



Information Theory Oriented Image Restoration

Cesario Vincenzo Angelino

► To cite this version:

Cesario Vincenzo Angelino. Information Theory Oriented Image Restoration. Signal and Image processing. Université Nice Sophia Antipolis, 2011. English. NNT: . tel-01362134

HAL Id: tel-01362134

<https://theses.hal.science/tel-01362134>

Submitted on 8 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Nice - Sophia Antipolis

ÉCOLE DOCTORALE STIC

Science et Technologies de l'Information et de la Communication

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis

Mention : Automatique, Traitement du Signal et des Images

présenté et soutenue par

Cesario Vincenzo ANGELINO

Laboratoire I3S (Informatique, Signaux et Systèmes de Sophia Antipolis)

**INFORMATION THEORY ORIENTED
IMAGE RESTORATION**

Thèse dirigée par Michel BARLAUD et Eric DEBREUVE

Soutenue publiquement le 9 septembre 2011 devant le jury composé de

Luigi PAURA	Professeur des Universités (Federico II Naples)	Rapporteur
Philippe SALEMBIER	Professeur des Universités (Barcelona)	Rapporteur
Michel BARLAUD	Professeur des Universités (Nice-Sophia Antipolis)	Directeur de thèse
Eric DEBREUVE	Chargé de recherche CNRS	Directeur de thèse
Giovanni POGGI	Professeur des Universités (Federico II Naples)	Directeur en co-tutelle

TESI DI DOTTORATO

UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”

DIPARTIMENTO DI INGEGNERIA ELETTRONICA
E DELLE TELECOMUNICAZIONI

DOTTORATO DI RICERCA IN
INGEGNERIA ELETTRONICA E DELLE TELECOMUNICAZIONI

INFORMATION THEORY ORIENTED IMAGE RESTORATION

CESARIO VINCENZO ANGELINO

Il Coordinatore del Corso di Dottorato

Ch.mo Prof. Niccoló RINALDI

I Tutori

Ch.mo Prof. Michel BARLAUD

Ch.mo Prof. Giovanni POGGI

Résumé : Cette thèse aborde la formulation par la théorie de l'information des problèmes de traitement d'image. Cette formulation exprime la solution au travers de la minimisation d'une énergie. Ces énergies appartiennent à la classe non paramétrique au sens où elles ne font aucune hypothèse paramétrique sur la distribution des données. Les énergies sont exprimées directement en fonction des données considérées comme des variables aléatoires. Toutefois, l'estimation non paramétrique classique repose sur des noyaux de taille fixe moins fiables lorsqu'il s'agit de données de grande dimension. En particulier, des méthodes récentes dans le traitement de l'image dépendent des données de type "patch" correspondant à des vecteurs de description de modèles locaux des images naturelles, par exemple, les voisinages de pixels. Le cadre des k -plus proches voisins résout ces difficultés en s'adaptant localement à la distribution des données dans ces espaces de grande dimension. Sur la base de ces prémisses, nous développons de nouveaux algorithmes qui s'attaquent principalement à deux problèmes du traitement de l'image : la déconvolution et le débruitage. Le problème de la restauration est développé dans les hypothèses d'un bruit blanc gaussien additif puis successivement adaptés à domaines tels que la photographie numérique et le débruitage d'image radar (SAR). Le schéma du débruitage est également modifié pour définir un algorithme d'inpainting.

Mots clefs : *restauration d'image, théorie de l'information, estimation non paramétrique, entropie, mean-shift, déconvolution, patch, débruitage non local, SAR, despeckling, inpainting d'image.*

Abstract: This thesis addresses informational formulation of image processing problems. This formulation expresses the solution through a minimization of an information-based energy. These energies belong to the nonparametric class in that they do not make any parametric assumption on the underlying data distribution. Energies are expressed directly as a function of the data considered as random variables. However, classical nonparametric estimation relies on fixed-size kernels which becomes less reliable when dealing with high dimensional data. Actually, recent trends in image processing rely on patch-based approaches which deal with vectors describing local patterns of natural images, e. g., local pixel neighborhoods. The k -Nearest Neighbors framework solves these difficulties by locally adapting the data distribution in such high dimensional spaces. Based on these premises, we develop new algorithms tackling mainly two problems of image processing: deconvolution and denoising. The problem of denoising is developed in the additive white Gaussian noise (AWGN) hypothesis and successively adapted to no AWGN realm such as digital photography and SAR despeckling. The denoising scheme is also modified to propose an inpainting algorithm.

Keywords: *image restoration, information theory, nonparametric estimation, entropy, mean-shift, deconvolution, patch, nonlocal denoising, SAR, despeckling, image inpainting.*

*To my mother,
Giovanna*

Contents

Acknowledgments	vi
List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Context	1
1.2 Contributions of the thesis	2
2 ESTIMATION OF SOME STATISTICAL MEASURES	5
2.1 Introduction	5
2.2 Kernel density estimation (KDE)	6
2.2.1 Multivariate KDE	6
2.2.2 Bandwidth selection	7
2.3 Nonparametric entropy estimation	10
2.3.1 Plug-in estimates	10
2.3.2 The k NN framework	10
2.4 Entropy derivative and Mean Shift	11
2.5 Mean-Shift approximation	13
2.5.1 Based on fixed kernel size	13
2.5.2 Based on variable kernel size	13
2.6 Conclusion	14
I Deconvolution	19
3 ENTROPY-BASED DECONVOLUTION	21
3.1 Introduction	21
3.2 State of the art	22
3.2.1 Deterministic approach	22
3.2.2 Statistical methods	23
3.2.3 Why using entropy?	24
3.3 Proposed method: MRED	24

3.3.1	Minimizing the residual entropy	24
3.3.2	Energy lower bound	25
3.3.3	Energy derivative	26
3.4	Experimental Results	27
3.5	Discussion and perspectives	29
II	Denoising	37
4	PATCH-BASED DENOISING	39
4.1	Introduction	39
4.2	Neighborhood Denoising: state of the art	40
4.2.1	The UINTA algorithm	40
4.2.2	Non-Local Means	40
4.2.3	Block Matching 3D (BM3D)	41
4.3	Proposed method: PCkNN	43
4.3.1	Entropy-based formulation	43
4.3.2	Energy lower bound	43
4.3.3	Energy derivative	45
4.3.4	Full patch denoising	47
4.3.5	Confidence-based patch combination	48
4.3.6	Summary of the algorithm	49
4.4	Experiments	49
4.5	Improving PCkNN with robust patch similarity	55
4.5.1	Robustness to noise	59
4.5.2	Transformation invariance	59
4.5.3	Effect of the transformations: A toy example	59
4.6	Conclusion and perspectives	60
III	Specific applications	63
5	DIGITAL PHOTOGRAPHY	65
5.1	Introduction	65
5.2	Noise and sensor	65
5.2.1	Sensor and image acquisition	65
5.2.2	Noise model	67
5.2.3	Parameters estimation	69
5.2.4	Demosaicing	70
5.3	Adapting PCkNN to CFA images	70
5.3.1	Intra-channel denoising	71
5.3.2	Inter-channel denoising	72

6	SAR despeckling	81
6.1	Introduction	81
6.2	State of the art	81
6.2.1	Spatial domain techniques	81
6.2.2	Wavelet based techniques	82
6.2.3	Non-local techniques	83
6.3	The BM3D algorithm and its SAR-oriented version	84
6.4	Proposed SAR-oriented modifications in detail	85
6.4.1	Block similarity measure	85
6.4.2	Group shrinkage	87
6.4.3	Aggregation	89
6.5	Experimental results	90
6.5.1	Gold standards and parameter setting	90
6.5.2	Results with simulated speckle	91
6.5.3	Results with actual SAR images	94
6.6	Conclusion and future work	97
7	Digital Image Inpainting	103
7.1	Introduction	103
7.2	State of the art	106
7.2.1	PDE based algorithms	107
7.2.2	Exemplar-based texture synthesis	108
7.3	Proposed algorithm and results	109
7.4	Conclusion and perspectives	110
	Conclusion	115
IV	Appendix	117
A	Derivative of the residual entropy E	119
A.1	Definitions and notations	119
A.2	Derivative of p	120
A.3	Derivative of E	121
B	Computation of the second order term $\chi(w)$ of the entropy derivation	125
B.1	Gaussian Residual PDF	125
B.1.1	Gaussian Kernel	125
B.1.2	Epanechnikov Kernel	126
B.2	Histogram Residual pdf	126
B.2.1	Gaussian Kernel	127
B.2.2	Epanechnikov Kernel	128
C	Derivative of Conditional entropy	129

D	PCKNN: comparative tables	131
----------	----------------------------------	------------

List of Figures

2.1	Kernel density estimation performance on a 1-D Gaussian mixtures for different bandwidth. Actual distribution is in black, kernel density estimate is in blue and kernels are in red. h is the plugin estimate using rule of thumb, from left to right, top to bottom: actual PDF, PDF estimated with $0.2h$, PDF estimated with h , PDF estimated with $5h$	8
2.2	Multivariate kernel density estimation performance on a 2-D Gaussian mixtures for different bandwidth. h is the plugin estimate using rule of thumb, from left to right, top to bottom: actual PDF, PDF estimated with $0.2h$, PDF estimated with h , PDF estimated with $5h$	9
2.3	Entropy estimation of different PDF with relative errors. Circles represent Parzen, <i>i.e.</i> fixed bandwidth, estimation. Squares represent k NN estimator ($k = 10$) and triangles represent the estimator presented in [NK07].	16
2.4	3D data uniformly distributed on embedded 2D manifolds. In lexicographic order: Plane, Spherical surface, Circle (intersection between the Plane and a sphere), Ring torus and Swiss roll. Estimated entropies are given in Table 2.6.	17
3.1	Residual Entropy as function of noise entropy. Initial residual entropy (blue), Final residual entropy (green), Theoretical lower bound (red).	26
3.2	Algorithm performances for gaussian noise (top) and uniform noise (bottom) as function of noise entropy. (a) Initial (blue) and Final (red) PSNR; (b) Initial (blue) and Final (red) SSIM.	29
3.3	Left: Degraded image with a gaussian filter ($\sigma^2 = 3$), and zero mean Uniform noise of entropy 2 bits. Right: Residual entropy as function of iterations.	31
3.4	Deconvoluted images comparison, Uniform noise, entropy 2 bit. .	32
3.5	Deconvoluted images comparison, Gaussian mixture noise, entropy 1.85 bits	33
3.6	Mixture noise. Left: Gaussian mixture pdf (entropy 1.85 bits). Right: Gaussian/Uniform mixture pdf (entropy 0.7 bits).	34

3.7	Deconvoluted images comparison, Gaussian + Uniform mixture noise, entropy 0.7 bits	35
4.1	Flowchart of BM3D algorithm [DFKE07b].	42
4.2	Image patch illustration.	44
4.3	Behavior of the conditional entropy before and after denoising. Squares represent the conditional entropy, $h(\tilde{X} \tilde{Y})$, of the noisy image. Diamonds represent the conditional entropy, $h(\hat{X} \tilde{Y})$, after denoising with proposed method. Circles denote the lower bound $h(X \tilde{Y})$. Left: Lena. Right: Mandrill.	46
4.4	Patch overlapping illustration.	48
4.5	Patch combination step illustration.	50
4.6	Block diagram for PCkNN algorithm.	50
4.7	Pseudo code for the proposed PCkNN algorithm.	51
4.8	PSNR for the image Elaine (left) and Lena (right).	52
4.9	In lexicographic order: Noisy, Original, BM3D, PCkNN, NL-means, and UINTA. The image Elaine was corrupted with an additive white Gaussian noise with standard deviation of $\sigma = 25$. . .	54
4.10	A close-up on the image Elaine. From left to right: Original, BM3D, and PCkNN. First row: image alone; second row: isolevel lines superimposed on the image.	55
4.11	A close-up on the image Lena. From left to right: Original, BM3D, and PCkNN. First row: image alone; second row: isolevel lines superimposed on the image.	56
4.12	A close-up on the image Barbara. From left to right: Original, BM3D, and PCkNN. First row: image alone; second row: isolevel lines superimposed on the image.	57
4.13	Normalized Total Variation. Left: Elaine. Right: Lena.	57
4.14	FT Amplitude of the error between the blurred original and the denoised image. Left BM3D, right PCkNN. First row: Elaine, second row: Lena.	58
4.15	Energy Spectrum Density (ESD) (dB) of the reconstruction error. Left: Elaine. Right: Lena.	58
4.16	Effect of the transformations: A toy example. In lexicographic order: Original, Noisy, Denoised with no transformations, Denoised with transformations.	60
4.17	PSNR plot for the image Elaine.	61
4.18	A close-up on the image Elaine of Figure 4.9. From left to right: Original, BM3D, PCkNN and PDC-RS. Top: image alone; bottom: isolevel lines superimposed on the image.	61
5.1	Physical properties of Silicon. The images above come from [Nak05].	66

5.2	©Foveon. Two different image sensor. Left Conventional CFA sensor. Right The layered Foveon X3 [®] sensor.	67
5.3	Optical system.	67
5.4	Properties of noise components. The pictures above are taken from [HK94].	69
5.5	Linear noise model.	70
5.6	PSD of the green channel (left) and a red/blue channel (right) from the demosaiced white noise patch using bilinear demosaicing [AGPP09].	71
5.7	Intra-channel denoising scheme. Each color channel of the RAW image is denoised separately. The two Green of the Bayer grid are tread separately as well.	72
5.8	Professional benchmark image (Courtesy of DxO Labs) used for test.	73
5.9	Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).	74
5.10	Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).	74
5.11	Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).	75
5.12	Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).	75
5.13	Comparison between RAW demosaiced images with no denoising (left) and with denoising (right). Particular of the image in Fig. 5.8. (Courtesy of DxO Labs).	76
5.14	Inter-channel denoising. The block matching step is modified in order to catch only coherent patches within the search window.	77
5.15	Comparison between intra-channel (left) and coherent (right) block matching.	77
5.16	Comparison between intra-channel block-matching (left) and coherent block-matching (right). Particular of the eyes. (Courtesy of DxO Labs).	78
6.1	Original images used in the experiments.	92
6.2	Zoom of filtered images with the various techniques for Lena corrupted by one-look speckle.	93
6.3	Zoom of filtered images with the various techniques for Napoli corrupted by one-look speckle.	94
6.4	Filtered images with the various techniques for target.	96
6.5	False alarms (FA) images with the various techniques for target.	97
6.6	Test SAR-X images (©Infoterra GmbH) with selected areas for ENL computation (white rectangle).	98
6.7	Filtered images with the various techniques for Rosen3.	99
6.8	A zoom of enhanced ratio between the noisy and denoised images for the various techniques.	100

7.1	The Last Supper (Italian: Il Cenacolo or L'Ultima Cena), Leonardo Da Vinci, 15th century.	103
7.2	An early example of inpainting. From “The commissar vanishes”, D. King, 1997.	104
7.3	Application of inpainting to restoration of an old photography. . .	105
7.4	An example of text removing. From Bertalmio [BSCB00].	105
7.5	Application to cinema post production.	106
7.6	Pseudocode for the proposed inpainting algorithm.	110
7.7	Photo of Abraham Lincoln taken by Alexander Gardner on February 5. 1865. Original image (left) and inpainted image (right). . .	111
7.8	Results on inpainting 2	112
7.9	Application to object removal.	113
B.1	$\chi(w)$ for a Gaussian and Epanechnikov Kernel, both with $\sigma = \text{std}(R)$	128
D.1	Image dataset used for algorithm comparison	132
D.2	Averaged PSNR and SSIM measures for the different algorithms. .	133

List of Tables

2.1	Estimation of Differential Entropy from 3D data embedded on 2D manifolds. Data are uniformly distributed on manifolds. Therefore entropy is $\log_2 S$, where S is the manifold surface. $N_{sample} = 5000$, $N_c = 10$ and $k = 1$.	15
3.1	Image quality measures with different algorithm and different noise statistics	30
3.2	Comparison of different algorithms with different noise mixtures.	34
6.1	PSNR results for Lena and Boat.	91
6.2	PSNR results for Napoli.	92
6.3	PSNR and detection results in terms of number of identified features for Target.	95
6.4	ENL for real SAR images.	99
D.1	Parameters setting for the PCkNN algorithm	131
D.2	PSNR (dB) results for the test images.	133
D.3	Tab. D.2 (continued). PSNR (dB) results for the test images.	134
D.4	SSIM ($\in [0, 1]$) results for the test images.	135
D.5	Tab. D.4 (continued). SSIM ($\in [0, 1]$) results for the test images.	136

Chapter 1

Introduction

The general context of this thesis is the Image Restoration. Part of the work presented here has been developed at the Napoli University. Indeed, my Ph.D. program has been developed between the University of Napoli “Federico II” (Italy) and the University of Nice-Sophia Antipolis (France), in the framework of a *co-tutelle* doctoral project. The topic I spent most of the time on, has been AWGN image denoising, at the University of Nice, while my time at University of Napoli was focused on application to SAR image denoising.

1.1 Context

Algorithms of image processing and computer vision can be classified in three main categories: low-level, mid-level and high-level algorithms. Low-level algorithms process basic operations on image pixels, *e.g.*, some pixels are moving in the image plane. Mid-level vision includes higher level processing like pixel grouping. High-level vision is the final stage which gives a semantic meaning to the scene. This document deals with low-level image processing tasks that may represent building bricks for content analysis or understanding. Mainly two problems will be studied: deconvolution and denoising. Image inpainting, will be briefly mentioned.

A central notion in this kind of problems is the notion of *similarity*. Many image and video processing problems can be solved by optimizing some cost functions. The notion of similarity is often behind these functions. It can be a *self-similarity* when a coherence is searched for within an object, or a *cross-similarity* between two objects, images, or videos. Image restoration and segmentation typically call upon self-similarity and content-based indexing and retrieval depend on the definition of a cross-similarity. Intermediately, tasks performed on video such as restoration, segmentation, tracking, and optical flow computation rely upon a similarity of an objet or a scene view with itself as observed on another frame.

1.2 Contributions of the thesis

The main contribution of this thesis is a new statistical framework inspired from Information Theory to address the problem of image restoration. Many problems of image and video processing can be expressed as the minimization of a data consistency residual and a term of mismatch with respect to a priori constraints. Traditionally, these functionals are based on penalization functions such as the ones defined for robust estimation, sometimes referred to as ϕ -functions. From a statistical point of view, recurring to these functions is equivalent to implicitly making assumptions on the probability density functions (PDFs) of the residual and the model mismatch, *e.g.*, Gaussian, Laplacian, or other parametric laws for the square function, the absolute value, or other ϕ -functions, respectively. Alternatively, it is interesting to adapt to (an estimation of) the true PDF. This nonparametric approach implies to define functionals which take PDFs as input. Entropy has been proposed in this context since, as a measure of dispersion of a PDF, its minimization leads the residual or model mismatch values to concentrate around narrow modes, the highest one normally corresponding to the annihilation of the residual or mismatch, the others corresponding to inevitable outliers.

Based on this considerations, a novel method for image deconvolution based on the minimization of the residual entropy has been proposed in the first part of this thesis. The use of entropy turns out to be robust with respect to different noise distributions applied to the observations. Indeed, the only hypothesis on the noise is the spatial independency, *i.e.*, noise samples in different spatial location are statistical independent.

In the same spirit of this Information Theory-driven approach the denoising problem has been tackled in the second part of the thesis. Here, we rely on the notion of *patch*, *i.e.*, a portion (usually a small square) of the image. Then the patch conditional entropy, which carries the residual information of the central pixel knowing its neighborhood, is considered. The denoising is performed in order to reduce this conditional entropy which has been leveled up by the presence of the noise. Developing this model leads to an information based interpretation of state of the art patch based denoising algorithms. Indeed, the derivation of the entropy criterion leads to a (weighted) average (or equivalently a filtering) of *similar* patches. This step is the core of state-of-the-art algorithms such as NLmeans and BM3D. Although the minimization of the conditional entropy has been introduced in UINTA, we proposed a different minimization scheme in the space of the patches coupled with the aggregation (or reprojection) step from the BM3D strategy.

The third part of this thesis deals with specific application of the denoising technique. In particular, in Chapter 5 we apply the denoising procedure developed in Chapter 4 to professional real world camera image. In this kind of images the noise can be considered still Gaussian, but *signal dependent*. Indeed the variance of the noise can be modeled as an affine function of the signal intensity. Furthermore, modern camera sensors produce RAW images which are *mosaiced*. This means

that in each spatial position (or pixel), only one color information is available. To have a complete RGB image a further step, called *demosaicing*, is required. This step combines the outputs of neighbors pixel in order to reconstruct the R,G, and B components in each spatial position. Denoising can be performed before or after the demosaicing step. However, demosaicing introduces correlation among neighborhood pixels and hence correlates the noise. The result is a “structured noise” which is not Gaussian and not independent anymore. Removing this noise is a harder task since algorithms usually rely on a hypothesis of independence. Therefore, denoising is performed before demosaicing. The algorithm has been then adapted in order to deal with the variable variance of the noise and the complex geometry of CFA matrix.

Chapter 6 presents a denoising scheme for Synthetic Aperture Radar (SAR) images which adapts the BM3D algorithm to SAR images peculiarities such as the multiplicative noise. This part of the work has been developed at the University of Naples. Although not stated in the variational framework, this scheme belongs to the general patch-based denoising context. The image self-similarity is exploited in order to filter the noise while preserving textures and edges.

Finally, Chapter 7 deals with the problem of digital image inpainting, which represents a growing area of image processing and computer vision research. Briefly, we adapted the denoising AWGN method in order to fill the damaged/missing regions in the image.

Chapter 2

ESTIMATION OF SOME STATISTICAL MEASURES

2.1 Introduction

The solution to image and video processing problems can often be formulated as follows

$$\hat{x} = \arg \min_x \phi_d(y - Hx) + \lambda \phi_r(\nabla x) \quad (2.1)$$

where ϕ_d (usually the L^p -norm operator) and ϕ_r correspond to data fidelity and regularization, respectively, and λ is the regularization parameter. Under some hypothesis one can note that (2.1) corresponds to the Maximum A Posteriori solution if the noise follows a generalized Gaussian law of shape parameter p and the a priori on the solution is given by a Gibbs PDF. These laws being defined by a small number of parameters, (2.1) can only adapt to the data to a limited extent. Moreover, such parametric assumptions may not be flexible enough to efficiently deal with outliers.

Even if one can pick functions proposed in robust estimation in order to reduce the bias introduced by outliers, these functions are still sensitive to the values of the outliers nonetheless. Moreover, they still represent an implicit assumption on the underlying distribution of the data.

On the contrary entropy and other related statistical measures are less sensitive to outliers because they deal with them in terms of frequency of occurrence as opposed to value. In addition, if the PDF(s) is/are estimated nonparametrically, then the measure makes no assumptions on the data or, otherwise stated, adapts to them.

In this chapter we resume the basics of nonparametric estimation of some statistical measures focusing on the k -Nearest Neighbors (k NN) framework which presents a lot of advantages in dealing with high dimensional spaces.

2.2 Kernel density estimation (KDE)

Kernel-based methods make no assumption about the actual PDF. Consequently, the estimated PDF cannot be described in terms of a small set of parameters, as opposed to, *e.g.*, a Gaussian PDF defined by its mean and variance. Such methods are then qualified as non-parametric. KDE is an important class of estimators since virtually all nonparametric algorithms are asymptotically kernel methods [TS92].

Let $\{s_1, s_2, \dots, s_n\}$ be a set of n independent observations of a random variable X with $p(\cdot)$ as PDF. The basic KDE may be written compactly as

$$\hat{p}(s) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{s - s_i}{h}\right) = \frac{1}{n} \sum_{i=1}^n K_h(s - s_i), \quad (2.2)$$

where $K_h(t) = K(t/h)/h$. Thus a kernel estimator is an equal mixture of n kernels, centered at the n data points.

Choosing a good value for the bandwidth, h , is the most difficult task. The choice of this parameter will be further discussed in Section 2.2.2

2.2.1 Multivariate KDE

The extension of the kernel estimator to the multivariate case where the samples are vector-valued data, $s \in R^d$, is straightforward. The KDE \hat{p} uses the multivariate d -dimensional kernels $K_H(\cdot)$, where the bandwidth H is a $d \times d$ covariance matrix. Therefore Eq.(2.2) writes

$$\hat{p}(s) = \frac{1}{n} \sum_{i=1}^n K_H(s - s_i), \quad (2.3)$$

where, in the most general case, $K_H(s) = |H|^{-1/2} K(H^{-1/2}s)$, with H being a $d \times d$ symmetric and positive definite *bandwidth* matrix, whose meaning will be clarified later, and $K(\cdot)$ being a d -variate kernel function, bounded and with compact support, satisfying the following set of conditions [WJ95]:

$$\begin{aligned} \int_{R^d} K(s) ds &= 1, & \lim_{\|s\| \rightarrow \infty} \|s\|^d K(s) &= 0, \\ \int_{R^d} s K(s) ds &= 0, & \int_{R^d} s s^T K(s) ds &= c_K I, \end{aligned} \quad (2.4)$$

where c_K is a constant and I is the identity matrix. It is convenient to separate the *size* of H from the *orientation* of H . To that end, write $H = h^2 A$, where $\det A = 1$. Thus, the size of H is $\det h^2 A = h^{2d}$. Commonly data are rotated by the transformation $A^{-1/2}$, then a normal kernel or, more generally, a product kernel, possibly with different smoothing parameter, h_k , in the k -direction:

$$\hat{p}(s) = \frac{1}{n} \sum_{i=1}^n \left[\prod_{k=1}^d K_{h_k}(s^{(k)} - s_i^{(k)}) \right]. \quad (2.5)$$

2.2.2 Bandwidth selection

As mentioned before, the critical parameter of KDE is the bandwidth h . In practice as long as in theory, this parameter should tend to zero when the number of samples tends to infinity. Larger bandwidth will capture overall structure while smaller bandwidth will get finer structure.

