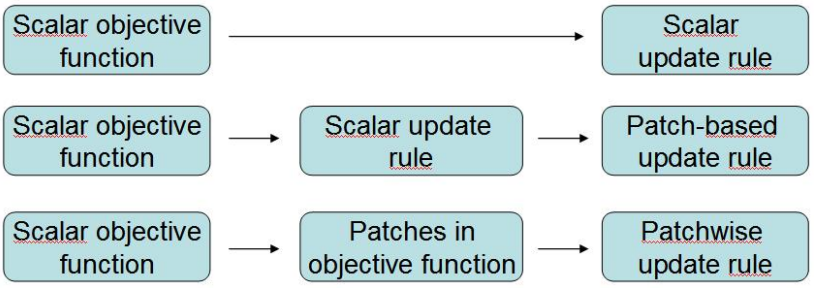


Figure 3.1: Summary of the derivation method for the scalar filters, patch-based filters, and patchwise filters.

3.1 Statistical denoising as a mode estimation task

Mode estimation [60, 20, 23, 70] and robust statistics [44, 43] constitutes a framework from which some of the most studied denoising methods can be easily derived. The underlying assumption is that the noisefree image should consist of a few repeated intensities or patterns, at least locally or temporarily. From a statistical point of view, the data distribution has therefore several modes, whose width depends on the noise standard deviation. Hence, under usual noise assumptions, the top of any mode indicates a location that should correspond to a noisefree pattern. Indeed, if the noisy intensities are evenly distributed around this mode, the top of this mode should provide a surrogate of the true, noise free mode. In practice, mode estimation is achieved by running a hill-climbing procedure [20, 23] on a kernel density estimator [60], as described in the next section.

3.1.1 Scalar filters in the mode estimation framework

Mode estimation is usually done by kernel density estimator methods. The most widely known kernel density estimator is undoubtedly Parzen's window estimator [60]. In the multidimensional case, this nonparametric estimator approximates an unknown probability density function $p(\mathbf{x}_i)$ defined on $\mathbb{R}^D$, from which a finite sample denoted by $\mathbf{X} = [\mathbf{x}_i]_{1 \leq i \leq N}$ is drawn. The estimator can then be written as

$$\hat{p}(\mathbf{x}) = \sum_{i=1}^{N} \frac{1}{NC\sigma^D} k(\|\mathbf{x} - \mathbf{x}_i\|_2 / \sigma) \ , \tag{3.1}$$

where k is a decreasing continuous function of its argument, $\|\cdot\|_2$ denotes the Euclidean norm, C is a normalization factor for k and σ is a bandwidth that controls the estimator smoothness. As the argument of k is an unsigned difference, the resulting function of $\mathbf{x}$ is a radial and local kernel. One usually choose k so that the kernel is a Gaussian function, that is, $k(u) = \exp(-u^2/2)$. For $C = (2\pi)^{D/2}$, this option amounts to building $\hat{p}(\mathbf{x}_i)$ as a mixture of equally weighted Gaussian probability density functions, which is simple and convenient in many respects.

Intuitively, the kernel density estimator transforms the discrete probability density function of the observed sample, which consists of a finite set of Dirac impulses, into a continuous probability density function. For this purpose, each impulse is blurred by replacing it with a weighted, smooth, narrow, and monomodal probability density function, such as a Gaussian one.

There exist of course many refinements of Parzen's window estimator. The main difference lie in the type of kernel or in the bandwidth determination [59, 15, 37]. For instance, the bandwidth can be adapted separately for each kernel, using a pilot estimator.

If the modes of the probability density function $p(\mathbf{x}_i)$ corresponds to its local maxima, then they can be approximated by locating the modes of the kernel density estimator $\hat{p}(\mathbf{x})$. For this purpose, we can run a so-called hill-climbing procedure on $\hat{p}(\mathbf{x}_i)$ [20, 23]. In practice, we can use simple techniques such as a gradient ascent or fixed-point iterations. The different maxima can be reached by changing the initialization point.

Obviously, the kernel density estimator smoothness is critical in this process. If it is too low, then the hill-climbing procedure is likely to get stuck in spurious local maxima, leading to insufficient noise reduction. Running the procedure several times with different initializations around the same actual mode of $p(\mathbf{x}_i)$ shows that the mode estimator has a high variance in this case. In contrast, if the kernel density estimator is too smooth, then close modes can overlap too much and their estimates are biased. Mode seeking requires a smoother kernel density estimator and thus a larger kernel bandwidth than probability density function approximation tasks. Only the accuracy of the mode locations is measured, not the discrepancy between the estimated probability density function and the true one. This section shows how the scalar filters can be obtained starting from a mode estimation framework.

Technically speaking, hill-climbing on a kernel density estimator is very close to robust statistics [44, 43]. Let us define a dataset $\mathbf{X}$ with a sample $\mathbf{x}_i$. In order to

estimate the expectation μ of $p(\mathbf{x}_i)$, one usually computes its sample mean $\hat{\mu}$, that is,

$$\mu \approx \hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \ . \tag{3.2}$$

The sample mean minimizes the sum of squared errors (SSE) written as

$$E(\mathbf{X};\hat{\mu}) = \sum_{i=1}^{N} (\mathbf{x}_i - \hat{\mu})^2 \ . \tag{3.3}$$

If the probability density function is Gaussian, that is

$$p(\mathbf{x}) = \frac{1}{C\sigma} \exp(-\frac{\|\mathbf{x}-\mu\|_2^2}{2\sigma^2}) \ , \tag{3.4}$$

with C being a normalizing constant. Then the log-likelihood of the dataset $\mathbf{X}$ with respect to μ can be written as

$$\ell(\mathbf{X};\mu) = \log \prod_{i=1}^{N} \frac{1}{C\sigma} \exp\left(-\frac{\|\mathbf{x}_i-\mu\|_2^2}{2\sigma^2}\right) \tag{3.5}$$

$$= \sum_{i=1}^{N} -\log(C\sigma) - \frac{\|\mathbf{x}_i-\mu\|_2^2}{2\sigma^2} \ . \tag{3.6}$$

If $p(\mathbf{x}_i)$ is not Gaussian, the SSE is no longer optimal. As a typical case, let us consider a sample $\mathbf{X}$ that should be Gaussian but is actually contaminated by some outliers. To account for tails that are heavier than expected, we can recompute the log-likelihood for Student's t distribution with F degrees of freedom [72]. We have

$$\ell(\mathbf{X};\mu) = \log \prod_{i=1}^{N} \frac{1}{S\sigma^D} \left(1 + \frac{\|\mathbf{x}_i-\mu\|_2^2}{F\sigma^2}\right)^{-\frac{F+D}{2}} \tag{3.7}$$

$$= \sum_{i=1}^{N} -\log(S\sigma^D) - \frac{F+D}{2} \log\left(1 + \frac{\|\mathbf{x}_i-\mu\|_2^2}{F\sigma^2}\right) \ . \tag{3.8}$$

where

$$S = (\pi F)^{D/2} \frac{\Gamma(F/2)}{\Gamma((F+D)/2)} \ . \tag{3.9}$$

is a normalization factor. Maximizing the likelihood in this case amounts to minimizing a sum of errors given by

$$E(\mathbf{X};\hat{\mu}) = \sum_{i=1}^{N} \log\left(1 + \frac{\|\mathbf{x}_i-\hat{\mu}\|_2^2}{F\sigma^2}\right) \ . \tag{3.10}$$

Each term is nearly quadratic close to $\mathbf{x}_i$ but it tends to be logarithmic when moving away from $\mathbf{x}_i$. This sum of errors cannot be minimized in closed form. Yet, equating the derivative with respect to $\hat{\mu}$ with zero leads to an iterative fixed-point update that quickly converges on a solution.

Accordingly, the estimate of the mode that lies the closest from $\mathbf{x}_i$ after the t-th iteration is given by

$$\hat{\mu}^{(t)} = \frac{\sum_{i=1}^{N} \left(1 + \frac{\|\mathbf{x}_i - \hat{\mu}^{(t-1)}\|_2^2}{F\tilde{\sigma}^2}\right)^{-\frac{F+D}{2}} \mathbf{x}_i}{\sum_{i=1}^{N} \left(1 + \frac{\|\mathbf{x}_i - \hat{\mu}^{(t-1)}\|_2^2}{F\tilde{\sigma}^2}\right)^{-\frac{F+D}{2}}} \quad , \tag{3.11}$$

where $\tilde{\sigma}$ is an oracle for the true unknown standard deviation σ, which does not vanish as in the usual quadratic estimator. The above iterative estimator turns out to be a weighted estimated, that is, a sample mean with unequally weighted terms. As can easily be seen, weights associated with outliers have a reduced influence, hence justifying the robustness of this kind of estimators [44, 43]. In order to identify the mode that lies the closest from each $\mathbf{x}_i$, one runs the procedure N times with the successive initializations $\hat{\mu}^{(0)} = \mathbf{x}_i$.

The idea of reducing the influence of outliers by a adapted error function can be pursued one step further by writing

$$E(\mathbf{X};\hat{\mu}) = \sum_{i=1}^{N} \left(1 - \exp\left(-\frac{\|\mathbf{x}_i - \hat{\mu}\|_2^2}{2\sigma^2}\right)\right) \quad , \tag{3.12}$$

where each term is a Leclerc function (or upside-down Gaussian function).

From a log-likelihood perspective, equation (3.12) rewrites as

$$\ell(\mathbf{X};\mu) = \log \prod_{i=1}^{N} \frac{1}{C\sigma} \exp\left(-1 + \exp\left(-\frac{\|\mathbf{x}_i - \mu\|_2^2}{2\sigma^2}\right)\right) \tag{3.13}$$

$$= \sum_{i=1}^{N} -\log(C\sigma) - \exp\left(-\frac{\|\mathbf{x}_i - \mu\|_2^2}{2\sigma^2}\right) \quad . \tag{3.14}$$

The functional involved in this log-likelihood is not a proper probability density function, because

$$\lim_{u \to \infty} \exp(-1 + \exp(-u^2/2)) = \exp(-1) \quad , \tag{3.15}$$

and thus

$$\int_{-\infty}^{\infty} \exp(-1 + \exp(-u^2/2)) du = \infty \quad . \tag{3.16}$$

In practice, however, we can adaptively trim the functional domain for abscissae below $\min_{1\leq i\leq N}\mathbf{x}_i$ and beyond $\max_{1\leq i\leq N}\mathbf{x}_i$. In this way, the inequality

$$\int_{\min_{1\leq i\leq N}\mathbf{x}_i}^{\max_{1\leq i\leq N}\mathbf{x}_i}\exp(-1+\exp(-\|\mathbf{x}-\mathbf{x}_i\|_2^2/2))d\mathbf{x}<\infty\ . \tag{3.17}$$

ensures that a finite normalization factor exists to build an actual probability density function, which looks like a uniform distribution between $\min_{1\leq i\leq N}\mathbf{x}_i$ and $\max_{1\leq i\leq N}\mathbf{x}_i$, topped with a nearly Gaussian bump centered on x_i. In this way, the effective part of the tails is heavier than for Student's t distribution. This allows the estimator to be even more robust to outliers.

The fixed-point update corresponding to (3.12) is

$$\hat{\mu}^{(t+1)}=\frac{\sum_{i=1}^{N}\exp\left(-\frac{\|\mathbf{x}_i-\hat{\mu}^{(t)}\|_2^2}{2\tilde{\sigma}^2}\right)\mathbf{x}_i}{\sum_{i=1}^{N}\exp\left(-\frac{\|\mathbf{x}_i-\hat{\mu}^{(t)}\|_2^2}{2\tilde{\sigma}^2}\right)}\ , \tag{3.18}$$

where the weights associated with the outliers decrease even faster than in the previous case. The last definition of $E(\hat{\mu};\mathbf{X})$ (equation (3.12)) is equivalent to a kernel density estimator with Gaussian kernels, up to a linear transformation. It is also noteworthy that with small changes ($\hat{\mu}^{(t)}=\hat{y}_{\mathbf{j}}^{(t)}$ and Ψ'_σ chosen as a Gaussian kernel), this equation reduces to the expression of the mean-shift shown in equation (2.24). If the local constraint w_{ij} is introduced into equation (3.12), the local M-smoother comes out of this framework. This illustrates how the statistical filters can be derived from mode estimation.

The above comparison yields some interesting conclusions. Though technically similar, hill-climbing on a kernel density estimator and robust statistics follow quite different philosophies. The kernel density estimator aims at approximating as well as possible an unknown probability density function $p(\mathbf{x}_i)$ in a nonparametric fashion. Usually, a quality criterion ensures that good accuracy is achieved everywhere, in high as well as low density regions. Minimizing the discrepancy between $p(\mathbf{x}_i)$ and $\hat{p}(\mathbf{x}_i)$ has the priority over smoothness. In contrast, the previous paragraph shows that the kernel type matters in mode estimation and numerous kernels have been investigated in the literature by Huber, Welsh and Leclerc, Geman and McClure, and Tukey [7, 8]. Smoothness also proves to be important: robust estimators detailed above involve an oracle for the unknown width σ of the sought mode, giving to mode estimation the flavor of a paramatric method. In kernel density estimators, the bandwidth is usually much smaller.

The usual interpretation of the kernel density estimator is that its value at $\mathbf{x}$ is determined by the sum of the contributions of all narrow kernels centered on each data point $\mathbf{x}_i$. In densely populated regions, the kernels overlap and the sum of contributions increases. A slightly different interpretation can be provided for mode estimation, which focuses only on the local maxima of the kernel density estimator $\hat{p}(\mathbf{x}_i)$. The kernel value used in the robust estimator is then a way to reflect the similarity between $\mathbf{x}$ and $\mathbf{x}_i$, as its value decreases when $\mathbf{x}$ moves away from $\mathbf{x}_i$.

3.2 Patch-based filters

The previous section showed how the scalar filters are in fact a mode estimation task: in the filtering process, the pixel values are replaced with the most probable mode in their neighbourhood. However, mode separation is often more salient in high-dimensional spaces. For this reason, in this section, the statistical filters presented in Chapter 2 are extended in order to use high-dimensional data in the mode estimation task.

Equation (3.18) estimates the modes using a probability density function estimate built from single pixels. In this case, the probability density function estimate space is 1-dimensional. The dimensionality of this space can be increased if the kernel probability density estimation is evaluated using a small block of pixels adjacents to pixel $x_{\mathbf{i}}$, called *patches*. More formally, we define the patch centered on the $\mathbf{i}$th pixel as

$$\mathbf{x}_{\mathbf{i}} = \{\mathbf{x}_{\mathbf{j}} \text{ s.t. } \|\mathbf{i}-\mathbf{j}\|_\infty \leq D\} \tag{3.19}$$

$$= \{\mathbf{x}_{\mathbf{j}} \in P_{\mathbf{i}}\} \ . \tag{3.20}$$

The intensities observed in patch $\mathbf{x}_{\mathbf{i}}$ are denoted by $\mathbf{x}_{\mathbf{i}}^k, 1 \leq k\ D$ with D the total number of pixels in the patches (i.e. a 5×5 patch has 25 pixels, and a dimensionality of 25). The $\mathbf{x}_{\mathbf{i}}^1$ is the upper left pixel of the patch $\mathbf{x}_{\mathbf{i}}$, and $\mathbf{x}_{\mathbf{i}}^P$ is the lower right pixel. The L_∞ norm (or city-block distance) ensures that $\mathbf{x}_{\mathbf{i}}$ represents square blocks in the image, but any other choice is possible. The toroidal topology of the image (or an appropriate padding) guarantees that a patch can be constructed for each pixel. The square brackets indicate that intensities are arranged in a vector, always in the same order. Similarly, symbols $\mathbf{y}_{\mathbf{i}}$ and $\hat{\mathbf{y}}_{\mathbf{i}}$ denote the noisefree value of the $\mathbf{i}$th patch and its estimate, respectively.

The representation of patches by vectors allows us to compute the pairwise dis-

tances in the usual (Euclidean) way:

$$d(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{\sum_{k=1}^{D} (\mathbf{x_i}^k - \mathbf{x_j}^k)^2} = \|\mathbf{x_i} - \mathbf{x_j}\|_2 \ , \tag{3.21}$$

with D being the dimensionality of the patches.

3.2.1 Patches in the litterature

In the recent years, filtering in the spatial domain has deeply evolved thanks to the introduction of patches. Extending filters to patches has considerably improved their denoising power. Examples of patch-based filters are the non-local means (NLmeans) [13, 12]), unsupervised information-theoric adaptative filtering (UINTA) [2] and Bayesian approaches [49]. Adaptive patch sizes [46] and iterative updates [46, 32, 42] have also been investigated.

Other developments of the non-local means are related to kernel regression [19], probabilistic weights [32], efficient implementations with tree structures [11] or dictionaries [49]. The connections between bilateral filtering, the mean-shift procedure and (iterative) NLmeans on one side, and with diffusion processes, partial differential equations, and spectral graph theory on the other side are pointed out in [41, 40]. Robust kernels [42], variable width kernels [3] and specific patch-to-patch distances [49, 68] are also reported in the literature.

While the filtering results of these patch-based filters significantly improves the filtering results, the introduction of the patches is usually heuristic and the resulting filter does not minimize an error function as the scalar filters did. The optimization procedure is usually not convincing (heuristic variance/bias trade-off controls the size of the patch, ...). The aim of this manuscript is to introduce the patches in a solid, theoretically sound way.

In this chapter, we propose two different ways of introducing patches in the statistical filters. The first one produces two new filters, and the already widely used non-local means filter. The in this filters are called *patch-based* filters. The second one produces the *patchwise filters* family, which contains three new filters. The patchwise filters are shown to be a generalization of the patch-based filters, with a sound theoretical justification.

3.2.2 Introducing the patches in the statistical filters

In this section patches are introduced into the scalar statistical filters presented in Chapter 2. The easiest and most intuitive way of introduce the patches in the statistical filters is to start from update rules (2.15) and (2.22). The pixel-to-pixel distance $\|\hat{y}^t_{\mathbf{i}} - x_{\mathbf{j}}\|_2^2$ is then replaced with a patch-to-patch distance $\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2$, as in

$$\hat{y}^{t+1}_{\mathbf{i}} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2) x_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2)} \tag{3.22}$$

and

$$\hat{y}^{t+1}_{\mathbf{i}} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \hat{\mathbf{y}}^t_{\mathbf{j}}\|_2^2/2) \hat{y}^t_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \hat{\mathbf{y}}^t_{\mathbf{j}}\|_2^2/2)} \quad , \tag{3.23}$$

for the local M-smoother and bilateral filter, respectively. This modification introduces patch comparisons in the tonal kernel Ψ'_σ. Choosing a patch size equal to one trivially brings back the usual scalar filters.

If $w_{\mathbf{ij}} = 1$ over the whole image, the considered filters correspond to the mean-shift filter, and the patch-based versions of this filter are

$$\hat{y}^{t+1}_{\mathbf{i}} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2) x_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2)} \tag{3.24}$$

and

$$\hat{y}^{t+1}_{\mathbf{i}} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \hat{\mathbf{y}}^t_{\mathbf{j}}\|_2^2/2) \hat{y}^t_{\mathbf{j}}}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \hat{\mathbf{y}}^t_{\mathbf{j}}\|_2^2/2)} \quad . \tag{3.25}$$

At the first iteration, equations (3.24) and (3.25) reduce to the non-local means [13]. To limit the diffusion process, only (3.24) will be used in the rest of the thesis. From now on, update rule 3.25 will be named *patch-based non-local means*.

3.3 Patchwise filtering

The introduction of patches directly into update rules (3.22), (3.23), (3.24), (3.25) breaks the connection with the objective functions E_{LMSs} (2.10), E_{BF} (2.23) and E_{MS} (2.25): one can no longer claim that these modified update rules actually optimize these error functions.

In the case of the bilateral filter, an attempt to relate patch-based filtering to an error function can be found in [50]. Unfortunately, it involves hybrid patches that mix together noisy constant intensities coming from $\mathbf{x}_{\mathbf{i}}$ and a central varying pixel taken from $\hat{\mathbf{y}}^t_{\mathbf{i}}$. A similar approach is developed in [46].

As an alternative approach, the next section shows how to introduce patches in the E_{LMSs} (2.10) and E_{BF} (2.23) error functions rather than in the update rules, and derive new update rules from these objective functions. The objective is to obtain patches in statistical filters that minimize a well identified error function. For reasons that will be explained in the next sections, these methods are called *patchwise* filters, in opposition with the *patch-based* filters of equations 3.22, 3.23 and 3.24.

3.3.1 Patchwise local M-smoother and bilateral filter

In the case of the local M-smoothers, let us start with the modified error function

$$E_{\text{PWLMSs}}(\hat{\mathbf{y}}) = \sum_{\mathbf{i}\in I}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\Psi_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2) \ . \tag{3.26}$$

Next, we minimize it by gradient descent. For this purpose, the gradient of the distance between two patches is written

$$\frac{\partial}{\partial y_{\mathbf{k}}}\frac{\|\mathbf{y}_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2}{2} = \begin{cases} y_{\mathbf{k}} - x_{\mathbf{j}+\mathbf{k}-\mathbf{i}} & \text{if } \mathbf{k}\in P_{\mathbf{i}} \\ 0 & \text{otherwise} \end{cases} \ . \tag{3.27}$$

The index of $\mathbf{x}$ is visually explained in Fig. 3.2.

Therefore, the partial derivative of E_{PWLMSs} with respect to $\hat{y}_{\mathbf{k}}$ is given by

$$\frac{\partial E_{\text{PWLMSs}}}{\partial \hat{y}_{\mathbf{k}}} = \sum_{\mathbf{i}\in P_{\mathbf{k}}}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\,\Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2)(\hat{y}_{\mathbf{k}} - x_{\mathbf{j}+\mathbf{k}-\mathbf{i}}) \tag{3.28}$$

and the gradient descent can be written as

$$y_{\mathbf{k}}^{t+1} = y_{\mathbf{k}}^t - \alpha^t \left.\frac{\partial E_{\text{PWLMSs}}}{\partial \hat{y}_{\mathbf{k}}}\right|_{\mathbf{y}^t} \tag{3.29}$$

$$= \left(1 - \alpha^t \sum_{\mathbf{i}\in P_{\mathbf{k}}}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\,\Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \mathbf{x}_{\mathbf{j}}\|_2^2/2)\right) y_{\mathbf{k}}^t + \alpha^t \sum_{\mathbf{i}\in P_{\mathbf{k}}}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\,\Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \mathbf{x}_{\mathbf{j}}\|_2^2/2)\, x_{\mathbf{j}+\mathbf{k}-\mathbf{i}} \ . \tag{3.30}$$

To preserve the global level of the image, the step size is fixed according to

$$\alpha^t = 1 \Bigg/ \sum_{\mathbf{i}\in P_{\mathbf{k}}}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\,\Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \mathbf{x}_{\mathbf{j}}\|_2^2/2) \tag{3.31}$$

leads to the update rule given by

$$y_{\mathbf{k}}^{t+1} = \frac{\sum_{\mathbf{i}\in P_{\mathbf{k}}}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\,\Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \mathbf{x}_{\mathbf{j}}\|_2^2/2)\, x_{\mathbf{j}+\mathbf{k}-\mathbf{i}}}{\sum_{\mathbf{i}\in P_{\mathbf{k}}}\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\,\Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \mathbf{x}_{\mathbf{j}}\|_2^2/2)} \ . \tag{3.32}$$

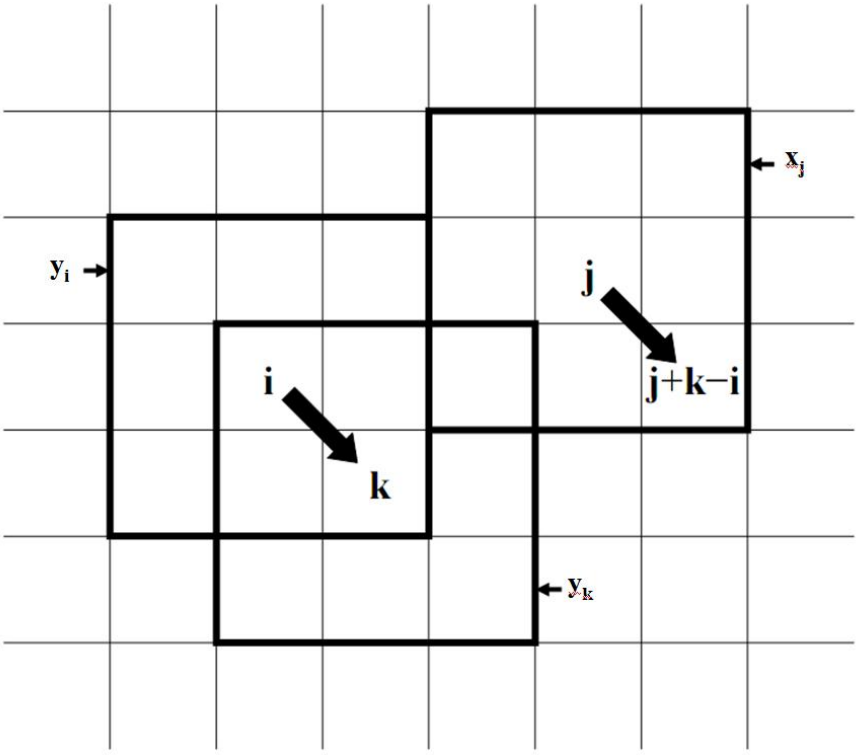


Figure 3.2: Differentiating $\|\mathbf{y_i} - \mathbf{x_j}\|_2^2$ with respect to $y_\mathbf{k}$ is zero except if $\mathbf{k} \in P_\mathbf{i}$. In this case, shifting from $\mathbf{i}$ to $\mathbf{k}$ towards $\mathbf{j}$ allows us to determine which index in $P_\mathbf{j}$ is involved in the partial derivative.