The problem of an automatic, data-driven choice of the bandwidth has actually more importance for the multivariate than for the univariate case. In one or two dimensions one can choose an appropriate bandwidth interactively just by looking at the plot of density estimates for different bandwidths. However, this task becomes very hard, if not impossible, in dimensions higher than 3. Two of the most frequently used methods of bandwidth selection are the plug-in method and the method of cross-validation, which can make the selection of this parameter automatic. Among the plug-in rules we recall here:

- a Parzen bandwidth selection is the Silverman's rule-of-thumb [Sil86]

$$\hat{h} \approx 1.06\hat{\sigma}n^{-1/5}, \quad (2.6)$$

where $\hat{\sigma}$ is an estimation of the unknown density standard deviation.

$$\hat{\sigma} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (s_i - \bar{s})^2}. \quad (2.7)$$

- In the multivariate case, it is not possible to derive the rule-of-thumb for general H and Σ . However, if we consider H and Σ to be diagonal matrices,

$$\hat{h}_j = \left(\frac{4}{d+2} \right)^{1/(d+4)} n^{-1/(d+4)} \sigma_j. \quad (2.8)$$

Note that this formula coincides with Silverman's rule of thumb in the case $d = 1$, see (2.6) and [Sil86]. Replacing the σ_j s with estimates and noting that the first factor is always between 0.924 and 1.059, we arrive at Scott's rule [Sco92]:

$$\hat{h}_j = \hat{\sigma}_j n^{-1/d+4}. \quad (2.9)$$

Equation (2.8) shows that it might be a good idea to choose the bandwidth matrix H proportional to $\Sigma^{-1/2}$, where Σ is the covariance matrix of the data. In this case we get as a generalization of Scott's rule

$$\hat{H} = n^{-1/(d+4)} \hat{\Sigma}^{-1/2}. \quad (2.10)$$

Using such a bandwidth corresponds to a transformation of the data, so that they have an identity covariance matrix. As a consequence we can use bandwidth matrices to adjust for correlation between the components of X .

Another method for automatic bandwidth selection is the double kernel estimates, which estimates the density with two different kernels (*e. g.*, Gaussian and Epanechnikov) and tries to find the bandwidth which minimizes the distance between these two estimation.

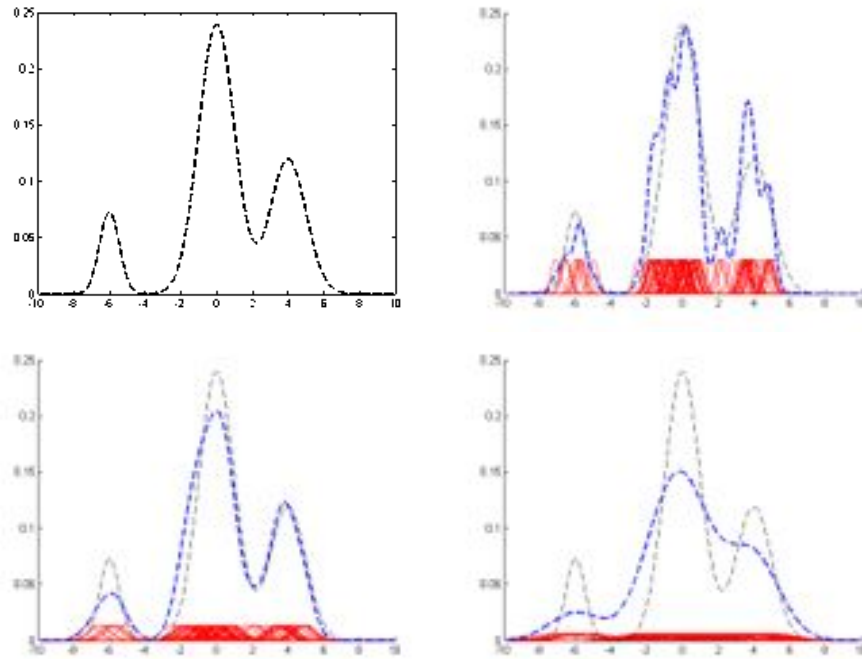


Figure 2.1: Kernel density estimation performance on a 1-D Gaussian mixtures for different bandwidth. Actual distribution is in black, kernel density estimate is in blue and kernels are in red. h is the plugin estimate using rule of thumb, from left to right, top to bottom: actual PDF, PDF estimated with $0.2h$, PDF estimated with h , PDF estimated with $5h$.

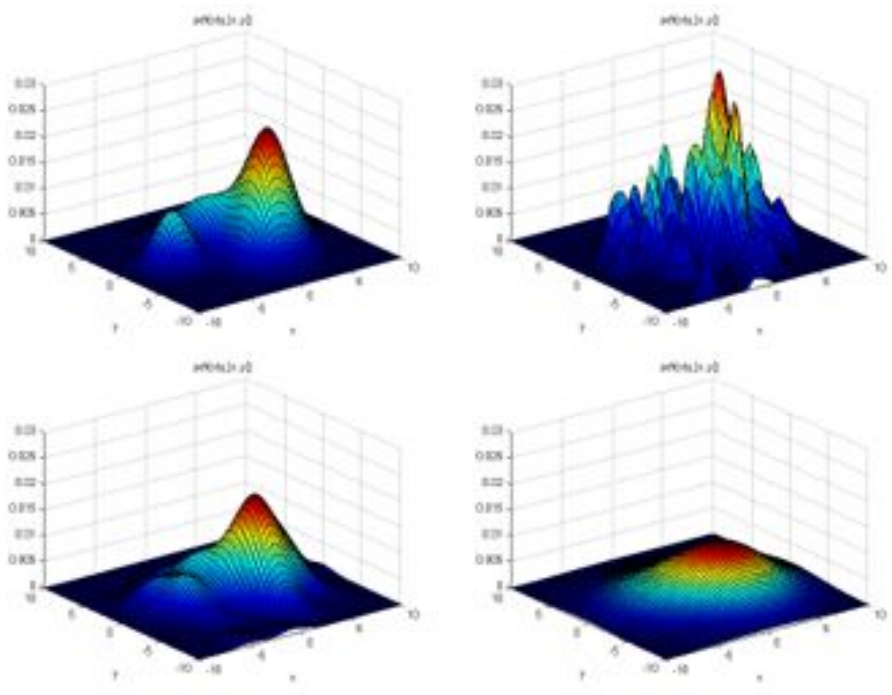


Figure 2.2: Multivariate kernel density estimation performance on a 2-D Gaussian mixtures for different bandwidth. h is the plugin estimate using rule of thumb, from left to right, top to bottom: actual PDF, PDF estimated with $0.2h$, PDF estimated with h , PDF estimated with $5h$.

2.3 Nonparametric entropy estimation

Entropy is a functional of the PDF and represents a measure of dispersion of a random variable. The differential entropy $H(p_X)$, or equivalently $H(X)$, of a continuous random variable X of R^d with PDF p , writes

$$H(X) = - \int_{R^d} p(\alpha) \log p(\alpha) d\alpha. \quad (2.11)$$

2.3.1 Plug-in estimates

The plug-in estimates of entropy are based on a consistent density estimate p_n of p such that p_n depends on X_1, \dots, X_n .

Integral estimate

The integral estimator is of the form

$$H_n = - \int_{A_n} p_n(t) \log p_n(t) dt, \quad (2.12)$$

where, with the set A_n one typically excludes the small or tail values of p_n . This estimator was first proposed by Dimitriev and Tarasenko [DT74], who proved the strong consistency of H_n for $d = 1$. The evaluation of the integral in (2.12) however requires numerical integration and is not easy if p_n is a KDE. Joe [Joe89] considers estimating $H(p)$ by (2.12) when p is a multivariate PDF, but he points out that the calculation of (2.12) when p_n is a KDE gets more difficult for $d \geq 2$.

The resubstitution estimate

Ahmad and Lin [AL76] proposed estimating $H(p)$ by

$$H_n = -\frac{1}{n} \sum_{i=1}^n \log p_n(X_i), \quad (2.13)$$

where p_n is a kernel density estimate. They showed the mean square consistency of (2.13) for $d = 1$. Joe [Joe89] considered the estimation of $H(f)$ for multivariate PDF's by an entropy estimate of the resubstitution type 2.13, also based on a KDE. His analysis and simulations suggest that the sample size needed for good estimates increases rapidly with the dimension d of the multivariate density.

2.3.2 The k NN framework

A consistent and unbiased entropy estimator was proposed for $k = 1$ [KL87]. This work was extended to $k \geq 1$ with a proof of consistency under weak conditions on

the underlying PDF [GLMI05]

$$H_{kNN}(U) = \log(v_d(|U| - 1)) - \psi(k) + \frac{d}{|U|} \sum_{s \in U} \log \rho_k(U, s), \quad (2.14)$$

where v_d is the volume of the unit ball in R^d , $|U|$ is the cardinality of the sample set U , ψ is the digamma function Γ'/Γ , and $\rho_k(U, s)$ is the distance to the k -th nearest neighbor of s in U excluding the sample located at s if any. Informally the main term in estimate (2.14) is equal to the mean of the log-distances to the k -th nearest neighbor of each sample. Note that (2.14) does not depend on the PDF p_U .

While the kNN PDF estimator is competitive in high dimensions only, the entropy estimator is accurate even in the univariate case [GLMI05] as shown in Figure 2.3.2 for several noise distribution. This might be explained by the smoothing effect of the log-distance averaging in (2.14). Moreover, the kNN entropy estimator also seems reasonably stable with respect to k until fairly high dimensions. Therefore, the choice of k does not appear to be really crucial, as opposed to the choice of h in the Parzen method. Actually, k must tend toward infinity when $|U|$ tends toward infinity, and such that $k/|U|$ tends toward zero when $|U|$ tends toward infinity. An admissible choice is $k = \sqrt{|U|}$.

2.4 Entropy derivative and Mean Shift

As mentioned in Section 2.1, a lot of image processing problems can be formulated as variational: a cost functional, often called energy, is minimized with respect to the unknowns of the problem in order to find a best solution. The minimization procedure often involves the calculus of the energy derivative. When the energy is entropy-based, the gradient of a PDF over the PDF (sometimes called normalized density gradient)

$$\frac{\nabla p}{p} = \nabla \log p \quad (2.15)$$

will be needed. PDFs have either a finite support or they tend toward zero at infinities, hence the question of stability or even existence of (2.15). Fortunately, this term can be approximated by the Mean Shift vector.

Mean Shift was first introduced in 1975 by Fukunaga and Hoestler [FH75] as a technique for the estimation of probability density gradients, but recently [CM02, Che95, KF99, LK06] the advantages of such approach both in density estimation and clustering have been newly recognized.

As for the non-parametric density estimation techniques, the main idea on which this approach is based lies on the fact that samples in an arbitrary feature space can be seen as an empirical probability density function, that is, local maxima of the probability should be observed in areas that have a dense concentration of data points.

The basic approach in the Parzen Window technique lies on the observation that, given a d -dimensional feature space and a set of n data points (s_1, \dots, s_n) ,

the probability density function $p(s)$ can be estimated as

$$\hat{p}_{H,K}(s) = \frac{1}{n} \sum_{i=1}^n K_H(s - s_i), \quad (2.16)$$

In [CM02], the authors pointed out that a family of kernel functions satisfying the conditions 2.4 and showing the sufficient property of radial symmetry can be obtained in the following way:

$$K(s) = c_{k,d} k(\|s\|^2), \quad (2.17)$$

with $c_{k,d}$ normalizing constant, that is to say defining a univariate *kernel profile* $k(x)$ for $x \geq 0$ and rotating it in the space R^d . It is further observed in [WJ95] that, in order to limit complexity in the density estimation procedure, a common practical choice is to set the bandwidth matrix H as proportional to the identity matrix, that is $H = h^2 I$, so that only one parameter should be provided in advance. Under this assumption, the formula of the estimator given in (2.16) becomes

$$\hat{p}_{h,K}(s) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{s - s_i}{h}\right\|^2\right) \quad (2.18)$$

Applying the gradient operator to both sides of (2.18) yields to the form of the *density gradient estimator*. Using $g(x) = -k'(x)$, we obtain

$$\begin{aligned} \hat{\nabla} p_{h,K}(s) &= \frac{2c_{k,d}}{nh^{d+2}} \sum_{i=1}^n (s_i - s) g\left(\left\|\frac{s - s_i}{h}\right\|^2\right) \\ &= \frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^n g\left(\left\|\frac{s - s_i}{h}\right\|^2\right) \right] \left[\frac{\sum_{i=1}^n s_i g\left(\left\|\frac{s - s_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{s - s_i}{h}\right\|^2\right)} - s \right]. \end{aligned} \quad (2.19)$$

Observe that the density estimate $\hat{p}(s)$ evaluated using the function $G(s) = c_{g,d} g(\|s\|^2)$ as a kernel (also called the *shadow* of kernel $K(s)$) is given by

$$\hat{p}_{h,G}(s) = \frac{c_{g,d}}{nh^d} \sum_{i=1}^n g\left(\left\|\frac{s - s_i}{h}\right\|^2\right), \quad (2.20)$$

Therefore it is possible to rewrite Eq. (2.19) as

$$\hat{\nabla} p_{h,K}(s) = \frac{2c_{k,d}}{h^2 c_{g,d}} \hat{p}_{h,G}(s) \mathbf{m}_{h,G}(s), \quad (2.21)$$

with the term

$$\mathbf{m}_{h,G}(s) = \frac{\sum_{i=1}^n s_i g\left(\left\|\frac{s - s_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{s - s_i}{h}\right\|^2\right)} - s, \quad (2.22)$$

being called the *mean shift* (MS) vector. Combining (2.21) and (2.22) we obtain an estimation of the normalized PDF gradient

$$\frac{\nabla p(s)}{p(s)} \propto \frac{\sum_{i=1}^n s_i g\left(\left\|\frac{s-s_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{s-s_i}{h}\right\|^2\right)} - s. \quad (2.23)$$

2.5 Mean-Shift approximation

2.5.1 Based on fixed kernel size

As shown in eq. (2.22), the MS vector in the point s involves a weighted average of samples s_i in the neighborhood of s . The size of the neighborhood depends on the bandwidth parameter h while the weights are given by the shadow $g(\cdot)$. In the Parzen Window approach (i.e., fixed bandwidth) it can be expressed, using an Epanechnikov [Epa69] kernel k (and hence its shadow $g(\cdot) = \text{rect}(\cdot)$), as

$$\frac{\nabla p(s)}{p(s)} = \frac{d+2}{h^2} \frac{1}{k(s, h)} \sum_{s_j \in S_h(s)} (s_j - s), \quad (2.24)$$

where d is the dimension of the feature space S , $S_h(s)$ is the support of the Parzen kernel centered at point s and of constant size h , $k(s, h)$ being the number of observation falling into $S_h(s)$. The choice of the kernel window size h is critical [Sco92]. If h is too large, the estimate will suffer from too little resolution, otherwise if h is too small, the estimate will suffer from too much statistical variability. As the dimension of the data space increases, the space sampling gets sparser (problem known as the curse of dimensionality). Therefore, less samples fall into the Parzen window centered on each sample, making the PDF estimation less reliable. Dilating the Parzen window does not solve this problem since it leads to over-smoothing the PDF. In a way, the limitations of the Parzen Method come from the fixed window size: the method cannot adapt to the local sample density. The k -th nearest neighbor (kNN) framework provides an advantageous alternative.

2.5.2 Based on variable kernel size

k -th Nearest Neighbors

Taking the kernel $K(\cdot)$ to be a uniform density on the unit sphere with $H(s) = \rho_k(s)I_d$ where $\rho_k(s)$ is the distance from s to the k -th nearest data point, one has the k -nearest neighbor estimator [LQ65] (kNN). In the Parzen-window approach, the PDF at sample s is related to the number of samples falling into a window of fixed size centered on the sample. The kNN method is the dual approach: the density is related to the size of the window necessary to include the k nearest neighbors of the sample. Thus, this estimator tries to incorporate larger bandwidths in the tails of the distributions, where data are scarce.

In kNN framework, the MS vector is given by [FH75]

$$\frac{\nabla p(s)}{p(s)} = \frac{d+2}{\rho_k^2} \left[\left(\frac{1}{k} \sum_{s_j \in S_{\rho_k}} s_j \right) - s \right], \quad (2.25)$$

where ρ_k is the distance of s to the k -th nearest neighbor.

The kNN estimator is not guaranteed to be one (hence, the kernel is not a density) and the discontinuous nature of the bandwidth function manifests directly into discontinuities in the resulting estimate. Furthermore, the estimator has severe bias problems, particularly in the tails [Hal83, MR79] although it seems to perform well in higher dimension [TS92].

Adaptive Weighted kNN Approach

The kNN method provides several advantages with respect to the Parzen Window method. For example, the number of samples falling in the window is fixed and known. Thus even if the sampling space gets sparser, we cannot have empty regions, i.e., no samples inside. Moreover, the window size is locally adaptive. However, as near the distribution modes there is an high density of samples, the window size associate to the k -th nearest neighbor could be too small. In this case the estimate will be sensible to statistical variations in the distribution.

To avoid this problem we would increase the number of nearest neighbors, to have an appropriate window size near the modes. However this choice would produce a window too large in the tails of the distribution. Thus very far samples would contribute to the estimation, producing severe bias problems.

We propose an alternative solution that keeps advantages from both Parzen and kNN approaches. The samples contribution is weighted by formally making the following substitution

$$\frac{1}{k} \sum_{s_j \in S_{\rho_k}} s_j \rightarrow \frac{1}{\sum_{j=1}^k w_j} \sum_{\substack{j=1 \\ s_j \in S_{\rho_k}}}^k w_j s_j. \quad (2.26)$$

Intuitively, the weights w_j must be a function of distance between the actual sample and the j th nearest neighbor, i.e., samples with smaller distance are weighted more heavily than ones with larger distance.

2.6 Conclusion

In this chapter we resumed the basics of non parametric estimation. In particular we pointed out that the normalized gradient of a PDF, which often needs to be calculated in entropy-based variational problems, can be locally approximated by means of a weighted average of samples falling in a local neighborhood. In addition we pointed out that the k NN framework easily adapts the neighborhood size

Manifold	Theoretical	$H_{manifold}$	H_{kNN}
Plane	2.0	2.0	-3.9
Circle	1.0	1.0	-5.0
Swiss roll	10.6	10.6	8.9
Ring torus	9.6	9.6	7.4
Sphere	5.7	5.2	1.0

Table 2.1: Estimation of Differential Entropy from 3D data embedded on 2D manifolds. Data are uniformly distributed on manifolds. Therefore entropy is $\log_2 S$, where S is the manifold surface. $N_{sample} = 5000$, $N_c = 10$ and $k = 1$.

to the local data density and provides more reliable results than a Parzen window approach for high dimensional feature spaces.

Finally, we want to point out that even if standard non parametric estimation tools provide very good estimates for *ordinary* distributions, they can fail if data are complexly structured. This is the case of high dimensional data embedded on lower dimensional manifolds. A novel technique for this kind of data has been proposed in [NK07]. An illustrative example is given in Figure 2.6, where 3D data are uniformly distributed on 2D manifold. The theoretical entropy of such a data is, in bits, $\log_2(S)$, where S is the surface of the manifold. Table 2.6 shows estimated entropies with the method proposed in [NK07] and the k NN estimator.

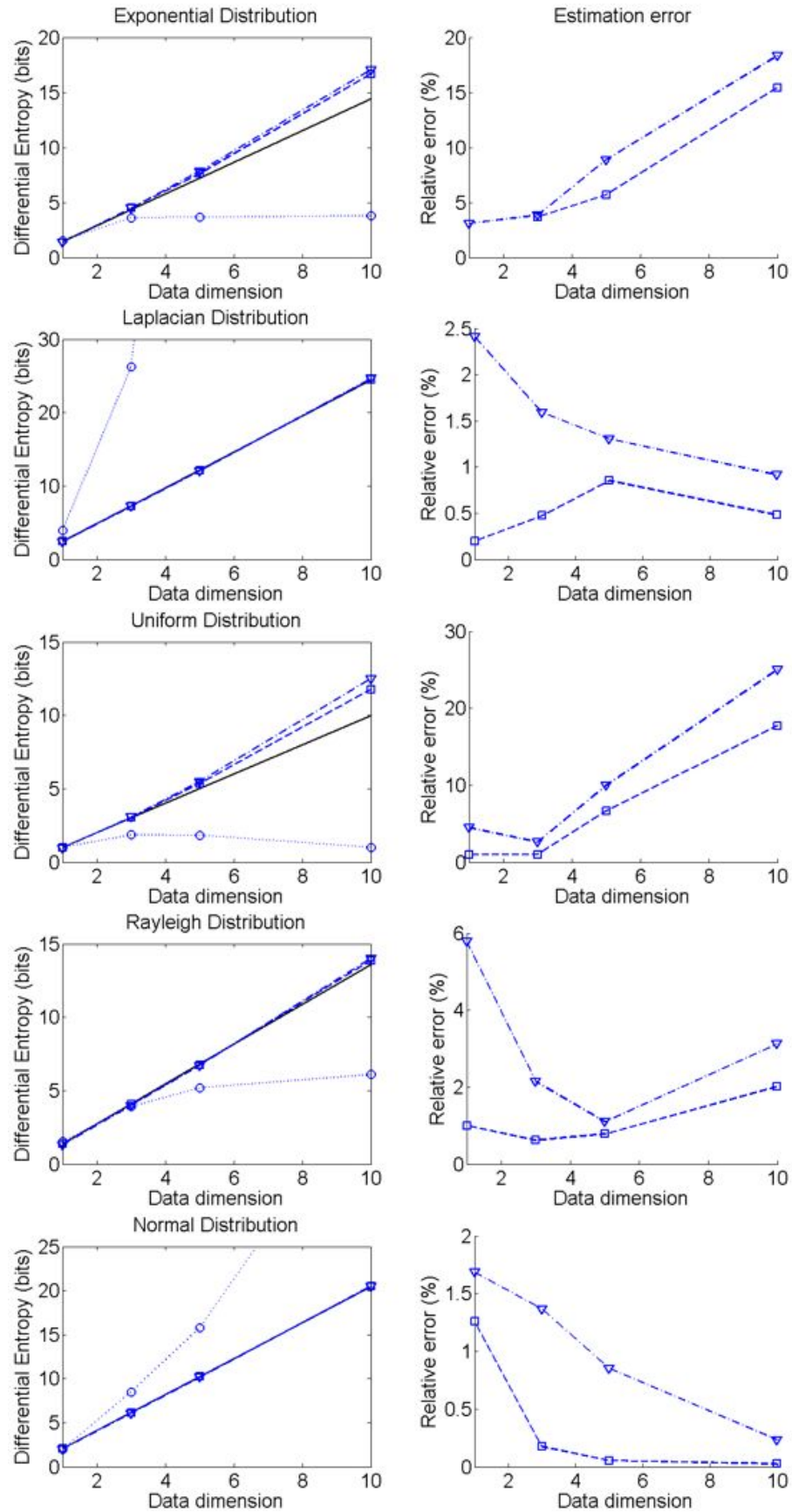


Figure 2.3: Entropy estimation of different PDF with relative errors. Circles represent Parzen, *i.e.* fixed bandwidth, estimation. Squares represent k NN estimator ($k = 10$) and triangles represent the estimator presented in [NK07].

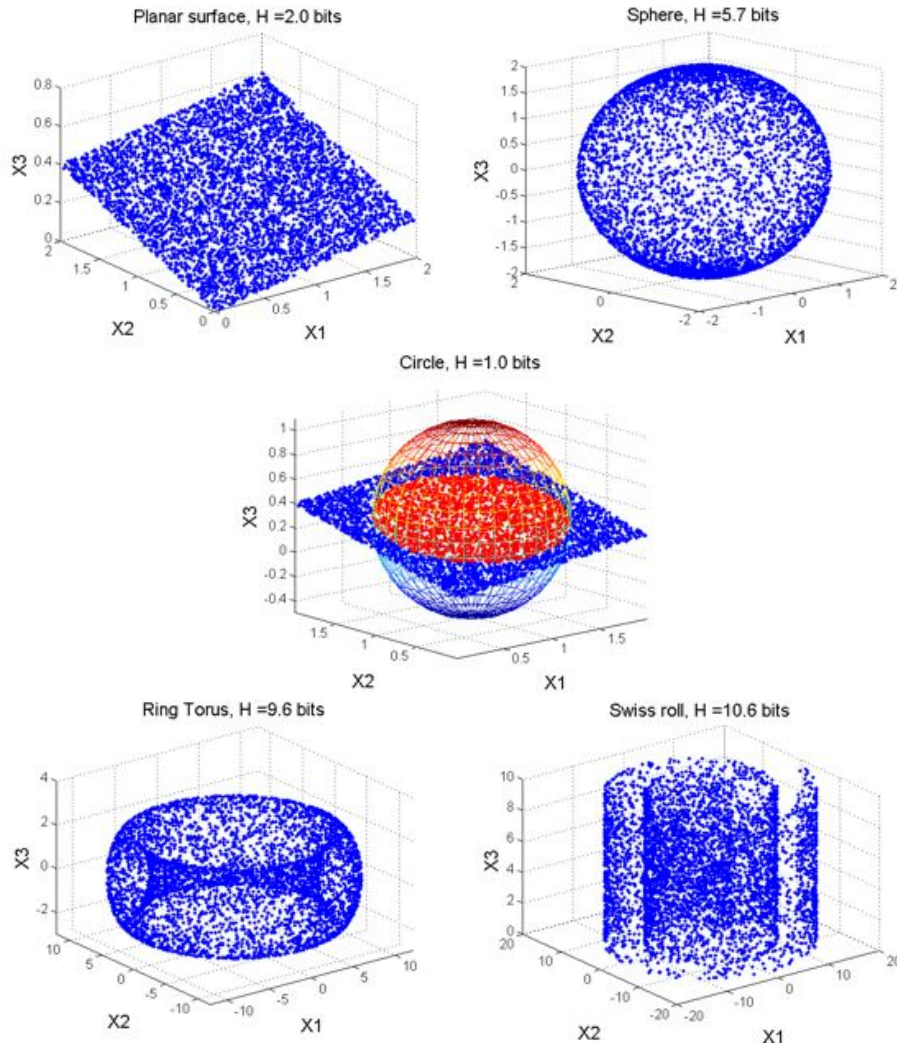


Figure 2.4: 3D data uniformly distributed on embedded 2D manifolds. In lexicographic order: Plane, Spherical surface, Circle (intersection between the Plane and a sphere), Ring torus and Swiss roll. Estimated entropies are given in Table 2.6.