As to patch-based bilateral filtering, the same reasoning starting from

$$E_{\text{PWPBBF}}(\hat{\mathbf{y}}) = \sum_{\mathbf{i}\in I} \sum_{\mathbf{j}\in N_\mathbf{i}} w_{\mathbf{ij}} \Psi_\sigma(\|\hat{\mathbf{y}}_\mathbf{i} - \hat{\mathbf{y}}_\mathbf{j}\|_2^2/2) \ . \tag{3.33}$$

leads to

$$y_\mathbf{k}^{t+1} = \frac{\sum_{\mathbf{i}\in P_\mathbf{k}} \sum_{\mathbf{j}\in N_\mathbf{i}} w_{\mathbf{ij}} \Psi'_\sigma(\|\hat{\mathbf{y}}_\mathbf{i}^t - \hat{\mathbf{y}}_\mathbf{j}^t\|_2^2/2)\, \hat{y}_{\mathbf{j}+\mathbf{k}-\mathbf{i}}^t}{\sum_{\mathbf{i}\in P_\mathbf{k}} \sum_{\mathbf{j}\in N_\mathbf{i}} w_{\mathbf{ij}} \Psi'_\sigma(\|\hat{\mathbf{y}}_\mathbf{i}^t - \hat{\mathbf{y}}_\mathbf{j}^t\|_2^2/2)} \ . \tag{3.34}$$

For both patch-based update rules, the initialization remains unchanged, i.e. $y_\mathbf{k}^0 = x_\mathbf{k}$ for all $\mathbf{k} \in I$. Scalar local M-smoothers and bilateral filters update rules are found back if the patch radius is set to zero. Just as with the previously described patch-

based filters, extending $N_{\mathbf{i}}$ to the whole image and letting $w_{\mathbf{ij}} = 1$ leads to a patchwise iterative version of the non-local means [13] as described in the next section.

3.3.2 Patchwise non-local means

As in section 3.3.1, non-locality can be achieved by choosing $w_{ij} = 1$ on the whole image. The error function 3.26 then writes

$$E_{\text{PWNLMEANS}}(\hat{\mathbf{y}}) = \sum_{\mathbf{i}\in I} \sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2) \ . \tag{3.35}$$

The derivation used in section 3.3.1 can be applied to this objective function to derive a patchwise version of the non-local means algorithm:

$$y_{\mathbf{k}}^{t+1} = \frac{\sum_{\mathbf{i}\in P_{\mathbf{k}}} \sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \mathbf{x}_{\mathbf{j}}\|_2^2/2)\, \mathbf{x}_{\mathbf{j}}}{\sum_{\mathbf{i}\in P_{\mathbf{k}}} \sum_{\mathbf{j}\in N_{\mathbf{i}}} \Psi'_\sigma(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \mathbf{x}_{\mathbf{j}}\|_2^2/2)} \ . \tag{3.36}$$

As in Section 3.3.1, the patchwise version of the non-local means is related to the classical non-local means in a simple way: if only the term associated with the central pixel of the patch is kept in the update rule, the patchwise non-local means reduces to the classical non-local means found in the literature. Note that the non-local means is traditionally used in a non-iterative way. Equation (3.36) supposes that this update rule is used iteratively, which can, in the case of a non-local filter, cause unwanted diffusion. In this work, we decide to use only (3.36) to minimize the effects of diffusion while using this algorithm iteratively. Update rule (3.36) will then be called *patchwise non-local means*.

In this thesis, filters (3.22), (3.23) and (3.24) will be referred to as *patch-based filters*, in opposition with filters (3.32), (3.34) and (3.36) that are called *patchwise patch-based filters*. The reason behind this name choice is that in the patch-based filters, the outer sum over $P_{\mathbf{k}}$ is dropped and only the term associated with the central pixel of the patch remains: the patch-based filters update one pixel at a time. The patchwise patch-based filters, however, change the value of the whole patch during one single iteration. The patch-based filters can thus be considered as approximations of the patchwise ones.

3.3.3 Computational complexity and implementation

At first glance, the patchwise updates seem to require more operations than the patch-based filters, because they involve an additional sum. In order to show that this in-

tuition is false, let us first compute the complexity of the local M-smoother and the bilateral filter without patches. Updates (2.15) and (2.22) are to be applied to whole image and are thus repeated $|J|$ times. Each update requires $O(|N_\mathbf{i}|)$ operations, in order to compute all filter weights and their sum in both their numerator and denominator. This leaves us with a total time complexity of $O(|J|\,|N_\mathbf{i}|)$.

The introduction of patches in the tonal kernel, such as in the pixelwise updates, increases the number of operations to compute each weight of the filter. Computing the patch-to-patch distance has a complexity of $O(|P_\mathbf{i}|)$, instead of $O(1)$ previously. This makes the total time complexity equal to $O(|J|\,|N_\mathbf{i}|\,|P_\mathbf{i}|)$.

As to the patchwise updates, let us first notice that the terms in the numerator and denominator of (3.32) and (3.34) depend on indexes $\mathbf{i}$ and $\mathbf{j}$. A given pair $(\mathbf{i},\mathbf{j})$ occurs in the computation of several filtered intensities. More precisely, looking once again at Figure 3.2 shows that this pair is involved in $\hat{y}_\mathbf{k}$ for $\mathbf{k} \in P_\mathbf{i}$. Starting from this observation, we rewrite the filtered intensity in (3.32) and (3.34) as

$$\hat{y}_\mathbf{i}^{t+1} = v_\mathbf{i}/n_\mathbf{i} \ , \tag{3.37}$$

where temporary variables $v_\mathbf{i}$ and $n_\mathbf{i}$ accumulate the terms of the numerator and denominator, respectively. Next, let us assume that these variables already account for all terms except those related to index pair $(\mathbf{i},\mathbf{j})$, with $\mathbf{i}$ fixed and $\mathbf{j}$ running in $N_\mathbf{i}$. Then, in the case of the local M-smoother, we accumulate the last terms by writing

$$n_\mathbf{i} \leftarrow n_\mathbf{i} + \sum_{\mathbf{j}\in N_\mathbf{i}} w_{\mathbf{ij}}\, \Psi'_\sigma(\|\hat{\mathbf{y}}_\mathbf{i}^t - \mathbf{x}_\mathbf{j}\|_2^2/2) \tag{3.38}$$

$$\mathbf{v}_\mathbf{i} \leftarrow \mathbf{v}_\mathbf{i} + \sum_{\mathbf{j}\in N_\mathbf{i}} w_{\mathbf{ij}}\, \Psi'_\sigma(\|\hat{\mathbf{y}}_\mathbf{i}^t - \mathbf{x}_\mathbf{j}\|_2^2/2)\, \mathbf{x}_\mathbf{j} \ , \tag{3.39}$$

where $\mathbf{n}_\mathbf{i}$ and $\mathbf{v}_\mathbf{i}$ are as usual the patch values centred on the $\mathbf{i}$th pixel. If other terms than those associated to the given value of $\mathbf{i}$ are missing, then the two last updates can be applied for them as well. Eventually, we conclude that the updates can be applied for all $\mathbf{i} \in J$, starting from the initialization $n_\mathbf{i} = 0, v_\mathbf{i} = 0$. This means that computing (3.38) and (3.39) for all $\mathbf{i} \in J$, followed by (3.37) for all $\mathbf{i} \in J$ is equivalent to update (3.32). A similar reasoning can be followed for the bilateral filter. In both cases, the computational complexity is distributed as follows. The computation of the patch-to-patch distance requires $O(|P_\mathbf{i}|)$ operations. This leaves us with a total time complexity of $O(|J|\,|N_\mathbf{i}|\,|P_\mathbf{i}|)$ in order to update $n_\mathbf{i}$ for all $\mathbf{i}$. On the other hand, updating $v_\mathbf{i}$ entails patch-to-patch additions. Fortunately enough, all patch elements are premultiplied by the same factor, meaning that the patch-to-patch distance and the patch-to-patch

accumulation can be carried out sequentially. Therefore, the time complexity remains the same and experimental runtimes for the patchwise and pixelwise filters are not significantly different.

Provided the patches are rectangular blocks, the patch-to-patch distances and the patchwise update can be sped up by carrying them out with separable convolutions. Moreover, the convolution along each dimension of the image with a constant window can be implemented in linear time, by using accumulators. These computational simplifications have been described in [25] for the patch-to-patch distances; we have implemented the same tricks for the patchwise updates. More precisely, in the one-dimensional case, let us assume that the image is a vector with L pixels, the support of the spatial kernel encompasses $2M+1$ pixels, and that the block-shaped patch contains $2K+1$ pixels. The image must be padded with $2K+M$ pixels at both ends, so that we can fetch pixel intensities x_i and $\hat{y}_i^t$ with $1-2K-M \leq i \leq L+2K+M$; $\hat{y}_i^t$ is initialized to f_i. If $d(i,m) = \|\mathbf{y}_i - \mathbf{x}_{i+m}\|_2^2$ denotes the squared Euclidean patch-to-patch distance, with $-M \leq m \leq M$, then for fixed m we can compute

$$d(1-K,m) = \sum_{k=-K}^{K} (\hat{y}_{1-K+k}^t - x_{1+m+k})^2 \tag{3.40}$$

and for all $1-K < i \leq L+K$ we iterate

$$\begin{aligned} d(i+1,m) = d(i,m) &+ (\hat{y}_{i+1+K}^t - x_{i+1+K+m})^2 \\ &- (\hat{y}_{i+1-K}^t - x_{i+1-K+m})^2 \ . \end{aligned} \tag{3.41}$$

Once this is done, we can obtain the terms involved in (3.38) and (3.39) by computing first

$$n(1,m) = \sum_{k=-K}^{K} w_m \Psi'_\sigma(d(1+k,m)/2) \ , \tag{3.42}$$

$$v(1,m) = \sum_{k=-K}^{K} w_m \Psi'_\sigma(d(1+k,m)/2)\, \hat{u}_{1+m}^t \ , \tag{3.43}$$

and next, for all $1 < i \leq L$,

$$\begin{aligned} n(i+1,m) = n(i,m) &+ w_m \Psi'_\sigma(d(i+1+K,m)/2) \\ &- w_m \Psi'_\sigma(d(i+1-K,m)/2) \ , \end{aligned} \tag{3.44}$$

$$\begin{aligned} v(i+1,m) = v(i,m) &+ w_m \Psi'_\sigma(d(i+1+K,m)/2) \hat{u}_{1+m}^t \\ &- w_m \Psi'_\sigma(d(i+1-K,m)/2) \hat{u}_{1+m}^t \ , \end{aligned} \tag{3.45}$$

where w_m is the spatial weight that corresponds to $w_{\mathbf{ij}}$ with $\mathbf{j}-\mathbf{i}=m$. An outer loop running over $-M \leq m \leq M$ allows us to compute

$$n_i = \sum_{m=-M}^{M} n(i,m) \quad , \quad v_i = \sum_{m=-M}^{M} v(i,m) \ , \tag{3.46}$$

and eventually $\hat{y}_i^{t+1} = v_i/n_i$ for $1 \leq i \leq L$. The sequence of all abovementioned steps amounts to a complexity of $O(LMK)$ in this one-dimensional case. The generalization to two or more dimensions is straightforward but it requires much space and complicated notations. The most noticible difference is the need to repeat the linear convolution for each dimension. The total time complexity is then $O(|I|\,|N_{\mathbf{i}}|\,D)$, where D is the image dimensionality. The complexity is linear in D and thus lower than that in [25] because we perform all subtractions in (3.41) rather than in each call to the kernel function. The complexity does not depend on the patch size and only factor D makes it higher than with scalar filters.

3.4 Experiments

In this section, the filters presented in Chapter 2 and Chapter 3 are compared in order to discuss their respective denoising performances:

- Scalar local M-smoother and bilateral filter,
- Patch-based (PB) local M-smoother, bilateral filter and non-local means,
- Patchwise (PW) local M-smoother, bilateral filter and non-local means.
- SAFIR, a state-of-the-art filter [48, 47]. SAFIR is a patch-based statisical deniosing algorithm that proposes an original statistical patch-based framework for noise reduction. The idea is to minimize an objective nonlocal energy functional involving spatio-temporal image patches. The minimizer has a simple form and is defined as the weighted average of input data taken in spatially-varying neighborhoods. The size of each neighborhood is optimized to improve the performance of the pointwise estimator reduction and preservation of space-time discontinuities.

The experiments are conducted on Lena, Barbara, and Boat images. For each image, the pixel intensities are independently perturbated by an additive Gaussian noise. Three noise levels are retained, with standard deviations equal to $\sigma_{noise} = 5$,

$\sigma_{noise} = 10$ and $\sigma_{noise} = 15$ respectively. Example of noisy pictures are shown in Figure 3.3.

Each experiment involves 5 noisy copies of each image for the optimisation of the parameters t (number of iterations), ρ (spatial kernel width), σ (radiometric kernel width) and p (patch size). For the final evaluation of the mean RMSE, mean PSNR and standard deviation of the PSNR, the filter is applied on 100 repetitions of the noisy image with the optimised values of the parameters.

Tables 3.1, 3.2 and 3.3 present the Root of the Mean Mean Square Error (RMMSE), mean PSNR (MPSNR), standard deviation of the PSNR and the values of the parameters, obtained for all the filtering algorithms and noise levels. The RMMSE is defined as:

$$RMMSE = \sqrt{\frac{1}{M}\sum_{k=1}^{M} MSE_k} \ , \tag{3.47}$$

where MSE_k is the mean square error of the k_{th} image, and M is the total number of repetitions of the image. Introducing the patches in the statistical filters provides the most striking performance gain in all these experiments. In most cases, the optimal denoising result is obtained after the first iteration. This is only the case if the paramters are optimized. When the parameter values have to be estimated, the best solution is often to use two or three iterations, with less denoising at each step than one badly tuned iteration. The reason why the fixed-point filters reach their best denoising value at the first iteration and not after an unlimited number of iteration is that the error function that the filter optimizes is not the RMSE. The minimum of the RMSE and of the error function could be found at different parameter values and number of iterations.

The non-local means obtains slightly less good performances than the patch-based local-M smoothers and bilateral filters. Taking into account pixels in the whole image could cause some bias in the mode estimation. The patchwise local M-smoothers, bilateral filters, and non-local means also provide a steady improvement over their patch-based equivalent, but their performance gain is less striking. This means that the approximation made by filtering the center pixel only is quite good for the choice of parameter used in the experiment. The patchwise filters results are close to those of SAFIR.

Example results for Lena with $\sigma_{noise} = 15$ are presented in Figures 3.4 and 3.5. The observations that can be made on these pictures are representative of the results obtained from the different denoising algorithms. For this reason, only the results for the Lena picture at the highest noise level are displayed. Numerical results are confirmed by the visual quality of the images. There is a visible improvement from the scalar fil-

(a) Original Lena image without noise

(b) Lena image with additive gaussian noise $\sigma_{noise} = 5$

(c) Lena image with additive gaussian noise $\sigma_{noise} = 10$

(d) Lena image with additive gaussian noise $\sigma_{noise} = 15$

Figure 3.3: Lena image with different noise levels

ter to their patch-based counterpoint. The differences between the patch-based filters, patchwise filters and SAFIR are too subtle to be noticed by visual analysis.

3.5 Conclusions

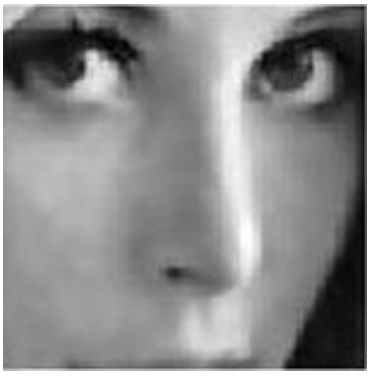
This chapter presents the introduction of the patches in the statistical filters. This introduction is motivated by mode estimation considerations: in high-dimensional spaces, the modes are better separated and it is therefore easier to identify to which mode one pixel belongs. Two methods have been used to introduce the patches in the filters. The first one is simple, but does not minimise an error function, while the second one introduces the patches in new error functions and derives new filters from them. These last filters are shown to be a generalization of the simpler patch-based ones.

Experiments are conducted on three different images, and feature eight variations of the scalar, patch-based, patchwise filters, and in addition SAFIR, a state-of-the-art filter. These nine filters are compared after an optimization of their parameters. All the patch-based filter denoised images show performance gain over the scalar filtered images. Even if the performance gain of the patchwise filters over the patch-based ones is not statistically significant, the patchwise approach is sounder and theoritically better motivated than the patch-based approach. It also provides slightly better results at the same computational cost.

(a) Local M-smoother

(b) Bilateral Filter

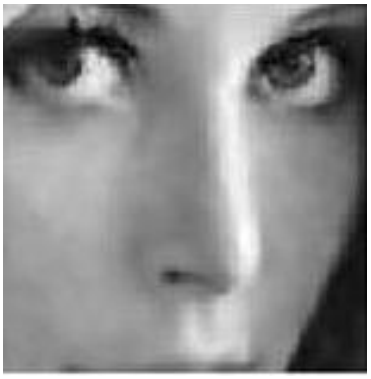
(c) Patch-based local M-smoother



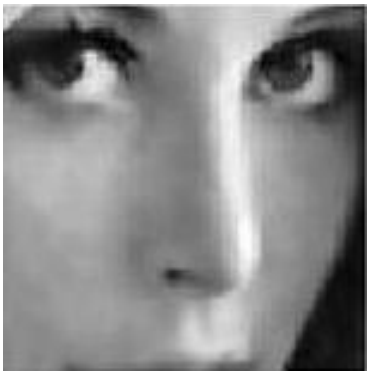
(d) Patch-based bilateral Filter

(e) Non-local means

Figure 3.4: Example results of Lena denoising with different filtering algorithms (part 1)



(a) Patchwise local M-smoother



(b) Patchwise bilateral filter

(c) Patchwise non-local means

(d) SAFIR

Figure 3.5: Example results of Lena denoising with different filtering algorithms (part 2)

$\sigma_{noise} = 5$							
Lena	RMMSE	MPSNR	std(PSNR)	t	σ	ρ	p
Noisy image	5.0032	34.1458	0.0749				
LMSs	3.4073	37.4826	0.0414	1	10.5968	4.8111	1
BF	3.4073	37.4826	0.0414	1	10.5968	4.8111	1
PBLMSs	3.0942	38.3198	0.0410	1	6.0353	4.4747	5
PWPBLMSs	3.0915	38.3274	0.0485	1	5.8082	1.4476	5
PBBF	3.0942	38.3198	0.0410	1	6.0353	4.4747	5
PWPBBF	3.0915	38.3274	0.0485	1	5.8082	1.4476	5
NLmeans	3.1777	38.0885	0.0349	1	5.2161	n.a	3
PWNLmeans	3.1726	38.1025	0.0361	1	5.0932	n.a	3
SAFIR	3.1523	38.1584	0.0371	4	n.a.	n.a.	13
Barbara							
Noisy image	4.9971	34.1564	0.0763				
LMSs	4.0012	36.0870	0.0638	1	9.2913	3.7327	1
BF	4.0012	36.0870	0.0638	1	9.2913	3.7327	1
PBLMSs	3.5035	37.2408	0.0545	1	6.4241	4.8099	5
PWPBLMSs	3.5014	37.2460	0.0585	1	6.4704	4.1992	5
PBBF	3.5035	37.2408	0.0545	1	6.4241	4.8099	5
PWPBBF	3.5014	37.2460	0.0585	1	6.4604	4.1992	5
NLmeans	3.5289	33.2780	0.0507	1	5.2578	n.a	3
PWNLmeans	3.5258	33.2829	0.0523	1	5.0766	n.a	3
SAFIR	3.5598	37.1024	0.031	4	n.a.	n.a.	13
Boat							
Noisy image	4.9967	34.1571	0.0638				
LMSs	3.9241	36.2560	0.0646	1	10.1310	2.9358	1
BF	3.9241	36.2560	0.0646	1	10.1310	2.9358	1
PBLMSs	3.6842	36.8039	0.0763	1	6.4056	3.2491	3
PWPBLMSs	3.6822	36.8087	0.0786	1	6.2953	3.1796	3
PBBF	3.6842	36.8039	0.0763	1	6.4056	3.2491	3
PWPBBF	3.6822	36.8087	0.0786	1	6.2953	3.1796	3
NLmeans	3.7213	36.7169	0.0669	1	5.3213	n.a	3
PWNLmeans	3.7177	36.7253	0.0695	1	5.1682	n.a	3
SAFIR	4.0460	35.9903	0.0431	3	n.a.	n.a.	13

Table 3.1: RMMSE, PSNR, standard deviation of the PSNR, and parameter values for all images with 100 repetitions, perturbated by additive i.i.d. Gaussian noise with standard deviation equal to 5. t = number of iterations, σ = radiometric kernel width, ρ spatial kernel width, p = patch size (in pixels).

$\sigma_{noise} = 10$							
Lena	RMMSE	mean(PSNR)	std(PSNR)	t	σ	ρ	p
Noisy image	10.0011	28.1298	0.0865				
LMSs	5.1218	33.9424	0.0591	1	25.9477	2.9327	1
BF	5.1218	33.9424	0.0591	1	25.9477	2.9327	1
PBLMSs	4.5251	35.0182	0.0542	1	12.3355	2.9156	5
PWPBLMSs	4.5153	35.0371	0.0576	2	8.2242	1.8088	5
PBBF	4.5251	35.0182	0.0542	1	12.3355	2.9156	5
PWPBBF	4.5150	35.0376	0.0590	2	8.2221	1.8301	5
NLmeans	4.6258	34.8271	0.0436	1	10.1293	n.a	5
PWNLmeans	4.6052	34.8658	0.0444	1	9.9683	n.a	5
SAFIR	4.4367	35.1897	0.0343	4	n.a.	n.a.	13
Barbara							
Noisy image	10.0027	28.1285	0.0696				
LMSs	6.0288	32.5262	0.0591	1	26.8511	2.8409	1
BF	6.0288	32.5262	0.0591	1	26.8511	2.8409	1
PBLMSs	5.5583	33.2320	0.0652	1	12.6075	3.4908	5
PWPBLMSs	5.5458	33.2515	0.0591	1	11.6827	3.6703	5
PBBF	5.5583	33.2320	0.0652	1	12.6075	3.4908	5
PWPBBF	5.5458	33.2515	0.0591	1	11.6827	3.6703	5
NLmeans	5.6171	33.1406	0.0494	1	10.6273	n.a	5
PWNLmeans	5.6013	33.1650	0.0429	1	10.1083	n.a	5
SAFIR	5.3935	33.4935	0.0343	3	n.a.	n.a.	13
Boat							
Noisy image	9.9983	28.1323	0.0844				
LMSs	6.8405	31.4290	0.0502	1	19.2729	4.0971	1
BF	6.8405	31.4290	0.0502	1	19.2729	4.0971	1
PBLMSs	5.5042	33.3169	0.0363	1	9.2067	11.2660	5
PWPBLMSs	5.4621	33.3836	0.0415	1	9.3434	8.6191	5
PBBF	5.5042	33.3169	0.0363	1	9.2067	11.2660	5
PWPBBF	5.4621	33.3836	0.0415	1	9.3434	8.6191	5
NLmeans	5.4968	33.3286	0.0346	1	9.5284	n.a	5
PWNLmeans	5.4782	33.3580	0.0362	1	9.1022	n.a	5
SAFIR	5.6425	33.1014	0.0385	3	n.a.	n.a.	13

Table 3.2: RMMSE, PSNR, standard deviation of the PSNR, and parameter values for all images with 100 repetitions, perturbated by additive i.i.d. Gaussian noise with standard deviation equal to 10. t = number of iterations, σ = radiometric kernel width, ρ spatial kernel width, p = patch size (in pixels).