Part I

Deconvolution

Chapter 3

ENTROPY-BASED DECONVOLUTION

3.1 Introduction

Image restoration attempts to reconstruct or recover an image that has been degraded by using a priori knowledge of the degradation phenomenon. The problem consists in the reconstruction of an original image x from an observed image y . The simplest model connecting x to y is the linear degradation model

$$y = Hx + n, \quad (3.1)$$

where H is a linear operator and n is the observation noise. When the operator H is space-invariant the model becomes

$$y = m * x + n, \quad (3.2)$$

where degradations are modeled as being the result of convolution together with an additive noise term, so the expression image deconvolution (or deblurring) is used frequently to signify linear image restoration [GW02]. Here m represents a known space-invariant blur kernel (point spread function, PSF), x is an ideal version of the observed image y and n is (usually Gaussian) noise.

The objective of restoration is to obtain an estimate \hat{x} as close as possible to the original image, by means of a certain criterion. We focus on variational methods, that have an important role in modern image research. Classically, the solution minimizes a certain functional, often called energy, which typically is the norm of the residual with some regularization term,

$$\hat{x} = \arg \min_x \underbrace{\|y - Hx\|_{\Sigma}^2}_{\text{Data-fidelity}} + \underbrace{J_{\chi}(x)}_{\text{Regularization}}, \quad (3.3)$$

where Σ and χ represent two suitable functional spaces.

The data-fidelity term measures a sort of *distance* between the observed and the reconstructed image, while the regularization term describes some properties (or constraints) of the image we are looking for. In stochastic based approaches, x , y and n are considered as realizations of random fields.

In this chapter, we develop a new deconvolution algorithm which minimizes the residual differential entropy.

3.2 State of the art

3.2.1 Deterministic approach

We consider a space-invariant imaging system, such that the model of image formation is given by¹

$$y(u) = \int m(u-v)x_0(v)dv + n(u). \quad (3.4)$$

Inverse filter

Suppose for simplicity that the signal-to-noise ratio (SNR) is sufficiently large such that we can reasonably neglect at first sight the noise term in eq.(3.4). If we use the Fourier transform (FT), eq.(3.4) becomes rather trivial since we get

$$Y(f) = M(f)X_0(f). \quad (3.5)$$

It is clear from equation(3.5) that the support of $M(f)$ plays an important role in the solution of the problem. Indeed, the uniqueness of the solution is closely related to the *null space* of the convolution operator, *i.e.* the solution of the equation

$$M(f)X(f) = 0. \quad (3.6)$$

If the support of $M(f)$ is the whole frequency space, then $X(f) = 0$. In this case the solution of the restoration problem is unique and is given by

$$X(f) = \frac{Y(f)}{M(f)}. \quad (3.7)$$

However if we are in the presence of noise we get

$$X(f) = X_0(f) + \frac{N(f)}{M(f)}. \quad (3.8)$$

The second term in equation(3.8) comes from the inversion of the noise contribution and it may be responsible for the non-existence of the solution. Indeed since the noise is a process independent of the image formation, there may be division

¹For notational compactness we use a single argument to indicate spatial location.

by zero and $X(f)$ has singularities at the zeros of $M(f)$. Moreover, even if $M(f)$ is not zero for some values of f , it tends to be zero when $|f| \rightarrow \infty$. Since the behavior of $N(f)$ is not related to $M(f)$, the ratio $N(f)/M(f)$ may not tend to zero. Hence, the solution may not exist or be not stable due to the noise amplification.

On the other hand, if the support of $M(f)$ is a bounded subset of the frequency space, the solution of the image deconvolution problem is not unique. In this case we take as solution the least-squares (LS) solution.

Wiener filter

The inverse filter divides in the frequency domain by numbers that are very small, which amplifies any observation noise in the image. A better approach is based on the Wiener filter [Wie49]. The goal is to find a filter which gives an estimate $\hat{x} = g * y$ such that it minimizes the mean square error. In the Fourier domain, the Wiener filter is given by

$$G(f) = \frac{M^*(f)S_x(f)}{|M(f)|^2S_x(f) + S_n(f)}, \quad (3.9)$$

where $S_x(\cdot)$ and $S_n(\cdot)$ are the mean power spectral density of the input signal $x(\cdot)$ and the noise $n(\cdot)$ respectively.

The operation of the Wiener filter becomes apparent when the filter equation above is rewritten:

$$G(f) = \frac{1}{M(f)} \left[\frac{|M(f)|^2}{|M(f)|^2 + SNR^{-1}(f)} \right], \quad (3.10)$$

where $SNR(f) = S_x(f)/S_n(f)$ is the signal-to-noise ratio. When there is zero noise (*i.e.* infinite signal-to-noise), the term inside the square brackets equals 1, which means that the Wiener filter is simply the inverse of the system, as we might expect. However, as the noise at certain frequencies increases, the signal-to-noise ratio drops, so the term inside the square brackets also drops. This means that the Wiener filter attenuates frequencies dependent on their signal-to-noise ratio. The Wiener filter equation above requires us to know the spectral content of a typical image, and also that of the noise. Often, we do not have access to these exact quantities, but we may be in a situation where good estimates can be made. For instance, in the case of photographic images, the signal (the original image) typically has strong low frequencies and weak high frequencies, and in many cases the noise content will be relatively flat with frequency.

3.2.2 Statistical methods

As opposed to deterministic approaches which do not take into account the random nature of noise, in stochastic approaches y , x , and n are considered as realization of random fields. If some statistical property of the noise, such as the expectation value or the correlation function or the probability distribution, is known,

then one can develop methods where this information is used for image deconvolution. In classical statistics, Maximum Likelihood (ML) is the most commonly used method for parameter estimation. Its application to image restoration is based on the knowledge of the random properties of noise so that the probability density $p_n(y|x)$ is known. Then the ML estimator looks for the image x which is most likely to produce the detected image y , *i. e.* the image which maximize the probability of observing y

$$\max_x p(y|x) \quad (3.11)$$

with $p(y|x) = p(y = Mx + n|x) = p(n = y - Mx)$. If the noise is white and gaussian, $p_N(t) = \frac{1}{2\pi\sigma} \exp -\frac{||t||^2}{2\sigma^2}$ and

$$\max_x p(y|x) = \max_x \frac{1}{2\pi\sigma} \exp -\frac{||y - Mx||^2}{2\sigma^2} \Leftrightarrow \min_x ||y - Mx||^2. \quad (3.12)$$

Thus, in the case of additive Gaussian noise, the ML-method is equivalent to the LS method.

3.2.3 Why using entropy?

As it is well known, LS estimation is sensitive to outliers, or deviations, from the assumed statistical model. In the literature other more robust estimators have been proposed, like M-estimators [BA93], involving non-quadratic and possibly non-convex energy functions. However, these methods rely on parametric assumptions on the noise statistics, which may be inappropriate in some applications due to the contribution of multiple error source, such as radiometric noise (Poisson), readout noise (Gaussian), quantization noise (Uniform) and "geometric" noise, the latter due to the non-exact knowledge of the PSF. Therefore density estimation using a nonparametric approach is a promising technique.

We propose to minimize a functional of the residual distribution, in particular the differential entropy of the residual. We use entropy because it provides a measure of the dispersion of the residual, in particular low entropy implies that the random variable is confined to a small effective volume and high entropy indicates that the random variable is widely dispersed [CT91]. Moreover, entropy criterion is robust to the presence of outliers in the samples. Experimental results with non gaussian distributions show the interest of such a nonparametric approach.

3.3 Proposed method: MRED

3.3.1 Minimizing the residual entropy

Image deblurring is an inverse problem, that can be formulated as a functional minimization problem. Let Ω denote a rectangular domain in \mathbb{R}^2 , on which the image function $x : u \in \Omega \rightarrow \mathbb{R}^d$ is defined, d being the image dimensionality.

Ideally, the recovered image \hat{x} satisfies

$$\hat{x} = \arg \min_x \int_{\Omega} \varphi(y - m * x) \, du, \quad (3.13)$$

where $\varphi(\cdot)$ is a metric representing data-fidelity. In the case of Gaussian noise, a quadratic function is used. However, parametric assumptions on the underlying noise density function are not always suitable, due to the multiple source of noise. We define as energy to be minimized a continuous version of the Ahmad-Lin [AL76] entropy estimator ($H_{A-L}(r)$), defined as:

$$\begin{aligned} E(x) &= |\Omega| H_{A-L}(r) \\ &= - \int_{\Omega} \log(p_x(r(u))) \, du. \end{aligned} \quad (3.14)$$

In order to solve the optimization problem a steepest descent method is used. The energy derivative has been analytically calculated and it is shown in section 3.3.3.

3.3.2 Energy lower bound

In this section we provide a lower bound (LB) to the energy in eq.(3.14), in order to check how our algorithm works on minimizing residual entropy (see Fig.3.1). The residual can be viewed as the sum of two random variables, namely, $R = N + \tilde{X}$. The first one is the noise, and the second one is the projection of the error by means of the operator $m(\cdot)$, i.e., $\tilde{x} = m * (x_0 - x)$.

Proposition 3.1. *The residual entropy $h(R)$ is lower bounded by the noise entropy $h(N)$.*

Proof. Let us consider the mutual information between R and \tilde{X} ,

$$\begin{aligned} I(R; \tilde{X}) &= h(R) - h(R|\tilde{X}) \\ &= h(R) - h(N|\tilde{X}). \end{aligned}$$

Since the noise N is independent from \tilde{X} , $h(N|\tilde{X}) = h(N)$, and by the non negativity property of mutual information we obtain

$$h(R) \geq h(N). \quad (3.15)$$

□

As it is well known, mutual information is a measure of the amount of information that one random variable contains about another random variable [CT91]. The closer x is to the original image x_0 , the less information on \tilde{X} is carried by the residual. Therefore entropy minimization can be interpreted as the process which uses the information carried by the residual to recover x_0 , until there is no more information, i.e., the residual entropy reaches the lower bound.

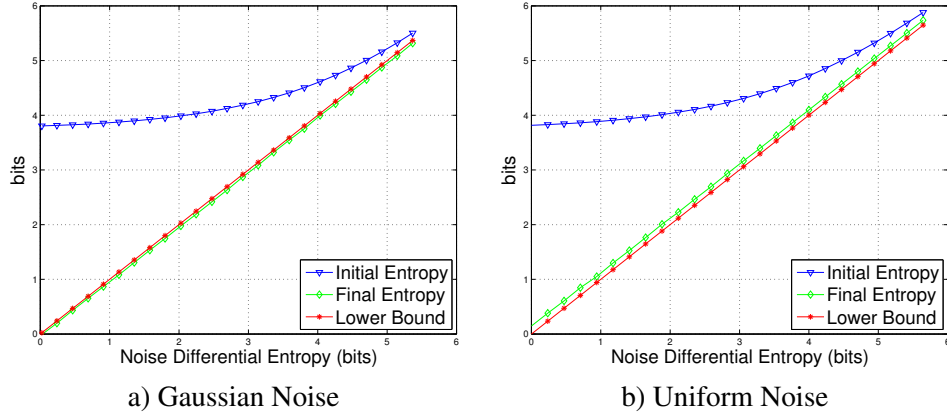


Figure 3.1: Residual Entropy as function of noise entropy. Initial residual entropy (blue), Final residual entropy (green), Theoretical lower bound (red).

3.3.3 Energy derivative

The residual pdf is estimate by using a nonparametric continuous version Parzen estimator, with symmetric kernel $K(\cdot)$,

$$p_x(s) = \frac{1}{|\Omega|} \int_{\Omega} K(s - r(u)) \, du. \quad (3.16)$$

Note that $p_x(s)$ is the residual pdf associated to the current estimate image x . Therefore changes in x provides changes in $p_x(s)$, hence changes in the residual entropy (energy). By taking the Gâteaux derivative of eq.(3.14) it can be shown (see the Appendix A for the demonstration.) that the gradient of $E(x)$ at $v \in \Omega$ is equal to

$$\nabla E(x)(v) = \int_{\Omega} m(v - w) k(w) \, dw, \quad (3.17)$$

with

$$k(w) = \frac{\nabla p_x(r(w))}{p_x(r(w))} + \chi(w) \quad (3.18)$$

and

$$\chi(w) = -\frac{1}{|\Omega|} \int_{\Omega} \frac{\nabla K(r(u) - r(w))}{p_x(r(u))} \, du. \quad (3.19)$$

The first term in (3.18) is the normalized gradient of the residual pdf and it is proportional to the local Mean-Shift (MS) [FH75]. In the following the MS estimation is addressed as well as the computation of Eq. 3.19.

Mean-Shift approximation

The first term in (3.18) is the normalized gradient of the residual pdf and it is proportional to the local mean-shift [FH75]:

$$\frac{\nabla p_x(X)}{p_x(X)} = \frac{d+2}{h^2} M_h(X), \quad (3.20)$$

where $M_h(X)$ is the local MS vector. MS estimation can be dealt with either a Parzen window or a k -th nearest neighbor (kNN) approach. Since the residual is scalar, the Parzen approach is more suitable. Therefore the MS is

$$M_h(X) = \frac{1}{k} \sum_{X_i \in S_h(X)} (X_i - X) \quad (3.21)$$

i.e., the sample mean shift of the observations in the small region $S_h(X)$ centered at X ($S_h(X) = \{Y : \|Y - X\|^2 \leq h^2\}$).

Second order term χ

The integral in (3.19) is difficult to calculate in the general case and its computation is developed in Appendix B. However, under some hypotheses, the residual pdf is approximatively

$$p_x(\alpha) \approx \frac{N(\alpha)}{|D|}, \quad (3.22)$$

where $N(\alpha)$ is the number of samples such that $r(w) = \alpha$. Note that, this approximation does not make any assumption on the underlying residual pdf. $\chi(w)$ is the sample mean of a function of the random variable R , i.e.,

$$\frac{1}{|D|} \int_D \frac{\nabla K_\sigma(r(u) - r(w))}{p_x(r(u))} du \approx \int_{supp R} \nabla K_\sigma(\alpha - r(w)) d\alpha. \quad (3.23)$$

This is function of $r(w)$, if $r(w) = 0$ and the support of R is symmetric, the value of $\chi(w)$ is 0 as long as it is an integral of an even function. By means of this considerations, we could expect a negligible value of $\chi(w)$ if $r(w)$ is small, and higher values near the boundary of the support of R .

3.4 Experimental Results

In this section, some results from MRED (Minimum residual Entropy Deconvolution) algorithm are shown and compared to some state-of-the-art techniques: the Wiener filter [Wie49], the Lucy-Richardson [Ric72, Luc74] algorithm, the regularized inverse filter, the regularized Tikonoff filter and the Truncated SVD [HNO06]

In order to measure the performance of our algorithm we blurred the Lena image (512x512 pixel) by convolving it with a 13x13 Gaussian PSF with standard deviation $\sqrt{3}$, and adding noise with different distributions, such as Gaussian,

Uniform, Gaussian mixture, Gaussian-Uniform mixture and with different entropy magnitudes. Residual entropy minimization is carried out via the gradient descent algorithm described in section 3.3.3. At each iteration the mean-shift kernel size h is proportional to the standard deviation of the residual, since this choice generally assures a good compromise between robustness and accuracy [Com03].

Figure 3.1 shows in blue the initial residual entropy in green the value attained when the algorithm converges and in red the theoretical LB. We considered Gaussian noise in Figure 3.1a and Uniform noise in Figure 3.1b. In the gaussian case the proposed algorithm achieves the lower bound of entropy. However, in the uniform case as well the final entropy is quite close to the LB with a maximum relative difference of 0.02%.

The performance is quantified by the Peak Signal-to-Noise Ratio,

$$\text{PSNR} = 10 \log_{10} \frac{|x|_{\max}^2}{\text{MSE}} \quad (3.24)$$

where $|x|_{\max}$ is the maximum value admitted by the data format and the mean-square error

$$\text{MSE} = \langle [x(n) - \hat{x}(n)]^2 \rangle \quad (3.25)$$

is computed as a spatial average $\langle \cdot \rangle$, with x and \hat{x} being the original and restored images, respectively. We also use a recent novel quality assessment measure: the Structure Similarity (SSIM) index [WBSS04], which proved to be more consistent with the human eye perception. The SSIM measure is calculated between small windows (usually 8×8) of an image, namely x and y , as follows,

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (3.26)$$

where μ . and σ . represent respectively the image average and the variance in the window and c_1, c_2 two constants to stabilize the division with weak denominator.

Figure 3.2 shows the PSNR and SSIM measures between the original image x_0 and the degraded image y (blue) and the restored image \hat{x} (red) in function of the noise entropy for Gaussian and Uniform distribution.

Figure (3.4) shows the restored images from different algorithms as Lucy-Richardson [Ric72, Luc74] and Truncated SVD [HNO06], with uniform noise (entropy 2 bits). The truncation parameter of TSVD is found with a generalized cross validation [HNO06]. MRED has roughly the same PSNR of the TSVD restored image, however the latter has a more pronounced grain effect. This is well caught by the SSIM measure, for which our method is considered of higher quality. SSIM indicates better results also in the experiment of Figure 3.7, where a gaussian mixture noise has been used (see Fig. 3.6), even if the PSNR is lower than the one provided by the TSVD restoration.

A full set of comparisons is shown in Tab. 3.1 and 3.2 for different noise distributions and entropy values. In particular, Tab. 3.2 reports results for the two *non canonical* noise distributions shown in Figure 3.6. For the sake of clarity,

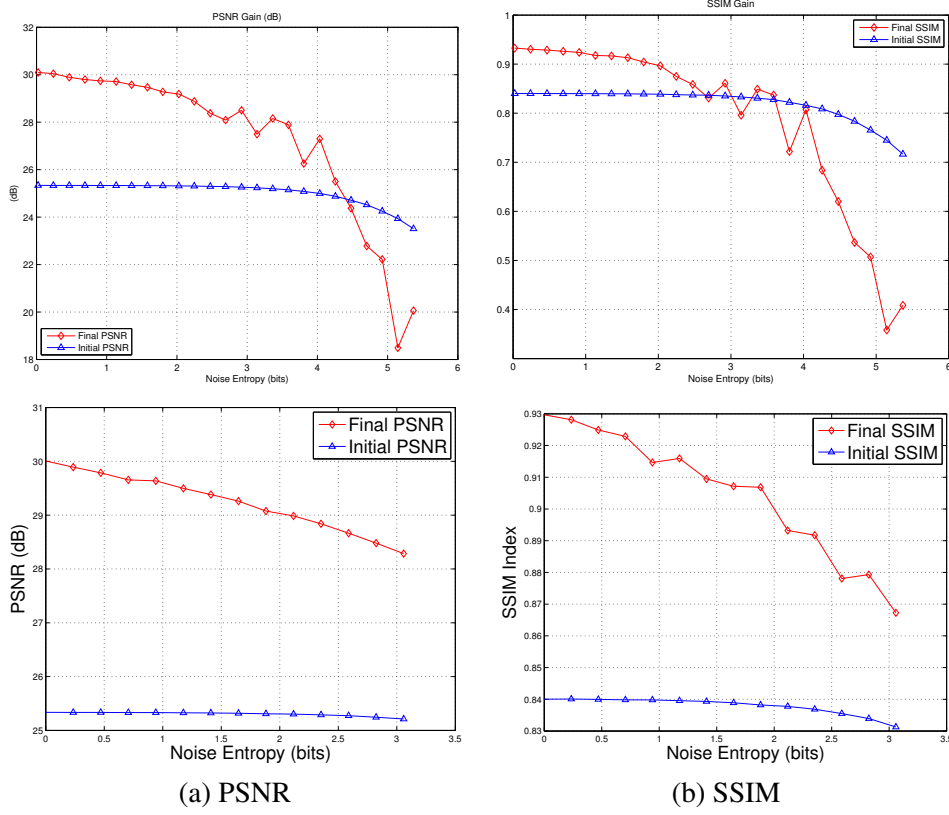


Figure 3.2: Algorithm performances for gaussian noise (top) and uniform noise (bottom) as function of noise entropy. (a) Initial (blue) and Final (red) PSNR; (b) Initial (blue) and Final (red) SSIM.

the best PSNR and SSIM value is put in boldface. The MRED algorithm always outperforms the other techniques except that in the two aforementioned cases for the PSNR value.

3.5 Discussion and perspectives

In this chapter we presented a deconvolution method in the variational framework based on the residual entropy minimization. We gave a theoretical meaning of the minimization procedure in terms of mutual information between random variables. The simulations indicated robust performance for different non-standard noise distribution probabilities. The robustness comes from the non parametric estimation which makes no assumptions on the noise PDF. Experiments show in many cases slightly better results w.r.t. some popular deblurring techniques. Results are even more promising considering that, contrarily to what happens in other techniques

Noise	Algorithm	PSNR (dB)	SSIM $\in [0, 1]$
Gaussian 1 bit	Wiener Filter	25.41	0.825
	Regularized Filter	25.42	0.828
	Lucy-Richardson	25.31	0.881
	Regularized Tikonov	25.79	0.837
	Truncated SVD	29.59	0.908
	MRED	29.65	0.922
Gaussian 2 bits	Wiener Filter	25.41	0.824
	Regularized Filter	25.41	0.825
	Lucy-Richardson	25.29	0.880
	Regularized Tikonov	25.60	0.804
	Truncated SVD	29.15	0.890
	MRED	29.16	0.895
Uniform 1 bit	Wiener Filter	25.41	0.824
	Regularized Filter	25.41	0.825
	Lucy-Richardson	25.31	0.881
	Regularized Tikonov	25.79	0.832
	Truncated SVD	29.53	0.906
	MRED	29.59	0.918
Uniform 2 bits	Wiener Filter	25.41	0.824
	Regularized Filter	25.41	0.825
	Lucy-Richardson	25.28	0.878
	Regularized Tikonov	25.39	0.786
	Truncated SVD	29.04	0.882
	MRED	29.03	0.900

Table 3.1: Image quality measures with different algorithm and different noise statistics

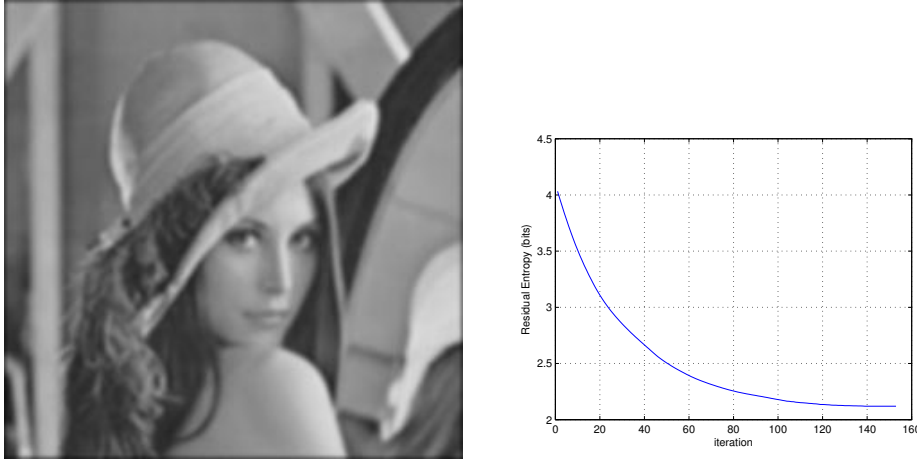


Figure 3.3: Left: Degraded image with a gaussian filter ($\sigma^2 = 3$), and zero mean Uniform noise of entropy 2 bits. Right: Residual entropy as function of iterations.

like Truncated SVD, MRED algorithm makes no use of regularization parameters. In most cases, iterative methods, converging to the ML-solutions, are used in such a way that regularization can be obtained by early stopping of the iterations. Indeed, these methods have the so-called semi-convergence property: the iterates first approach the "correct" solution and then go away. As future work, a possible regularization method is being taken into account that makes use of the Kullback-Leibler divergence between the residual distribution and the noise model, under the hypothesis that some a priori knowledge is available on the noise. A further remarkable property of MRED is its possible extension to the case of multispectral images.



a) Entropy based algorithm



b) Wiener filter



c) Regularized inverse filter



d) Lucy-Richardson



e) Tikonov FFT



f) Truncated SVD

Figure 3.4: Deconvoluted images comparison, Uniform noise, entropy 2 bit.



Figure 3.5: Deconvoluted images comparison, Gaussian mixture noise, entropy 1.85 bits

Noise	Algorithm	PSNR (dB)	SSIM $\in [0, 1]$
Mixture Gaussian 1.85 bits	Wiener Filter	25.43	0.824
	Regularized Filter	25.41	0.824
	Lucy-Richardson	25.31	0.879
	Regularized Tikonov	25.40	0.785
	Truncated SVD	28.93	0.876
	MRED	25.94	0.899
Mixture Gaussian Uniform 0.7 bits	Wiener Filter	25.41	0.824
	Regularized Filter	25.41	0.824
	Lucy-Richardson	25.32	0.881
	Regularized Tikonov	25.63	0.837
	Truncated SVD	29.66	0.913
	MRED	29.80	0.926

Table 3.2: Comparison of different algorithms with different noise mixtures.

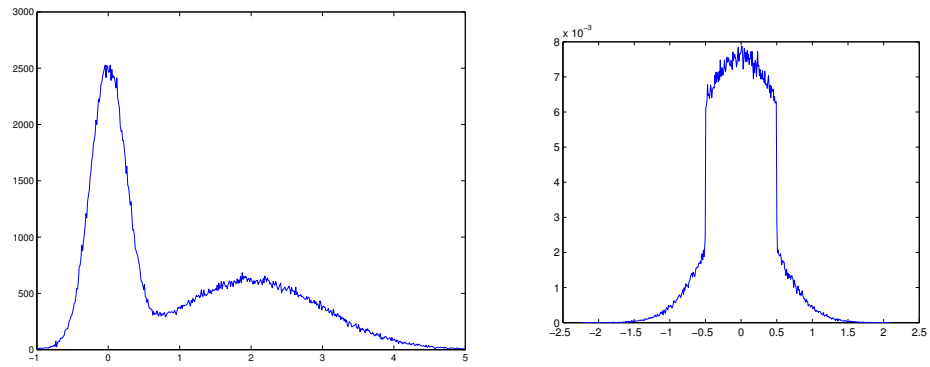


Figure 3.6: Mixture noise. Left: Gaussian mixture pdf (entropy 1.85 bits). Right: Gaussian/Uniform mixture pdf (entropy 0.7 bits).

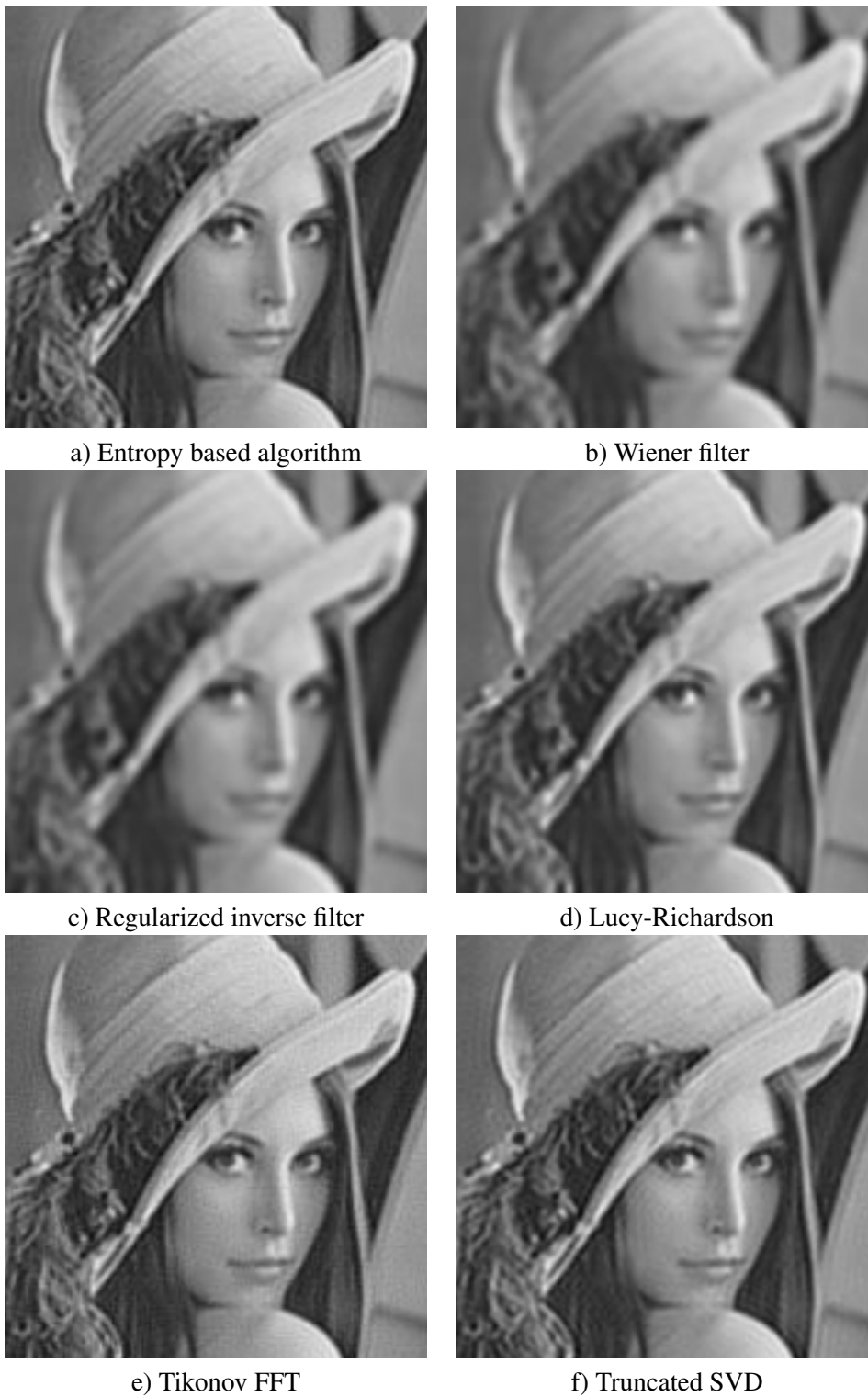


Figure 3.7: Deconvoluted images comparison, Gaussian + Uniform mixture noise, entropy 0.7 bits

Part II

Denoising

Chapter 4

PATCH-BASED DENOISING

4.1 Introduction

The ineffectiveness of many denoising techniques lies in the inadequacy of the model assumed for the image. In fact, the strategy adopted by various algorithms, is based on the assumption that the noise has a flat spectrum and the original image has significant spectral components only at low frequencies. Following this reasoning, the noise can be suppressed by attenuating the high frequencies while leaving the lower ones. The result produced is unsatisfactory, for several reasons. First, the high frequency signal components are suppressed together with noise, since it is not possible to distinguish them. It follows that the strong discontinuities or edges of objects, being concentrated at high frequencies, are not correctly reconstructed. In addition, the recovered image still contains a residual lowpass filtered noise.

Progress in denoising methods underwent a significant leap forward with non-local, patch-based methods, even compared with wavelet-based denoising and variational approaches calling upon sophisticated regularization. Based on distinct points of view, the methods UINTA [AW06] (Unsupervised, Information-Theoretic, Adaptive Image Filtering) and NL-means [BCM05] (Non Local means) pioneered this field in which BM3D [DFKE07a] (3D transform-domain collaborative filtering) represents the latest, “still to be overcome” improvement, at least in terms of the classical performance measure PSNR¹ (Peak Signal-to-Noise Ratio).

The central idea of this technique is the notion of self-similarity: given a small region of an image, a so-called patch, it is highly probable that other patches in the image are very similar. All these similar patches are degraded by noise. Nevertheless, if the noise is independent and identically distributed, then the correlations between them can be used to get rid of the noise.

In this chapter we develop a patch-based denoising algorithm following the stochastic variational approach. The chosen energy is information theory oriented, involving entropy measures on image patches.

¹Although this measure is widely used, it is known that it does not always reflect accurately the visual quality of the denoised (or, more generally, restored) image.

4.2 Neighborhood Denoising: state of the art

Classical denoising techniques employ a *local* approach in that they operate only in proximity of the pixel to be reconstructed. This strategy is unable to correctly reconstruct the texture and characteristics of small image details. An innovative idea came from some studies on the distribution of patches forming natural images [CidSZ08, HM99, LPM03] which have motivated recent patch-based processing methods, for example, for image and video denoising [BCM05, AW06, DFKE07a, BKB07] or inpainting [CPT04]. Indeed, these studies showed that there exist correlations among patches composing images. As a consequence, the probability is high that patches similar to a given image patch be encountered in the image itself. It is reasonable to think that these correlations remain in the presence of (some reasonable amount of) noise or after similar image degradations, offering the opportunity to reduce noise or to recover missing information in a patch. As a matter of fact, the nonlocal means algorithm (NL-means) [BCM05] and BM3D (3D transform-domain collaborative filtering) [DFKE07a] proved to be successful in image denoising.

In the following, we briefly describe three major techniques which represent the state-of-the-art of patch-based denoising.

4.2.1 The UINTA algorithm

The unsupervised information-theoretic adaptive filter (UINTA) [AW06] restores pixels by comparing pixel values with other pixels in the image that have similar neighborhoods. The UINTA strategy is to reduce the entropy, $h(\tilde{X}|\tilde{Y} = \tilde{y})$, of the conditional PDF for each pixel-neighborhood pair, by manipulating the value of each center pixel. For this, UINTA employs an iterative gradient descent strategy. At each step the PDF is estimated with a Parzen window approach using a multivariate Gaussian kernel where the bandwidth parameter is estimated at each step. Then the derivative of the entropy is calculated in each pixel and the descent step is performed. Being an iterative method, the stopping criteria plays an important role. Indeed, if entropy reduction counteracts the randomness introduced by the noise it reduces, however, the inherent randomness in the signal. The authors found empirically that UINTA can iterate until the root-mean-square (RMS) difference (residual) between input and the processed image equals the noise level, when the latter is known.

4.2.2 Non-Local Means

The Non-Local Means algorithm filters the noisy image $\tilde{X}(i) = X(i) + N(i)$ with a weighted mean. Unlike classical spatial techniques, the processing of the pixel $\tilde{X}(i)$ involves an extended image portion S_i called search window and centered at

i ². The pixels which fall into S_i are weighted according to the similarity between their neighborhood (usually, a square window \mathcal{N} surrounding the pixel) and that of $\tilde{X}(i)$:

$$NL[\tilde{X}](i) = \sum_{j \in S_i} w(i, j) \tilde{X}(j) \quad (4.1)$$

where the weights are defined as:

$$w(i, j) = \frac{1}{Z(i)} \exp \left[-\frac{d_{i,j}^2}{h^2} \right], \quad (4.2)$$

where $Z(i) = \sum_{j \in S_i} \exp \left[-d_{i,j}^2/h^2 \right]$ is the normalizing factor and the parameter h controls the decay of the exponential function and hence the level of filtering. The term $d_{i,j}$ represents the Gaussian weighted Euclidean distance between the two neighborhoods of $\tilde{X}(i)$ and $\tilde{X}(j)$

$$d_{i,j} = \|\tilde{X}[\mathcal{N}_i] - \tilde{X}[\mathcal{N}_j]\|_a. \quad (4.3)$$

In summary the Non-Local Means (NLM) algorithm [BCJ05] performs a weighted mean of *all* the pixels in a certain neighborhood; the weight associated with each given pixel, however, depends not on its geometrical distance from the target pixel but on its similarity with it, measured by the mean square error between the patches surrounding the selected pixel and the target. An evolution towards a multipoint rather than pointwise filtering has been proposed in the Block-Matching 3D (BM3D) algorithm [DFKE07b] which will be discussed in the next Section.

4.2.3 Block Matching 3D (BM3D)

Under some restrictive conditions, the AWGN denoising problem has simple solutions. For example, if the source is wide-sense stationary, with perfectly known statistics, the optimum linear MMSE estimator is the well-known Wiener filter. Unfortunately, real-world images are *never* stationary (the information actually lies in non-stationarities), and their statistics are not easily estimated from noisy sources, which is why more sophisticated techniques are needed.

The wavelet transform (WT) represented a major step forward in this direction. In fact, WT provides a sparse representation of images [Mal98] where large coefficients correspond region boundaries, while small coefficients contain mostly noise. Therefore, some simple form of coefficient thresholding allows for a strong noise rejection with a good preservation of image details.

Shortly after Donoho first introduced wavelet shrinkage [Don95], Baraniuk *et al.* proposed [GSB97] a two-step filtering procedure in the wavelet domain, which will be later reprised in BM3D. The first step is a hard thresholding which

²The search window could be the whole image, but it is limited in order to reduce the computational burden.

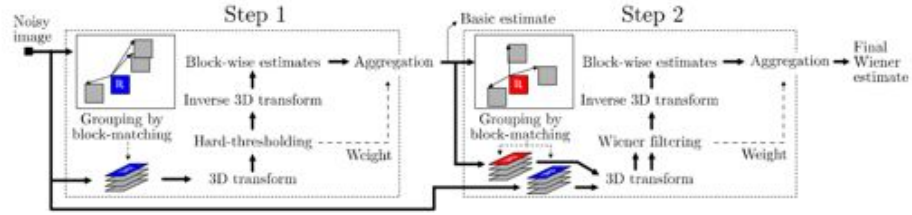


Figure 4.1: Flowchart of BM3D algorithm [DFKE07b].

provides a basic estimate of the clean image; such an estimate, however, is used only to compute the statistics for an empirical Wiener filtering operating in the transform domain [GSB97] which performs the actual denoising of the original image. Back to Wiener, then, but in the transform domain and with a preliminary hard thresholding that provides the basis for better estimating the relevant statistics.

A further change of perspective came with the non-local filtering approach, recently introduced by Buades *et al.* [BCJ05], inspired in turn by image inpainting literature. The non-local approach relies on the observation that most images exhibit clear self-similarities, as most patches repeat almost identically over and over in the image. Once these similar patches are identified, one can carry out the filtering along such patches, wherever they are, rather than in a local neighborhood of the pixel, mimicking a true statistical, as opposed to spatial, filtering.

The BM3D algorithm [DFKE07b] operates a very effective synthesis of all these ideas. Just like in [GSB97], it works in two steps: the first one uses hard thresholding to build a relatively clean image for estimating statistics, while the second one performs the actual denoising through empirical Wiener filtering in the transform domain. Both steps, however, work not on local neighborhoods, but on groups of blocks drawn from different image locations and collected on the basis of their similarity, in the spirit of the non-local approach. Therefore, the resulting 3D groups are highly redundant allowing for a sparser WT representation and a more effective separation between signal and noise through hard thresholding in the first step; as a further consequence, statistics can be more reliably estimated, and the Wiener filtering of the second step (always working on the 3D groups) turns out to be extremely effective. We can now summarize, at a very high level, the processing flow of BM3D also shown in Figure 4.1. The first step, operating on the noisy image, comprises three stages

- grouping: for each reference block, the most similar blocks are located in the image according to a minimum Euclidean distance criterion;
- collaborative filtering: each 3D group undergoes WT, hard thresholding and inverse WT;
- aggregation: all filtered blocks are returned to their original location and contribute with suitable weights to the basic estimate of the image.

The second step comprises the same three stages, with the following differences

- grouping: blocks are located based on the basic estimate provided by the first step;
- collaborative filtering: each 3D group (of noisy blocks) undergoes DCT/WT, *Wiener filtering* and inverse transform;
- aggregation: like in step one.

At present, BM3D can be arguably [KFEA10] considered the state of the art for AWGN denoising. The reader is referred to [DFKE07b] for more details.

4.3 Proposed method: PCkNN

4.3.1 Entropy-based formulation

The inverse problem of image restoration can be formulated as a minimization problem. As mentioned in Section 4.2, natural images exhibit correlation among the patches which compose them. This correlation should be accounted for in deriving a restoration procedure.

We consider as in UINTA [AW06] the conditional entropy functional, i.e., the uncertainty of the random pixel X when its neighborhood is given, as a suitable measure for denoising applications. The use of conditional entropy is justified in section 4.3.2.

Let us model an image as a random field X . Let T be the set of pixels of the image and N_t be a neighborhood of pixel $t \in T$. We define a random vector $Y(t) = \{X(\tau)\}_{\tau \in N_t}$, corresponding to the set of (color) intensities at the neighbors of pixel t . We also define a random vector $Z(t) = (X(t), Y(t))$ to denote image regions or patches, i.e., pixels combined with their neighborhoods (see Figure 4.2).

The recovered image ideally satisfies

$$\hat{X} = \arg \min_X h(X|Y = y_i), \quad (4.4)$$

where h represents the differential entropy. The minimization is carried out via a gradient descent and the energy derivative is calculated in Section 4.3.3.

4.3.2 Energy lower bound

In this section we present motivations for use conditional entropy as suitable measure for denoising applications. To do this, we provide an information theoretic result on conditional entropy in noisy images.

The conditional entropy of patches represents the uncertainty on the color X of a pixel when its neighborhood Y is known. Due to the spatial correlation between a pixel and its neighborhood, this conditional uncertainty is generally small in average. When adding noise to the image, some of the information carried by the

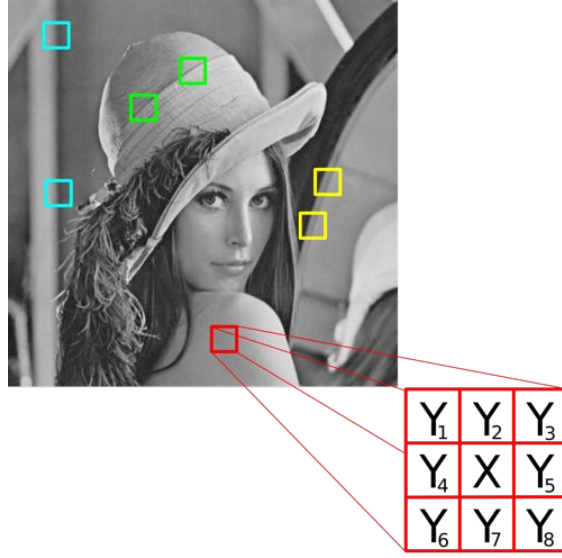


Figure 4.2: Image patch illustration.

neighborhood is lost, so that the uncertainty of a pixel knowing its neighborhood tends to be higher in average. This is formally stated by the following proposition.

Proposition 4.1. *Let X be a random variable and Y a random vector representing its neighborhood. Let \tilde{X} be the sum of X with a white³ noise N independent of X . Similarly, let \tilde{Y} be a noisy neighborhood vector. Then,*

$$h(\tilde{X}|\tilde{Y}) \geq h(X|Y). \quad (4.5)$$

Proof. By definition,

$$h(\tilde{X}|\tilde{Y}) = h(\tilde{X}) - I(\tilde{X}; \tilde{Y}) \quad (4.6)$$

and

$$h(X|Y) = h(X) - I(X; Y), \quad (4.7)$$

where $I(\cdot; \cdot)$ denote the mutual information. First note that $h(\tilde{X})$ is greater than $h(X)$ since the addition of two independent random variables increases the entropy [CT91]. Then we have to prove that $I(X; Y) \geq I(\tilde{X}; \tilde{Y})$. Note that $X \rightarrow Y \rightarrow \tilde{Y}$ forms a Markov chain, since \tilde{Y} and X are conditionally independent given Y . Thus, the data processing inequality [CT91] reads

$$I(X; Y) \geq I(X; \tilde{Y}). \quad (4.8)$$

Now, since mutual information is symmetric and $\tilde{Y} \rightarrow X \rightarrow \tilde{X}$ forms a Markov chain itself (since the noise is white), than we obtain

$$I(\tilde{Y}; X) \geq I(\tilde{Y}; \tilde{X}). \quad (4.9)$$

³The samples are assumed to be statistically independent.

By combining Eq. (4.8) and Eq. (4.9) we have

$$I(X; Y) \geq I(\tilde{X}; \tilde{Y}). \quad (4.10)$$

□

Proposition 4.1 supports the intuition that the minimization of the conditional entropy is an appropriate denoising approach. However, in practice, X must be recovered while the noiseless neighborhood Y is also unknown. \tilde{Y} can be inferred from realizations of the observation \tilde{X} , though. Hence, an inequality involving $h(X|\tilde{Y})$ would better justify an algorithm based on conditional entropy.

Proposition 4.2. *Let $\delta X = X + \delta$ with δ be independent from Y and N . Then*

$$h(\delta X|\tilde{Y}) \geq h(X|\tilde{Y}). \quad (4.11)$$

Proof. The proof is the same as for prop. 4.1. Note that $\delta X \rightarrow X \rightarrow \tilde{Y}$ forms a Markov chain. □

As a consequence, the random variable X associated with the noiseless image is also a minimizer of the conditional entropy when the noisy neighborhood \tilde{Y} is known.

Fig. 4.3 illustrates the behavior of the conditional entropy before and after denoising with respect to the lower bound $h(X|\tilde{Y})$. As expected, the conditional entropy $h(\tilde{X}|\tilde{Y})$ of the noisy image is greater than or equal to the conditional entropy $h(\hat{X}|\tilde{Y})$ of the denoised image for all noise levels.

Unlike UINTA, the lower bound provides a natural stopping criterion for an iterative algorithm. Such a lower bound does not exist in UINTA because the context \tilde{Y} changes with iterations. However, the conditional entropy of the denoised image is occasionally lower than the theoretic lower bound. This is explained by the fact that rapidly varying textures are interpreted as noise and therefore partially degraded by the denoising process. Therefore, iterations should be stopped before the lower bound is attained.

4.3.3 Energy derivative

Classically, we propose to use a gradient descent procedure to solve the optimization problem in (4.4). As a consequence, we need to determine the derivative of the conditional entropy of the color of a pixel knowing its neighborhood.

Entropy, whether conditional or not, can be approximated by the Ahmad-Lin estimator [AL76]

$$h(X|Y = y_i) \approx -\frac{1}{|T|} \sum_{t_j \in T} \log p(x_j|y_i), \quad (4.12)$$

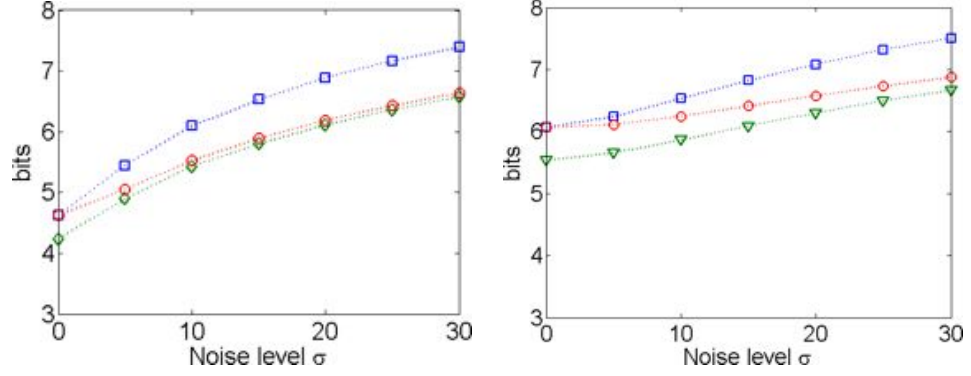


Figure 4.3: Behavior of the conditional entropy before and after denoising. Squares represent the conditional entropy, $h(\tilde{X}|\tilde{Y})$, of the noisy image. Diamonds represent the conditional entropy, $h(\hat{X}|\tilde{Y})$, after denoising with proposed method. Circles denote the lower bound $h(X|\tilde{Y})$. Left: Lena. Right: Mandrill.

where the color x_j is encountered at pixel t_j , y_i is the set of (noisy) colors in the neighborhood of t_j , and

$$p(s|y_i) = \frac{1}{|T_{y_i}|} \sum_{t_m \in T_{y_i}} K(s - x_m), \quad (4.13)$$

is the kernel estimate of the probability density function (PDF), with T_{y_i} the set of pixels which have the same neighborhood y_i , $K(\cdot)$ a symmetric kernel, and x_m the color encountered at pixel t_m .

The energy derivative of (4.12) is (see Appendix C for the demonstration)

$$\frac{\partial h(X|Y = y_i)}{\partial x_i} \approx -\frac{1}{|T|} \frac{\nabla p(z_i)}{p(z_i)} \frac{\partial z_i}{\partial x_i}, \quad (4.14)$$

where $p(Z)$ is the joint PDF of the high dimensional random vector Z . The solution to (4.4) can be computed by gradient descent

$$\begin{cases} X^{(0)} = X_0 \\ x_i^{(n+1)} = x_i^{(n)} - \alpha \frac{\partial h(X^{(n)}|\tilde{Y} = y_i)}{\partial x_i} \\ \quad = x_i^{(n)} + \beta \frac{\nabla p(z_i^{(n)})}{p(z_i^{(n)})} \cdot \frac{\partial z_i^{(n)}}{\partial x_i} \end{cases}. \quad (4.15)$$

Equation (4.14) shows the energy derivative is the product of a gradient vector on the space of patches Z and a projection term which projects the gradient onto the pixel space X . This means that we first calculate a patch of which we retain only the central pixel in the update equation 4.15. In the next Section we extent this pixel-based procedure to a full patch processing.