$\sigma_{noise} = 15$								
Lena		RMMSE	mean(PSNR)	std(PSNR)	t	σ	ρ	p
Noisy image	(b)	14.9981	24.6101	0.0826				
LMSs	(c)	6.4146	31.9874	0.0512	1	44.2942	3.2892	1
BF	(d)	6.4146	31.9874	0.0512	1	44.2942	3.2892	1
PBLMSs	(e)	5.5889	33.1843	0.0531	2	12.5253	3.5749	5
PWPBLMSs	(f)	5.5462	33.2509	0.0594	2	11.7869	4.6313	5
PBBF	(g)	5.5887	33.1846	0.0523	2	12.5114	3.5811	5
PWPBBF	(h)	5.5464	33.2506	0.0678	2	11.7814	4.6281	5
NLmeans	(i)	5.7116	32.9956	0.0418	1	13.2817	n.a	5
PWNLmeans	(j)	5.7003	33.0128	0.0421	1	12.4728	n.a	5
SAFIR	(k)	5.5541	33.2389	0.0764	4	n.a.	n.a.	13
Barbara								
Noisy image	(b)	15.0007	24.6086	0.0821				
LMSs	(c)	9.2642	28.7946	0.0677	1	28.5618	4.8083	1
BF	(d)	9.2642	28.7946	0.0677	1	28.5618	4.8083	1
PBLMSs	(e)	7.1058	31.0985	0.0741	1	12.2734	13.0870	5
PWPBLMSs	(f)	7.0581	31.1570	0.0770	1	12.4715	10.3464	5
PBBF	(g)	7.1058	31.0985	0.0741	1	12.2734	13.0870	5
PWPBBF	(h)	7.0581	31.1570	0.0770	1	12.4715	10.3464	5
NLmeans	(i)	7.1158	31.0863	0.0379	1	13.2215	n.a	5
PWNLmeans	(j)	7.0992	31.1066	0.0404	1	12.7798	n.a	5
SAFIR	(k)	6.9016	31.3516	0.0601	4	n.a.	n.a.	13
Boat								
Noisy image	(b)	14.9953	24.6117	0.0979				
LMSs	(c)	7.6514	30.4560	0.0485	1	42.8622	3.0811	1
BF	(d)	7.6514	30.4560	0.0485	1	42.8622	3.0811	1
PBLMSs	(e)	6.9901	31.2411	0.0508	2	14.1054	2.9948	5
PWPBLMSs	(f)	6.9442	31.2984	0.0550	2	12.9953	3.1038	5
PBBF	(g)	6.9895	31.2419	0.0468	2	14.1169	2.9893	5
PWPBBF	(h)	6.9440	31.2986	0.0414	2	13.0046	3.1070	5
NLmeans	(i)	7.3249	30.8348	0.0370	1	13.8132	n.a	3
PWNLmeans	(j)	7.2949	30.8704	0.0332	1	12.0148	n.a	5
SAFIR	(k)	6.999	31.2301	0.0449	4	n.a.	n.a.	13

Table 3.3: RMMSE, PSNR, standard deviation of the PSNR, and parameter values for all images with 100 repetitions, perturbated by additive i.i.d. Gaussian noise with standard deviation equal to 15. t = number of iterations, σ = radiometric kernel width, ρ spatial kernel width, p = patch size (in pixels).

Chapter 4

High dimensionality in the patch-based filters

In the statistical filters based on mode estimation methods, pixels (or patches) are compared using a notion of similarity. This similarity is a function (typically a Gaussian kernel) of the Euclidean distance between the intensities of the patches to be compared. The literature contains many publications that give recommendations about the choice of the optimal kernel and bandwidth [59, 15, 37] for the one-dimensional case (scalar version of the mode estimation filters), but in the high-dimensional cases, the same kernels are usually chosen without much questioning.

This chapter shows that the choice of the Euclidean distance in the patch-based filters has non-negligible consequences on the resulting similarity measure. Indeed, in the patch-based and patchwise filters, the patches are high-dimensional vectors. The Euclidean distance between those patches is measured in a high-dimensional space; it is thus subject to the curse of dimensionality [5, 34] and in particular, to the phenomenon of norm and distance concentration [38]. The norm concentration phenomenon states that when the dimensionality of the space increases, the mean of usual norms and distances increases, but their variance does not change. As a result, the ratio between the variance of the norms and their mean tends towards zero as the dimensionality increases. The practical consequence is that in a high-dimensional space, all distances between points look similar. The discrimination power of the Euclidean norm is thus less important in high-dimensional spaces. As a result, in a high-dimensional space, a useful similarity kernel should be adapted to this phe-

nomenon. This chapter proposes new similarity kernels, designed in order to use the patch-based and patchwise filters in high-dimensional spaces without suffering from the phenomenon of norms and distances concentration.

In the literature, the fact that the use of patches induces the need for a special treatement has already been mentioned. [77] and [24, 57] use PCA to reduce the dimensionality of the patches, and [49] subtracts a distance in the radiometric kernel to reduce the effect of the norm concentration. [36] proposes to improve patch-based denoising with learned dictionaries and redundant information. While these approaches provide good results, they can be seen as clever ways of minimizing the effects of high-dimensionality instead of adapting completely the filters to the high-dimensional case. Indeed, if the filters use high-dimensional data as input, it is elegant and natural to adapt the distance and similarity definitions of these filters to the high-dimensional case. In contrast, plugging some additional transformations or projections of the data in the filters to minimize the effects of the high dimensionality is a way of minimizing the effects of the distance concentration phenomenon, but does not solve the problem in itself: for example the Gaussian kernel is not an adapted similarity measure in high-dimensional spaces. For this reason, we propose to adapt the similarity kernel used in the filters to the case of high-dimensional data. The new similarity kernel takes the distance concentration phenomenon into account and increases the performances of the patch-based and patchwise filters.

4.1 Distance concentration phenomenon

When the distances between patches $\mathbf{x_i}$ and $\mathbf{x_j}$ are calculated in a filter, the considered pixels form a finite sample of data. That special case has to be analyzed: indeed, in this case, the norm of the vectors $\mathbf{x_i}$ and $\mathbf{x_j}$ is a random variable. The realisation of this random variable has a certain distribution. If that random variable is a distance instead of a norm, the distribution stays the same unless when the number of data points is finite. Let us study the distribution of $[\mathbf{x_i}]$, a finite sample of $\mathbf{X}$, with $1 \leq \mathbf{i} \leq N$, in the following practical situation: $\hat{\mu} = \mathbf{x_j}$, and $\mathbf{x_i} \sim G(\hat{\mu}, \sigma\mathbf{I})$. The difference between two Gaussian distribution follows a χ^2 distribution. Thus we can write

$$\frac{\|\mathbf{x_i} - \hat{\mu}\|_2}{\sigma\sqrt{2}} \sim \begin{cases} G(0,0) & \text{if } \mathbf{x_i} = \mathbf{x_j} \quad (\text{with probability } \frac{1}{N}) \\ \chi_D & \text{otherwise} \quad (\text{with probability } \frac{N-1}{N}) \end{cases}, \tag{4.1}$$

where χ_D denotes a chi distribution with D degrees of freedom, and $G(0,0)$ is a zero mean, zero variance Gaussian distribution, which amounts to a single peak placed on

0.

The special case $\mathbf{i} = \mathbf{j}$ has to be considered because $\mathbf{x_i}$ is a finite sample. In this particular case, the pairwise Euclidean distance has a distribution which is a mixture of a zero-mean zero-variance Gaussian distribution with probability $1/N$, and a χ_D distribution with probability $(N-1)/N$. The zero-mean zero-variance Gaussian distribution only exists if the sample is finite.

The distribution corresponding to the finite sample situation can be written as

$$f(u,N,D) = \frac{1}{N}\delta(u) + \frac{N-1}{N}\frac{\sqrt{2}}{\Gamma(D/2)}(u^2/2)^{\frac{D-1}{2}}\exp(-u^2/2) \quad . \tag{4.2}$$

The χ_D part of this function is illustrated in Figure 4.1 for $N = 10$ and several values of D.

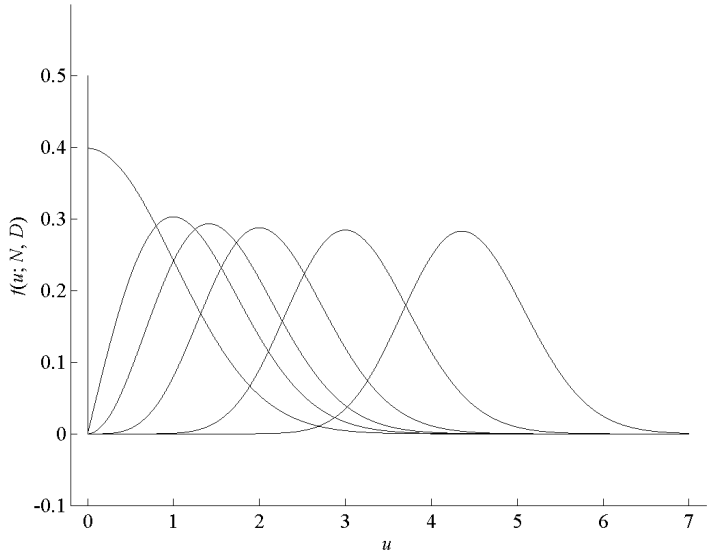


Figure 4.1: χ_D distribution, plotted for $N = 10$ and $D \in \{1,2,3,5,10,20\}$, from left to right.

In high-dimensional spaces, pairwise distances between data points from a finite sample, drawn from the same Gaussian distribution, have a distribution given by $f(u;N,D)$. This probability density function has two modes located in 0 and $\sqrt{D-1}$.

If we neglect the first mode ($G(0,0)$), to simulate a non-finite number of sample points, the expected value and the variance are given by $\mathrm{E}[\chi_D] = \sqrt{2}\Gamma((D+1)/2)/\Gamma(D/2)$ and $\mathrm{Var}[\chi_D] = D - \mathrm{E}^2[\chi_D]$, respectively. As the dimensionality increases, the variance of the bell of the distance distribution stays constant and the top of the bell (average distance) moves away from zero. The distance is said to *concentrate*. As the ratio between the variance and the average distance tends to zero, the points in the bell of $f(u;N,D)$ are considered to be more and more similar to each other.

With these kind of norm distribution, how can one define an efficient similarity measure? Intuitively, an efficient similarity measure should consider as similar the data points with a distance comprised between 0 and the most probable value given by the mode. For higher values of the distance, the similarity should decrease rapidly, following the slope of the right tails in Figure 4.1. A similarity measure is usually a function of the norm that maps the distance between two data points between 0 and 1. Similar (or close) points should have a similarity close to 1; dissimilar points should have a similarity close to 0. The most common choice of similarity measure in the statistical filters is the Gaussian kernel. Intuitively, the slope of a good similarity measure should follow closely the curve of the distance distribution to provide a good estimation of the similarity.

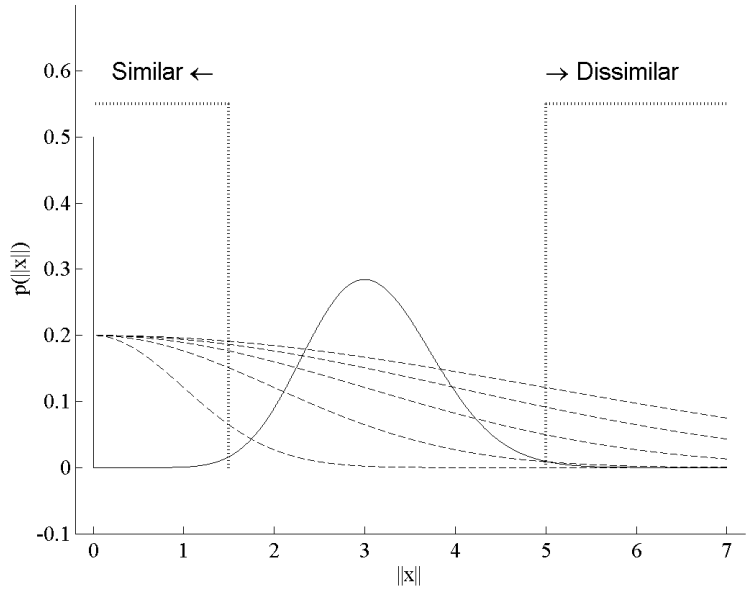
When a Gaussian kernel is used as a similarity measure in a high-dimensional space, it is unable to decrease quickly enough to follow the curve between the main mode of $f(u,N,D)$ and the right tail, whatever the kernel width is. As a result, dissimilar data points are mapped close to 1 and seen as similar.

This is illustrated in Figure 4.2a, in which a Gaussian kernel used as a similarity measure for $f(u,2,10)$ and $\sigma \in \{1,2,3,4,5\}$. For all kernel widths, the slope of the kernel is unable to be steep enough to follow the distribution of the distances. Figure 4.2b illustrates a kernel that has such behaviour. It has a flat top up to left side of the distribution bell, then decreases in a steep slope following the distance distribution shape. For obvious reasons, these kernels are called *flat top* kernels in the rest of this work.

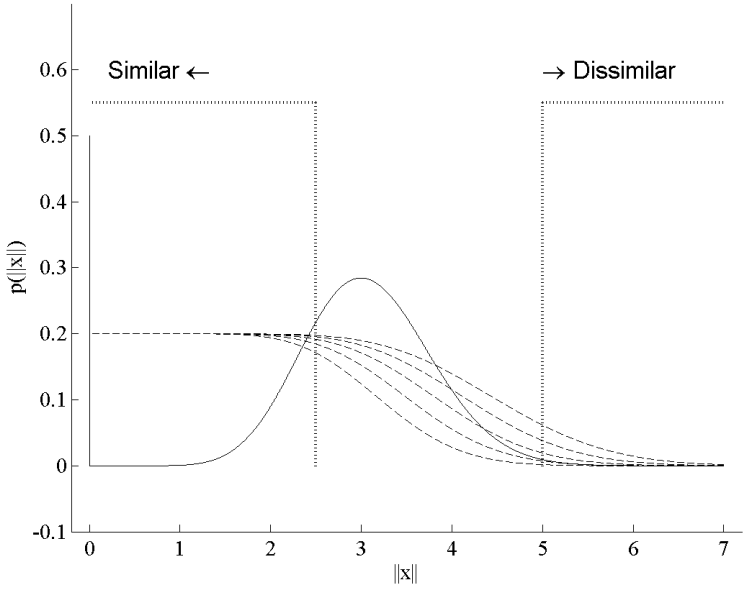
In the next section, several types flat top kernels are proposed.

4.2 Flat top Similarity Kernels

To further define the flat top similarity kernels, the notion of similarity has to be detailed first. As highlighted in the previous chapter, hill-climbing a kernel density estimator to find modes uses kernels placed on high-dimensional patches of the image.



(a) Plain line: $f(u,N,D)$. Dotted line: Gaussian similarity kernels with different widths. The Gaussian kernel is too smooth to be able to drop quickly enough between the main mode and the right tail, whatever the bandwidth is.



(b) Plain line: $f(u,N,D)$. Dotted line: Flat top similarity kernels with different widths, taking $f(u,N,D)$ into account. The slope of the flat top kernels follows much more closely the distance distribution.

Figure 4.2: $f(u,N,D)$ and various similarity kernels.

The parametric interpretation of the procedure, through the log-likelihood, shows the importance of using a well-chosen similarity measure. Intuitively, choosing a similarity kernel amounts to choose a function which associates a value between 0 and 1 to the distance between two points or vectors. In this section, we will further investigate the choice of efficient kernels as similarity measures in high-dimensional spaces.

4.2.1 Similarity definition

Let us assume that the data consist of a sample drawn from a mixture of M modes located at μ_j, with $1 \leq j \leq M$. Each mode is associated to a radial probability density function $q(\mathbf{x};\mu_j,\sigma) = k(\|\mathbf{x}-\mu_j\|_2/\sigma)$ centered on μ_j.

The similarity between any point $\mathbf{x}$ and μ_j should at the same time be a decreasing function of the distance $\|\mathbf{x}-\mu_j\|_2$ and take into account the information conveyed by the probability density function $q(\mathbf{x};\mu_j,\sigma)$.

In many situations, $q(\mathbf{x};\mu_j,\sigma)$ decreases with respect to $\|\mathbf{x}-\mu_j\|$ and the similarity is usually defined as a function of the radial probability density function:

$$s_\sigma(\|\mathbf{x}-\mu_j\|_2) = g(q(\mathbf{x};\mu_j,\sigma)) \ , \tag{4.3}$$

where g is monotonically decreasing on $\mathbb{R}^+$.

In the case of a Gaussian similarity kernel, for instance, one can simply take $g(u) = \tilde{\sigma}u$, which leads to

$$s_{\tilde{\sigma}}(\|\mathbf{x}-\mu_j\|_2) = \tilde{\sigma}q(\mathbf{x};\mu_j,\tilde{\sigma}) = \exp\left(-\frac{\|\mathbf{x}-\mu_j\|_2^2}{2\tilde{\sigma}^2}\right) \ , \tag{4.4}$$

where the true standard deviation is replaced with its oracle $\tilde{\sigma}$.

However, building a similarity that is directly proportional to the mode probability density function does not work if the latter is not a decreasing function of $\|\mathbf{x}-\mu_j\|_2$. To overcome this shortcoming, we suggest to define the similarity as follows:

$$s_{\tilde{\sigma},P}(\|\mathbf{x}-\mu_j\|_2) = \frac{\int_{\|\mathbf{x}-\mu_j\|_2/\tilde{\sigma}}^{\infty} k(u)u^P du}{\int_0^\infty k(u)u^P du} \ , \tag{4.5}$$

where $P \in \mathbb{N}^+$ is the order of the similarity. With this definition, the similarity has the following intuitive properties; for any k: $s_{\tilde{\sigma},P}(0) = 1$, and $u \leq v \Rightarrow s(u) \geq s(v)$. Furthermore, with the Gaussian case, that is, $k_{\tilde{\sigma}}(u) = \exp(-u^2/2\tilde{\sigma}^2)$, one can observe that $s_{\tilde{\sigma}}(\|\mathbf{x}-\mu_j\|) = s_{\tilde{\sigma},1}(\|\mathbf{x}-\mu_j\|)$. In other words, in the Gaussian case, the order $P = 1$ is then equivalent to the classical definition of the similarity.

4.2.2 χ_D kernel

Let us then study the use of other functions k in the similarity definition (4.5). Instead of choosing k according to the probability density function of $(\mathbf{x}_i - \mu^{(t)})/\sigma$, one can alternatively use the probability density function of $\|\mathbf{x}_i - \hat{\mu}^{(t)}\|_2/\sigma$ given by (4.2) , which follows closely the χ_D distribution. For this reason this similarity measure will be called the χ_D similarity. It is written

$$s_{\tilde{\sigma},P}(\|\mathbf{x}_i - \hat{\mu}^{(t)}\|_2) = \frac{\int_{\|\mathbf{x}_i-\hat{\mu}^{(t)}\|_2/\tilde{\sigma}}^{\infty} f(u;N,D)u^P du}{\int_0^{\infty} f(u;N,D)u^P du} \tag{4.6}$$

$$= Q\left(\frac{\|\mathbf{x}_i - \hat{\mu}^{(t)}\|_2^2}{2\tilde{\sigma}^2}, \frac{D+P}{2}\right) \ , \tag{4.7}$$

where Q is the regularized upper incomplete Gamma function.

In the $P = 1$ case, the similarity is the complementary cumulative distribution function of $f(u;N,D)$, that is, the probability to observe a larger distance than the observed one $\|\mathbf{x}_i - \hat{\mu}^{(t)}\|_2$. As this kernel is built from the distribution of the distances between two samples from the same distribution, this similarity measure follows the complementary cumulative distribution function of this kernel, and has the 'flat top' shape shown in Figure 4.2b.

4.2.3 Burr kernel

To avoid computational burden, the χ_D similarity can be approximated by simpler and 'all purpose' distributions defined on $\mathbb{R}^+$, such as the Burr type XII distribution [14]. The complementary cumulative distribution function of this distribution is much easier to compute than the regularized uppper incomplete Gamma function involved in the χ_D similarity. Furthermore, the Burr distribution has additional parameters that allow in particular the right tail thickness to be adjusted; this can be useful for non-Gaussian noise. The scaled complementary cumulative distribution function of the Burr type XII distribution is given by

$$F(b,\lambda,\tau,\theta) = (1-(b/\lambda)^{\tau})^{-\theta}\,. \tag{4.8}$$

A good approximation of the complementary cumulative distribution function of the χ_D distribution can be found with $\tau = D$, $\theta = -1$, and $\lambda = \sqrt{2}\sigma$. With these values, the Burr complementary cumulative distribution function reproduces the shape of the χ_D complementary cumulative distribution function, with first a flat top, a steep descent, and a thin tail.

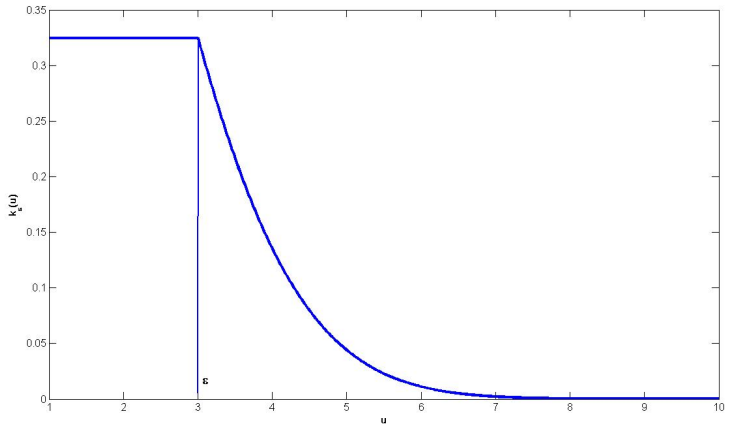


Figure 4.3: Topped Gaussian kernel for $\sigma = 2$, $\varepsilon = 3$, $0 \leq u \leq 10$.

4.2.4 Topped Gaussian kernel

The χ_D similarity stems from the combination of a multivariate Gaussian noise model with the Euclidean norm. For another noise or norm, the similarity function is expected to have a different formulation while keeping the same visual shape, that is, a flat top, followed by a steep descent towards zero. The phenomenon of norm concentration remains indeed valid for a wide range of distributions and norms.

Another flat top function can be defined as follows: a function constant between 0 and ε that becomes a negative exponential after ε. This can be written

$$k_{\varepsilon,\sigma}(u) = \exp\left(\frac{-\max(u,\varepsilon)^2}{2\sigma^2}\right) \quad , \tag{4.9}$$

where parameter ε is larger than or equal to zero and defines the width of the plateau in a topped Gaussian function, and σ controls the slope of the kernel. These two parameters ensure that this kernel behaves as a χ_D kernel. Figure 4.3 illustrates how the topped Gaussian kernel behaves for $\sigma = 2$, $\varepsilon = 3$, $0 \leq u \leq 10$.

4.2.5 Non-reflexive kernels

From Figure 4.1, another intuitive way of getting rid of the distance concentration comes to mind: it seems that a simple simple Gaussian kernel $\Psi_\sigma(u) = \exp(-\frac{u^2}{2\sigma^2})$

would a be good enough similarity measure if the minimum mode value was subtracted from the distance in order to recenter it around zero.

First, let us consider the radiometric kernel in update rule (3.25): $\Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2)$. The argument in Ψ'_σ is the distance between two patches $\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2$. Let us choose a distance Z to subtract to $\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2$.

The radiometric kernel is now written $\Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2 - Z)$. The argument of Ψ'_σ is a distance. To have a physical meaning, it should be positive for all possible values of Z. This brings up two conditions: $Z \leq \min(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2)$ and $\mathbf{i} \neq \mathbf{j}$. We can now rewrite the update rule (3.25) with the subtraction of the Z distance:

$$\hat{y}^{t+1}_\mathbf{i} = \frac{\sum^{(\mathbf{i}\neq\mathbf{j})}_{\mathbf{j}\in N_\mathbf{i}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2 - Z)\hat{y}^t_\mathbf{j}}{\sum^{(\mathbf{i}\neq\mathbf{j})}_{\mathbf{j}\in N_\mathbf{i}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2) - Z} \tag{4.10}$$

$$= \frac{\sum^{(\mathbf{i}\neq\mathbf{j})}_{\mathbf{j}\in N_\mathbf{i}} \exp\left(-\frac{Z}{2\sigma^2}\right) \Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2)\hat{y}^t_\mathbf{j}}{\sum^{(\mathbf{i}\neq\mathbf{j})}_{\mathbf{j}\in N_\mathbf{i}} \exp\left(-\frac{Z}{2\sigma^2}\right) \Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2)} \tag{4.11}$$

$$= \frac{\sum^{(\mathbf{i}\neq\mathbf{j})}_{\mathbf{j}\in N_\mathbf{i}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2)\hat{y}^t_\mathbf{j}}{\sum^{(\mathbf{i}\neq\mathbf{j})}_{\mathbf{j}\in N_\mathbf{i}} \Psi'_\sigma(\|\hat{\mathbf{y}}^t_\mathbf{i} - \hat{\mathbf{y}}^t_\mathbf{j}\|_2^2/2)} \tag{4.12}$$

$$\tag{4.13}$$

With these conditions, distance Z disappears naturally as the exponential is developed.

In other words, if $\mathbf{i} \neq \mathbf{j}$, the Gaussian kernel behaves as if a distance Z had been subtracted in order to bring the distances as close as possible to zero, and thus reduces the effects of the curse of dimensionality. The burden of using a similarity based on a topped Gaussian function can be avoided just by excluding the 'reflexive' ($\mathbf{i} = \mathbf{j}$) term in the corresponding gradient with usual Gaussian similarities. Conversely, the naive operation of dropping the reflexive term with usual Gaussian similarities amounts to implicitly using an topped Gaussian kernel, as the plateau width is then automatically adjust

4.3 Experiments

The experiments feature eight images with 512×512 pixels and 256 grey levels (the benchmark image, Lena, Barbara, Couple, Hill, House, Fingerprint and Boat).

These images are polluted by additive Gaussian white noise with standard deviation $\sigma_{\text{noise}} = 20$, $\sigma_{\text{noise}} = 30$ and $\sigma_{\text{noise}} = 40$. The performances of the patchwise non-local means with different kernels are compared. The tested kernels are

- Gaussian kernel ,
- Gaussian kernel without the reflexive term,
- χ_D kernel,
- Burr kernel,
- Topped Gaussian.

For each image, several parameters of the filter are to be optimized : k, the patch size varies from 3×3, to 11×11 by steps of 2 pixels. ρ, the width of the spatial kernel (Gaussian), is fixed as $\rho = 10$ pixels. The number of iteration is fixed at 1 as Chapter 3 showed us that it is often the best possible choice when the parameters are optimised. The width σ of kernel Ψ is optimized in order to minimize the mean RMSE on 100 independent repetitions of the polluted images, for each patch size (the topped Gaussian kernel also needs the additional parameter ε to be optimized). The optimized parameter are then used to evaluate the mean root mean square error (MRMSE), its standard deviation (stdRMSE) and the mean peak signal to noise ratio (MPSNR) on a new set of 100 images polluted with the same noise model. Results are reported in Tables 4.1 to 4.8: for each image it presents the MRMSE, stdRMSE, MPSNR, and the optimal σ, ε (if needed), and patch radius r used to obtain these results.

A statistical test has been performed to evaluate the significance of these results: for all images and all noise levels, the denoising effects resulting from the use of all the flat top kernels are significantly better than in the simple Gaussian kernel case ($p < 0.01$ on all cases). The optimal value of σ is larger in the case of the flat top kernels, because of the steeper slope of these kernels. The difference between the results increases as the noise increases, suggesting that the gain of performances is more important if the noise is more intense. The optimal patch size is also often smaller for the Gaussian kernel than for the other kernels. This behaviour is explained by the smoother slope of this kernel.

Some example results for the Lena image are shown in Figures 4.4, 4.5, 4.6, 4.7 ($\sigma_{noise} = 20$), and 4.8, 4.9, 4.10, 4.11 ($\sigma_{noise} = 40$). The images were cropped in order to be able to better see the fine details of the images. However, despite this manipulation, it is still very hard to see a visual difference between the different kernels used

Lena: $\sigma_{noise} = 20$						
kernel	MRMSE	MPSNR	stdRMSE	σ	ε	r
Gaussian	6.73	31.57	0.0245	0.80		2
Gaussian (NR)	6.34	32.08	0.0223	0.42		4
Topped Gaussian	6.34	32.08	0.0245	0.42	1.08	4
Chi	6.22	32.26	0.0264	1.59		4
Burr	6.23	32.23	0.0244	1.64		4
$\sigma_{noise} = 30$						
Gaussian	8.54	29.50	0.0282	0.75		3
Gaussian (NR)	7.77	30.32	0.0282	0.41		4
Topped Gaussian	7.77	30.32	0.0282	0.41	1.07	4
Chi	7.64	30.47	0.0331	1.52		4
Burr	7.60	30.51	0.036	1.57		4
$\sigma_{noise} = 40$						
Gaussian	10.11	28.03	0.0424	0.77		3
Gaussian (NR)	9.14	28.91	0.0469	0.42		4
Topped Gaussian	9.14	28.91	0.0469	0.42	1.05	4
Chi	9.00	29.04	0.0489	1.50		4
Burr	8.93	29.12	0.048	1.56		4

Table 4.1: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the Lena image.

for the denoising process. It is however possible to see that the images denoised with the classical Gaussian kernel seem less smooth than the other images using kernels specifically made to cope with the high dimensionality of the patches. The fine details and textures are lost in the noise in the $\sigma_{noise} = 40$ case.

4.4 Conclusions

Denoising through mode estimation achieves better denoising result when used with high-dimensional data. However, [6, 38] showed that in high-dimensional spaces, the distances between data points concentrate; the Euclidean distance is not an efficient

Barbara: $\sigma_{noise} = 20$						
kernel	MRMSE	PSNR	stdRMSE	σ	ε	r
Gaussian	8.46	29.58	0.0223	0.80		2
Gaussian (NR)	7.81	30.28	0.0345	0.47	2.43	2
Topped Gaussian	7.80	30.29	0.0245	0.46		3
Chi	7.75	30.34	0.0264	1.61		3
Burr	7.73	30.36	0.0282	1.67		3
$\sigma_{noise} = 30$						
Gaussian	10.96	27.34	0.0346	0.73		3
Gaussian (NR)	9.59	28.50	0.0374	0.39	1.10	3
Topped Gaussian	9.59	28.50	0.0374	0.39		4
Chi	9.63	28.45	0.0.04	1.54		4
Burr	9.60	28.49	0.0435	1.59		4
$\sigma_{noise} = 40$						
Gaussian	13.19	25.73	0.0479	0.72		3
Gaussian (NR)	11.30	27.07	0.05	0.39	1.10	4
Topped Gaussian	11.30	27.07	0.05	0.39		4
Chi	11.39	27.00	0.0547	1.50		4
Burr	11.30	27.07	0.0583	1.56		4

Table 4.2: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the Barbara image.

similarity measure anymore. This chapter presents new similarity measures that can be used in order to provide a better distance estimation in high-dimensional spaces, and thus a more efficient mode discrimination, and better denoising results.

Four new kernels are proposed : the Gaussian with non-reflexive term, the topped Gaussian kernel, the χ_D kernel, and the Burr kernel. To compare these filters, 8 images have been filtered using the same filter, and these different kernels, in addition to the classical Gaussian kernel. Results show that the best results are obtained using the χ_D kernel. A statistical test has been performed between the RMSE results obtained by using each one of the new kernels, and those obtained using the classical Gaussian kernel; each one of the new kernels performs better that the Gaussian one, for all images and all noise levels.

House: $\sigma_{noise} = 20$						
kernel	MRMSE	PSNR	stdRMSE	σ	ε	r
Gaussian	8.54	29.49	0.0213	0.73		2
Gaussian (NR)	8.26	29.78	0.0231	0.48	1.13	2
Topped Gaussian	8.26	29.78	0.0233	0.48		2
Chi	8.22	29.82	0.0245	1.5		3
Burr	8.20	29.85	0.0223	1.55		2
$\sigma_{noise} = 30$						
Gaussian	10.47	27.72	0.0264	0.75		2
Gaussian (NR)	9.86	28.24	0.0282	0.37	1.1	4
Topped Gaussian	9.86	28.24	0.0281	0.37		4
Chi	9.86	28.25	0.0316	1.48		4
Burr	9.86	28.24	0.0331	1.53		4
$\sigma_{noise} = 40$						
Gaussian	11.93	26.59	0.0346	0.80		2
Gaussian (NR)	11.23	27.11	0.0374	0.38	1.09	4
Topped Gaussian	11.23	27.11	0.0370	0.38		4
Chi	11.21	27.13	0.0374	1.46		4
Burr	11.20	27.14	0.0461	1.52		4

Table 4.3: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the House image.

Benchmark: $\sigma_{noise} = 20$						
kernel	MRMSE	PSNR	stdRMSE	σ	ε	r
Gaussian	9.33	28.73	0.2293	0.95		1
Gaussian (NR)	8.87	29.17	0.2271	0.68	1.32	1
Topped Gaussian	8.92	29.12	0.2295	0.73		1
Chi	8.60	29.44	0.2193	1.73		2
Burr	8.51	29.53	0.2167	1.78		2
$\sigma_{noise} = 30$						
Gaussian	13.68	25.41	0.3173	0.88		1
Gaussian (NR)	12.46	26.21	0.3686	0.51	2.01	2
Topped Gaussian	12.56	26.15	0.3701	0.54		2
Chi	12.65	26.09	0.3624	1.56		2
Burr	12.44	26.23	0.3728	1.62		2
$\sigma_{noise} = 40$						
Gaussian	18.29	22.88	0.4031	0.84		1
Gaussian (NR)	16.06	24.01	0.4572	0.49	1.86	2
Topped Gaussian	16.13	23.97	0.4577	0.50		2
Chi	16.53	23.76	0.4182	1.55		2
Burr	16.24	23.92	0.4222	1.62		2

Table 4.4: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the Benchmark image.

Boat: $\sigma_{noise} = 20$						
kernel	MRMSE	PSNR	stdRMSE	σ	ε	r
Gaussian	8.63	29.41	0.0218	0.76		2
Gaussian (NR)	8.39	29.66	0.0235	0.49	1.10	2
Topped Gaussian	8.39	29.66	0.0237	0.49		2
Chi	8.27	29.78	0.0231	1.51		2
Burr	8.24	29.81	0.0239	1.58		2
$\sigma_{noise} = 30$						
Gaussian	10.82	27.44	0.0281	0.75		2
Gaussian (NR)	10.19	27.97	0.0317	0.38	1.12	4
Topped Gaussian	10.19	27.97	0.0316	0.38		4
Chi	10.11	28.03	0.0365	1.51		4
Burr	10.10	28.04	0.0411	1.57		4
$\sigma_{noise} = 40$						
Gaussian	12.73	26.03	0.0435	0.76		2
Gaussian (NR)	11.76	26.73	0.051	0.38	1.07	4
Topped Gaussian	11.76	26.73	0.0511	0.38		4
Chi	11.68	26.78	0.0517	1.49		4
Burr	11.64	26.81	0.0537	1.54		4

Table 4.5: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the Boat image.

Couple: $\sigma_{noise} = 20$						
kernel	MRMSE	PSNR	stdRMSE	σ	ε	r
Gaussian	8.95	29.10	0.0221	0.75		2
Gaussian (NR)	8.57	29.47	0.0214	0.48	1.06	2
Topped Gaussian	8.57	29.47	0.0214	0.48		2
Chi	8.44	29.60	0.0241	1.54		3
Burr	8.45	29.60	0.0248	1.60		3
$\sigma_{noise} = 30$						
Gaussian	11.43	26.97	0.0316	0.74		2
Gaussian (NR)	10.52	27.69	0.0331	0.37	1.13	4
Topped Gaussian	10.52	27.69	0.0332	0.37		4
Chi	10.53	27.68	0.0441	1.51		4
Burr	10.56	27.66	0.0397	1.56		3
$\sigma_{noise} = 40$						
Gaussian	13.43	25.57	0.0313	0.75		2
Gaussian (NR)	12.24	26.37	0.0378	0.37	1.10	4
Topped Gaussian	12.24	26.37	0.0375	0.37		4
Chi	12.30	26.33	0.0474	1.47		4
Burr	12.27	26.36	0.0497	1.53		4

Table 4.6: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the Couple image.

Fingerprint: $\sigma_{noise} = 20$						
kernel	MRMSE	PSNR	stdRMSE	σ	ε	r
Gaussian	8.95	29.10	0.223	0.75		2
Gaussian (NR)	8.57	29.47	0.0245	0.48	1.06	2
Topped Gaussian	8.57	29.47	0.0244	0.48		2
Chi	8.44	29.60	0.0245	1.54		3
Burr	8.45	29.60	0.0243	1.60		3
$\sigma_{noise} = 30$						
Gaussian	11.43	26.97	0.0312	0.74		2
Gaussian (NR)	10.52	27.69	0.0334	0.37	1.13	4
Topped Gaussian	10.52	27.69	0.0335	0.37		4
Chi	10.53	27.68	0.0346	1.51		4
Burr	10.56	27.66	0.0360	1.56		3
$\sigma_{noise} = 40$						
Gaussian	13.43	25.57	0.0331	0.75		2
Gaussian (NR)	12.24	26.37	0.0387	0.37	1.10	4
Topped Gaussian	12.24	26.37	0.0386	0.37		4
Chi	12.30	26.33	0.0471	1.47		4
Burr	12.27	26.36	0.0478	1.53		4

Table 4.7: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the Fingerprint image.

Hill: $\sigma_{noise} = 20$						
kernel	MRMSE	PSNR	stdRMSE	σ	ε	r
Gaussian	8.54	29.50	0.0211	0.74		2
Gaussian (NR)	8.26	29.79	0.0213	0.49	1.14	2
Topped Gaussian	8.26	29.79	0.0211	0.49		2
Chi	8.23	29.83	0.0251	1.50		3
Burr	8.20	29.85	0.0221	1.55		2
$\sigma_{noise} = 30$						
Gaussian	10.48	27.73	0.0265	0.75		2
Gaussian (NR)	9.87	28.25	0.0281	0.37	1.11	4
Topped Gaussian	9.87	28.25	0.0278	0.37		4
Chi	9.86	28.25	0.0315	1.48		4
Burr	9.87	28.25	0.0283	1.54		4
$\sigma_{noise} = 40$						
Gaussian	11.94	26.59	0.0341	0.80		2
Gaussian (NR)	11.24	27.12	0.0375	0.39	1.09	4
Topped Gaussian	11.24	27.12	0.0377	0.39		4
Chi	11.22	27.13	0.0386	1.47		4
Burr	11.20	27.14	0.0385	1.53		4

Table 4.8: For all listed kernels: Mean Root Mean Square Error (MRMSE), Mean Peak Signal to Noise Ratio (MPSNR), standard deviation of Root Mean Square Error (stdRMSE), optimal σ, optimal ε, and optimal patch radius (r) for 100 repetitions of the Hill image.

(a) Original Lena image without noise

(b) Lena image with additive gaussian noise $\sigma_{noise} = 20$

Figure 4.4: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 20$.

(a) Denoised Lena image with Gaussian kernel

(b) Denoised Lena image with non-reflexive Gaussian kernel

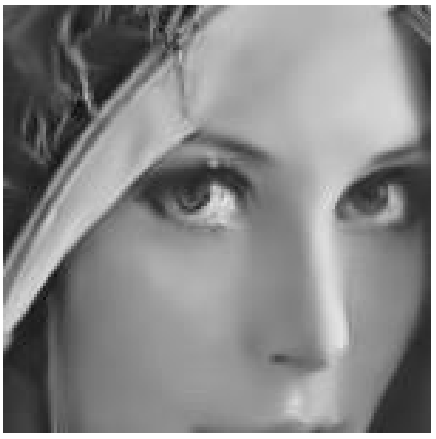
Figure 4.5: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 20$.

(a) Denoised Lena image with topped Gaussian kernel

(b) Denoised Lena image with χ_D kernel

Figure 4.6: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 20$.

(a) Denoised Lena image with non-reflexive χ_D kernel



(b) Denoised Lena image with Burr kernel

(c) Denoised Lena image with non-reflexive Burr kernel

Figure 4.7: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 20$.

(a) Original Lena image without noise

(b) Lena image with additive gaussian noise $\sigma_{noise} = 40$

Figure 4.8: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 40$.

(a) Denoised Lena image with Gaussian kernel

(b) Denoised Lena image with non-reflexive Gaussian kernel

Figure 4.9: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 40$.

(a) Denoised Lena image with topped Gaussian kernel

(b) Denoised Lena image with χ_D kernel

Figure 4.10: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 40$.

(a) Denoised Lena image with non-reflexive χ_D kernel

(b) Denoised Lena image with Burr kernel

(c) Denoised Lena image with non-reflexive Burr kernel

Figure 4.11: Examples of results obtained for the Lena image with $\sigma_{\text{noise}} = 40$.

Chapter 5

Estimation of the local noise for local parameter adjustment in the patch-based filters

The previous chapters detailed the use and performances of the statistical filters in low- and high-dimensional spaces. Even if these methods prove to be efficient denoising algorithms, the choice of parameters strongly influences the results of the denoising process. For instance, one should be particularly careful when choosing the width σ of the radiometric kernel: with too small of a width, most of the noise remains on the filtered image, with too large a width, the features of the images are blurred and details are lost.

In the previous chapters, an optimisation procedure ensured that the best possible choice of parameters was used in all situations. However, in real life situations and in particular in the case of medical images, it is often impossible to optimally tune these parameters. Since the noise arises during the acquisition process, the optimisation method used in the previous chapters cannot be used here as the true image is unknown, making it impossible to optimize the parameters and to calculate the RMSE or PSNR of the denoised images.

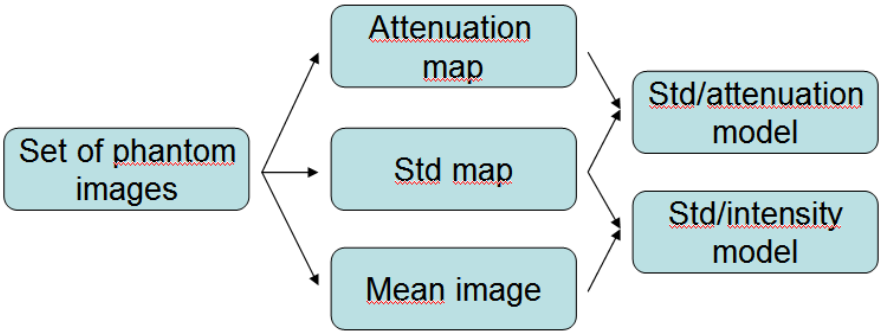
The noise in each image depends of the scanner and of the parameters used for acquisition. In practice, when a physician needs to filter an image, he uses a set of default parameters that is proposed for each scanner. This is based on the rather strong assumption that the noise observed on the image uniquely depends of the scanner, and

not, for example of the object to be scanned, the dose of radiation used, and other hidden factors. For this reason, a set of default parameters is certainly not optimal for a particular image.

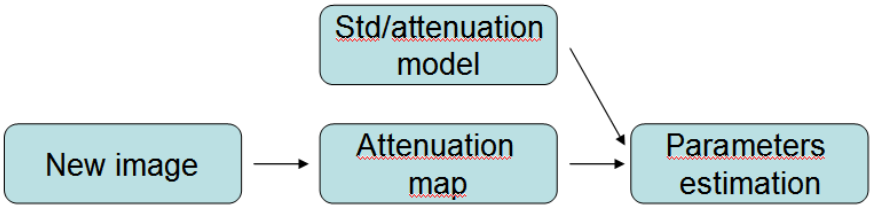
Furthermore, the noise found on a CT scan image is more complex than a simple Gaussian noise. Indeed the noise is composed of noise caused by the stochastic fluctuations of the X-rays absorption and scattering, and noise caused by the detectors. In most cases these two noise sources are combined and the noise is modeled as additive noise. However, the statistical properties of this noise are not constant over the whole image. In this case, a constant standard deviation may be unable to provide the best results on all parts of the image. A trade-off has to be found between smoothing the low intensity parts of the images and removing the noise on high intensity areas.

The literature already proposes some ways of dealing with noise on CT scan images: [55] proposes to use a partial differential equation based method using correlations analysis to reduce the noise on a CT scan image. [75] uses partial differential equations on projection images rather than on the reconstructed images. In [62], the curvelet transform is adapted to denoise CT images and the results are compared to wavelet transform based denoising. Wavelets tresholding in windows of the image is also used in [10, 1]. All these methods need a set of parameters to work properly. One of the main challenges today is to choose the set of parameters giving the best denoising results with the best preservation of the image quality. Nowadays, the medical practitioners usually use a fixed set of parameter depending of the scanner that took the image to analyze. However, this set of parameters is usually far from perfect, and if the noise depends on the intensity of the pixels or some more complex parameter, these parameters might need to be changed for each images.

In this chapter, we propose to remove the noise on CT scan images using the statistical filters presented above. We also propose to estimate the parameters of the filter by a statistical model of the noise built from other images acquired with the same scanner. With this model the best choice of parameters is automatically computed for each image, providing the best possible denoising results. The statistical filters are a good choice for estimating the parameters of the filters, since these parameters have a direct link with the standard deviation of the noise present on the image. It is not the case for the methods presented in the literature. For this specific scanner, this model gives an estimate of the standard deviation of each pixel; the standard deviations are gathered in a so-called *standard deviation map*. The standard deviation map is used to estimate the value that σ should have to filter this pixel efficiently, or to provide an estimate of a global σ for the whole image. The model design and parameter



(a) Summary of the model design steps



(b) Summary of the paramter estimation steps

Figure 5.1: Summary of the steps in this chapter.

estimation steps proposed in this chapter are summarized in Figures 5.1a and 5.1b.

Section 5.1 contains a brief presentation of the CT scan images and their characteristics. Section 5.2 presents how statistics can be extracted from a set of phantom images. These statistics provide a surrogate of the true image, that can be used as ground truth to estimate the filtering results, and the standard deviation of each pixel over the whole set of images. Section 5.3 presents various standard deviation models and how they fit the data. Section 5.4 shows how the statistical filters presented in the previous chapters can be modified in order to use the standard deviation map. Section 5.5 proposes an experiment designed in order to estimate the filtering results with the standard deviation models on real CT scan images and presents the results. Finally, Section 5.6 draws conclusions about this chapter.

5.1 Simplified model of the CT scan

The CT scan is a structural imaging method using X-rays to construct an accurate 3-dimensional image of the anatomy of the patient. Images are taken from a lot of different angles around the patient. These images are then combined to provide a precise 2-dimensional image of a very fine layer of the patient's anatomy. A set of 2-dimensional layers, combined along the third axis, gives a 3-dimensional image on the organs, which can be used to detect the tumor positions and volumes. The scanners can be used for diagnostic (low-energy scanners, called KVCT, for kilovoltage CT), or for careful positioning of the patient before treatment, for example in radiotherapy (high-energy scanners, called MVCT megavoltage CT). The MVCT images usually have a higher noise level than the KVCT images. This is because the scanner used for the positioning of the patients uses the accelerator built in the treatment device. This accelerator generates rays of higher energy for treatment purpose.

5.1.1 Presentation of the scanner

The X-rays used by the CT scan are generated by a X-rays tube. A detector is placed on the opposite side of the tube to detect the rays after they went through the patient (see Figure 5.2). An anti-deviation grid is placed in front of the detector to ensure that the deflected rays are not detected.

The detector measures a value called the *coefficient of attenuation*, which is used to calculate the *CT number*, measure in Hounsfield unity (H). The CT number is defined as

$$CT(\mathbf{i}) = 1000\left(\frac{\mu^t_{\mathbf{i}} - \mu^w}{\mu^w}\right) \quad , \tag{5.1}$$

where $\mu^t_{\mathbf{i}}$ is the attenuation coefficient of the pixel at coordinates $\mathbf{i}$, while μ^w is the attenuation coefficient of water (typically equals to 0.195). The observed values span between 0H for air to 4000H for dense bones. The actual values observed from a scanner can be different depending of its calibration. For instance, the CT images used in this thesis span between -1000H and 3000.

While the CT scan is an extremely useful tool for physicians, it still presents some intrinsic limitations. In particular, the contrast of the images, and the noise are three major issues with this imaging modality.

The spatial resolution determines the size of the small objects that will be visible in the image. A better spatial resolution is also useful in order to clearly see the

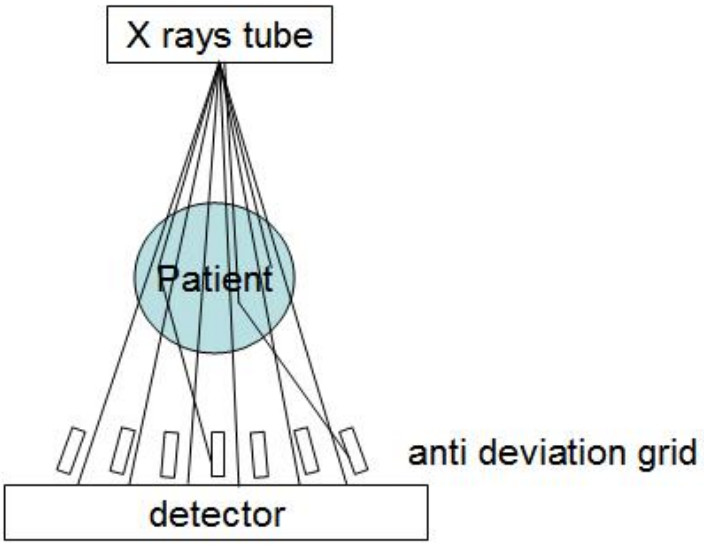


Figure 5.2: Simplified CT scan principle

border between two organs. The spatial resolution changes for lots of reasons: patient movements, sampling of the signal, size of the detector, reconstruction algorithms, etc.

The contrast measures how well the image is able to differentiate between tissues of almost similar density. It depends of the signal to noise ratio, which depends on the quantity of detected rays.

The noise found on CT scan images varies widely depending on the parameters of the acquisition: the energy of the rays, the time of exposition, and the slice thickness play a major role on the noise intensity.

In radiotherapy, the quality of the CT image is very important for the physicians in order to decide of the doses to use and of the patient positioning. The filtering algorithms provide useful tools to get rid of most of the noise, as long as the parameters are well chosen. Unfortunately, as there is no ground truth for the CT images, it is impossible in practice to estimate the quality of an image and thus to estimate the parameters needed to obtain a well denoised image.

The next section proposes to use a set of phantom images, taken with the same scanner, with different acquisition parameters. These images are used to construct a surrogate of the true phantom image and its standard deviation. These statistics are also used in order to build a model of the standard deviation for this scanner, and model the noise of any new image taken from this scanner.

5.2 Statistics from the phantom image

Firstly, a model of the standard deviation of all the pixels in an image taken with the chosen scanner has to be estimated. In order to do this, a set of images of a phantom (an artifical object with constant piecewise areas of different densities) has been taken. Each image of the phantom contains 40 2-dimensional layers of 512×512 pixels. Two parameters are changed between the images: the slice thickness can be $\{0.5, 2, 8\}$ mm, and the effective current is taken in $\{37, 75, 112, 150, 187, 225\}$ mA/s. The combinations of these parameters provide 18 images of the same phantom, with different noise characteristics for a total of $18 \times 40 = 720$ 2-dimensional layers (see Figure 5.3).