4.3.4 Full patch denoising

Unlike denoising methods such as [AW06, BCM05] which actually implement pixel-based iterative procedures relying on patches, as done by (4.15), we propose to convert (4.15) into a patch-based iterative procedure. Discarding the projection term $\partial z_i^{(n)} / \partial x_i$, we get

$$\begin{cases} X^{(0)} = X_0 \\ z_i^{(n+1)} = z_i^{(n)} + \beta \frac{\nabla p(z_i^{(n)})}{p(z_i^{(n)})} \end{cases} \quad (4.16)$$

The normalized derivative in (4.16) can be approximated by a mean shift term [FH75, CM02] on the high dimensional joint PDF of Z . The mean shift gives an estimation of $\nabla \log p(z_i)$ [FH75]. In the k -th nearest neighbor (kNN) framework, it can be expressed as [ADB08]

$$\frac{\nabla p(z_i)}{p(z_i)} = \frac{d+2}{\rho_k^2(z_i)} \frac{1}{k} \sum_{z_j \in \text{kNN}(z_i)} (z_j - z_i) \quad (4.17)$$

where d is the dimension of Z , $\text{kNN}(z)$ denotes the set of the k closest patches of z and $\rho_k(z_i)$ is the kNN patch distance.

By setting $\beta = \rho_k^2(z_i)/(d+2)$ and using (4.17), it is clear that the iterative procedure of (4.16) is equivalent to

$$z_i^{(n+1)} = \frac{1}{k} \sum_{z_j \in \text{kNN}(z_i)} z_j^{(n)} \quad (4.18)$$

In practice, we have noticed that performing only one iteration is sufficient. Therefore, for each patch z_i , we have

$$z_i^* = \frac{1}{k} \sum_{z_j \in \text{kNN}(z_i)} z_j. \quad (4.19)$$

Furthermore, to account for the fact that, among the patches of $\text{kNN}(z_i)$, patches farther away might not correspond as well to noisy versions of z_i as do closer patches (they may also be slightly structurally different), the average in (4.19) is weighted as follows

$$z_i^* = \frac{1}{\sum_{z_j \in \text{kNN}(z_i)} w_j} \sum_{z_j \in \text{kNN}(z_i)} w_j z_j \quad (4.20)$$

where, similarly to NL-means [BCM05],

$$w_j = \exp(-|z_i - z_j|^2 / \sigma_w^2), \quad (4.21)$$

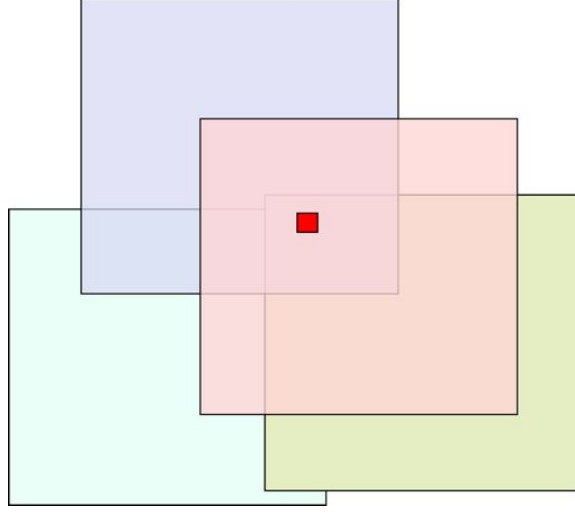


Figure 4.4: Patch overlapping illustration.

σ_w being a parameter. To reduce the effect of noise, the distances between patches are computed as in [Tas09] after a Principal Component Analysis (PCA): $|z_i - z_j| \rightarrow |\text{PCA}(z_i) - \text{PCA}(z_j)|$.

Dealing with full patch denoising leads to an estimation problem. In fact, a single pixel in the restored image belongs to several denoised patches, each of them carrying its own value. As an example, Figure 4.4 shows such a situation. Therefore the question on how to combine these different estimates arises. The next Section is devoted to the denoised patch combination.

4.3.5 Confidence-based patch combination

Denoised patches obtained in (4.20) overlap each other. In consequence, there is some redundancy in the denoising process. Indeed, for a given pixel x_i , we obtain several estimators, one for each patch to which x_i belongs (if the patch size is $N_p \times N_p$, we have N_p^2 estimators.). This is shown in Figure 4.4.

The simplest procedure to come back from patch space to pixel space is to retain for each denoised patch only the central pixel. This is employed for example in NLmeans [BCM05]. A further step is try to combine the different estimates coming from denoised patches. Among a plethora of methods for combining estimators, we used, as in [DFKE07a], a linear combination of denoised patches. Clearly, among all the patches containing x_i , some will lead to an accurate denoising at x_i , some might not while leading to an accurate denoising in other pixels. Therefore we give to each denoised patch z_i a weight c_i which represents its reliability. We

call c_i the patch denoising confidence. For a given pixel i , the aggregation writes

$$x_i = \frac{1}{\sum_{q=1}^{N_p^2} c_{i,q}} \sum_{q=1}^{N_p^2} c_{i,q} x_{i,q}^*, \quad (4.22)$$

where $c_{i,q}$ is the confidence of patch q among the N_p^2 patches which contain pixel i .

We chose as patch denoising confidence

$$c_i = \frac{\left(\sum_{z_j \in \text{kNN}(z_i)} w_j \right)^2}{\sum_{z_j \in \text{kNN}(z_i)} w_j^2}. \quad (4.23)$$

In fact, since the noise is assumed to be uncorrelated, the synchronous summation (4.20) reduces the amount of noise (as characterized by its variance) by a factor of $1/c_i$.

In summary, for each patch z_i of the noisy image, we are able to compute with the confidence c_i (see (4.23)) a denoised patch z_i^* (see (4.20)). These patches are then combined (or aggregated) according to their confidence term. In Figure 4.5 the patch combination procedure is sketched.

The denoised patches z_i^* are then aggregated as follows: starting from an aggregation image of zeros and a confidence map of zeros, a denoised patch z_i^* is added, after weighting by c_i , to the aggregation image at its original location, and a constant patch equal to c_i is added to the confidence map at the same location. After dealing with all the patches, the denoised image is defined as the pointwise division between the aggregation image and the confidence map.

4.3.6 Summary of the algorithm

The Patch Confidence k NN denoising (PCkNN) algorithm tries to minimize the conditional entropy of a pixel knowing its neighborhood. To do this, we estimate the mean shift vector in the high dimensional feature space Z , which represents the space of noisy patches. Estimation is carried out by searching the k nearest neighbors in such a high dimensional space. Then for each pixel in the noisy image, a denoised patch is estimated as a weighted summation of the k nearest patches (see (4.20)). The weights depend on the distance between patches (see (4.21)) and they contribute to the confidence c_i (see (4.23)). Then denoised patches are aggregated as in Section 4.3.5 (see (4.22)) to form the final denoised image. Figure 4.7 shows the pseudo code for the proposed algorithm and in Figure 4.6 a block diagram of the proposed PCkNN algorithm is sketched.

4.4 Experiments

We compare the proposed technique with three state-of-the-art denoising algorithms: the UINTA algorithm [AW06], the Non Local Means (NL-

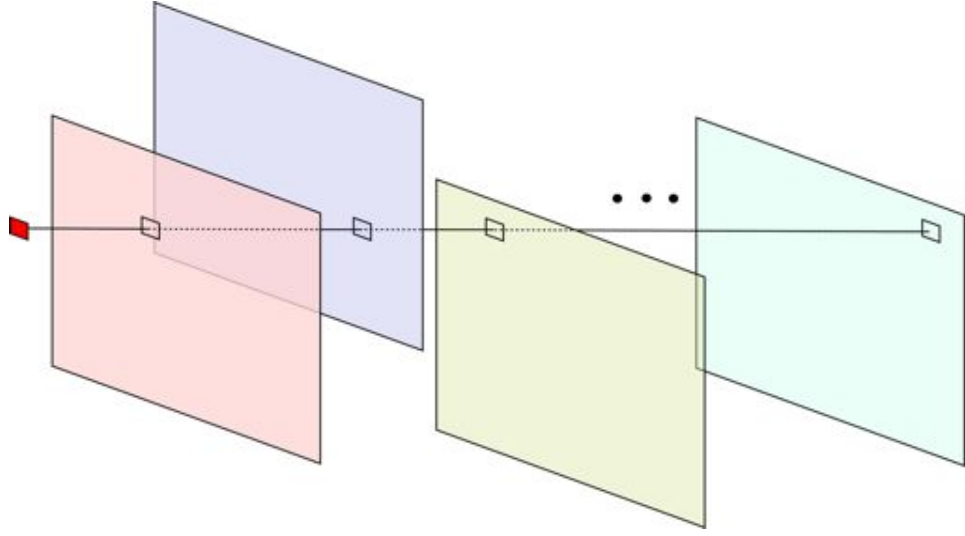


Figure 4.5: Patch combination step illustration.

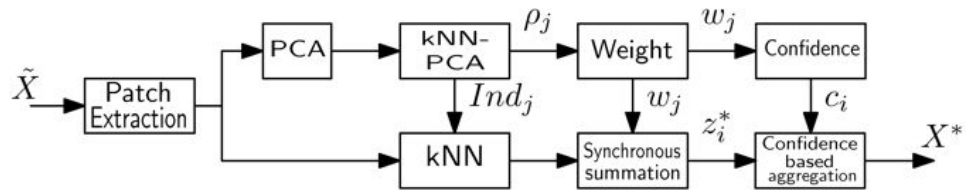


Figure 4.6: Block diagram for PCKNN algorithm.

Pseudo code

For each pixel x :

- $A(x, r)$ (search Area);
- $W(x, f)$, i.e., the patch of radius f surrounding x ;
- For each pixel $y \in A(x, r) - \{x\}$:
 - $W^*(\cdot, f) = W(\cdot, f) \cdot \text{Mask}$;
 - $d(x, y) = \|W^*(x, f) - W^*(y, f)\|^2$;
 - choose the first k nearest elements, such as $0 \neq d(x, y_1) \leq d(x, y_2) \leq \dots \leq d(x, y_k)$;
 - $w(x, y_i) = \exp -d(x, y_i)/\sigma_w^2$ with $\sigma_w = \sigma$;
 - then the denoised patch is

$$W^*(x, f) = \frac{1}{\sum_{i=1}^k w(x, y_i)} \sum_{i=1}^k w(x, y_i) W(y_i, f) \quad (4.24)$$

- assign to each pixel of the denoised patch the confidence

$$c(x) = \frac{\left(\sum_{i=1}^k w(x, y_i)\right)^2}{\sum_{i=1}^k w(x, y_i)^2}$$

- replace the denoised patch in the image

$$U(x, f) \leftarrow c(x) W^*(x, f) \quad (4.25)$$

$$C(x, f) \leftarrow c(x) \quad (4.26)$$

Then the final image is:

$$U \leftarrow U./C \quad (4.27)$$

Figure 4.7: Pseudo code for the proposed PCKNN algorithm.

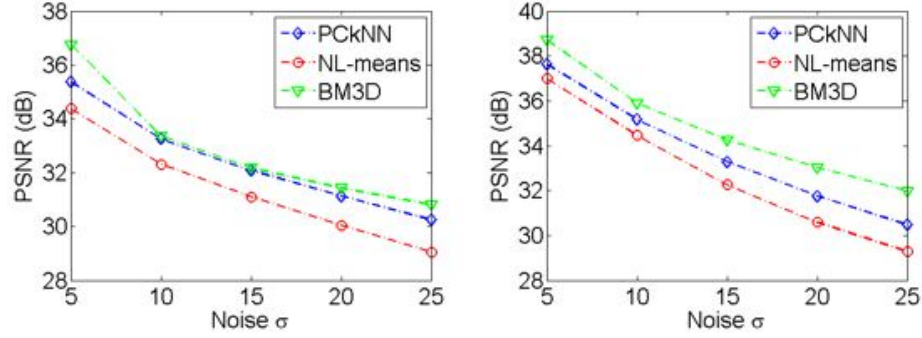


Figure 4.8: PSNR for the image Elaine (left) and Lena (right).

means) [BCM05], and the Block Matching 3D (BM3D) algorithm [DFKE07a], all briefly described in Section 4.2. Such techniques have been chosen because of their competitive performance and (not least) for the availability of software code to run the experiments. Experimental results have in fact been obtained by using the Authors' own code available online. For all these algorithms, if not stated otherwise, the free parameters are set as suggested in the reference papers. However, we will highlight the common parameters among the different methods.

We considered several images commonly used in the AWGN denoising and shown in Figure D.1. Noisy images are obtained by adding a simulated white Gaussian noise with standard deviation σ of 5, 10, 15, 20 and 25. We set BM3D in normal profile with a patch size of 8 and a search window of 39, while UINTA set the neighborhood size to 9. We chose for both PCKNN and NL-means a patch radius of 7 and a search radius of 15. Furthermore, as suggested from authors, we set $h = \sigma$, h being the filtering parameter of NL-means and σ the noise standard deviation. The performance is quantified by the PSNR and SSIM measures defined in Section 3.4. PSNR measure and the related mean square error (MSE) are the simplest and most widely used quality metrics in image processing, with clear physical meanings, but they are not very well matched to perceived visual quality [WBSS04]. Human visual system is highly adapted for extracting structural information, and SSIM, which compares local patterns of pixel intensities, is more suitable for image quality assessment.

As shown in Figure 4.8⁴, PCKNN outperforms both UINTA and NL-means and is very close to BM3D which provides consistently the best performance in terms of PSNR. This is not surprising since BM3D is basically a two step algorithm, where the second one performs a Wiener filtering. As it is well known, Wiener filter minimizes the average squared distance between the filter output and the desired signal, which, in the case of denoising, turns out to be the noise-free signal. Since PSNR is equivalent to MSE (all images have the same dynamic), BM3D is optimal in the sense that maximizes the PSNR. However, PSNR does not provide a

⁴A full set of comparison is reported in Appendix D.

stand-alone measure for quality assessment. Indeed, the image quality of PCkNN is definitely better than the other algorithms as shown in Figure 4.9 for the image Elaine. In particular, the residual noise is quite natural and does not exhibit spurious patterns, thus leading to a denoised image with a very natural appearance. NL-means is clearly oversmoothed and BM3D presents many flattened regions in smoothly varying areas (see Fig. 4.10), giving a somewhat unnatural, cartoon effect to the denoised image. This is the major drawback of this algorithm for high noise levels that might be due to the thresholding in the wavelet domain.

To illustrate the cartoon effect, Figure 4.10 shows a close-up on the image Elaine and the corresponding isolevel lines. The orientation and density of these lines provide an indication on the direction and the norm of the gray level gradient. PCkNN preserved very well the original isolevel line configuration while BM3D created a “patchwork” of flattened regions. This behavior is very well known in the data compression communities since 80’s and the same effect can be observed also in Figures 4.11 and 4.12 for the images Lena and Barbara respectively.

In order to measure the flattening behavior, we rely on a Total Variation (TV) based criterion that we call Normalized TV error (nTV). It is the integral of the absolute relative error between the norm gradients of the denoised and the original image respectively. Namely, if u and \hat{u} are respectively the original and the denoised image, nTV is given by

$$\text{nTV}(\hat{u}) = \frac{1}{\Omega} \int_{\Omega} \frac{||\nabla \hat{u}(x)| - |\nabla u(x)||}{\frac{1}{|\mathcal{N}(x)|} \int_{\mathcal{N}(x)} |\nabla u(y)| dy} dx, \quad (4.28)$$

where the average value in a neighborhood $\mathcal{N}(x)$ of x is used in order to avoid division by zero. Lower values of nTV indicate a better reconstruction of image gradient and hence fine details in the denoised image. Figure 4.13 shows the nTV criterion for the image Elaine and Lena.

A further analysis can also be done in the Fourier domain. According to the model

$$\hat{u} = u * G_{\sigma} + \epsilon \quad (4.29)$$

where the denoised image is modeled as a blurred version of the original image plus some noise which represents the model noise, we can analyze the Fourier Transform (FT) of the error image which comes out from the difference between the algorithm output and a blurred version of the original image. Figure 4.14 shows the modulus of the error FT for both BM3D and PCkNN corresponding to both the image Elaine and Lena corrupted with noise with standard deviation $\sigma = 25$. The figure clearly shows that the BM3D FT error has high energy in the low frequency range. For the sake of clarity, we plotted the equivalent energy density distribution in Figure 4.15. The curves of Figure 4.15 have been obtained by integrating over growing rings the Energy Density Spectrum (ESD) (*i.e.*, the square modulus of the FT) of the error. We then normalized by the error total energy in order to have both curves integrate to 1 (Parseval theorem). Finally, we showed the densities in the log-scale (dB) in order to highlights the differences.



Figure 4.9: In lexicographic order: Noisy, Original, BM3D, PCKNN, NL-means, and UINTA. The image Elaine was corrupted with an additive white Gaussian noise with standard deviation of $\sigma = 25$.

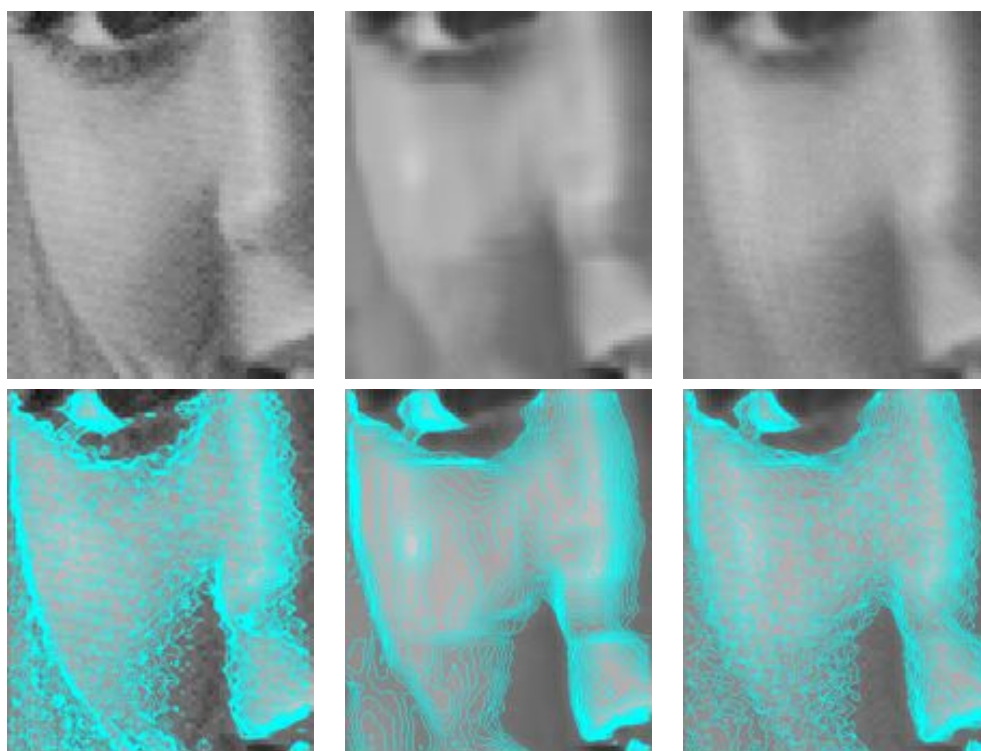


Figure 4.10: A close-up on the image Elaine. From left to right: Original, BM3D, and PCKNN. First row: image alone; second row: isolevel lines superimposed on the image.

Figure 4.15 clearly shows that PCKNN error is more concentrated in high frequencies. On the contrary, BM3D error presents higher energy at low frequencies. At first sight, the human visual system (HVS) acts as a low-pass filter, hence, it is more sensitive to low-frequency errors. Therefore, the perceived quality of PCKNN is better than BM3D. This does not mean that in general PCKNN removes more noise than BM3D, but the residual noise of PCKNN appears more natural since it is masked by the HVS. This is a well-known characteristic in speech audio processing domain in which noise is masked near the speech formants. Thus, a good denoising algorithm should mask the inevitable residual noise as much as possible.

4.5 Improving PCKNN with robust patch similarity

The key idea of patch-based denoising algorithm is image self similarity. On an image, one can find several small areas or patches that are *similar* to each other. Suppose that we have several noisy realizations of the same patch. Assuming an additive zero mean noise, we can denoise such a patch by taking the average of



Figure 4.11: A close-up on the image Lena. From left to right: Original, BM3D, and PCkNN. First row: image alone; second row: isolevel lines superimposed on the image.

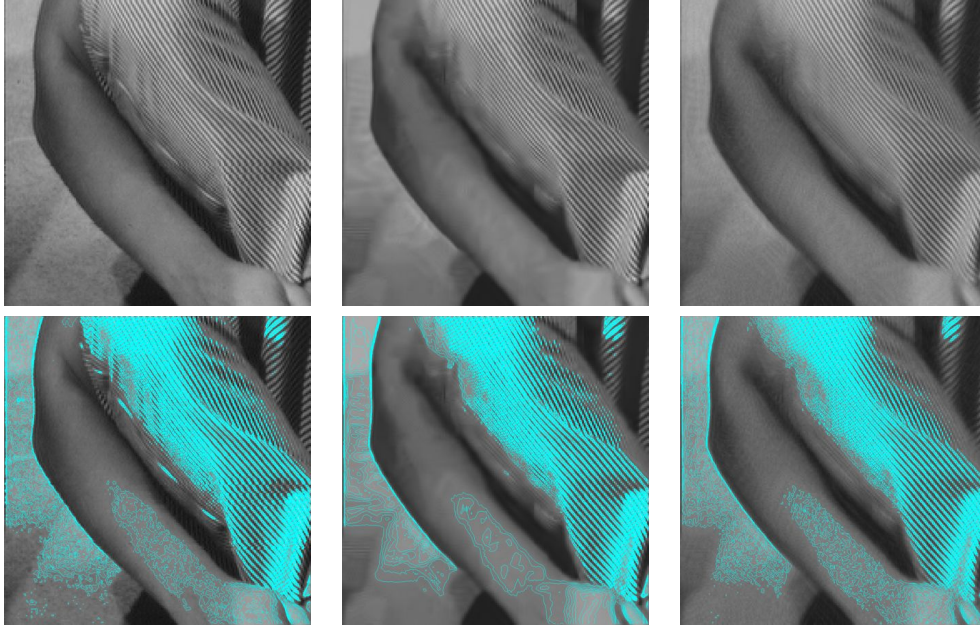


Figure 4.12: A close-up on the image Barbara. From left to right: Original, BM3D, and PCKNN. First row: image alone; second row: isovalue lines superimposed on the image.

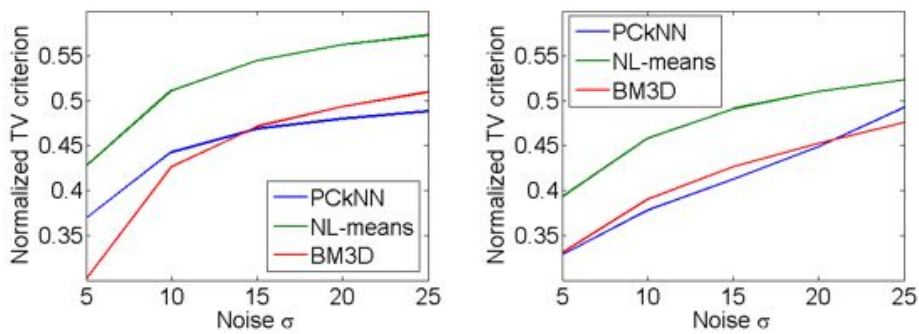


Figure 4.13: Normalized Total Variation. Left: Elaine. Right: Lena.

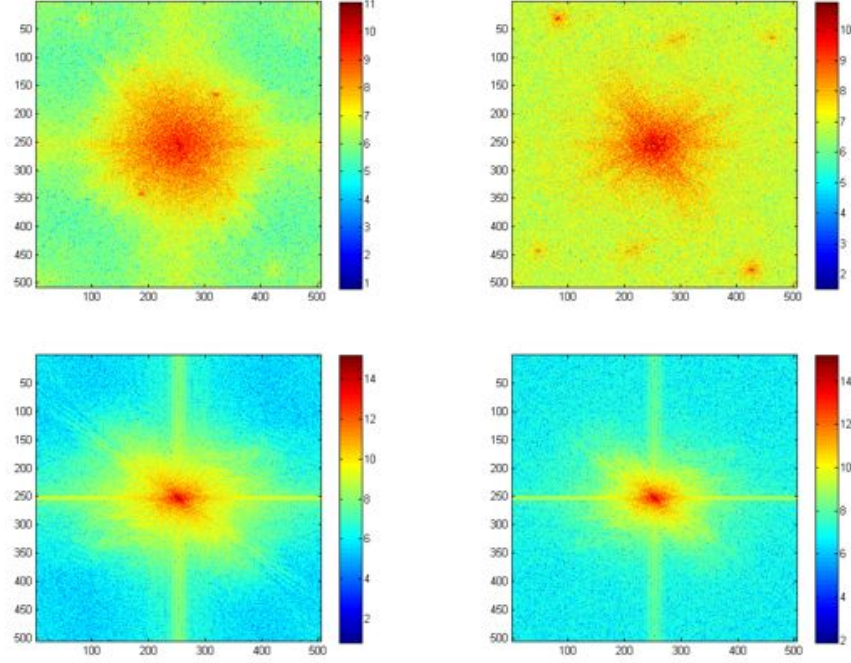


Figure 4.14: FT Amplitude of the error between the blurred original and the denoised image. Left BM3D, right PCKNN. First row: Elaine, second row: Lena.

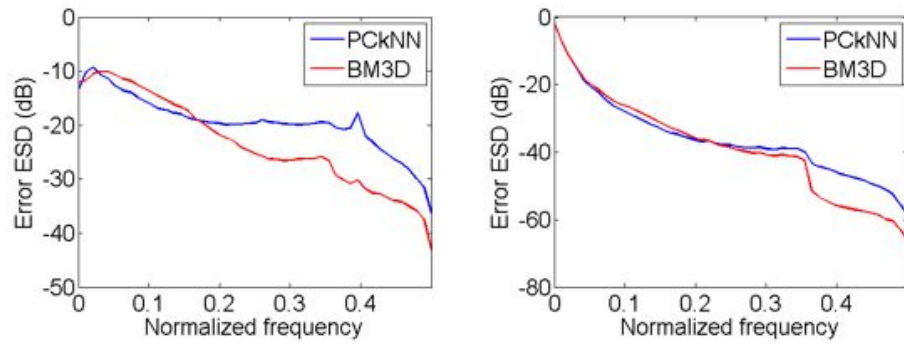


Figure 4.15: Energy Spectrum Density (ESD) (dB) of the reconstruction error. Left: Elaine. Right: Lena.

the noisy patches. The (dis)similarity function between two patches plays an important role in choosing the best *candidates* (in our case the k nearest neighbors) for the denoising task. The most popular choice is the \mathcal{L}^2 -distance between the patches. However, when the noise is high, such a distance becomes unreliable. Moreover, two patches can be similar only up to some transformation such as rotation. Therefore, we propose to define a patch similarity invariant to some patch transformations and applied to polynomial approximations of the patches. This represents an improvement over PCKNN.

4.5.1 Robustness to noise

Before trying to look for similar patches, the patches are modified in order to ensure some level of robustness with respect to noise. Given a noisy patch considered as a surface, a polynomial surface approximation is computed with independent degrees in both directions. Each degree can range from 1 to 10. The best approximation has been defined as the one producing a zero-mean noise between the noisy patch and its approximation. The approximations of all the patches are then used, possibly after geometrical transformations (see Section 4.5.2), to look for self similarities.

4.5.2 Transformation invariance

Standard patch-based denoising algorithms compute the \mathcal{L}^2 -distance blockwise. This means that the similar patch search is performed allowing only for patch translation. However, more similar patches can often be found if allowing for rotation or symmetry of the patches [SPH09]. Building on this idea, we propose an extended set of patch transformations:

- symmetries with respect to the horizontal and vertical axes and the diagonals;
- rotations by $\pi/2$, π , and $3\pi/2$;
- rotations between $-\alpha$ and α with a step of $d\alpha$.

Note that when comparing two patches, the rotations $\pi/2$, π , and $3\pi/2$ are often not the transformation leading to the best matching (contrary to the rotations between $-\alpha$ and α with α around $\pi/6$). However, they are computed cheaply and exactly (no interpolation required).

In summary, let S be the set of patches in the image and \mathcal{T} the set of transformations (symmetries and rotations), then the set in which we look for similar patches is the augmented set $S \cup \mathcal{T}(S)$.

4.5.3 Effect of the transformations: A toy example

A white Gaussian noise was added to a synthetic image of a disk. The PSNR of the noisy image was 14.9. This image was denoised without and with transformation invariant patch similarity (see Section 4.5.2). The positive effect of using patch

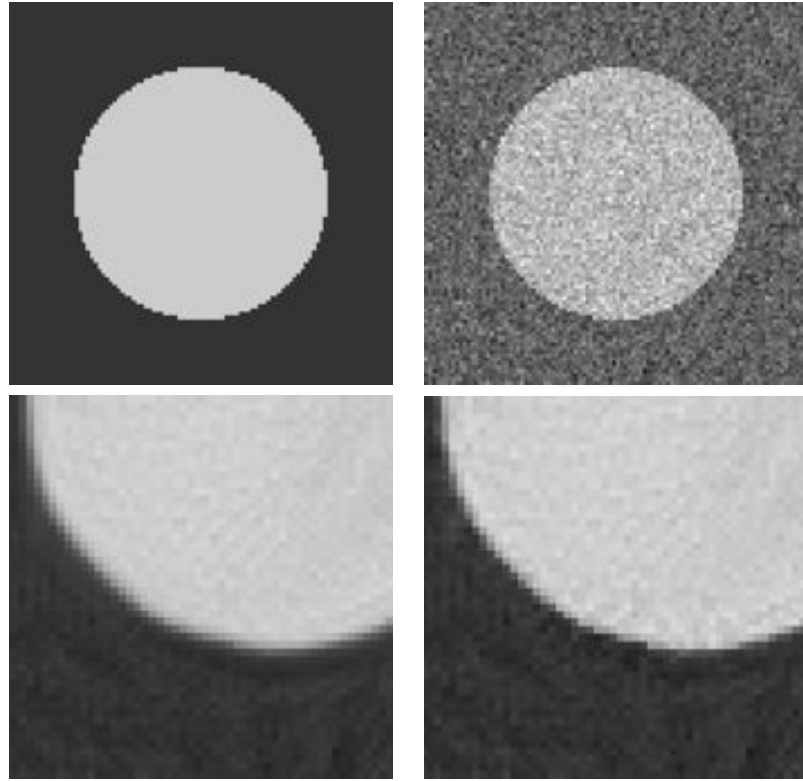


Figure 4.16: Effect of the transformations: A toy example. In lexicographic order: Original, Noisy, Denoised with no transformations, Denoised with transformations.

transformations can be seen on Figure 4.16. Objectively, the denoised image obtained with transformations has a PSNR of 29.1 while the image obtained without transformations has a PSNR of 27.7.

We also tested this improved algorithm PDC-RS on the image elaine for several level of noise. Figure 4.17 shows the Peak Signal to Noise Ratio (PSNR) as function of the noise standard deviation for the compared algorithms. PDC-RS still outperforms UINTA and NLmeans and despite a slightly lower score w.r.t. PCkNN and BM3D has an improved image quality as highlighted in Figure 4.18.

4.6 Conclusion and perspectives

In this Chapter we proposed a novel algorithm for image denoising. The development started from Information Theory concepts and we saw that entropy minimization leads to non-local filtering approaches. Thus, we also provided a new variational interpretation of non-local filtering such as NL-means. However, key

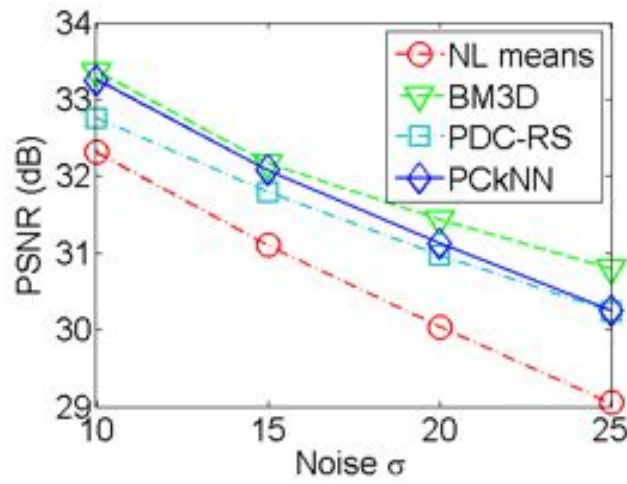


Figure 4.17: PSNR plot for the image Elaine.

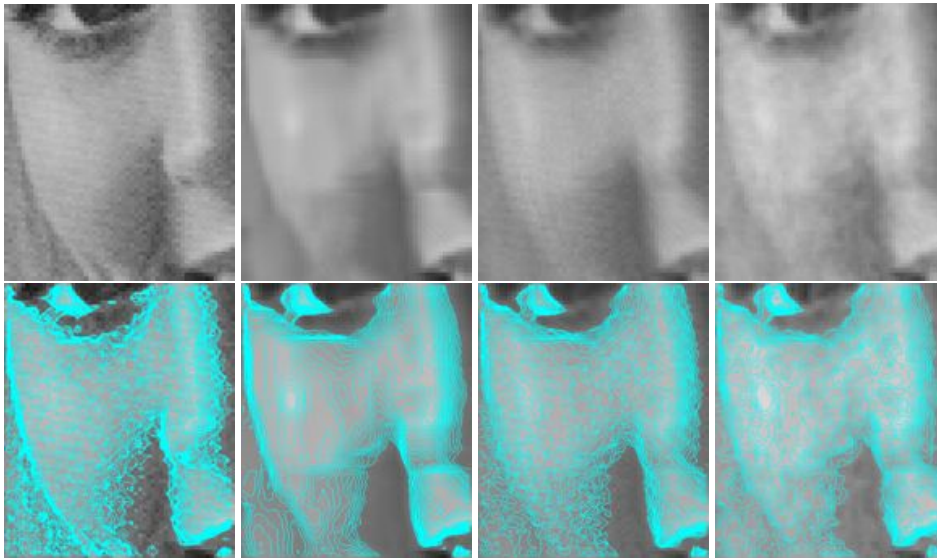


Figure 4.18: A close-up on the image Elaine of Figure 4.9. From left to right: Original, BM3D, PCKNN and PDC-RS. Top: image alone; bottom: isolevel lines superimposed on the image.

philosophical issues remain to be solved.

First, what is the optimal patch size? Classically, this size is a parameter. However, it is clear that the number and quality of patches similar (with some tolerance) to a patch centered on a given pixel vary with the patch size: (a) as the patch gets smaller, more similar patches are found but they are less and less useful for denoising; (b) for some range of patch size, the number of similar patches appropriate for denoising should be sufficient; (c) as the patch gets larger, less patches are found and they are less and less useful for denoising.

The second key issue is the definition of a good measure of similarity between patches. Normally, the \mathcal{L}_2 -norm is used. Besides the fact that the patches are degraded by noise, this measure does not, as one can expect, correspond to a reliable hint of visual similarity. Which alternatives can help solving these questions? Concerning the patch size, one can think of using the local information scale such as the one provided by keypoint detectors (SIFT (Scale-invariant feature transform)...). However, how many times some local pattern is repeated in the image is what counts for denoising and such a scale is not linked to this repetition. Yet, a multiscale decomposition of the whole image can be used to infer the local degree of repetition at the appropriate scale.

Concerning the measure of similarity between patches, without pretending to find a measure expressing visual similarity (something virtually impossible for this notion being subjective), a measure based on the decomposition of patches on some basis is a possible direction of research. A truncated PCA (Principal component analysis) and SVD (Singular value decomposition) have been proposed for that purpose [Tas09, OEW08], although with moderate success. Another option would be to really exploit a coarse-to-fine decomposition to define a similarity measure.

The method will be further developed to deal with color images. Although it would be straightforward to simply apply the method to each channel independently, it is clear that a joint processing would be more appropriate. It appears to us that the proposed method can achieve this goal since (1) the mean shift principle applies to any dimension and (2) the kNN framework allows to deal with high dimensional data efficiently.

Another aspect that will be tackled is the definition of a data fidelity term. Referring to the proposed method as an operator on an image, one can note that this operator is not idempotent. Applying it again and again will produce images less and less noisy but also, eventually, degraded. In such an iterative scenario, the denoising operator should therefore be balanced with data fidelity. This latter term should probably follow the same information theoretic inspiration as the former one.

Part III

Specific applications

Chapter 5

DIGITAL PHOTOGRAPHY

5.1 Introduction

This chapter and the next two chapters deal with application of denoising to digital photography, synthetic aperture radar (SAR) images and to digital image inpainting (which can be viewed as a high-level denoising task). The growth of compact digital cameras and the use of satellite applications as well as implementation of inpainting algorithm in prosumer softwares show how denoising is still an hot topic in the image processing and computer vision communities. These *real world* applications point out that the noise is not only AWGN. Indeed, in modern digital cameras the noise can be approximated by an additive random variable with the variance depending on the signal intensity. In SAR images the noise is still signal dependent but it is also a multiplicative noise and not Gaussian. Finally, in digital inpainting the “noise” has no statistical characterization but only spatial, let say we know *where* it is but we do not know *what* it is.

This chapter is devoted to the digital photography. In the next Sections the noise model of modern cameras is introduced. Then, an adapted version of our PCkNN algorithm is presented.

5.2 Noise and sensor

5.2.1 Sensor and image acquisition

The heart of any digital camera is the sensor which acquires the images. It consists of a great number of small photo detectors, which are often realized with a CCD technology. The image acquisition converts the light in a 2-dimensional signal which measures the light intensity in each spatial position. This task is performed by the image sensor, a $M \times N$ matrix of photo detectors. A photo detector is an electronic device which stores an electrical charge proportional to the energy of the

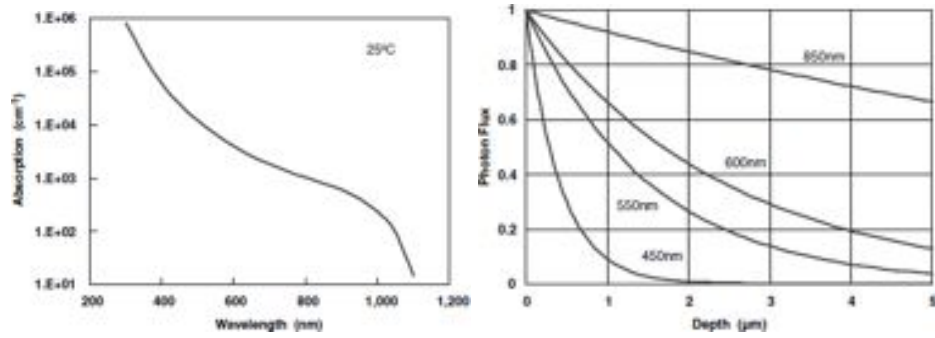


Figure 5.1: Physical properties of Silicon. The images above come from [Nak05].

incident light, with the following relation [HK94]:

$$I = T \int_{\Lambda} B(\lambda) \eta(\lambda) d\lambda, \quad (5.1)$$

where T is the exposition time, B is the incident light energy, η is the charge density per energy unit and Λ is the visible wavelength spectrum. The parameter η depends on the number of photons that the silicon is able to absorb [Nak05]. Figure 5.1a shows the silicon sensibility as function of the wavelength. The graph points out that a photo detector presents a monochromatic behavior. Figure 5.1b shows the penetration ability of a photon in the silicon. As we can expect, the higher is the incident radiation wavelength the higher is the penetration depth. This two properties of the silicon allows two different type of sensors:

1. **Color Filter Arrays (CFA).** The photo detectors are installed on the silicon surface following a bi-dimensional grid. In order to separate the colors, each photosite is covered by a color filter. The most used mask is the so-called *Bayer grid* where each 2×2 block receive 1 red pixel, 2 green pixel and 1 blue pixel (see Figure 5.2 left). To have a complete RGB image a further step, called *demosaicing*, is required. This step combines the outputs of neighbors pixel in order to reconstruct the R,G, and B components in each spatial position;
2. **Layered sensors.** The photo-detectors are installed in the silicon at three different depth (see Figure 5.2 right). The key point is the penetration ability of photons. In fact, the first layer catches all visible spectral components. The second one, receives the red and green, but not the blue, which is filtered by the silicon itself. The the last layer receives only the red light. A linear combination of the three outputs give the color information in each spatial position.

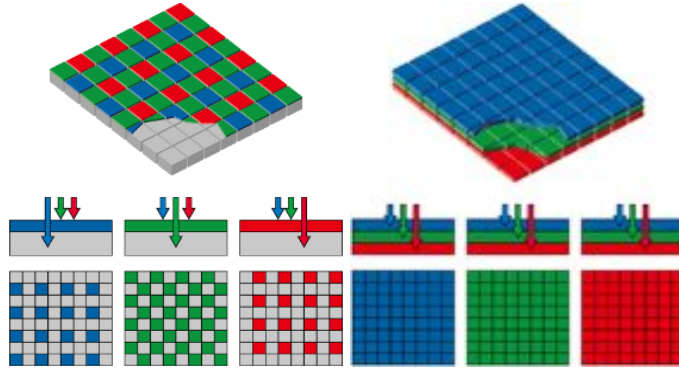


Figure 5.2: ©Foveon. Two different image sensor. **Left** Conventional CFA sensor. **Right** The layered Foveon X3[®] sensor.

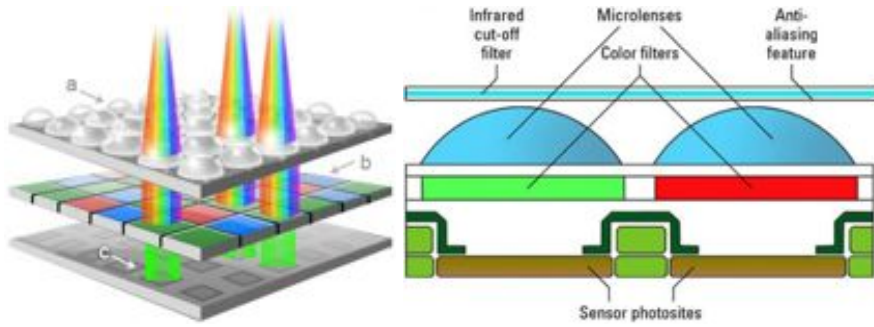


Figure 5.3: Optical system.

5.2.2 Noise model

The image acquisition starts when the external light hits the optical system. This latter is composed by an external lens, which filters infra-red wavelengths, and by a grid of micro-lenses on the photodiode which guide the light through its photo-site. These lenses produce geometric distortions and color aberrations. However, these effects do not enter in the noise model. The incident light passes through the optical system and reaches the CFA, a filter which allows only one color to pass. Although the CFA does not introduce any distortion, the CFA image requires an additional color interpolation step, called *demosaicing*. Once filtered by the CFA, the light beams reach the photo-detectors. Each photo-detector generates a current proportional to the number of incident photons [HK94] following approximately the law in Eq. (5.1). The ideal response of the photo-detector is altered by several disturbs which give a noisy response modeled as:

$$I_N = (1 + K) \cdot I + N_{DC} + N_I, \quad (5.2)$$

where K , N_{DC} and N_I are random variables which represent, respectively, the following noisy components [HK94]:

1. **photo response non-uniformity.** The manufacturing process is subject to small precision errors. These errors lead to small physical variations which change the photosite response. Therefore, if the sensor is uniformly illuminated, the charges stored in each photo-detector will be slightly different. This non uniformity is called PRNU. The variation of charge with respect to its nominal value I is modeled as a random variable K with zero mean and variance σ_K^2 (equal for all photo-detectors);
2. **dark current.** The silicon thermal energy produces free electrons which increase the photosite stored charge. This flux of free electrons is called *dark current*. The name recall the fact that in dark condition a photo-detector stores only electrons generated by thermal excitation.
3. **photonic noise.** The quantized nature of light leads to a certain degree of uncertainty on the stored charge in each photo-detector. The number of electrons actually received follows a Poisson distribution. The *photonic noise* is then modeled as a random variable with zero mean and variance equal to the ideal number of electrons that a photo-detector should receive (included the dark current). This yields to a *signal-dependent* noise N_I .

In order to obtain the RAW image, the sensor output is converted in tension by the output amplifier and then is quantized by the A/D converter. These operations introduce other disturbs [HK94], following the relation

$$D = A \cdot I_N + N_A + N_Q, \quad (5.3)$$

where A is the amplifier gain, N_A and N_Q are random variables with zero mean (independent from I_N) which represent, respectively, the amplification and quantization noise.

By the way, the PRNU is constant in time, since it depends on the sensor physical properties. Therefore, K can be considered deterministic, hence the noise model becomes

$$D = \underbrace{I_A + K \cdot I_A + E_{DC}}_{\mu} + \overbrace{N_S + N_T + N_A + N_Q}^N \quad (5.4)$$

N_G

where I_A is the amplified value of the light intensity, $E_{DC} = A \cdot E[N_{DC}]$ is the expected value of the (amplified) dark current, $N_T = A \cdot N_{DC} - E_{DC}$ is the thermal noise and $N_S = A \cdot N_I$ is the amplified photonic noise.

This model points out that the observed pixel D can be divided into a deterministic component μ , linearly dependent on the scene luminance, and a random component N with zero mean which account for the noise. This latter is the sum

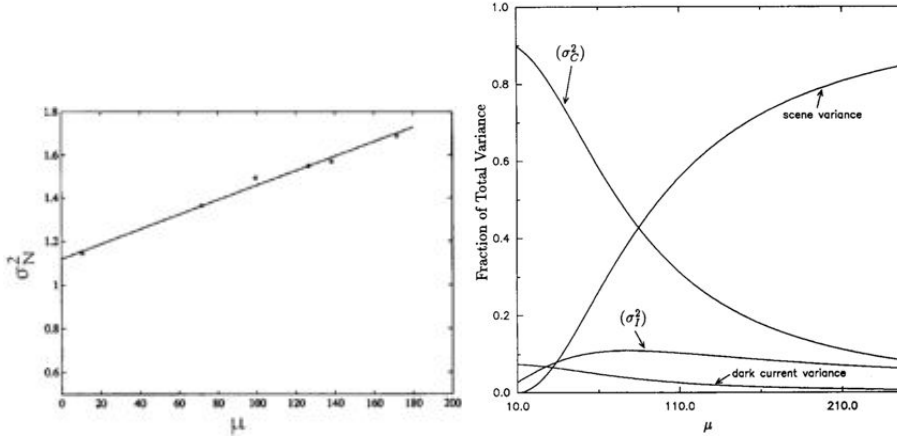


Figure 5.4: Properties of noise components. The pictures above are taken from [HK94].

of a Poisson random variable N_S and of a combination N_G which can be modeled as a Gaussian random variable [FTKE08]. Since N_S and N_G are statistical independent the overall variance of the additive noise is $\sigma_N^2 = \sigma_S^2 + \sigma_G^2$, where

$$\sigma_S^2 = A^2 \cdot \sigma_I^2, \quad \text{with} \quad \sigma_I^2 = (1 + K)I + E[N_{DC}] \quad (5.5)$$

$$\sigma_G^2 = \sigma_T^2 + \sigma_A^2 + \frac{q^2}{12} \quad (5.6)$$

Equations (5.5) and (5.6) show that the additive noise N is *signal-dependent*, as shown in Figure 5.4 (left). Moreover, Figure 5.4 (right) show that the overall noise is dominated by $K \cdot I$ in region of strong intensity ($I \gg 0$), with a variance $\sigma^2 = I^2 \sigma_K^2$, while is dominated by the Gaussian noise N_G in dark regions [HK94].

5.2.3 Parameters estimation

The parameters of the noise model can be found through a calibration procedure [HK94]. Under the hypothesis that D is linearly dependent on I_A , the calibration gives a good estimate of the dark current and of the PRNU. These can be used to subtract the deterministic components of the noise (*flat-fielding*)

$$D_C = \frac{D - \hat{E}_{DC}}{1 + \hat{K}} \approx I_A + \frac{N_S + N_G}{1 + \hat{K}} \quad (5.7)$$

The new image D_C is dominated by the photonic noise instead of $K \cdot I$, while the Gaussian noise still dominates dark regions. However, flat-fielding can be applied only on RAW images, since D is linearly dependent on I_A .

Flat-fielding is usually performed on astronomic images, where the PRNU is remarkable due to the long exposition time. Digital cameras simply removes dark

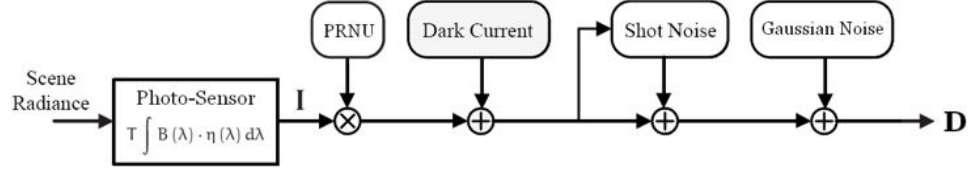


Figure 5.5: Linear noise model.

current [LFG06], which can be easily estimated capturing a completely dark frame. The final model is

$$D_C = D - \hat{E}_{DC} = \underbrace{(1 + K) \cdot I_A}_{I_K} + \underbrace{N_S + N_G}_N \quad \sigma_N^2 = a \cdot I_A + b \quad (5.8)$$

where $a, b \in \mathbb{R}$ represent the degree of dependence between signal and noise.

5.2.4 Demosaicing

The acquisition process ends with the *post-processing* step. This first removes the dark-frame. Then the RAW image is demosaiced in order to reconstruct the image R, G and B channels. However, here we want to point out that demosaicing introduces correlation into the noise model because of the interpolation between neighboring pixels. Indeed let us consider the bilinear interpolation, the simplest demosaicing approach. When using patches of white noise as input data for a CFA sensor the PSD of the demosaiced noise is shown in Figure 5.6 [AGPP09], resulting in a “colored”, *i.e.* correlated, noise. Since the red (or blue) data is more subsampled, the demosaicing treats both color planes differently, explaining the difference in PSD bandwidth. Figure 5.6 also shows that the noise energy is concentrated at low frequencies. This explains the tendency to favor smooth color patches in demosaiced images.

There exist a wide range of techniques for reconstructing the full color image from mosaiced image data. However, since all these techniques perform interpolation in some way, similar conclusion can be drawn for all demosaicing algorithms.

After demosaicing, color and gamma correction are usually performed [GN04, LSK⁺08]. Finally, the image is compressed.