The layers of the same image are considered as repetitions of the same object acquired with the same parameters values. Depending of the parameter choice, the images can be polluted by heavy to light noise (see Fig 5.4). These 720 layers can be used to estimate the mean and the standard deviation generated by this scanner during the acquisition.

5.2.1 Mean phantom image

The mean of the 720 layers (see Fig. 5.5) is used as a surrogate of the ground truth image. The mean phantom image provides a useful surrogate of the true image and it will be used as a reference for assessing the filters quality in the rest of this chapter.

5.2.2 Standard deviation map

The standard deviation map for each pixel, displayed in Figure 5.6, is constructed from the standard deviation of each pixel over the 720 layers. In addition to the estimation of the standard deviation, this map provides useful information on the nature of the noise. The standard deviation is higher in the center of the phantom than near its borders. This suggests that it depends not only on the density of the object to be imaged, but

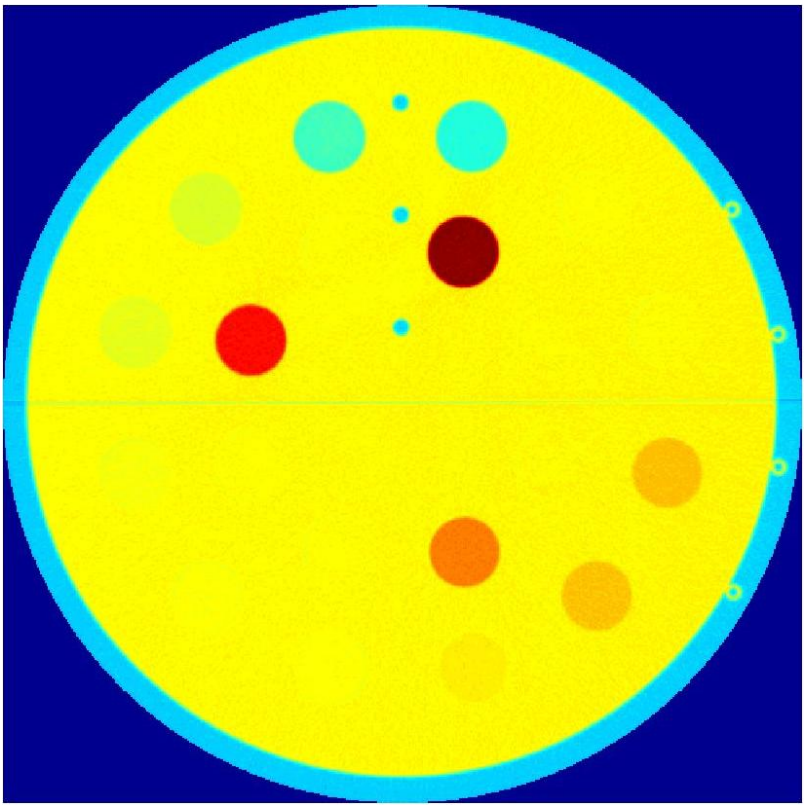
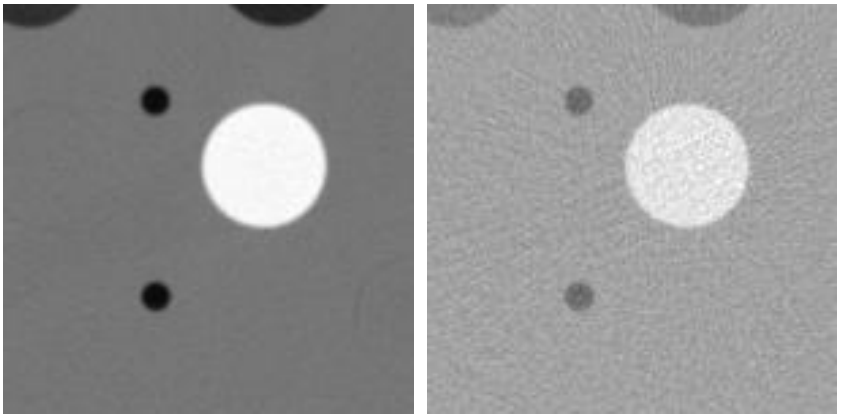


Figure 5.3: Example of phantom image



(a) Phantom image with low noise (8mm layer, 225 mA)

(b) Phantom image with high noise (0.5mm layer, 37 mA)

Figure 5.4: Examples of noise on the phantom image.

also on the distance travelled through it. This also shows that the hypothesis of a constant standard deviation over the whole image is not valid in the case of real CT images.

The observation made on Figure 5.6 strongly suggests that using only the pixels intensities to model the standard deviation might not be sufficient. In order to be able to model the noise correctly, the data should also depend of the distance travelled into the phantom. The attenuation of the rays is a quantity that depends on the path travelled through the phantom, and it determines the pixel intensities. The attenuation could then prove to be a useful data to model the standard deviation. The next section introduces the notion of attenuation and explains how it can be extracted from any CT image.

Sources of standard deviation

This section studies the different sources of standard deviation in the phantom images. We assume that the sources of variance consists in the variance between the layers of each one of the 18 images (intra-images), the variance between the means of the different images (inter-images), and the variance between the spatial locations of the pixels (in 3 dimensions). As there is only one image taken for each scanner setting, it is unfortunately not possible to explore the last source of variance. However, by

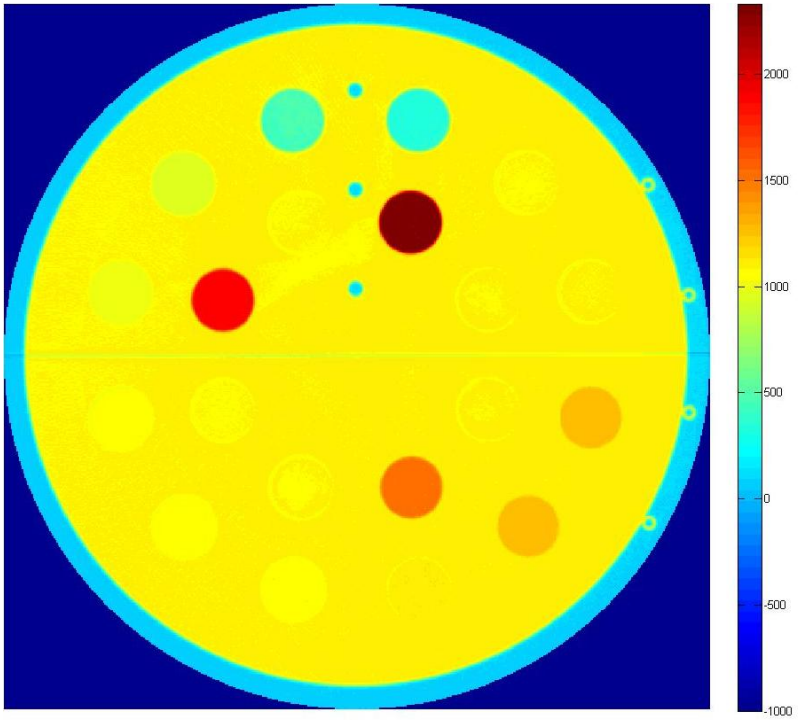


Figure 5.5: Mean phantom image (in Hounsfield unity)

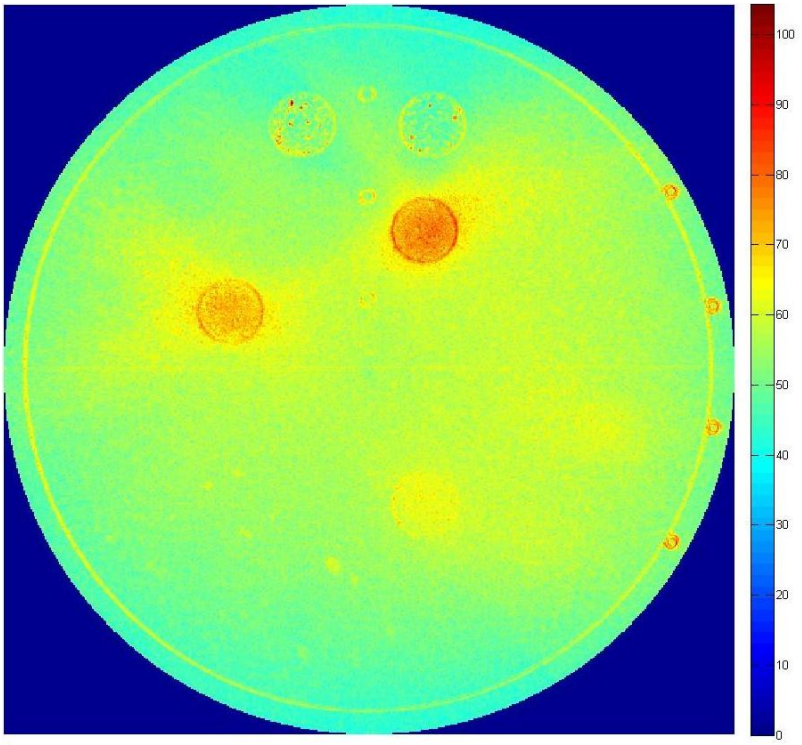


Figure 5.6: Standard deviation map.

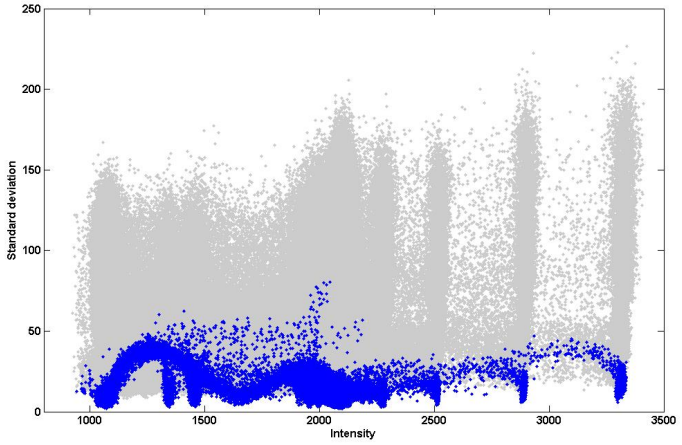


Figure 5.7: Contributions of standard deviation.

comparing the intra-images and inter-images variances, it is possible to evaluate the contribution of the first two sources of noise to the CT images.

Figure 5.7 shows the standard deviation of each pixel in function of its intensity. The pale gray dots displays the standard deviation of each pixel, between the layers of the same image, while the dark gray dots present the standard deviation between the mean images. This figure shows that the contribution of standard deviation coming from the different settings of the scanner is much smaller than the standard deviation between the layers of the same image. The noise between the layers of the same image is much more important than the noise coming from different settings on the scanner.

5.2.3 Attenuation map

The CT images are reconstructed from the information gathered in the multiple images taken around the patient. This information corresponds to the attenuation of the X-rays by the objects and organs through which the rays went. The attenuation measures are then used to calculate the intensities of the pixels of the image. From the image, it is thus possible to reverse this step and find the attenuation of the rays observed on each pixel. Once this is done, it is possible to try to find a model of the standard deviation of a pixel in function of this attenuation map.

Radon transform

The projection of a two-dimensional object with density $\lambda(x,y)$ is a set of line integrals. The Radon transform computes the line integrals from multiple parallel beams, in a certain direction. If $\lambda(x,y)$ represents the unknown density of the organs to be imaged (see Figure 5.8), then the Radon transform of this density represents the attenuation of the X-rays at the detector, in one particular direction L. The projection of $\lambda(x,y)$ along L gives the projection of the function, on the direction p, orthogonal to L. The Radon transform is written

$$RT\{\lambda(x,y)\} = \{RT\,\lambda\}(p,\phi) = \int_L \lambda(x,y)dl \ . \tag{5.2}$$

In this expression, L is defined as

$$x\cos\phi + y\sin\phi = p \ . \tag{5.3}$$

For a fixed ϕ, $\{RT\lambda\}(p,\phi)$ is a projection of the density $\lambda(x,y)$ along the path L and the angle of projection ϕ.

With (5.3), (5.4) becomes

$$\{RT\,\lambda\}(p,\phi) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \lambda(x,y)(x\cos\phi + y\sin\phi - p)dxdy \ . \tag{5.4}$$

To illustrate this, let us imagine an object to scan, which has the form of a disc. During the acquisition process, a set of projections of this image is taken at different angles ϕ (see Figure 5.9)[1]

These projections are then concatenated and the set of the projections at different angles ϕ is called a *sinogram* (see Figure 5.10)

Figure 5.11 illustrates how the resolution of the sinogram increases with the number of projections.

A single pixel in the spatial domain (x,y) corresponds to a sine wave in the Radon space (p,ϕ). To illustrate this, let us look at a 10×10 matrix with one single pixel with a non zero intensity (see Figure 5.12a). The sinogram of this matrix is displayed in Figure 5.12b, for $0 \leq \phi \leq 1000$ degrees.

The actual CT scan images have to be reconstructed from the sinograms obtained from the detectors. This operation is called the retroprojection and consists in projecting the value of the attenuation along the beam on the spatial space. The combination

[1]Source of images 5.9, 5.10, 5.11 and 5.13: `http://www.cnebmn.jussieu.fr/partagechups/aurengo/tomonum-p1/tomonum.pdf`, 01 February 2011

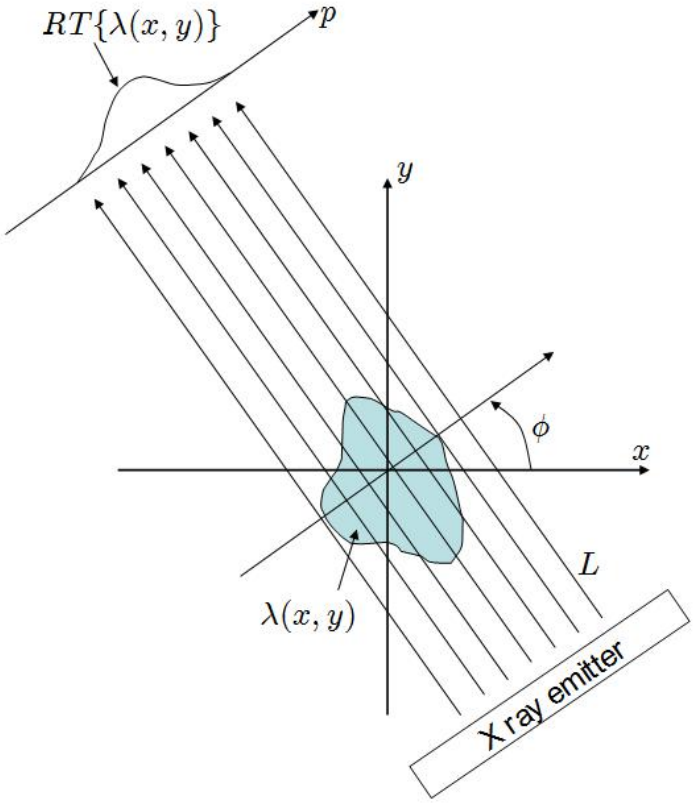


Figure 5.8: Radon transform. The attenuation $RT\{\lambda(x,y)\}$ is the integral of the density $\lambda(x,y)$ along the beams L.

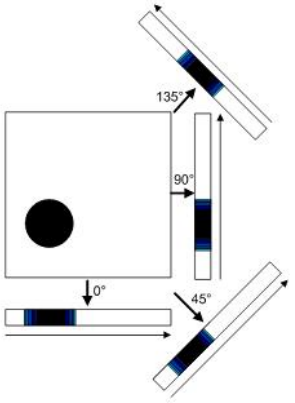


Figure 5.9: Projections of the object to be imaged at different angles.

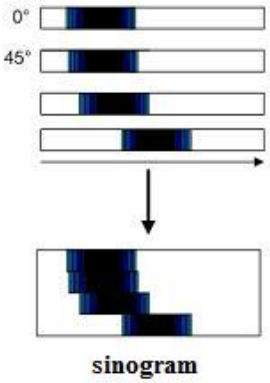


Figure 5.10: Construction of a sinogram.

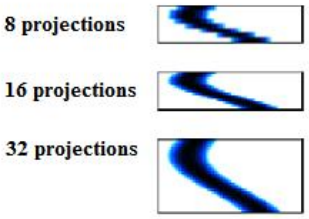
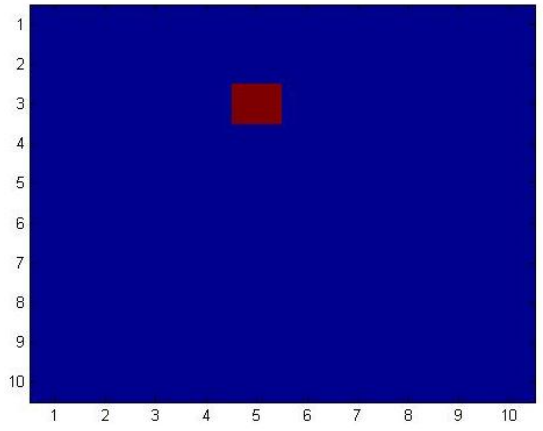
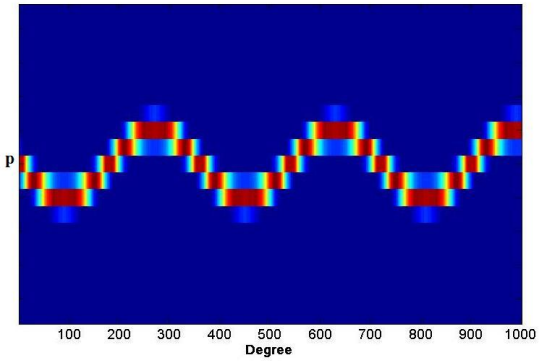


Figure 5.11: Resolution of a sinogram depending of the number of projections.



(a) Matrix with a simple pixel in the spatial domain



(b) Sinogram of that matrix

Figure 5.12: Example of sinogram from a single illuminated pixel matrix (rotation of 90 degrees in comparison with Figure 5.10).

of the inverse projections from different angles give the original object (see Figure 5.13). The larger the number of projections is, the finer the reconstructed image. Conversely, using the Radon transform on a reconstructed image gives back the sinogram from which the image was reconstructed. This inverse Radon transform is used to construct the attenuation map.

Fan-beam transformation

The Radon transform makes the hypothesis that parallel beams are used for the acquisition. In practice, the beams are emitted as a fan, from a single point on the emitter (see Figure 5.14). The *fan-beam transform* is able to take that geometry into account. The principle of the fan-beam transform is very similar to the Radon transform [45], adapted for the fan-beam geometry. However, it is out of the scope of this thesis and will not be detailed here.

The CT images that we have obtained are reconstructed using the inverse fan-beam transform of the sinogram.

Attenuation map

As the scanner image is reconstructed from the attenuation values with the inverse fan-beam transform, it is possible to find the attenuation values back from the scanner image using the fan-beam transform. The total attenuation can then be calculated for each pixel separately in order to create an *attenuation map*. The first step in order to compute the attenuation map is to calculate the sinogram of the whole image (see Figure 5.15)

To each pixel in the image corresponds a sinusoid on the sinogram. In order to evaluate the attenuation on a single pixel, it must be isolated from the rest of the image. The steps required to achieve this are illustrated on the cropped (50×50) mean phantom image.

1. A matrix of the same size as the original image is created, filled with zeros except for one single pixel and its direct neighbours (with smaller intensity) (see Figure 5.16)

2. The sinogram for the matrix in Figure 5.16 is estimated (see Figure 5.17).

3. The intensities of that sinogram are multiplied by the intensities of the sinogram of the image as illustrated on Figure 5.18. This step ensures that only the contributions of this particular pixel are found on the resulting matrix.

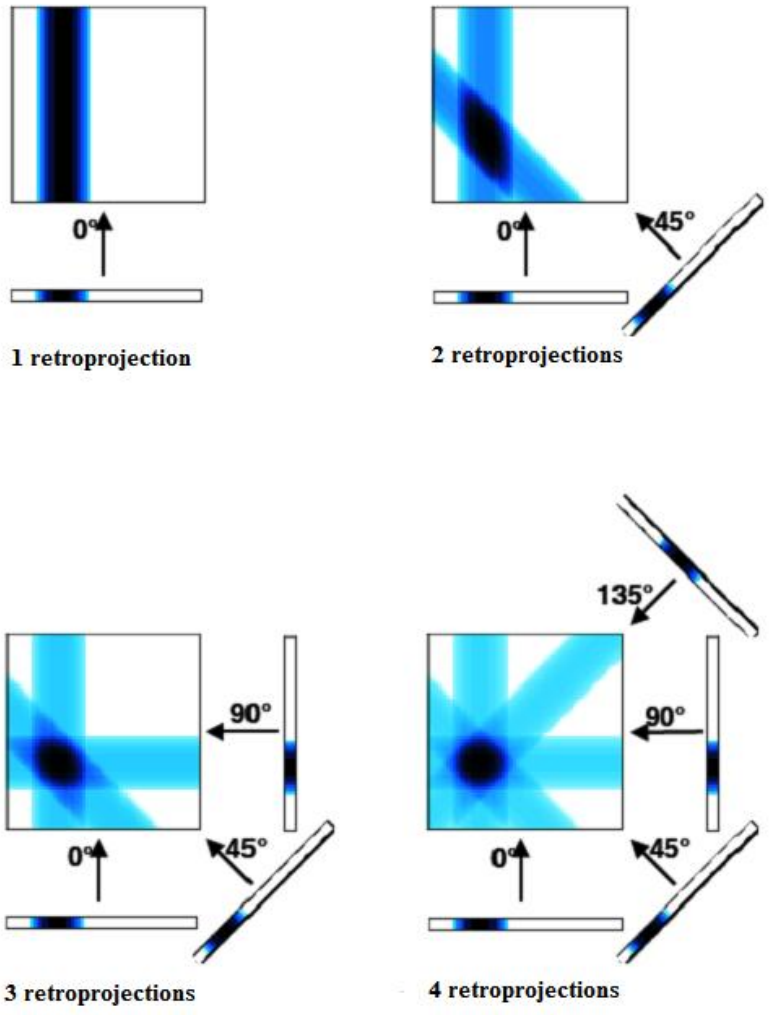


Figure 5.13: Reconstruction of the original object by retroprojection.

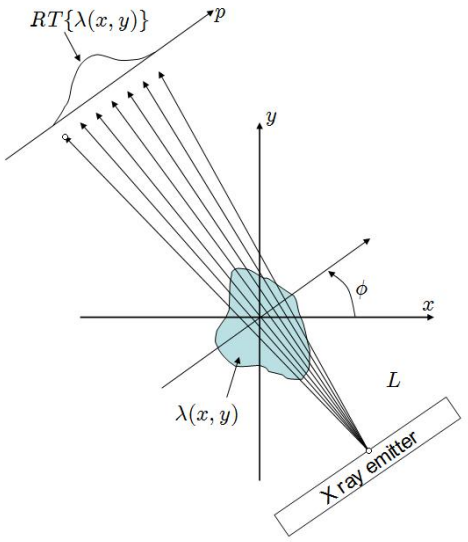


Figure 5.14: Fan beam transform. The attenuation $RT\{\lambda(x,y)\}$ is the integral of the density $\lambda(x,y)$ along the beams L.

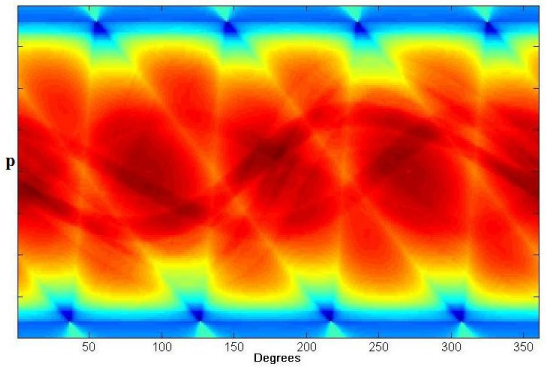


Figure 5.15: Sinogram of the mean phantom image.

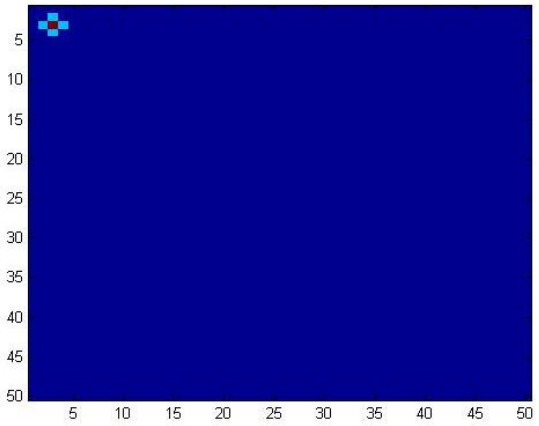


Figure 5.16: Selection of one single pixel

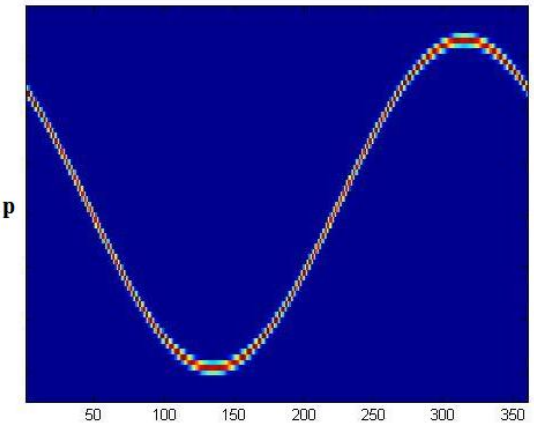


Figure 5.17: Sinogram of the single pixel matrix.

4. The attenuation map contains the sum of the attenuation observed in a particular pixel, at all angles ϕ.

The attenuation map is constructed by applying the previous set of operations, pixel by pixel, until the whole image is treated. Figure 5.19 illustrates the attenuation map obtained from the mean phantom image using this procedure.

On this map, it is visible that the attenuation depends on two factors:

- The distance to the center of the phantom: The longer the rays travel through

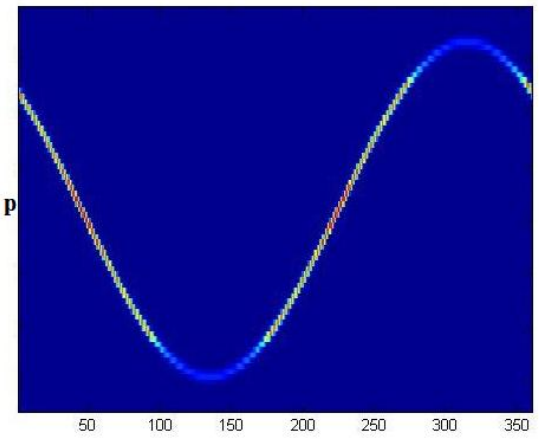


Figure 5.18: Sinogram of the single pixel matrix after multiplication by the sinogram of the whole image.

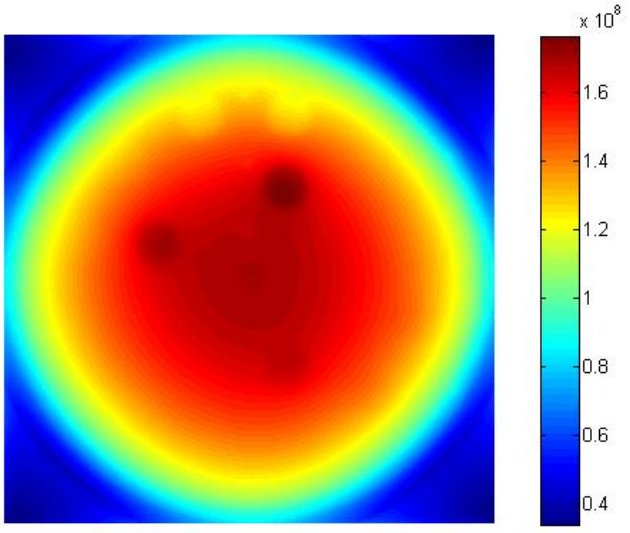


Figure 5.19: Attenuation map of the mean phantom image

tissues, the more attenuated they are. The attenuation is less intense on the sides of the cylinder than on the center, even if the density of the object is the same.

- The density of the objects the rays went through: higher density areas corresponding to the cylinder in the phantom have a higher attenuation than the areas with only low density material. It shows that the higher the density is, the higher the attenuation is.

Since the attenuation map contains the informations about the intensity of the pixels, but also on their position in the image, it could prove useful to model the noise. The next section exploits this attenuation map and the pixel intensities to build a model of the standard deviation of the noise from the set of phantom images.

5.3 Modelling the standard deviation of the noise

The aim of this section is to provide models of the standard deviation of the noise in function of the intensities and attenuation. The first model estimates the standard deviation from the intensity of the pixels (it does not use the attenuation map). The standard deviation-intensity model is written

$$\tilde{\sigma}_{\mathbf{i}} = f(y_{\mathbf{i}}) \ , \tag{5.5}$$

where $\tilde{\sigma}_{\mathbf{i}}$ is the estimated standard deviation at pixel $\mathbf{i}$.

The second model uses the attenuation map to model the noise. The intensity of a pixel depends on its attenuation, but the attenuation also contains information about the localisation of the pixel. The attenuation could then provide more useful information in the standard deviation model. The standard deviation-attenuation model is

$$\tilde{\sigma}_{\mathbf{i}} = g(a_{\mathbf{i}}) \ , \tag{5.6}$$

where $a_{\mathbf{i}}$ is the attenuation at the $\mathbf{i}th$ pixel.

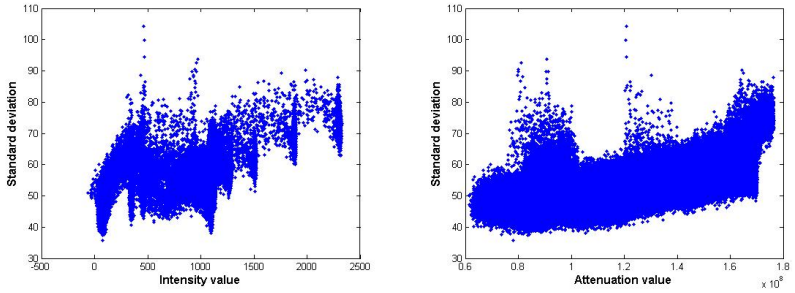
Figure 5.20 represents the standard deviation with respect to the intensity and the attenuation. Visually, it seems that the attenuation provides a better approximation of the standard deviation than the intensity. This is to be confirmed by the regression model results.

The models chosen are simple polynoms of order 1, 2 and 3. These models can be written as

$$\tilde{\sigma}_{\mathbf{i}} = \sum_{n=0}^{N} \alpha_n y_{\mathbf{i}}^n \tag{5.7}$$

or

$$\tilde{\sigma}_{\mathbf{i}} = \sum_{n=0}^{N} \alpha_n a_{\mathbf{i}}^n \ , \tag{5.8}$$



(a) Standard deviation with respect to the intensity

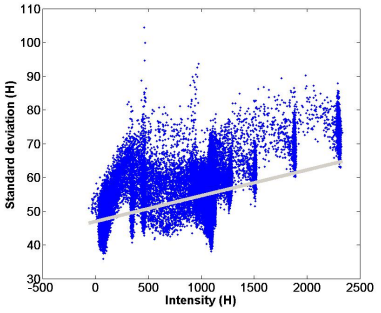
(b) Standard deviation with respect to the attenuation

Figure 5.20: Standard deviation with respect to the intensity and attenuation

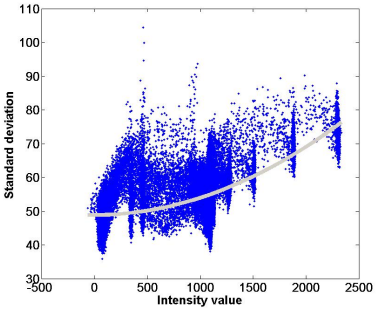
where $N = 1,2,3$ is the order of the models. To construct the models, the pixels corresponding to a zero variance have been deleted from the data to limit the influence of these outliers.

Table 5.1 presents the parameters these models need to minimize the RMSE of the standard deviation. For all models, the errors are lower for the attenuation model than for the intensity model. Concerning the standard deviation/intensity models, Figure 5.21a shows that the linear model is too simple to estimate the standard deviation correctly. This model also has the highest RMSE of all models. The quadratic and third order models (Figures 5.21b and 5.21c) are almost similar both in terms of visual fit and in terms of RMSE. For this reason, in the standard deviation-intensity model, the quadratic model is chosen for sparseness. In the standard deviation-attenuation case, the linear model (Figure 5.22a) is unable to model the data accurately and has the highest error of this family of models. The third order model behaves similarly to the quadratic model, and both have similar RMSE. The quadratic model is chosen again, for the sake of sparseness.

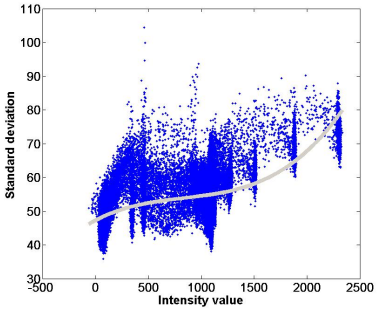
With these models, it is possible to provide an estimation of the standard deviation of the noise on pixel of a new image to construct the standard deviation map, providing it comes from the same scanner.



(a) Standard deviation/intensity model: order 1

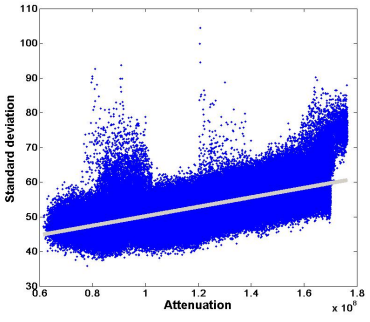


(b) Standard deviation/intensity model: order 2

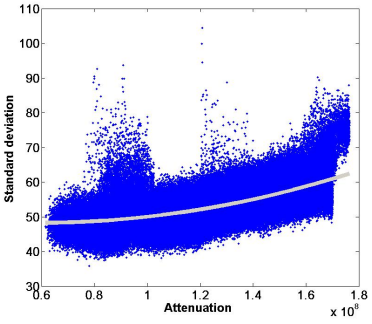


(c) Standard deviation/intensity model: order 3

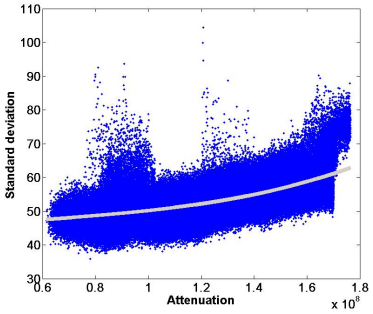
Figure 5.21: Standard deviation-intensity models



(a) Standard deviation/attenuation model: order 1



(b) Standard deviation/attenuation model: order 2



(c) Standard deviation/attenuation model: order 3

Figure 5.22: Standard deviation-attenuation models

std/int	α_0	α_1	α_2	α_3	RMSE
order 1 :	$7.64.10^{-3}$	46.90			5.1931
order 2:	$4.99.10^{-6}$	$1.11.10^{-4}$	48.83		5.0510
order 3:	$6.10.10^{-9}$	$-1.53.10^{-5}$	$1.64.10^{-2}$	47.23	5.0193
std/att					
order 1 :	$1.36.10^{-7}$	36.536			4.5376
order 2 :	$1.02.10^{-15}$	$-1.20.10^{-7}$	51.787		4.4709
order 3 :	$5.38.10^{-24}$	$-9.63.10^{-16}$	$1.1.*10^{-7}$	42.83	4.4696

Table 5.1: Parameters of the standard deviation-intensity and standard deviation-attenuation models. $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ are the parameters of the polynomial model.

5.4 Integration of the standard deviation model into the statistical filters

The previous section details how the standard deviation of the noise on each pixel can be estimated by a model depending on the intensity or on the attenuation. However, the statistical filters presented earlier only use a constant radiometric kernel width over the whole image and cannot take the information from the standard deviation map into account without modifications. This section proposes a variation of these filters that use the standard deviation map as an estimate of the kernel width.

Scalar filters

Classical statistical filters have two main parameters: the spatial kernel width and the radiometric kernel width.

The spatial kernel width controls the weight of a patch depending on the distance between this patch and the patch to be filtered:

$$w_{\mathbf{ij}} = \exp\left(-\frac{\|\mathbf{i}-\mathbf{j}\|_2^2}{2\rho^2}\right) \quad . \tag{5.9}$$

The radiometric kernel width controls the weight of a patch depending of the distance between the intensities of this patch and those of the patch to be filtered.

$$\text{Local M-smoother: } \Psi'_\sigma\left(\|\hat{y}^t_{\mathbf{i}} - x_{\mathbf{j}}\|_2^2/2\right) = \exp\left(-\frac{\|\hat{y}^t_{\mathbf{i}} - x_{\mathbf{j}}\|_2^2}{2\sigma^2}\right) \tag{5.10}$$

$$\text{Bilateral filter: } \Psi'_\sigma\left(\|\hat{y}^t_{\mathbf{i}} - \hat{y}^t_{\mathbf{j}}\|_2^2/2\right) = \exp\left(-\frac{\|\hat{y}^t_{\mathbf{i}} - \hat{y}^t_{\mathbf{j}}\|_2^2}{2\sigma^2}\right) \tag{5.11}$$

The widths of the spatial and radiometric kernels are respectively denoted by ρ and σ.

The radiometric kernel σ in (5.10) and (5.11) uses a constant value over the whole image. The rest of this section shows how to introduce the information provided by the standard deviation map in these filters. For two pixels

$$y_{\mathbf{i}} \sim G(\mu_{\mathbf{i}}, \sigma^2_{\mathbf{i}}) \text{ and } y_{\mathbf{j}} \sim G(\mu_{\mathbf{j}}, \sigma^2_{\mathbf{j}}) \ , \tag{5.12}$$

we can write

$$y_{\mathbf{i}} - y_{\mathbf{j}} \sim G(\mu_{\mathbf{i}} - \mu_{\mathbf{j}}, \sigma^2_{\mathbf{i}} + \sigma^2_{\mathbf{j}}) \ . \tag{5.13}$$

From the usual Gaussian distribution rules, the difference between two pixels, as used in the radiometric kernel of the scalar statistical filters, has a distribution as in (5.13). We suggest to modify (5.10) and (5.11) in the following way in order to adapt to standard deviation of the difference between the two pixels:

$$\text{Local M-smoother:} \phi'_\sigma\left(\|\hat{y}^t_{\mathbf{i}} - x_{\mathbf{j}}\|_2^2/2\right) = \exp\left(\frac{\|\hat{y}^t_{\mathbf{i}} - x_{\mathbf{j}}\|_2^2}{2\sigma_R(\sigma^2_{\mathbf{i}} + \sigma^2_{\mathbf{j}})}\right) \ , \tag{5.14}$$

$$\text{Bilateral filter:} \phi'_\sigma\left(\|\hat{y}^t_{\mathbf{i}} - \hat{y}^t_{\mathbf{j}}\|_2^2/2\right) = \exp\left(\frac{\|\hat{y}^t_{\mathbf{i}} - \hat{y}^t_{\mathbf{j}}\|_2^2}{2\sigma_R(\sigma^2_{\mathbf{i}} + \sigma^2_{\mathbf{j}})}\right) \ . \tag{5.15}$$

In (5.14) and (5.15), σ_R is a global width for the radiometric kernel that allows a global control of the radiometric kernel width over the whole image.

Patch-based and patchwise filters

In the case of the patch-based filters, each pixel has a different variance. This has to be taken into account it the kernel width: for the patch-based local M-smoother, the radiometric kernel has the following form (the same development can be done for the other patch-based filters and the patchwise filters):

$$\Psi'_\sigma(\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2/2) = \exp\left(\frac{\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2^2}{2\sigma^2}\right) \ , \tag{5.16}$$

with

$$\|\hat{\mathbf{y}}^t_{\mathbf{i}} - \mathbf{x}_{\mathbf{j}}\|_2 = \sqrt{\sum_{k=1}^{D} (\hat{\mathbf{y}}^k_{\mathbf{i}} - \mathbf{x}^k_{\mathbf{j}})^2} \ . \tag{5.17}$$

To use the information from the standard deviation map, these equations are modified in the following way: The noise standard deviation is now taken into account in the distance computation.

$$\|\hat{\mathbf{y}}_\mathbf{i}^t - \mathbf{x}_\mathbf{j}\|_2 = \sqrt{\sum_{k=1}^{D} \left(\frac{(\hat{\mathbf{y}}_\mathbf{i}^k - \mathbf{x}_\mathbf{j}^k)^2}{(\sigma_\mathbf{i}^k)^2 + (\sigma_\mathbf{j}^k)^2} \right)} \tag{5.18}$$

In this expression, $\sigma_\mathbf{i}^k$ is the standard deviation of the kth pixel of the $\mathbf{i}$th patch.

This distance is divided by D, the dimensionality of the patches, in order to make it independent from the patch size. The radiometric kernel is now

$$\Psi'_\sigma(\|\hat{\mathbf{y}}_\mathbf{i}^t - \mathbf{x}_\mathbf{j}\|_2^2/2) = \exp\left(\frac{\|\hat{\mathbf{y}}_\mathbf{i}^t - \hat{\mathbf{x}}_\mathbf{j}\|_2^2}{2\sigma_R^2 D} \right) \quad . \tag{5.19}$$

With these new radiometric kernel and distance definitions, the patch-based filters use the standard deviation map in the filtering process.

5.5 Experiments

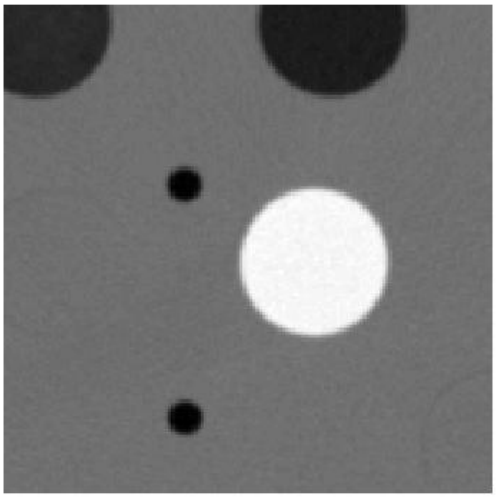
The experiment section aims at testing the performances of the modified statistical filters using the standard deviation models. In real life situations, then mean image is not known, and it is not possible to optimize the parameters. For this reason, the performances of the filters will be tested on the phantom image, with fixed values of σ_R, instead of an optimized one. The experiments should determine which model is robust to a non optimal choice of σ_R in order to get the best results without knowing the ground truth image.

5.5.1 Design of the experiment

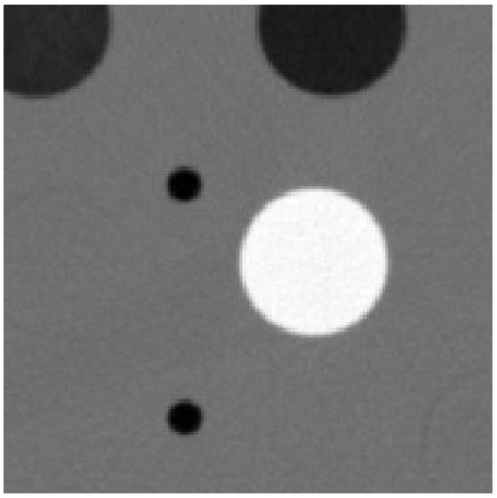
The images chosen for this experiment (see Figure 5.23) are produced by the noisiest (0.5mm layer depth, 75mA) and less noisy (8mm layer depth, 225mA) combination of parameters. In the rest of this chapter they will be called *low noise image* and *high noise image* respectively.

For each image, three models are compared:

- The standard deviation-intensity model;
- The standard deviation-attenuation model;



(a) Cropped image 15



(b) Cropped image 16

Figure 5.23: Selected parts of images 15 and 16

- A constant standard deviation model: the width of the radiometric kernel is constant over the whole image and is evaluated as the mean of the standard deviation maps from the two other models.

These three models are tested with different values of σ_R: $\sigma_R = \{0.2, 0.4, 0.6 \dots 3.8, 4\}$, and the Gaussian and χ_D kernels presented in Chapter 4. The filters used are patch-based bilateral filter with 7×7 patches and a spatial kernel of 15 pixels.

5.5.2 Results

This section presents the results of the experiments conducted on these two images. The lowest mean RMSE, highest mean PSNR, standard deviation of this RMSE, and σ_R giving these results are presented on Tables 5.2 and 5.3 for the low noise image and high noise image respectively. For all models, there is a significant improvement from the RMSE of the original image to the RMSE of all denoised images. For each kernel, the best denoising result are obtained with the constant radiometric kernel over the whole image. However, the standard deviation of the RMSE over the 40 layers shows that the differences between the models are not statistically significant. From these tables, all kind of models and kernels provides the same denoising quality. However, this table only displays the best denoising results while, in practice, the best choice of σ_R is unknown and has to be chosen blindly. If all the methods provide the same denoising quality, they should also prove to be robust to a non optimized choice of σ_R.

low noise image $RMSE = 14.41$					
Kernel	Model	MRMSE	MPSNR	std(RMSE)	σ_R
Gaussian	std-attenuation	7.52	49.84	1.07	0.4
	std-intensity	7.47	49.80	1.03	0.2
	constant	7.44	49.83	1.11	0.4
χ_D	std-attenuation	7.53	49.84	1.06	0.6
	std-intensity	7.38	49.91	1.04	0.6
	constant	7.36	49.92	1.11	0.6

Table 5.2: mean RMSE, mean PSNR, std RMSE, and σ_R giving the best denoising results on the low noise image.

Figures 5.24 and 5.25 present the mean RMSE (MRMSE) of the low noise image and high noise image for $0.2 \leq \sigma_R \leq 4$, for the Gaussian and χ_D kernel. For all images,

high noise image $RMSE = 15.78$					
Kernel	Model	MRMSE	MPSNR	std(RMSE)	σ_R
Gaussian	std-attenuation	8.02	49.20	1.03	0.4
	std-intensity	7.99	49.24	1.05	0.4
	constant	7.90	49.33	1.04	0.4
χ_D	std-attenuation	8.07	49.16	1.01	0.6
	std-intensity	7.92	49.31	1.05	0.6
	constant	7.81	49.44	1.04	0.6

Table 5.3: mean RMSE, mean PSNR, std RMSE, and σ_R giving the best denoising results on the high noise image.

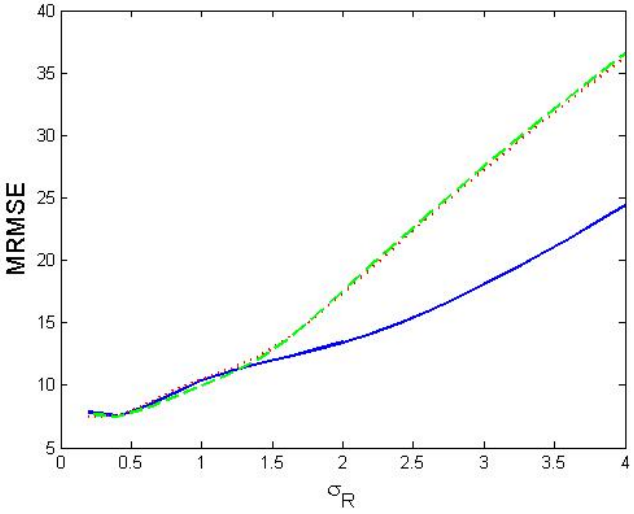
the constant variance and standard deviation-intensity models results decrease rapidly as σ_R increases, while the standard deviation-attenuation model performs better even at high values of σ_R. In the χ_D kernel case, for all models, the MRMSE is more robust that in the Gaussian kernel case but the standard deviation-attenuation model still provides the lowest MRMSE at high σ_R values.

Figure 5.26 shows the mean RMSE of all the models with the Gaussian and the χ_D kernels for the low noise (Figure 5.26a) and high noise (Figure 5.26b) images. Even if the Gaussian kernel best results provide similar RMSE as those from the χ_D kernel, the RMSE increases much faster with σ_R for the Gaussian kernel than for the χ_D kernel. For this reason, the χ_D kernel should be used with the patch-based denoising methods.

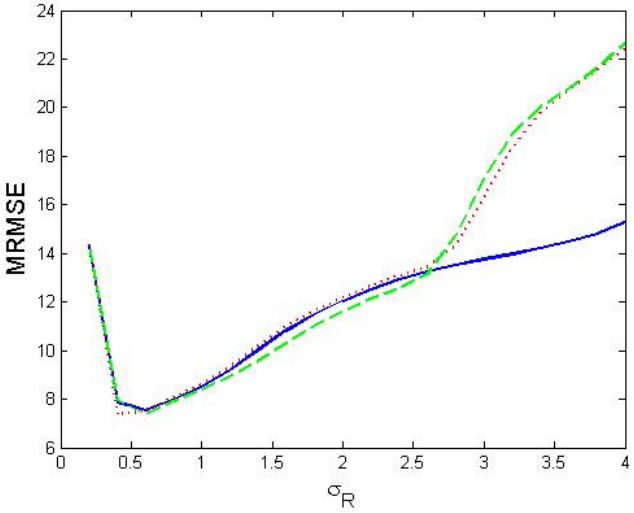
From these experiments, if it is not possible to choose the best σ_R without a ground truth image; the best choice is then to use a χ_D kernel, with the standard deviation-attenuation model, and $\sigma_R = 0.5$. This combination minimizes the loss of performances while σ_R is not optimally chosen.