5.3 Adapting PCkNN to CFA images

The image acquisition process introduces a noise component, as discussed in Section 5.2.2, which is very well approximated with an independent Gaussian stochastic process. However, contrary to the classical additive white Gaussian noise model with constant variance (used in Section 4.4 for comparison purposes), the variance of the digital camera noise can be modeled as an affine function of the signal intensity x :

$$\sigma^2(x) = \gamma x + \delta. \quad (5.9)$$

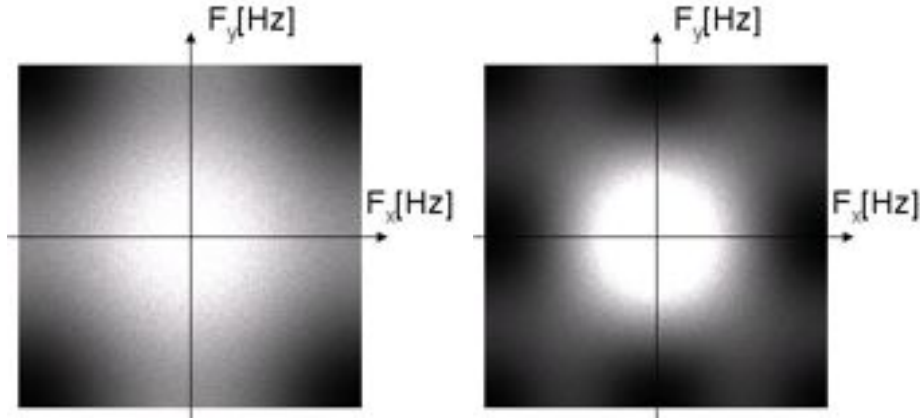


Figure 5.6: PSD of the green channel (left) and a red/blue channel (right) from the demosaiced white noise patch using bilinear demosaicing [AGPP09].

Therefore a denoising step should be included in the image acquisition pipeline. Moreover, most digital cameras acquire images using a single image sensor overlaid with a color filter array (CFA) which produces images with a single red, blue, or green component per pixel. The process of computing the missing 2 color components at each pixel is called demosaicing. Denoising can be performed before or after the demosaicing step. However, as explained in Section 5.2.4, demosaicing introduces correlation among neighborhood pixels and hence correlates the noise. The result is a “structured noise” which is not Gaussian and not independent anymore. Removing this noise is a harder task since algorithms usually rely on a hypothesis of independence. Therefore, we perform denoising before demosaicing.

In this Section we will focus on two main adaptation of the PCKNN algorithm presented in Chapter 4.

5.3.1 Intra-channel denoising

A naive and straightforward extension of PCKNN to color images is to treat the color image as the superposition of grayscale images, each of them representing the intensity in a color channel. We then applied the PCKNN algorithm to each color channel of the raw image, i.e., before demosaicing as sketched in Figure 5.7. Each channel acts as a down-sampled grey level image. Furthermore, we adapted our denoising algorithm to the varying variance model by making σ_w^2 in (4.21) equal to $\gamma\bar{x} + \delta$, \bar{x} being the average patch intensity. After denoising the image has been demosaiced.

We tested our algorithm on the 4416x3312-DxO Labs professional benchmark image of Figure 5.8 which is taken with a Canon G10 at ISO 1600. In this kind

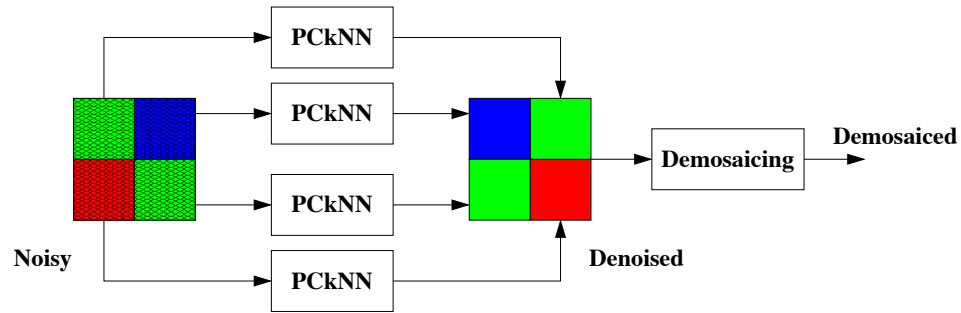


Figure 5.7: Intra-channel denoising scheme. Each color channel of the RAW image is denoised separately. The two Green of the Bayer grid are tread separately as well.

of experiments, given the lack of the original noiseless image, performance assessment is quite a challenging task. Furthermore, the comparison with other algorithms make no sense, since they rely on a constant noise variance hypothesis. However, a detailed and extensive comparison between PCKNN and the state-of-the-art has been developed in Chapter 4.

We set for all experiments a patch radius of 4 and a search radius of 20. The filtering parameter h follows the affine law of Eq.(5.9) with $\gamma = 116.49$ and $\delta = 618483$. Figures 5.9 to 5.13 show results on several images cropped out of the DxO image. Again, the denoised images have a very natural appearance without flattened regions. It is clear that strong noise reduction comes at the price, in general, of some loss of details. PCKNN seems to offer a good compromise between these contrasting needs.

However, PCKNN applied separately to each channel has a major drawback: it exploits only intra-channel correlation discarding any inter-channel dependency. This fact sometimes may cause a color mystification as shown in Figure 5.15 (left). To avoid this drawback, we deal with all channels jointly as discussed in the next section.

5.3.2 Inter-channel denoising

The processing presented in the previous Section treats each color channel independently. However, this might no be the best choice for several reasons. First of all, we process 4 down-sampled images and we do no consider the inter-channel correlation. This sometimes causes an evident preponderance of a color w.r.t. the others, with a next worsening of denoised image quality. In order to solve this problem, we process the RAW image taking into account all the color information. To do this, we have to modify the block-matching step (i.e. the k n.n. search). Indeed, when we look for similar patches in the search window, we have to keep in mind that each pixel carries an information on a specific color. Therefore, only a



subset of patches within the search window can be considered, in particular, those which have a coherent neighborhood with the reference patch (see Figure 5.14). The PCkNN algorithm is then modified as follows:

- These changes improve the denoised image quality removing colors artifacts. Such an enhancement is clearly visible in Figures 5.15 and 5.16 where the green color artifacts around the eyes has been reduced.



Figure 5.9: Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).

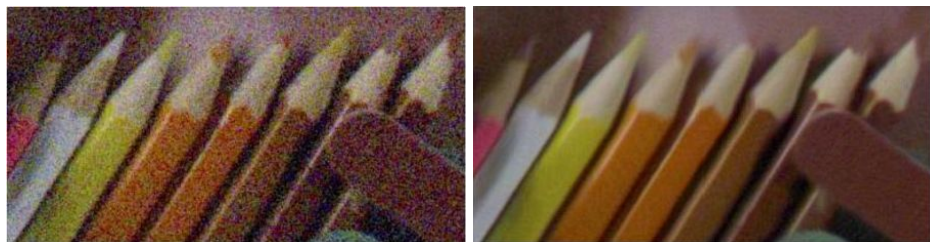


Figure 5.10: Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).



Figure 5.11: Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).

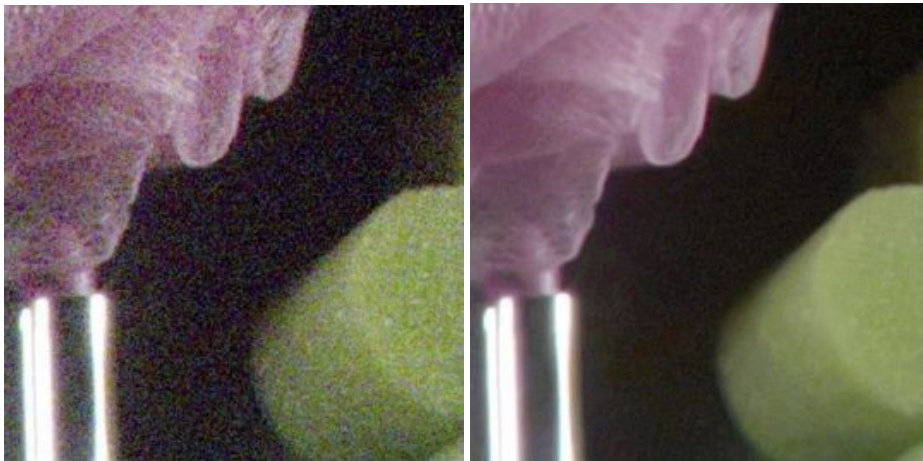


Figure 5.12: Professional benchmark image (Courtesy of DxO Labs). RAW image (left) and denoised image (right).



Figure 5.13: Comparison between RAW demosaiced images with no denoising (left) and with denoising (right). Particular of the image in Fig. 5.8. (Courtesy of DxO Labs).

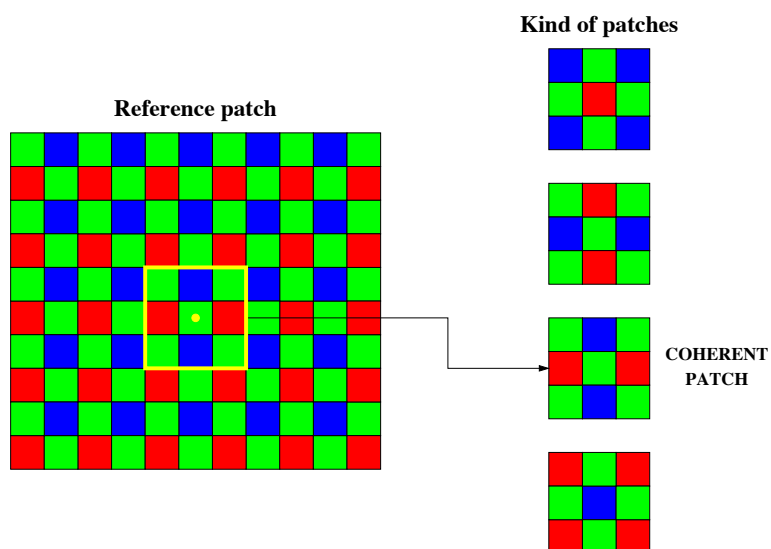


Figure 5.14: Inter-channel denoising. The block matching step is modified in order to catch only coherent patches within the search window.



Figure 5.15: Comparison between intra-channel (left) and coherent (right) block matching.



Figure 5.16: Comparison between intra-channel block-matching (left) and coherent block-matching (right). Particular of the eyes. (Courtesy of DxO Labs).

Acknowledgment

We would like to thank DxO Labs for providing the raw test image of Figure [5.8](#) and the demosaicing software.

Chapter 6

SAR despeckling

6.1 Introduction

Synthetic Aperture Radar (SAR) images are inherently affected by multiplicative speckle noise, which is due to the coherent nature of the scattering phenomena. Even though speckle carries itself information about the illuminated area, it degrades the appearance of images and affects the performance of scene analysis tasks carried out by computer programs (*e.g.*, segmentation and classification) or even by human interpreters [OQ04]. To counter this problem, users resort often to the multilook technique, which amounts to incoherently averaging a certain number (defined by the *number of Looks*) of independent images, thus reducing noise intensity, but often at the cost of a clear loss in image resolution. Therefore, it is certainly preferable to develop suitable filtering techniques, which reduce noise significantly but, at the same time, preserve all the relevant scene features, such as radiometric and textural information.

6.2 State of the art

6.2.1 Spatial domain techniques

Some of the early speckle reduction techniques, *e.g.* [AL84, YC86], use the so-called homomorphic approach, taking the log of the data, so as to obtain a more tractable additive model, and then applying some well-known method drawn from the AWGN (additive white Gaussian Noise) denoising literature. Such a “lazy” approach has the undeniable merit of simplicity but turns out to be largely suboptimal as it neglects some basic properties of speckle. In fact, the log-transformed speckle is definitely non-Gaussian, except for the case of a large number of looks [AA76], and has non-zero mean, in general, a bias to be corrected before any other processing. More important, the logarithm changes radically the data dynamics, leading to unavoidable radiometric distortions during the denoising process.

In the same period, more ambitious techniques are proposed [Lee80] [Lee81]

[FSSH82] [KSSC85] [TLB88] [LTN90] which tackle despeckling in the original domain (non log-transformed), as a statistical estimation problem, based on the multiplicative speckle model. All such techniques operate in the spatial domain with linear filters developed under a MMSE (minimum mean-square error) approach and simplifying the multiplicative noise model in various ways, *e.g.* through its linear approximation [Lee80] or recasting it as a signal-dependent additive noise model [KSSC85]. These early papers make already clear that some form of local adaptivity is necessary to account for the non-stationarity of the image: the intense smoothing required to reduce speckle in homogeneous areas cannot be applied in edge and textured regions lest important structural information gets lost. Contextual information is hence taken into account, in various ways [Lee81] [TLB88] [LTN90] to adapt the filters to local image behavior.

In [KSSC87] [LNTH90] [NLT91] the simple MMSE estimation is replaced by the more sophisticated and promising Bayesian MAP approach which, however, brings with it the problem of providing an accurate statistical description of the SAR image. Several competing models have been proposed in the literature, *e.g.*, the Gamma distribution considered in [LNTH90] which lead to the GMAP algorithm. As a matter of fact, this is a challenging and still open problem, and no parametric model, to date, seems able to account for the variety of situations encountered in SAR images [Gao10]. In addition, parameter estimation is by itself a tricky problem, being especially sensitive to the volume of available data (think of local estimation windows), with all inaccuracies translating in artifacts and artificial biases in the reconstructed scene [Tou02].

6.2.2 Wavelet based techniques

The diffusion of wavelets in the signal processing community, in the early 90's, opens the way to a new generation of despeckling techniques operating in the transform domain. Indeed, wavelet shrinkage image denoising, based on AWGN hypothesis, has been one of the first and most successful applications of this transform. Wavelet shrinkage can be readily applied to SAR despeckling after an homomorphic transformation which leads to an additive (though non-Gaussian) noise model. This approach is followed in [HJM⁺94], and again in [GJ97], where both hard and soft thresholding are tested. Despite the empirical selection of the threshold and the intrinsic limits of the homomorphic approach experiments on both synthetic and real SAR images show already a clear performance gain w.r.t. spatial-domain adaptive filters, especially in the absence of multilook. Further improvements are obtained by optimizing the shrinkage parameter through a statistical Bayesian approach. Again, if a MAP approach is chosen, an appropriate model of the log-transformed reflectance in the wavelet domain is needed, like the alpha-stable distribution proposed in [ATB03], the normal inverse Gaussian used in [SE04], or the simpler Cauchy distribution adopted in [BAS07] which leads to the derivation of both a MAP and a MMAE (minimum mean absolute error) Bayesian estimators, the latter proving superior in most experiments.

In order to overcome the drawbacks of the homomorphic approach several researchers resort to the additive signal-dependent speckle model in the wavelet domain. In [HLF02], inspired by [OE99], a low complexity MMSE estimation procedure is proposed to derive the shrinkage factor for each wavelet coefficient. In [FA02] a multiscale local coefficient of variation is defined to handle non-stationarities, while [DPCL04] proposes a modified ratio edge detector to the same end. Also in this setting, of course, a MAP estimation approach can be considered as in [SGJ01] where a Gamma distribution is used for the underlying radar reflectivity, or [FTA06] where a generalized Gaussian with spatially-varying parameters is considered. This latter work is further improved in [FTA08] where wavelet coefficients are classified based on their level of heterogeneity, so as to incorporate this information in the filtering procedure.

Although hardly accountable in such a short survey, all the wavelet-domain techniques, just as it happened for spatial-domain techniques, try to embody some forms of spatial adaptivity in the filtering process in order to better preserve image boundaries and textures. Therefore, the suitable use of contextual information keeps being a topic of central importance.

6.2.3 Non-local techniques

Given these premises, the “non-local” approach, recently proposed in [BCJ05] for AWGN denoising, looks like a potential breakthrough. The basic idea is to take advantage of the self-similarity commonly present in natural as well as SAR images: certain image patches tend to repeat over and over, with minor modifications, throughout the scene, a circumstance to be exploited in view of the boundary preservation goal. In the Non-Local Means (NLM) algorithm [BCJ05] filtering is carried out, as usual, through the weighted mean of all the pixels in a certain neighborhood; the weight associated with each given pixel, however, depends not on its geometrical distance from the target pixel but on its similarity with it, measured by the mean square error between the patches surrounding the selected pixel and the target. This principle has inspired several extensions, among which the evolution towards a multipoint rather than pointwise filtering, as proposed in the Block-Matching 3D (BM3D) algorithm [DFKE07b] where the non-local approach is combined with wavelet shrinkage and Wiener filtering in a two-step process. At present, BM3D can be arguably [KFEA10] considered the state of the art for AWGN denoising.

The NLM algorithm has been readily extended to SAR despeckling [CHKB08, HJL09, CLF09] with suitable modifications aimed at taking into account the problem peculiarities. The Probabilistic Patch-Based (PPB) algorithm [CLF09] is especially interesting, both for its theoretical contribution, with the development of a similarity measure well suited to SAR images, and for the excellent performance on test images.

Based on the conceptual path described above, and the related experimental evidence, in this chapter we go one step further and propose a SAR-oriented version

of BM3D, with the obvious goal of replicating the competitive advantage it enjoys in the AWGN context. By taking into account the peculiar features of SAR images in the original domain we derive a new despeckling algorithm, called SAR-BM3D, which exhibits an objective performance comparable or superior to all competing techniques on simulated speckled images, and guarantees a very good subjective quality on actual SAR images.

6.3 The BM3D algorithm and its SAR-oriented version

The despeckling algorithm we propose here can be seen as a SAR-oriented version of BM3D, since we use the same algorithmic structure as the original BM3D but modify most of the individual processing steps in order to take into account the peculiarities of SAR data. Therefore, ideas and tools come from both the AWGN denoising and the SAR despeckling fields. In the following of this Section, we justify and briefly outline the major modifications that lead to its SAR-oriented version, leaving for the next Section all detailed developments.

BM3D was developed in AWGN hypotheses, and using it with SAR images, characterized by multiplicative noise, makes little or non sense. Of course, one can always resort to the homomorphic approach, converting the multiplicative noise to additive, and using BM3D on the transformed data, before going back to the original domain. Indeed, this simple approach provides sometimes surprisingly good results. Nonetheless, the log-transform modifies the dynamics of the data, introducing unwanted artifacts, and the noise remains markedly non-Gaussian (especially for the single-look case) with a sure loss of performance. Therefore, in this work we decided to use the BM3D filtering structure because of its compelling rationale, but also to adapt it to the specific characteristics of the data, modifying the various processing steps so as to take into account the actual statistics of SAR noise. To this end we introduce two major modifications.

First of all, we adapt the criterion used to collect blocks in the 3D groups to the actual data statistics. For each reference block, BM3D looks (in a suitable search area) for those blocks which are closest to the reference in terms of Euclidean distance. In the AWGN setting this makes perfect sense because a smaller Euclidean distance corresponds to a higher likelihood that the two signal blocks (without noise) be equal, which is what the collaborative filtering needs. However, once the noise statistics change, as happens with SAR images, the Euclidean distance loses its significance and we need a different *ad hoc* similarity measure in order to keep identifying the signal blocks that are more likely to be equal to the reference one.

Our second modification stems from the same line of reasoning and concerns the collaborative filtering itself. In fact, hard thresholding is a reasonable choice in AWGN, since it is the minimax estimator of the uncorrupted group [Don95], but this is no longer true with multiplicative noise where a more suitable wavelet shrinkage strategy can be devised. In this work, in particular, we adopt the local linear minimum mean-square error (LLMMSE) solution, discussed in depth in

next Section. Together with this “compelling” modification, we introduce a further change consisting in the use of the undecimated WT (UDWT), aimed at obtaining more reliable estimates in the first step, especially needed in the presence of such intense noise. UDWT is not without drawbacks: it is computationally intensive, gives rise to correlated coefficients, and is not unitary, thus uncoupling optimality in the original and transform domain. Nonetheless, it has been shown experimentally [LGO96], and justified theoretically [Ela06], to provide better results than nonredundant WT, and has already been successfully applied to LLMMSE shrinkage in the case of speckle [FA02, ZBW05].

6.4 Proposed SAR-oriented modifications in detail

In this section we analyze in some depth the modifications adopted in BM3D in order to deal effectively with speckled SAR images. Under the hypothesis of fully developed speckle, the observed backscattered signal, $z(n)$ ¹, can be expressed as

$$z(n) = x(n)u(n) \quad (6.1)$$

where $x(n)$ is the noise-free reflectance and $u(n)$ the speckle, characterized by a unitary mean and independent of x . Equation (6.1) can be rewritten in terms of signal plus signal-dependent additive noise $v(n)$,

$$z(n) = x(n) + [u(n) - 1]x(n) = x(n) + u'(n)x(n) = x(n) + v(n), \quad (6.2)$$

It is worth noting that, due to the independence of x and u , and the fact that u' has zero mean, the additive noise v is zero-mean and appears to be uncorrelated with x . In the following, starting from the above model, and with the further assumptions that both signal and noise are spatially uncorrelated, we will first introduce the new similarity measure, and then the LLMMSE shrinkage in the transform domain for the two steps of the algorithm.

6.4.1 Block similarity measure

The non-local approach can be regarded as an attempt (limited by complexity and data scarcity) to carry out truly statistic, as opposed to spatial, averages. Assuming one is able to collect an arbitrary number of blocks with the same signal component and differing only in the noise realization, one can easily remove most noise (all of it in the limit) with simple filtering operations. Therefore, the block matching phase of BM3D aims at locating the blocks most likely to have the same signal component as the reference which, in AWGN hypotheses, coincide with those having the smallest Euclidean distance from the reference in the data space.

Outside of the AWGN realm, the Euclidean distance is not optimal anymore, but one can follow the same probabilistic principle to devise a new similarity measure based on the actual noise distribution. This is done for example in [ML07]

¹For notational compactness we use a single argument to indicate spatial location.

and in [CLF09] where a function of the intensity fluctuations in the image is used, which represents the likelihood that two intensity observations correspond to the same noise-free scene radiance. Mathematically, given two observed values $z(s)$ and $z(t)$, it results

$$p[z(s), z(t)|x(s) = x(t)] = \int_D p[z(s)|x(s) = \alpha] p[z(t)|x(t) = \alpha] p(\alpha) d\alpha \quad (6.3)$$

where $x(s)$ and $x(t)$ are the corresponding values of the noise-free signal, defined over the domain D , $p(\cdot)$ indicates a probability density function, and we have assumed $z(s)$ and $z(t)$ to be conditionally independent given x . This expression further simplifies to

$$p[z(s), z(t)|x(s) = x(t)] \propto \int_D p[z(s)|x(s) = \alpha] p[z(t)|x(t) = \alpha] d\alpha \quad (6.4)$$

if we assume, lacking any prior knowledge, $p(\cdot)$ to be uniform over D .

Considering that for an L -look amplitude SAR image speckle can be modeled [OO04] [XPU02] by a square-root gamma distribution with order L

$$p(z|x) = \frac{2}{\Gamma(L)} \left(\frac{L}{x}\right)^L z^{2L-1} \exp\left(-L\frac{z^2}{x}\right) \quad z \geq 0 \quad (6.5)$$

equation (6.4) reads as

$$p[z(s), z(t)|x(s) = x(t)] \propto \int_0^\infty \frac{4L^{2L}}{\Gamma^2(L)\alpha^{2L}} [z(s)z(t)]^{2L-1} \exp\left\{-\frac{L}{\alpha} [z^2(s) + z^2(t)]\right\} d\alpha \quad (6.6)$$

with the integral equal to [CLF09],

$$4L \frac{\Gamma(2L-1)}{\Gamma^2(L)} \left[\frac{z(s)z(t)}{z^2(s) + z^2(t)} \right]^{2L-1} \quad (6.7)$$

To translate this result into a manageable block similarity measure we must rewrite equation (6.3) with vectors drawn from the blocks B_s and B_t in place of scalars, and assume again the conditional independency of the observed values given the noise-free signal. Then we define the block similarity measure as

$$\begin{aligned} d[z(B_s), z(B_t)] &= -\log \left\{ \prod_k p[z(s+k), z(t+k)|x(s+k) = x(t+k)] \right\} \\ &= -\log \left\{ \prod_k 4L \frac{\Gamma(2L-1)}{\Gamma^2(L)} \left[\frac{z(s+k)z(t+k)}{z^2(s+k) + z^2(t+k)} \right]^{2L-1} \right\} \end{aligned} \quad (6.8)$$

where $z(B_s)$ is the vector of observed values drawn from block B_s , s is the reference pixel of the block and k is used to scan the whole block. Finally, discarding

the constant term, the block similarity measure reduces to [CLF09]

$$d_1[z(B_s), z(B_t)] = (2L - 1) \sum_k \log \left[\frac{z(s+k)}{z(t+k)} + \frac{z(t+k)}{z(s+k)} \right] \quad (6.9)$$

where the subscript 1 indicates that this measure is used in the first step.