5.6 Conclusions

While the patch-based and patchwise filters are very efficient filtering algorithms, they are designed to filter a noise with a constant variance over the whole image. In practice however, the noise is far from being constant everywhere on the image; the constant σ and ρ used in the statistical filter have then to be chosen as compromise to maximise the quality over the whole image. Moreover, as the ground truth image is not known it is in practice impossible to find the optimal parameters and they often are chosen by

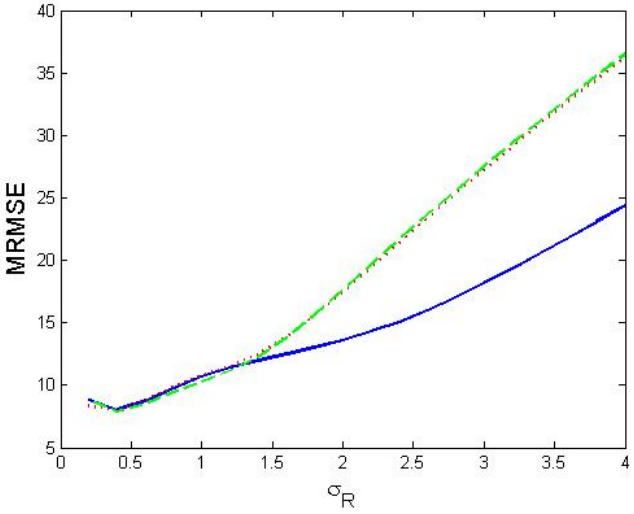


(a) Low noise image: MRMSE of the images denoised with the Gaussian kernel. Continuous line: standard deviation-attenuation model. Dotted line: standard deviation-intensity model. Dashed line: constant standard deviation model.

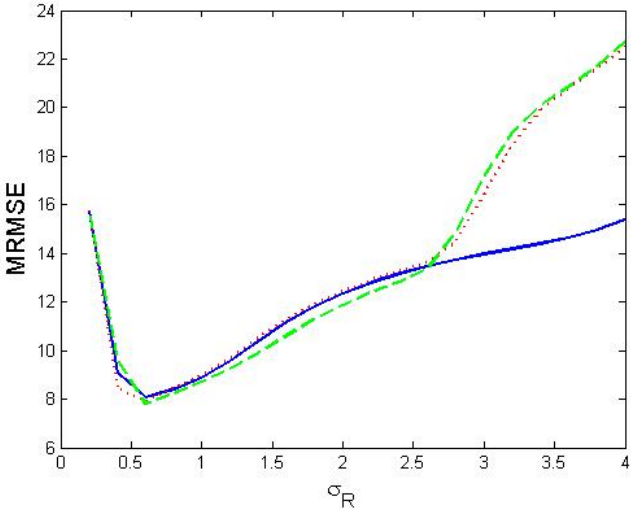


(b) Low noise image: MRMSE of the images denoised with the χ_D kernel. Continuous line: standard deviation-attenuation model. Dotted line: standard deviation-intensity model. Dashed line: constant standard deviation model.

Figure 5.24: Low noise image: MRMSE of the images denoised with different values of σ_R.

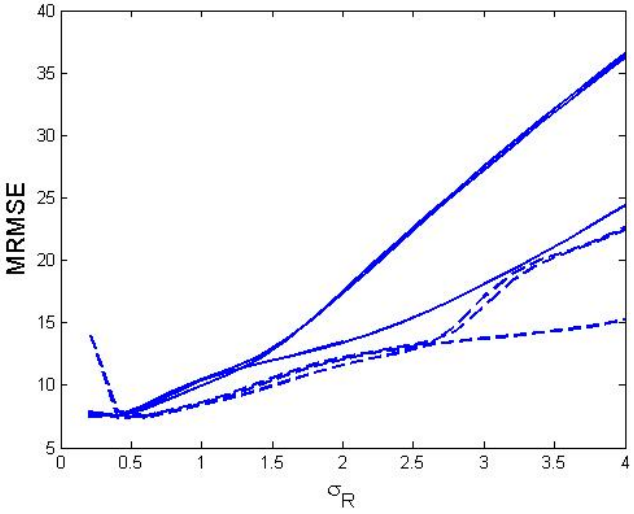


(a) High noise image: MRMSE of the images denoised with the Gaussian kernel. Continuous line: standard deviation-attenuation model. Dotted line: standard deviation-intensity model. Dashed line: constant standard deviation model.

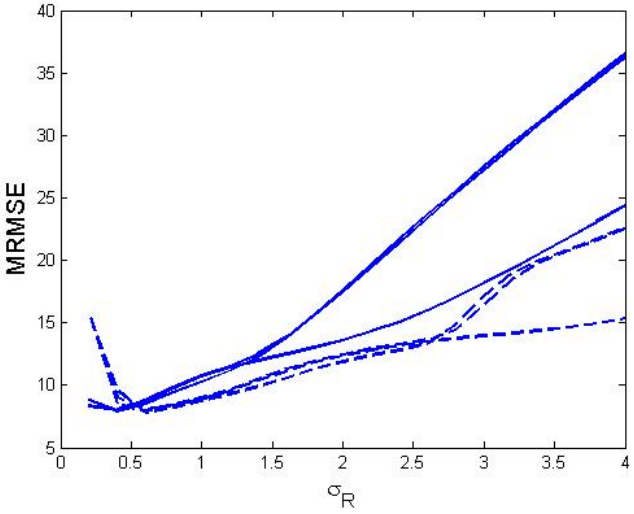


(b) High noise image: MRMSE of the images denoised with the χ_D kernel. Continuous line: standard deviation-attenuation model. Dotted line: standard deviation-intensity model. Dashed line: constant standard deviation model.

Figure 5.25: High noise image: MRMSE of the images denoised with different values of σ_R.



(a) Low noise image: Plain lines: all Gaussian kernel models. Dashed lines: all χ_D kernel models



(b) High noise image: Plain lines: all Gaussian kernel models. Dashed lines: all χ_D kernel models

Figure 5.26: Comparison between the Gaussian kernel models and the χ_D kernel models

default.

This chapter proposes to use a set of images taken from a specific CT scanner, and to model the noise on these images. With such a model, the noise on any new image from the same scanner can be evaluated, and the parameters of the filters can be chosen from this evaluation. For each pixel, the standard deviation of the noise is modeled in function of the pixels intensities and attenuation. A standard deviation map associating the standard deviation of the noise with each pixel is created. The filters are modified in order to be able to use the standard deviation map information.

Experiments are conducted to evaluate the performances of the modified filters, using a standard deviation map, with the filters with a constant σ, estimated from the models of the standard deviation of the noise. A phantom CT scan image is used for the experiments. The experimental section shows that while using a standard deviation map does not significantly improve the result of the denoising in terms of RMSE, it provides a much more robust denoising method. The standard deviation-attenuation model in particular is robust to a non optimal choice of the parameter σ_R. In practice, it is often not even possible to estimate the order of magnitude of the standard deviation from a CT image. Using the information from the standard deviation models to provide an estimation of σ value for the constant model means using these models indirectly. Considering that building the attenuation map is not computationally expensive, applying the standard deviation-attenuation model is competitive once the model is computed for a particular scanner. For these reasons, even if these is no visible improvement at the best situation, the standard deviation-attenuation model, with the χ_D kernel provides better results than the other methods on all non optimal cases and is thus recommended.

Chapter 6

Adaptation of the statistical filters to Poisson noise

The filters used in the previous chapters are adapted for the Gaussian noise case, with constant or variable variance. However, for some kind of images, the hypothesis of Gaussian noise is not valid (on PET scan images for instance). In the case of Poissonian noise, the noise depends on the intensity of the pixels. In this case, the Gaussian hypothesis made in the case of the statistical filter means that the σ parameter will be too large on low intensity areas, and too small in high intensity areas: the statistical filters are unable to filter the noise while preserving the edges and details of the image.

This chapter deals with the problem of the non-Gaussian noise, and in particular the Poissonian noise. In the literature, [54] proposes a procedure based on the minimization of an unbiased estimate of the MSE for Poisson noise. This problem is adressed by [53, 52], with a general methodology (PURE-LET) to design and optimize a wide class of transform-domain thresholding algorithms for denoising images corrupted by mixed Poisson-Gaussian noise. Total-variation regularization has also been adapted to the case of Poisson image denoising in [66, 16, 65]. Wavelets are also used to cope with the Poisson polluted images [17].

Rather than developing a whole new set of algorithms for the specific case of Poissonian noise, we propose to treat the images with the statistical filters detailed above in the thesis, and to use a variance-stabilizing transform in order to make the noise Gaussian on the transformed image. This strategy means that all the advantages of the statistical filters are kept: strong theoretical background, edge preservation, patch-

based and patchwise filters, reasonable computational load, models for estimation of the parameters, flat-top kernels, ...

There are several types of variance stabilizing transforms. In this chapter, two types of transforms are presented: the Anscombe transform [39], and the Fisz transform [39]. This chapter shows how the these transforms can be integrated in the filtering process (in the case of the Anscombe transform) or in the filters (for the Fisz transform). The experimental section uses artificial images with additive Poissonian noise in order to compare filtering results without variance stabilization transformation, and with the Anscombe or Fisz transformation. The images are filtered with the scalar filters and with the patchwise filters. Experiments have been performed on real phantom PET images and the residuals (original image minus filtered image) are displayed to compare filtering results.

6.1 Variance stabilizing transforms

A variance stabilizing transform is a mathematical transformation of the data designed to change the data variance. The variance of the transformed vector should be unrelated to their means. In this chapter, we propose to study the Anscombe transform and the Fisz transform.

6.1.1 Anscombe transform

If $x_{\mathbf{i}} \sim Pois(\lambda)$, the Anscombe transform is defined as [39, 51]

$$n_{\mathbf{i}} = A(x_{\mathbf{i}}) = 2\sqrt{x_{\mathbf{i}} + \frac{3}{8}} \ . \tag{6.1}$$

After this transformation, $n_{\mathbf{i}}$ follows approximately $N(O,1)$. This approximation, however, is not valid for small values of λ [76].

6.1.2 Fisz transform

The Fisz transform is based on the Fisz theorem [39, 51]: Let us define two independent vectors $X_i \sim Pois(\lambda_i)$, $i = 1,2$, and the function $\zeta : \Re^2 \to \Re$

$$\zeta(X_1,X_2) = \begin{cases} 0 & \text{if } X_1 = X_2 = 0 \\ \frac{(X_1 - X_2)}{(X_1 + X_2)} & \text{otherwise} \end{cases} \tag{6.2}$$

If $(\lambda_1, \lambda_2) \to (\infty, \infty)$ and $\lambda_1/\lambda_2 \to 1$ then

$$\zeta(X_1, X_2) - \zeta(\lambda_1, \lambda_2) \to N(0,1) \ . \tag{6.3}$$

Equation (6.3) is called the *Fisz transform*.

[76] proved that the transformed data is closer to a normal distribution than the output of the Anscombe transform. The same paper also showed that the stabilisation effect of the Fisz's transform is still good even if λ_1 and λ_2 are small.

6.2 Introduction of the variance stabilizing transform in the statistical filters

The two variance stabilizing transforms are able to effectively transform any Poisson vector or pair of vectors into a Gaussian vector. This section shows how they can be integrated into the filtering process in order to make the statistical filters able to cope with Poisson noise.

6.2.1 Anscombe transform

Using the Anscombe transform to filter an image polluted by Poisson noise is very easy. Firstly the Anscombe transform is applied on the image before filtering. Secondly, the transformed image is filtered as if it was polluted by Gaussian noise. Finally, the inverse Anscombe transform is applied on the filtered image, to bring it back to the real intensity values.

6.2.2 Fisz transform

This section shows how the Fisz transform can be used to filter images with the statistical filters.

To apply the Fisz transform: $\zeta(X_1, X_2) - \zeta(\lambda_1, \lambda_2)$, λ_1 and λ_2 are needed for each pixel. These values are usually not known and can not be extracted from the image. However, for the filtering application, the variance has to be stabilized, but the distribution does not have to be centered around zero. Using $\zeta(\lambda_1, \lambda_2)$ is then not necessary. The Fisz transform can therefore be reduced to $\zeta(X_1, X_2)$ and there is no need to estimate λ_1 and λ_2. The Fisz transform uses the subtraction between two Poisson vectors to stabilize the variance. A subtraction of the same type is also used during the intensity distance evaluation in the radiometric kernel. This suggests that the Fisz transform

could be introduced into the radiometric kernel in an elegant way. In the case of the bilateral filter, the update rules becomes

$$\hat{y}_{\mathbf{i}}^{t+1} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}}\, \Psi'_{\sigma}(\zeta(\hat{y}_i^t, \hat{y}_j^t))^2/2)\, \hat{y}_{\mathbf{j}}^t}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}(\zeta(\hat{y}_i^t, \hat{y}_j^t))^2/2)} \quad . \tag{6.4}$$

For the patch-based bilateral filter, the update rule is unchanged:

$$\hat{y}_{\mathbf{i}}^{t+1} = \frac{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \hat{\mathbf{y}}_{\mathbf{j}}^t\|_F^2/2) \hat{y}_{\mathbf{j}}^t}{\sum_{\mathbf{j}\in N_{\mathbf{i}}} w_{\mathbf{ij}} \Psi'_{\sigma}(\|\hat{\mathbf{y}}_{\mathbf{i}}^t - \hat{\mathbf{y}}_{\mathbf{j}}^t\|_F^2/2)} \quad . \tag{6.5}$$

but the definition of the distance between two patches becomes

$$d_F(\mathbf{y_i}, \mathbf{y_j}) = \sqrt{\sum_{k=1}^{D} (\zeta(\hat{y}_i, \hat{y}_j))^2} = \|\mathbf{y_i} - \mathbf{y_j}\|_F \quad . \tag{6.6}$$

With the Fisz transform, the variance stabilization is fully integrated into the statistical filter and there is no need of manipulating the image before and after filtering.

6.3 Experiments

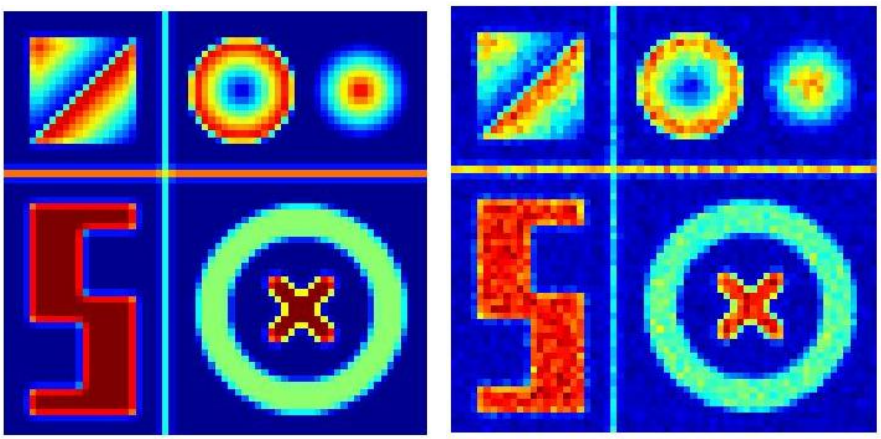
This sections aims at comparing the performances of the statistical filters on Poisson polluted images, using the two types of variance stabilizing transforms on an artifical and a real phantom image

6.3.1 Artificial image

The experiments compare the filtering performances of the patch-based filters without variance stabilizing transform, with the Anscombe transform, and with the Fisz transform. Figure 6.1 shows the artificial image used in the experiments, firstly without noise, then polluted with Poissonian noise.

The noisy image is filtered using the bilateral filter and the patch-based bilateral filter for 1×1 (scalar), 3×3, 5×5 and 7×7 patches, with 15 pixels as spatial kernel width and one single iteration. Each one of these filters uses one of following data processing scheme:

- No variance stabilizing transform: the data is not transformed and the filter treats it as if it was polluted by Gaussian noise.



(a) Original artificial image without noise (b) Artificial image with Poisson noise

Figure 6.1: Example of original and noisy artificial image.

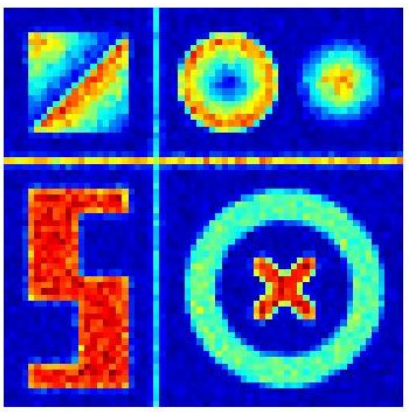
- Anscombe transform: The data is transformed with the Anscombe transform, then the filtering process is applied, and the inverse transform is applied.
- Fisz transform: The Fisz transform is integrated into the filters.

A set of 100 images is generated, polluted by the same noise model. The best algorithm is the one that minimizes the mean RMSE over the 100 repetitions. The parameter σ is optimized for each combination of patch sizes and type of variance stabilizing transform. Table 6.1 reports the results of this experiment. This table shows that the mean RMSE is significantly lower when a variance stabilizing transform is used. Between the variance stabilization transforms, the Fisz transforms performs statistically better than the Anscombe transform, even if the differences between the two are small in practice.

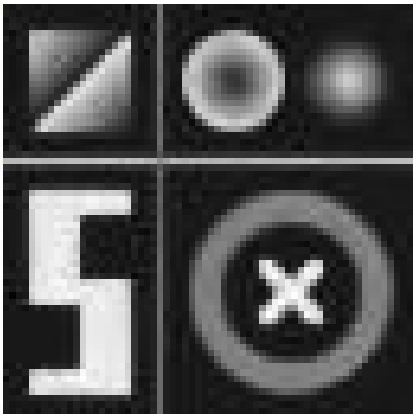
Figure 6.2 shows the noisy image, and the images filtered with each variance stabilizing transform. The visual differences between these images are too subtle to be noticed on visual evaluation.

6.3.2 Real phantom PET image

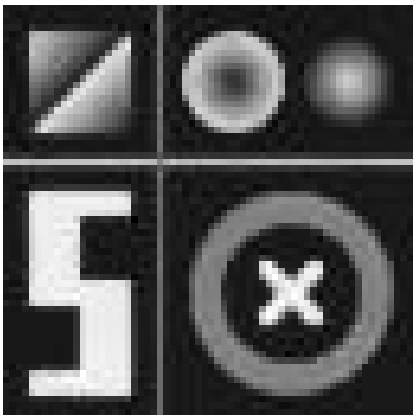
In this section, a phantom PET image is used to estimate the quality of the denoising process. As the ground truth image is not known (as the noise arises during the aqui-



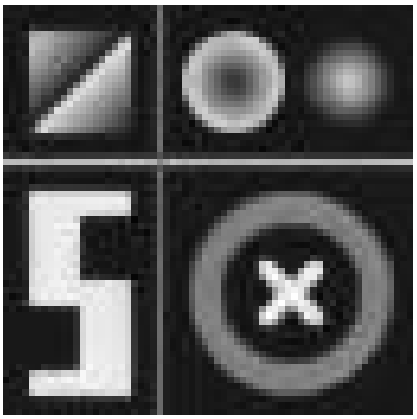
(a) Artificial image with Poisson noise



(b) Filtered image with no variance stabilizing transform



(c) Filtered image with Anscombe variance stabilizing transform



(d) Filtered image with Fisz variance stabilizing transform

Figure 6.2: Examples of filtered images.

Artificial image: MRMSE= 9.1503					
VST	MRMSE	MPSNR	std(RMSE)	σ_R	r
none	6.08	32.45	0.0172	12.58	5
Anscombe	5.20	33.80	0.0138	1.121	3
Fisz	5.13	33.92	0.0141	1.303	5

Table 6.1: For each variance stabilizing transform: mean RMSE (MRMSE), mean PSNR (MPSNR), standard deviation of the RMSE (std(RMSE)), σ_R, and patch size (r) giving the best denoising results on the Poisson polluted image.

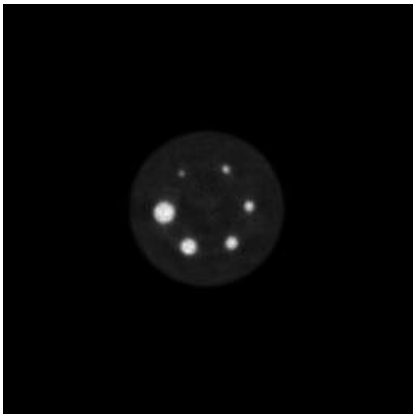


Figure 6.3: Original PET phantom.

sition process, the phantom image is also polluted by noise), the residuals (defined here as the original image minus the filtered image: $x - \hat{y}$) can be used to asses the quality of the filtering process. With this definition, the residual image displays what was filtered during the processing of the image. The visualisation of this residual image gives an insight on how the filters react in combination with the various variance stabilizing transforms.

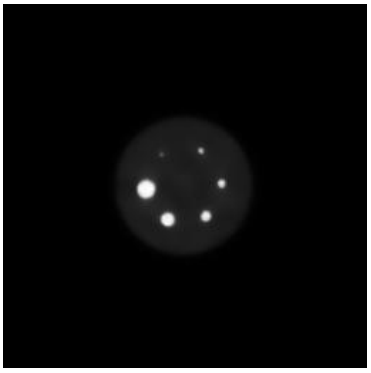
The phantom PET image used is displayed in Figure 6.3.

In the experiment, we propose to compare the denoising results on the phantom PET image using a scalar filter, and a patch-based filter. Each of these filters is tested with no variance stabilizing transform, the Anscombe, and Fisz variance stabilizing transforms. This gives six different ways of treating the images. The results are displayed on Figures 6.4 and 6.5. When no variance stabilizing transform is used, the

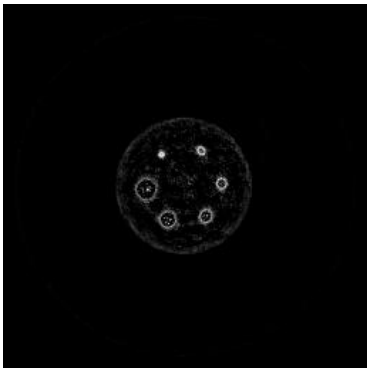
radiometric kernel width needed to get rid of most of the noise also blurs the edges of the cylinders of the phantom. The visual analysis of the residuals of the images pretreated with Anscombe transform shows that the features of the image are better respected with the Anscombe transform than without transform. Still the visible outline of the cylinders shows that even with the Anscombe transform, some details of the structure of the image are lost in the filtering process. This is not the case when the Fisz transform is used. In this case, the residual image does not show the contours of the cylinders. This proves that the only elements that the filtering process removes is the noise. For a similar denoising effect, the details and features of the images are better preserved when the Fisz transform is used than when no transform, or the Anscombe transform are used. For this reason, the Fisz transform is chosen as the best choice of variance stabilizing transform on real PET images.

6.4 Conclusion

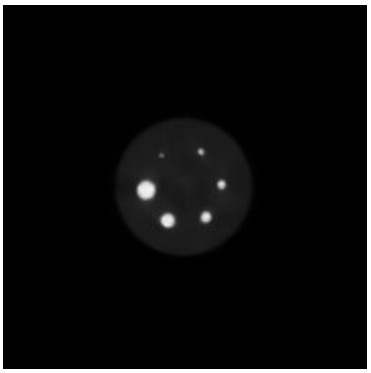
This chapter shows how the statistical filters, based on the hypothesis of Gaussian noise, can be used to filter images polluted with Poissonian noise. A variance stabilizing transform can be used to transform the data into a Gaussian vector. A methodology to use the Anscombe and the Fisz transforms on images has been proposed. Experiments are conducted on an artifical image polluted by Poissonian noise. This image is filtered with statistical filters, without variance stabilizing transform, with the Anscombe transform, and with the Fisz transform. For both variance stabilizing transforms, the mean RMSE of the transformed data is significantly lower than for the non-transformed data. The Fisz transform is also proven to perform better than the Anscombe transform, even if the performance gain is small enough to be invisible on visual analysis.



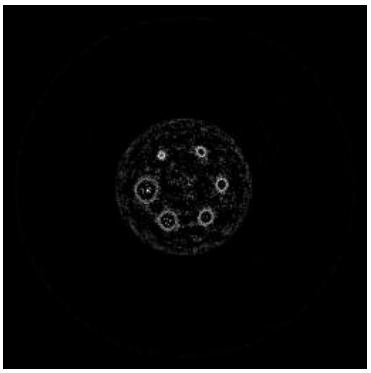
(a) Bilateral filter, no variance stabilizing transform.



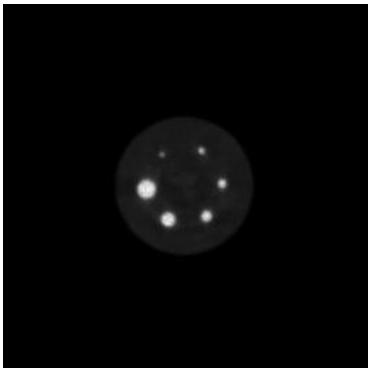
(b) Residuals of bilateral filter, no variance stabilizing transform.



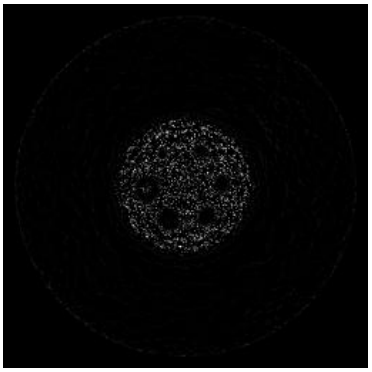
(c) Bilateral filter, Anscombe variance stabilizing transform.



(d) Residuals of bilateral filter, Anscombe variance stabilizing transform.

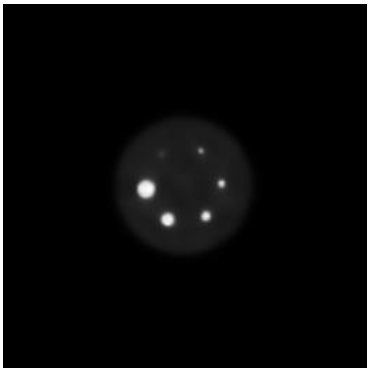


(e) Bilateral filter, Fisz variance stabilizing transform.

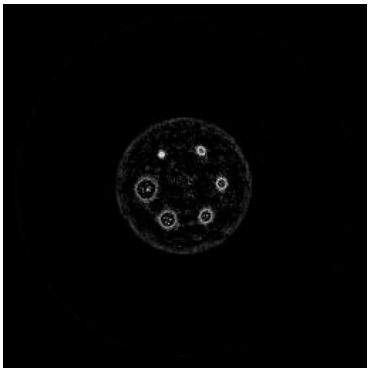


(f) Residuals of bilateral filter, Fisz variance stabilizing transform.

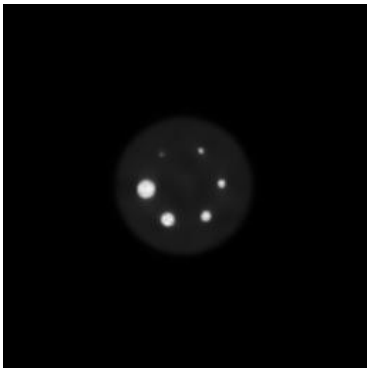
Figure 6.4: Denoising of the PET phantom image with the bilateral filter and residuals



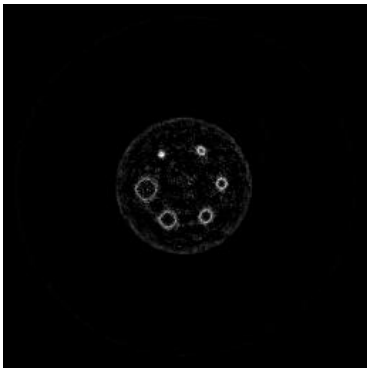
(a) Patch-based bilateral filter, no variance stabilizing transform.



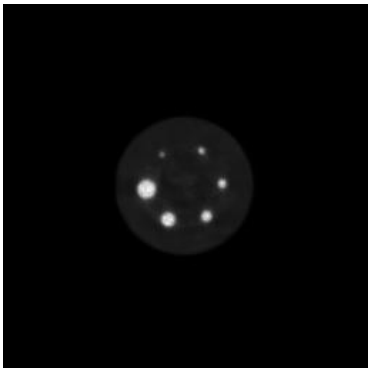
(b) Residuals of patch-based bilateral filter, no variance stabilizing transform.



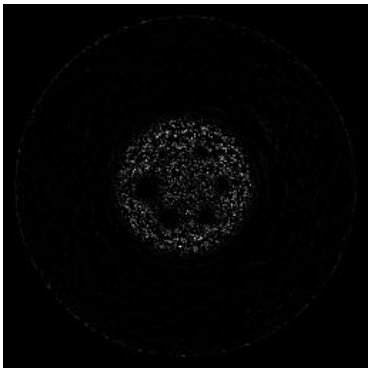
(c) Patch-based bilateral filter, Anscombe variance stabilizing transform.



(d) Residuals of patch-based bilateral filter, Anscombe variance stabilizing transform.



(e) Patch-based bilateral filter, Fisz variance stabilizing transform.



(f) Residuals of patch-based bilateral filter, Fisz variance stabilizing transform.

Figure 6.5: Denoising of the PET phantom image with the patch-based bilateral and residuals

Chapter 7

Contributions

The aim of this chapter is to summarize the contributions of this thesis.

7.1 Contributions in Chapter 3

In this chapter, the scalar filters are modified to use high-dimensional data. The state of the art proposes several local (bilateral filter, local M-smoother) and non-local (mean shift) scalar filters, and a non-local patch-based filter (non-local means). In Chapter 3, two local patch-based filters are introduced: the patch-based bilateral filter and local M-smoother. The link between the mean-shift and the non-local means has been used to propose an iterative version of the non-local means. Chapter 3 also proposes new types of patch-based filters that are called patchwise filters. Two local patchwise filters are proposed (patchwise bilateral filter and patchwise local M-smoother). A patchwise version of the non-local means is also mentioned. These contributions are summarized in Table 7.1.

The experiments compare the denoising between scalar filters, the patch-based filters, and the patchwise filters.

The developments presented in this chapter are published in [28, 29, 31].

7.2 Contributions in Chapter 4

Chapter 4 states that the high-dimensional distance function used in the patch-based and patchwise filters is subject to the curse of dimensionality and in particular to the

	Local	Non-local
Scalar	Bilateral filter Local M-smoother	Mean-shift
Patch-based	Patch-based bilateral filter Patch-based non-local means	Non-local means
Patchwise	Patchwise bilateral filter Patchwise non-local means	Non-local means

Table 7.1: Summary of the state-of-the-art statistical filters and the contributions. The state of the art filters appear in gray. The contributions are written in black.

phenomenon of norm concentration. As a consequence, the similarity kernel used to find out how similar two patches are does not differentiate the close and well separated patches. Chapter 4 proposes to use new types of similarity kernels, called *flat-top* kernels, in order to counter the phenomenon of norm concentration. These new kernels follow closely the shape of the complementary cumulative distribution function of the norm distribution in high-dimensional spaces. Four types of flat-top kernels are proposed and compared in the experiments. The study shows that all the flat-top kernels statistically outperform the classical Gaussian kernel in high-dimensional spaces.

The developments presented in this chapter are published in [26, 27].

7.3 Contributions in Chapter 5

In Chapter 5, a method for estimating the standard deviation from a set of images from one scanner is proposed. The standard deviation of each pixel is modelled with respect to the pixels intensities and attenuations. The models is used to construct a standard deviation map which provides an estimate of the standard deviation for each pixel of the image. In the rest of the chapter, the statistical filters are modified in order to adapt the radiometric kernel width associated with each pixel using the information contained in the standard deviation map. Experiments are performed on CT scan images. The denoising results between filters with a constant radiometric width, and filters with an adaptive kernel width using the models of the variance of the noise are compared. While the denoising results are the same with a constant or variable radiometric width, the experiments showed that the filtering results are much more robust to a non-optimal parameter choice when the model of the variance of the

noise is used.

7.4 Contributions in Chapter 6

Chapter 6 proposes to use a variance stabilizing transform to adapt the filters to the case of Poisson noise. Two different variance stabilizing transforms are used: the Anscombe and Fisz transform. This chapter details how these variance stabilizing transforms can be used in the field of image filtering. In particular, the introduction of the Fisz transform into the scalar, patch-based, and patchwise filters is detailed. Experiments show that using a variance stabilizing transform significantly improves the denoising results when the noise of the image follows a Poisson distribution.

The developments presented in this chapter are published in [30].

Chapter 8

Conclusions

Image analysis has become a very active and important research field in the last years. The evergrowing use of digital images in many demanding applications makes it necessary to refine and improve the quality of the images. Of course, the increasing quality of the acquisition devices plays a major role in the quality of the resulting images, but the need for efficient denoising algorithms is nonetheless there as the acquisition conditions can not always be controlled.

Many paradigms have been applied to denoising algorithms. This thesis focuses on denoising methods based on local or non-local statistics of the images. These *statistical filters* are mainly used on low-dimensional data, although nowadays, some statistical filters started using high-dimensional vectors to improve the filtering results; they are using blocks of images, called patches in the mode estimation procedure. However, high-dimensional spaces have strange and counter-intuitive properties, and one of them, the distance concentration phenomenon, is not taken into account in the traditional filters. Moreover, in real-life situations, the noise encountered often follows a more complex law than a simple Gaussian noise and might not be constant over the whole image. This invalidates the hypothesis of Gaussian noise. Finally, the parameters of the filters are often impossible to estimate in real life situations.

The goal of this thesis is to develop new denoising algorithms based on the state-of-the-art statistical filters, and to adapt these new filters and some existing filters to cope with several of the problems mentioned above: high dimensionality, non-Gaussian noise and parameter estimation.

8.1 Patch-based and patchwise filters

More specifically, we propose to adapt the existing low-dimensional (also called *scalar*) filters to the high-dimensional case. This is motivated by an analogy with the mode estimation procedure. The modes are usually more separated in high-dimensional spaces. Two paradigms have been used to introduce high dimensionality in the scalar statistical filters. The first one is heuristic and consists in replacing pixel-to-pixel distances by patch-to-patch distances in the scalar filters update rules. The main problem with this approach is that the new update rules do not minimize an error function anymore. The second paradigm solves that problem: the patches are introduced into the error functions and the update rules are derived from these new error functions. These update rules, called patchwise filters, are proven to be more general versions of the patch-based filters derived from the first paradigm.

Experiments are conducted on three images with additional Gaussian noise. Eight variations of the scalar, patch-based, and patchwise filters are compared; in addition, SAFIR, a state-of-the-art filter is also tested. The parameters of these filters are optimized for the lowest mean RMSE over 100 images. Results show that the patch-based and patchwise filters all provide a significant improvement over the scalar filters. With the optimised parameters, most of the filters have the lowest mean RMSE after the first iteration while the literature usually uses 3 iterations, with heuristically chosen parameters. All the patch-based filter denoised images show performance gain over the scalar filtered images. Even if the performance gain of the patchwise filters over the patch-based ones is not statistically significant, the patchwise approach is sounder and theoretically better motivated than the patch-based approach. It also provides slightly better results at the same computational cost.

8.2 Flat top kernels

Denoising through mode estimation achieves better denoising results when used in high-dimensional spaces. However, the patch-based and patchwise filters use the Euclidean distance and the Gaussian kernel as a similarity measure. In high-dimensional spaces, the Euclidean distance is subject to the so-called norm concentration phenomenon. As a result, the Gaussian kernel is not able to successfully make the difference between the close and well separated data points. In this thesis, we propose new similarity kernels called flat top kernels designed to cope with the norm concentration phenomenon in high-dimensional spaces.

Four new kernels are proposed : the Gaussian kernel with non-reflexive term, the topped Gaussian kernel, the χ_D kernel, and the Burr kernel. The experiment section compares these new filters: eight images have been filtered with the same filter using four different flat top kernels, and the traditional Gaussian kernel. The best results are obtained using the χ_D kernel, with or without the reflexive term depending of the image. For all images and all noise levels, the RMSE obtained for each one of the new kernels is lower than for the classical Gaussian kernel.

8.3 Estimation of the local noise for local parameter adjustment in the patch-based filters

The statistical filters are designed to filter noise with a constant variance over the whole image. This hypothesis is not valid in practical cases, in particular in the case of CT scan images. Moreover it is often impossible to find the optimal parameters for filtering a specific image as the ground truth is not known.

To solve this problem, we propose to build a model of the standard deviation of the noise for a particular scanner. From a set of phantom images, two models of the standard deviation of the noise are evaluated: the standard deviation as a function of the pixel intensities, and the standard deviation as a function of the pixel attenuations. With these models, the standard deviation of the noise can be estimated for each pixel of the image. The filters are modified in order to use this new information.

A study is conducted on phantom CT images to compare the denoising performances of the filters using the standard deviation map to those of the filters with a constant radiometric kernel width. The experiments show that while the RMSE results are not lower for the filters with the standard deviation map than for the filter with a constant radiometric width, the filters using the standard deviation map are much more robust. As in practice is it often impossible to estimate the correct value of this parameter, the robustness of the models is an important feature.

8.4 Adaptation of the statistical filters to Poisson noise

The noise on images is often more complex than a simple additive Gaussian noise. For instance, in the case of PET scan images, noise approximately follows a Poisson noise, as the intensities result from a count of photons in the sensors. The traditional statistical filters are designed for Gaussian noise denoising only. This means that when

applied to images contaminated with Poissonian noise, the statistical filters are not able to remove the noise efficiently. We propose to use variance stabilizing transforms to stabilize the variance of the noise on the images. The statistical filters can then be applied on the images as if the noise was Gaussian. Two variance stabilizing transforms are suggested: the Anscombe transform and the Fisz transform. A methodology to use the Anscombe transform is proposed, and the statistical filters are modified in order to integrate the Fisz transform.

Experiments are made on an artificial image polluted with Poisson noise. The study compares results from filters without variance stabilizing transform, and with the Anscombe and Fisz transforms. Scalar and patchwise filters are used. For both variance stabilizing transforms, the mean RMSE of the transformed data is significantly lower than for the non-transformed data. The Fisz transform is also proven to perform better than the Anscombe transform.

8.5 Take home message

The aim of this thesis is to modify the existing statistic filters in order to obtain better filtering results, to be more robust to parameter choices, and to be able to cope with other noise models than additive Gaussian noise. Several improvement have been proposed in order to cope with these issues. Firstly we showed that using high-dimensional data in the statistical filters is a sure way to significantly improve the filtering performances on all types of images. Secondly, when patches (high-dimensional data) are used, some precautions have to be taken, especially because of the behaviour of the Euclidean norm. When patches are used, we propose to optimize the filtering results by using a flat-top kernel in the filters instead of the classically used Gaussian kernel.

Thirdly, in the case of real world CT images where default parameters are used for the denoising algorithms, we propose to use a phantom image to build a model of the noise created by this scanner. This model can then be used to estimate the variance of the noise on any new image taken with this scanner. With these modifications, the filters use this variance estimation to adapt the radiometric kernel width for each pixel instead of using a constant radiometric width on the whole image. The filters using a variable radiometric kernel width are much more robust to a suboptimal parameter choice than the filters with a constant one. Fourthly, in the case of Poissonian noise, we propose to use the Fisz variance stabilizing transform, integrated into the filters, to adapt the filter to the hypothesis of Poissonian noise and optimize the filtering results.

Bibliography

[1] S.A. Ali, S. Vathsal, and Lal kishore K. An efficient denoising technique for ct images using window-based multi-wavelet transformation and thresholding. *European Journal of Scientific Research*, 48(2):315–325, 2010.

[2] S. A. Awate and R. T. Whitaker. Image denoising with unsupervised, information-theoric, adaptative filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(3):364–376, 2006.

[3] N. Azzabou, N. Paragios, F. Cao, and F. Guichard. Variable bandwidth image denoising using image-based noise models. In *IEEE Conf. in Computer Vision and Pattern Recognition*, pages 1–7, Minneapolis, MN, June 2007.

[4] D. Barash and D. Comaniciu. A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean-shift. *Image and Video Computing*, 22(1):73–81, 2004.

[5] R. Bellman. *Adaptative Control Processes: A Guided Tour*. Princeton University Press, Princeton, NJ, 1961.

[6] K. Beyer, S. Goldstein, R. Ramakirshan, and U. Shaft. When is "nearest neighbor" meaningful? In Springer-Verlag, editor, *Seventh international conference on database theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235, Jerusalem, Israel, 1999.

[7] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision archive*, 19(1):57–91, 1996.

[8] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. on Image Processing*, 7(3):421–432, 1998.

[9] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. on Image Processing*, 7(3):421–432, 1998.

[10] Anja Borsdorf, Rainer Raupach, Thomas Flohr, and Joachim Hornegger. Wavelet based noise reduction in ct-images using correlation analysis. *IEEE Trans. Med. Imaging*, 27(12):1685–1703, 2008.

[11] T. Brox, O. Kleinschmidt, and D. Cremers. Efficient non-local means for denoising of textural patterns. *IEEE Trans. on Image Processing*, 17(7):1083–1092, 2008.

[12] A. Buades, B. Coll, and J.-M. Morel. The staircasing effect in neighborhood filters and its solution. *IEEE Trans. on Image Processing*, 15(6):1499–1505, 2006.

[13] 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.

[14] I.W. Burr. Cumulative frequency functions. *Annals of Mathematical Statistics*, 13:215–232, 1942.

[15] R. Cao, A. Cuevas, and W. G. Manteiga. A comparative study of several smoothing methods in density estimation. *Computational Statistics and Data Analysis*, 17:153–176, 1994.

[16] R. H. Chan and K. Chen. Multilevel algorithm for a poisson noise removal model with total-variation regularization. *Int. J. Comput. Math.*, (5):1–18, May 2007.

[17] C. Charles and J. P. Rasson. Wavelet denoising of poisson-distributed data and applications. *Comput. Stat. Data Anal.*, 43:139–148, June 2003.

[18] R. Charnigo, J. Sun, and R. Muzic. A semi-local paradigm for wavelet denoising. *IEEE Trans. on Image Processing*, 15(3):666–677, Mar 2006.

[19] P. Chatterjee and P. Milanfar. A generalization of non-local means via kernel regression. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6814, Mar 2008.

[20] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[21] C.K. Chu, I. Glad, F. Godtliebsen, and Marron J.S. Edge-preserving smoothers for image processing. *Journal of the American Statistical Association*, 93, 1998.

[22] R. Coifman and D. Donoho. Translation invariant denoising. *Wavelets and statistics*, pages 125–150, 1995.

[23] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, may 2002.

[24] K. Dabov, A. Foi, and K. Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. pages no. 6812–07, San Jose, California, USA, January 2008.

[25] J. Darbon, A. Cunha, T.F. Chan, S. Osher, and G. Jensen. Fast nonlocal filtering applied to electron cryomicroscopy. In *IEEE Int. Symposium on Biomedical Imaging*, pages 1331–1334, Paris, France, May 2008.

[26] A. de Decker, J.A. Lee, D. Francois, and M. Verleysen. Mode estimation in high-dimensional spaces with flat-top kernels : Application to image denoising. In *European Symposium on Artificial Neural Networks (ESANN)*, pages 411–417, Bruges, Belgium, April 2010.

[27] A. de Decker, J.A. Lee, D. Francois, and M. Verleysen. Image denoising through mode estimation in high-dimensional spaces with flat-top kernels. *Neurocomputing*, 74, Issue 9, April 2011.

[28] A. De Decker, J.A. Lee, and M. Verleysen. Patch-based bilateral filter and local m-smoother for image denoising. In *European Symposium on Artificial Neural Networks – Advances in Computational Intelligence and Learning*, pages 95–100, Bruges (Belgium), April 2009.

[29] A. de Decker, J.A. Lee, and M. Verleysen. Performance assessment of patch-based bilateral denoising. In *Workshop on Medical Image Analysis and Description for Diagnosis Systems,*, pages 52–61, Porto, Portugal, 2009.

[30] A. de Decker, J.A Lee, and M. Verleysen. Variance stabilizing tranformations in patch-based bilateral filters for poisson noise image denoising. In *Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3673–6, Minneapolis, Minnesotta, USA, 2-6 September 2009 2009.

[31] A. de Decker, J.A. Lee, and M. Verleysen. A principled approach to image denoising with similarity kernels involving patches. *Neurocomputing*, 73, Issues 7–9(4):1199–1209, March 2010.

[32] C.-A. Deledalle, L. Denis, and F. Tupin. Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Trans. on Image Processing*, vol. 18, no. 12:2661–2672, 2009.

[33] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[34] D.L. Donoho. High-dimensional data analysis: The curse and blessings of dimensionality. Aide-memoire for a lecture for the American Math. Society : Math. Challenges of the 21st Century, 2000.

[35] M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Trans. on Image Processing*, 11(10):1141–1151, 2002.

[36] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736 – 3745, Dec. 2006.

[37] M. Farmen and J. S. Marron. An assessment of finite sample performance of adaptive methods in density estimation. *Computational Statistics and Data Analysis*, 30:143–168, 1999.

[38] D. Francois, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, July 2007.

[39] P. Fryzlewicz and G. P. Nason. A haar-fisz algorithm for poisson intensity estimation. *Journal of Computational and Graphical statistics*, 13(3):621–638, 2004.

[40] G. Gilboa and S. Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6(2):595–630, 2007.

[41] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.

[42] B. Goossens, H. Luong, A. Pizurica, and W. Philips. An improved non-local denoising algorithm. pages 25–29, Lausanne, Switzerland 2008.

[43] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. Wiley & Sons, New York., 1986.

[44] P. J. Huber. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. Wiley & Sons, New York, 1981.

[45] A.C. Kak and M. Slaney. *Principles of Computerized Tomographic Imaging*. IEEE Press, 1988.

[46] C. Kervrann and J. Boulanger. Optimal spatial adaptation for patch-based image denoising. *IEEE Trans. on Image Processing*, 15(10):2866–2878, 2006.

[47] C. Kervrann and J. Boulanger. Optimal spatial adaptation for patch-based image denoising. *IEEE Trans. on Image Processing*, 15(10):2866–2878, 2006.

[48] C. Kervrann and J. Boulanger. Local adaptivity to variable smoothness for exemplar-based image denoising and representation. *International Journal of Computer Vision*, 79(1):45–69, August 2008.

[49] C. Kervrann, J. Boulanger, and P. Coupé. Bayesian non-local means filter, image redundancy and adaptative dictionaries for noise removal. In *Conf. Scale-Space and Variational Methods*, volume 4485, pages 520–532, Ischia, Italy, May 2007.

[50] S. Kindermann, S. Osher, and P. W. Jones. Deblurring and denoising of images by non-local functionals. *Multiscale Modeling & Simulation*, 4(4):1091–1115, 2005.

[51] J.A. Lee, X. Geets, V. Gregoire, and A. Bol. Edge-preserving filtering of images with low photon counts. *IEEE Trans. on pattern analysis and machine intelligence*, 30(6):1014–1027, 2008.

[52] F. Luisier, T. Blu, and M. Unser. Image denoising in mixed poisson?gaussian noise. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 20(3):696–708, MARCH 2011.

[53] F. Luisier, T. Blu, and M. Unser. Image denoising in mixed poisson?gaussian noise. *IEEE Transactions on image Processing*, 20(3):696 – 708, March 2011.

[54] F. Luisier, C. Vonesch, T. Blu, and M. Unser. Fast interscale wavelet denoising of poisson-corrupted images. *Signal Processing*, 90:415–?427, 2010.

[55] Markus Mayer, Anja Borsdorf, Harald Köstler, Joachim Hornegger, and Ulrich Rüde. Nonlinear diffusion vs. wavelet based noise reduction in ct using correlation analysis. In *VMV'07*, pages 223–232, 2007.

[56] P. Mrazek, J. Weickert, and Bruhn A. *Geometric Properties for Incomplete data*, chapter On Robust Estimation and smoothing with Spatial and Tonal Kernels, pages 335–352. Springer, 2006.

[57] D. Darian Muresan. Adaptive principal components and image denoising. In *in Proc. Int. Conf. Image Processing*, pages 101–104, 2003.

[58] S. Osher, M. Burger, Goldfarb D., Xu J., and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modelling & Simulation*, 4(2):460–489, 2005.

[59] B. U. Park and B. A. Turlach. Practical performance of several data driven bandwidth selectors. *Computational Statistics*, 7:251–270, 1992.

[60] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, Sep 1962.

[61] P. Perona and J. Malik. Scale-space and edge-detection using anisotropic diffusion. *Trans. on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[62] Sivakumar R. Denoising of computer tomography images using curvelet transform. volume 2(1), pages 21–26, February 2007.

[63] S. Roth and M.J. Black. Steerable random fields. volume 17(1), pages 1–8, October 2007.

[64] S. Roth and M.J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009.

[65] A. Sawatzky, C. Brune, J. Müller, and M. Burger. *Computer analysis of images and patterns*. 2009.

[66] R. Chartrand T. Le and T. J. Asaki. A variational approach to reconstructing images corrupted by poisson noise. *J. Math. Imaging Vis.*, 27(3):257–263, Apr 2007.

[67] E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (BV, L^2) decompositions. *Multiscale Modeling & Simulation*, 2(4):554–579, 2004.

[68] T. Tasdizen. Principal components for non-local means image denoising. In *IEEE Int. Conf. on Image Processing*, pages 1728–1731, San Diego, California, Oct 2008.

[69] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Int. Conf. on Computer Vision*, pages 893–846, Bombay, India, 1998.

[70] R. van den Boomgaard and J. van de Weijer. On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. In *IEEE Conf. on Pattern Recognition*, volume 3, pages 927–930, Quebec, Canada, August 2002.

[71] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *Journal on Scientific Computing*, 17(1):227–238, 1996.

[72] Dennis D. Wackerly, William Mendenhall III, and Richard L. Scheaffer. *Mathematical Statistics with Applications*. Duxbury Advanced Series, sixth edition edition, 2002.

[73] J. Weickert. *Anisotropic Diffusion in Image Processing*. ECMI Series, 1998.

[74] G. Winkler, V. Aurich, K. Hahn, A. Martin, and K. Rodenacker. *Noise reduction in images: Some recent edge-preserving methods.* Mathematik, Informatik und Statistik, Statistik. Sonderforschungsbereich, 1998.

[75] Q Xia, Joseph Y Lo, Kai Yang, Carey E Floyd, and John M Boone. Dedicated breast computed tomography: volume image denoising via a partial-diffusion equation based technique. *Med Phys*, 35(5):1950–1958, 2008.

[76] B. Zhang, B. Zhang, M.J. Fadili, and J.-L. Starck. Multi-scale variance stabilizing transform for multi-dimensional poisson count image denoising. In M.J. Fadili, editor, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2006*, volume 2, pages 1035–1046, 2006.

[77] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43:1531–1549, 2010.