In the second step, in fact, the similarity measure must take into account the additional information provided by the first step, that is a coarse estimate of the noiseless signal \hat{x} . Therefore, following [CLF09], where this approach is used for iterative denoising in a Bayesian framework, we define the similarity measure in the second step as:

$$d_2[z(B_s), z(B_t)] = \sum_k \left[(2L - 1) \log \left(\frac{z(s+k)}{z(t+k)} + \frac{z(t+k)}{z(s+k)} \right) + L \frac{|\hat{x}(s+k) - \hat{x}(t+k)|^2}{\hat{x}(s+k)\hat{x}(t+k)} \right] \quad (6.10)$$

6.4.2 Group shrinkage

The hard thresholding used by BM3D in the first step is a reasonable choice in the AWGN context, but not anymore in the presence of SAR speckle. Therefore, we address the shrinkage problem in the framework of statistical estimation, with the noise model of (6.2), and look for the optimum linear estimator in the minimum MSE sense. It is worth emphasizing that WT and shrinkage take place on each 3D group individually, and hence, in this subsection, the group will be our basic data unit. After the linear wavelet transform, we obtain

$$\mathbf{Z} = \mathbf{X} + \mathbf{V} \quad (6.11)$$

where we have used capital letters for the transformed data, and boldface to indicate the vectors formed by all the coefficients of the group. Under the constraint of linearity, the optimal MMSE estimator is [Kay93]

$$\hat{\mathbf{X}} = E[\mathbf{X}] + (\mathbf{C}_{XZ})(\mathbf{C}_Z)^{-1}(\mathbf{Z} - E[\mathbf{Z}]) \quad (6.12)$$

where \mathbf{C}_Z is the covariance matrix of \mathbf{Z} , and \mathbf{C}_{XZ} the cross-covariance matrix of \mathbf{X} and \mathbf{Z} . Since signal and noise are uncorrelated in the spatial domain they remain uncorrelated also after the linear transform, with noise still zero-mean, therefore (6.12) simplifies to [Kay93]

$$\hat{\mathbf{X}} = E[\mathbf{X}] + (\mathbf{C}_X)(\mathbf{C}_X + \mathbf{C}_V)^{-1}(\mathbf{Z} - E[\mathbf{Z}]) \quad (6.13)$$

If we further assume that the covariance matrices are diagonal, the estimation acts separately on each coefficient of the group

$$\hat{X}(i) = E[X(i)] + \frac{\sigma_X^2(i)}{\sigma_X^2(i) + \sigma_V^2(i)}(Z(i) - E[Z(i)]) \quad (6.14)$$

and we obtain a local LMMSE filter, which is indeed an adaptive Wiener filter in the transform domain [HLF02] [OE99].

The hypothesis that both signal and noise coefficients are uncorrelated is quite reasonable when a wavelet transform is used, since it tends to decorrelate the data, and in fact the local Wiener filter has been used extensively in the AWGN context [GSB97] [MKR99] [JFWJ03] [Kaz03] providing a performance typically superior to that of classical thresholding. Of course, such hypothesis does not hold anymore with the UDWT, which is non-orthogonal and introduces some redundancy among the coefficients. Nonetheless, even in this case such an assumption is typically convenient, as the cost for the imperfect modeling is more than compensated by the opportunity to use a local estimator and by the significant reduction in complexity.

Since the shrinkage is applied only to the coefficients of the detail subbands, which can be reasonably considered to have zero mean, (6.14) becomes eventually

$$\hat{X}(i) = \frac{E[X^2(i)]}{E[X^2(i)] + E[V^2(i)]} Z(i) \quad (6.15)$$

or equivalently

$$\hat{X}(i) = \frac{E[Z^2(i)] - E[V^2(i)]}{E[Z^2(i)]} Z(i) \quad (6.16)$$

The problem now comes down to the estimation of the second order moments in the above formulas. In the literature, working with large images, these quantities are typically computed by means of sliding-window averages running on the various detail subbands of the wavelet transform. In our case, however, we deal with rather small groups (*e.g.*, $8 \times 8 \times 16$ coefficients) which, after an ordinary WT, would be decomposed in tiny detail subbands, making any such estimate totally unreliable. This is why we turn to UDWT for the first shrinkage step, as it provides us with subbands large enough to carry out reliable estimates.

First step

to carry out the estimates required in (6.16), we assume that the second order statistics of the observed signal, given the limited size of the 3D group, are constant over the whole group in the spatial domain, and over each subband in the transform domain. Therefore, we have

$$E[Z^2(i)] = \langle Z^2 \rangle_{\text{SB}(i)} = \frac{1}{|\text{SB}(i)|} \sum_{j \in \text{SB}(i)} Z^2(j) \quad (6.17)$$

where $\langle \cdot \rangle_{\text{SB}(i)}$ indicates average over the subband comprising the i -th coefficient.

As for the noise, this problem was addressed in [FA02] with reference to the UDWT case, obtaining

$$E[V^2(i)] = \frac{\sigma_u^2}{(1 + \sigma_u^2)} \sum_k h^2(k) E[z^2(i - k)] \quad (6.18)$$

where h is the subband equivalent filter, σ_u^2 a known parameter depending on speckle format and number of looks [XPU02], and k spans a 7×7 local window. Adapting the formula to our case we readily obtain

$$E[V^2(i)] = \frac{\sigma_u^2}{(1 + \sigma_u^2)} \langle z^2 \rangle_G \quad (6.19)$$

where $\langle \cdot \rangle_G$ indicates the average over the whole group. It is worth observing that the increase in complexity due to the use of an undecimated transform is compensated by the use of subband-wise and group-wise, as opposed to sliding-window, averages.

Eventually we have

$$\hat{X}_1(i) = \max \left(0, \frac{\langle Z^2 \rangle_{\text{SB}(i)} - \frac{\sigma_u^2}{(1 + \sigma_u^2)} \langle z^2 \rangle_G}{\langle Z^2 \rangle_{\text{SB}(i)}} \right) Z(i) \quad (6.20)$$

where the subscript indicates first step, and the max operator accounts for a possible sign inversion due to estimation errors.

Second step

the collaborative filtering in the second step has also a LLMMSE nature with the major difference that now an estimate of the noiseless signal coefficient is already available. As a first consequence, we can use simpler non redundant transforms, thus reducing complexity. In addition, with reference to (6.15), we estimate $E[X^2(i)]$ simply as $\hat{X}_1^2(i)$ where $\hat{X}_1(i)$ is the coefficient computed from the partially denoised signal $\hat{x}_1(n)$ provided by the first step. This amounts to using the empirical Wiener filtering, proposed in [GSB97] for the AWGN context and also used in the original BM3D [DFKE07b]. Finally, to estimate $E[V^2(i)]$ we assume it is constant over the group, and exploit again the first-step estimate $\hat{X}_1^2(i)$ carrying out an average over the whole group of the difference between the observed coefficient and its noiseless estimate as

$$E[V^2(i)] = \langle V^2 \rangle_G = \frac{1}{|G|} \sum_{i \in G} [Z(i) - \hat{X}_1(i)]^2 \quad (6.21)$$

In conclusion the second-step estimate reads

$$\hat{X}_2(i) = \frac{\hat{X}_1^2(i)}{\hat{X}_1^2(i) + \langle V^2 \rangle_G} Z(i) \quad (6.22)$$

6.4.3 Aggregation

To conclude the description of our algorithm let us focus on the aggregation phase. Since a given pixel can be included in more than one group, and hence estimated

several times, each time with a possibly different value, such values must be averaged using suitable weights. We follow here the approach of [DFKE07b] where the various estimates are weighted according to their presumed reliability, related in turn to the average noise power of the group after shrinkage. In formulas, the weight w_G associated with the estimate provided from group G is

$$w_G \propto \frac{1}{\langle V^2 \rangle_G \langle S^2(i) \rangle_G} \quad (6.23)$$

where $S(i)$ is the shrinkage factor for the i -th coefficient of the group. The same formula is used in both steps, while the expressions for $\langle V^2 \rangle_G$ and for the shrinkage factors are obviously different.

6.5 Experimental results

In SAR image denoising, given the lack of the original noiseless signal, performance assessment is quite a challenging task. Different indicators have been proposed to measure smoothness of smooth areas as well as sharpness of edges and details, but they are largely empirical, and provide little insight about how to balance image cleanness and preservation of diagnostic information. Therefore, following an approach widespread in the literature [BAS07], [FA02], [FTA06], [CLF09], we start with experiments carried out on optical images corrupted by simulated speckle, obtaining objective performance figures which allow a sound comparison among different denoising algorithms. Then, in the last part of the Section, we discuss experiments concerning actual SAR images.

6.5.1 Gold standards and parameter setting

We compare the proposed technique with three state-of-the-art despeckling algorithms: the spatially adaptive wavelet homomorphic shrinkage algorithm (SA-WBMAE) [BAS07], the wavelet-based MAP filtering algorithm (MAP-S) [FTA08], and the Probabilistic Patch Based (PPB) nonlocal filter [CLF09], all briefly described in Section I. Such techniques have been chosen because of their competitive performance and (not least) for the availability of software code to run the experiments. Experimental results have in fact been obtained by using the Authors' own code available online, or run by the Authors themselves on our test images. We also include in the comparison two state-of-the-art AWGN techniques used in a homomorphic setting, the AWGN versions of PPB (H-PPB) and BM3D (H-BM3D), which are especially interesting for images with a large number of looks. Finally, we consider also the well-known Frost filter [FSSH82] which, although pretty aged, is a *de-facto* standard, included in many image processing software packages, and used routinely by photo-interpreters of military and civil space agencies.

For all these algorithms, if not stated otherwise, the free parameters are set as suggested in the reference papers. As for the proposed SAR-BM3D algorithm,

	Lena				Boat			
	L=1	L=2	L=4	L=16	L=1	L=2	L=4	L=16
<i>Noisy</i>	12.11	14.90	17.84	23.79	11.76	14.52	17.47	23.45
<i>Frost</i>	19.72	23.06	26.03	30.48	19.18	22.21	24.81	28.08
<i>SA-WBMMAE</i>	25.10	27.33	29.04	32.47	23.36	25.11	26.76	29.98
<i>MAP-S</i>	26.40	28.08	29.72	33.16	23.93	25.40	27.06	30.51
<i>PPB</i>	26.68	28.45	29.88	32.70	23.97	25.51	26.95	29.83
<i>SAR-BM3D</i>	27.91	29.61	31.22	34.18	25.49	26.86	28.41	31.45
<i>H-PPB</i>	25.31	27.85	29.75	32.82	23.40	25.37	26.98	29.98
<i>H-BM3D</i>	26.46	29.21	31.26	34.52	24.49	26.67	28.56	31.76

Table 6.1: PSNR results for Lena and Boat.

in the first step we use a Daubechies-8 UDWT transform with a maximum-level decomposition, and fixed groups of dimension $8 \times 8 \times 16$. Just like in BM3D, the computational burden is reduced by using a relatively small search area, 39×39 , and by selecting reference blocks only on every third row and column. Similar choices apply to the second step except for the transform, which is a spatial DCT followed by a Haar DWT along the blocks with a maximum-level decomposition, and for the group dimensions that grow to $8 \times 8 \times 32$.

6.5.2 Results with simulated speckle

In order to obtain reliable results, we considered a variety of sources, including some general-purpose images commonly used in the AWGN denoising literature, some aerial photographs which better resemble SAR images in terms of scene structure, and a synthetic image, first introduced by Lee in [LJDaA94], in order to test structure preservation. SAR-like images are obtained by multiplying optical images by simulated white speckle in amplitude format (square root intensity model) [XPU02] with pdf's corresponding to the cases of 1, 2, 4 and 16 looks.

In Tab. 6.1 we report results for two general-purpose 512×512 -pixel images, Lena and Boat (Fig. 6.1), widely used as benchmark in the denoising community, for $L=1, 2, 4$, and 16 looks. The best PSNR for each case is put in boldface for the sake of clarity. Although the Frost filter does already a good job, with an improvement of several dBs w.r.t. the noisy image, more sophisticated techniques prove definitely superior, especially for the most critical case of $L = 1$ (no multilook), where an additional gain of 6-8 dB is achieved. SAR-BM3D provides consistently the best performance, gaining from 1 to 1.5 dB w.r.t. PPB which looks as the second best. The only exception to this rule is represented by the homomorphic version of BM3D (H-BM3D) which, for large L , is slightly superior even to the proposed dedicated technique. As a matter of fact, the two algorithms based on the homomorphic approach exhibit quite a similar behavior, becoming more and more competitive with increasing L . This is not surprising, however, since the noise in the log image tends to become Gaussian as L increases, in which case a general-purpose AWGN denoising algorithm in the homomorphic setting

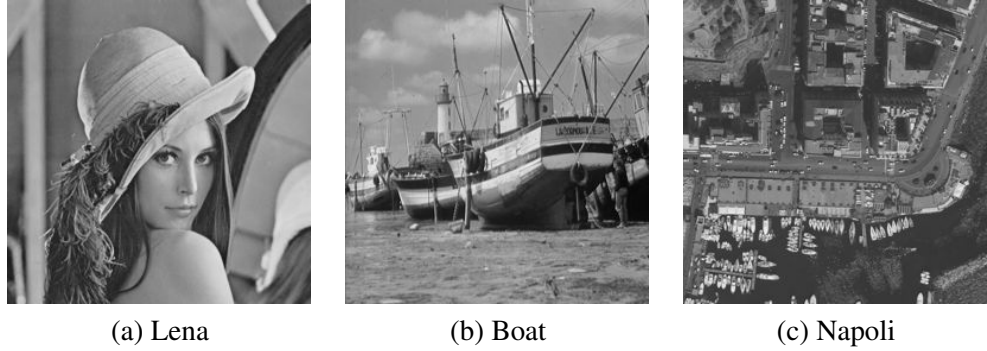


Figure 6.1: Original images used in the experiments.

	L=1	L=2	L=4	L=16
<i>Noisy</i>	14.28	17.05	19.99	25.98
<i>Frost</i>	20.81	23.08	24.64	26.03
<i>SA-WBMAE</i>	22.05	23.41	24.78	27.74
<i>MAP-S</i>	22.09	23.45	25.03	29.11
<i>PPB</i>	21.33	22.58	24.38	28.17
<i>SAR-BM3D</i>	23.57	25.02	26.64	30.14
<i>H-PPB</i>	19.63	22.09	24.50	28.50
<i>H-BM3D</i>	22.89	24.68	26.36	29.98

Table 6.2: PSNR results for Napoli.

becomes a perfectly sensible choice. In the absence of multilook, instead, the proposed SAR-dedicated algorithm provides a clear advantage over the homomorphic approach. Fig. 6.2 shows the zoom of the denoised images provided by all algorithms for Lena with $L = 1$. It is clear that strong noise reduction comes at the price, in general, of some loss of details, most notable in the PPB image. SAR-BM3D seems to offer the best compromise between these contrasting needs.

Tab. 6.2 gives results for a 512×512 -pixel section of an aerial photo showing a prevalently urban scene in the city of Naples (Italy) (Fig. 6.1). The general behavior of the PSNR is quite similar to that of the previous experiments, except for the gap between SAR-BM3D and the reference techniques which grows slightly larger. It is worth taking a closer look, instead, at the zoom of denoised images shown in Fig. 6.3. Here, given the wealth of fine details in the original, the smoothing provided by some filtering techniques is particularly annoying, with many individual objects, both cars and boats, merged together or even lost in the background. SAR-BM3D instead, and to a lesser extend H-BM3D and PPB (with a modified setting proposed by the Authors) provide an acceptable balance between smoothing and detail preservation.

Tab. 6.3 finally presents results for the synthetic 256×256 -pixel Target image, reported for the first time in [LJDaA94], which contains points and strips of increasing dimensions. The point targets have size of 1×1 , 3×3 , and 5×5 pixels,

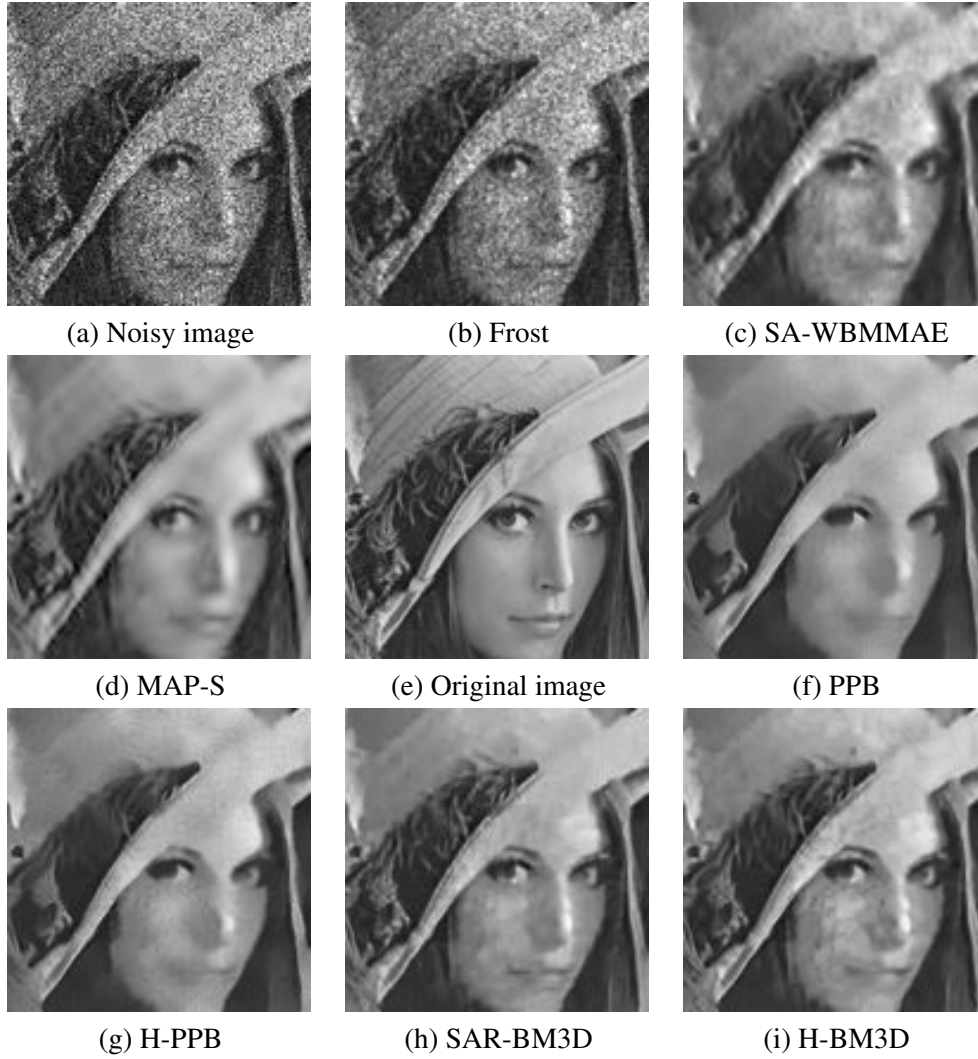


Figure 6.2: Zoom of filtered images with the various techniques for Lena corrupted by one-look speckle.

while the strip width goes from 1 to 13 pixels in 2-pixel increments. All target pixels have value 120, while the background pixels have value 60. In terms of PSNR, the most significant difference w.r.t. previous experiments is the larger gain of the BM3D-based techniques over the others. This is probably due to the block-wise processing used in BM3D, which allows to treat coherently neighboring pixels. Filtered images are shown in Fig. 6.4. To test feature preservation, we decided to process them with a simple detector (a gaussian filter followed by a threshold operator), declaring the detection of a target whenever an above-threshold region superimposed the target. Results, in terms of number of detected features, are re-

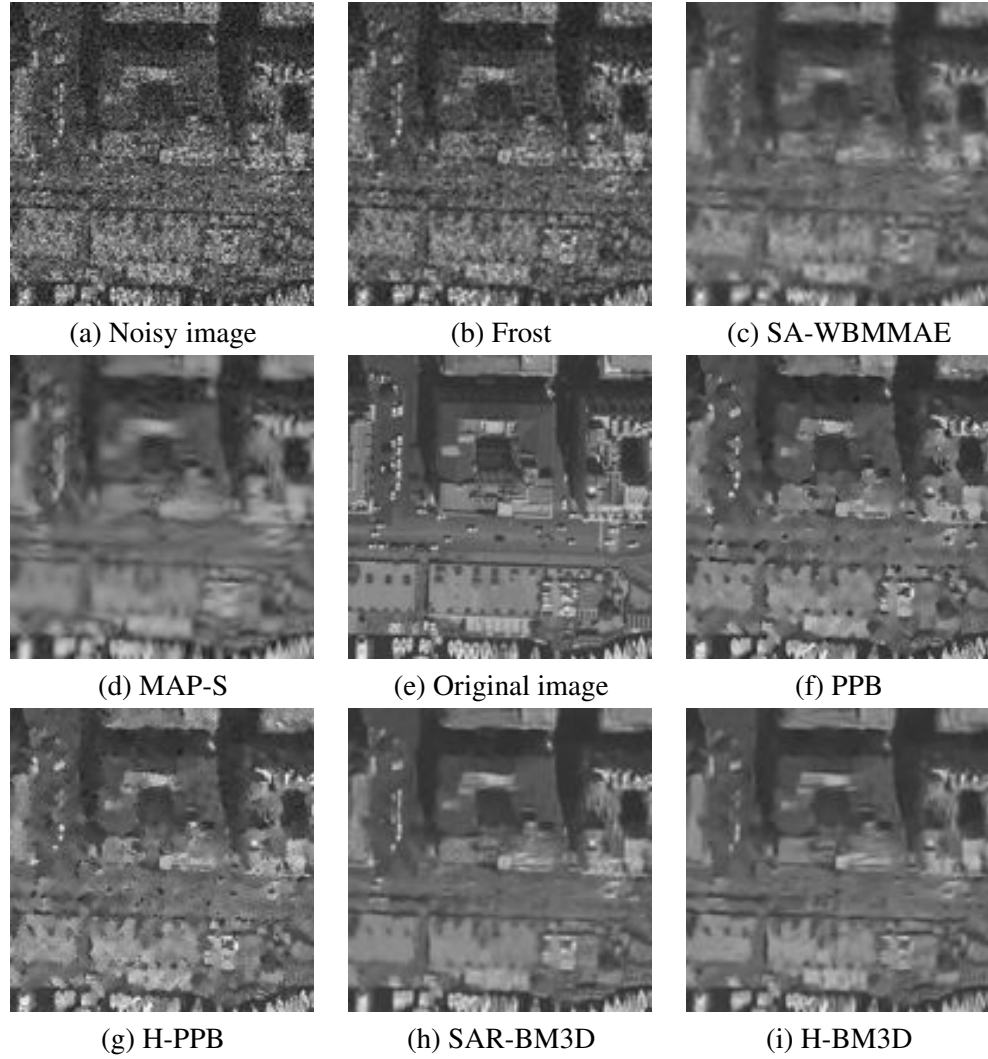


Figure 6.3: Zoom of filtered images with the various techniques for Napoli corrupted by one-look speckle.

ported again in Tab. 6.3, and confirm what visual inspection also suggests, namely, that all filters behave about equally well on point targets (lost) and bars (saved), but only SAR-BM3D and Frost save all 5×5 and most 3×3 targets, with the latter generating however an inordinate amount of false alarms (F.A.).

6.5.3 Results with actual SAR images

For this set of experiments we considered five single-look and one 6-look TerraSAR-X images in amplitude format taken over Rosenheim (Germany) and

	L=1	L=2	L=4	L=16	bars	5×5	3×3	1×1	F.A.
<i>Noisy</i>	17.66	20.45	23.41	29.39	–	–	–	–	–
<i>Frost</i>	24.78	27.62	29.85	32.23	7	7	7	2	~ 100
<i>SA-WBMAE</i>	28.02	29.78	31.28	35.17	6	7	1	0	~ 10
<i>MAP-S</i>	28.84	30.28	31.98	36.91	7	7	5	0	~ 10
<i>PPB</i>	29.78	32.56	35.46	40.70	6	5	0	0	0
<i>SAR-BM3D</i>	31.93	35.80	39.80	45.75	6	7	5	0	0
<i>H-PPB</i>	27.71	31.33	34.57	40.16	5	4	0	0	0
<i>H-BM3D</i>	30.45	34.76	38.73	45.05	6	5	2	0	~ 10

Table 6.3: PSNR and detection results in terms of number of identified features for Target.

Toronto. Fig. 6.6 shows 512×512 -pixel sections drawn from such images covering heterogeneous sceneries: urban areas, fields, woods, a lake. For these images we computed the ENL (Equivalent Number of Looks), a standard parameter widely used in the remote sensing community which measures the speckle reduction in homogeneous areas. Once selected an apparently homogeneous region in the image, like those in the red boxes in Fig. 6.6, the ENL is computed as

$$\text{ENL} = \mu_x^2 / \sigma_x^2 \quad (6.24)$$

with μ_x^2 the average intensity of the selected area and σ_x^2 its variance. Larger ENL values indicate stronger speckle rejection and, consequently, an improved ability to tell apart different gray levels. Tab. 6.4 reports the ENL values for the proposed and reference algorithms (we discard PPB and BM3D in the homomorphic context, by now). Results are quite consistent, indicating PPB² by far as the technique with the strongest speckle rejection ability, followed by MAP-S, SAR-BM3D and the others. On the other hand, this is immediately obvious by visual inspection of results, like those for the Rosen3 image, whose filtered versions are shown in Fig. 6.7. The PPB image looks more pleasant than the others and is probably more helpful to gain a quick insight of the scene. On the downside, it presents widespread artifacts resembling watercolor strokes but, with neither the noiseless image nor an expert interpreter, it is difficult to decide whether this implies any loss of details. Some help comes from the analysis of ratio images obtained, as proposed in [OQ04], as the pointwise ratio between the SAR original z and denoised \hat{x} images

$$R = z / \hat{x} \quad (6.25)$$

Given a perfect denoising, that is $\hat{x} = x$, the ratio image should contain only speckle. On the contrary, the presence of geometric structures or details correlated to the original image indicates that the algorithm has removed not only noise but

²In the case $L = 6$ (low-resolution images) the parameter setting is the same used for Naples as indicated by the authors.

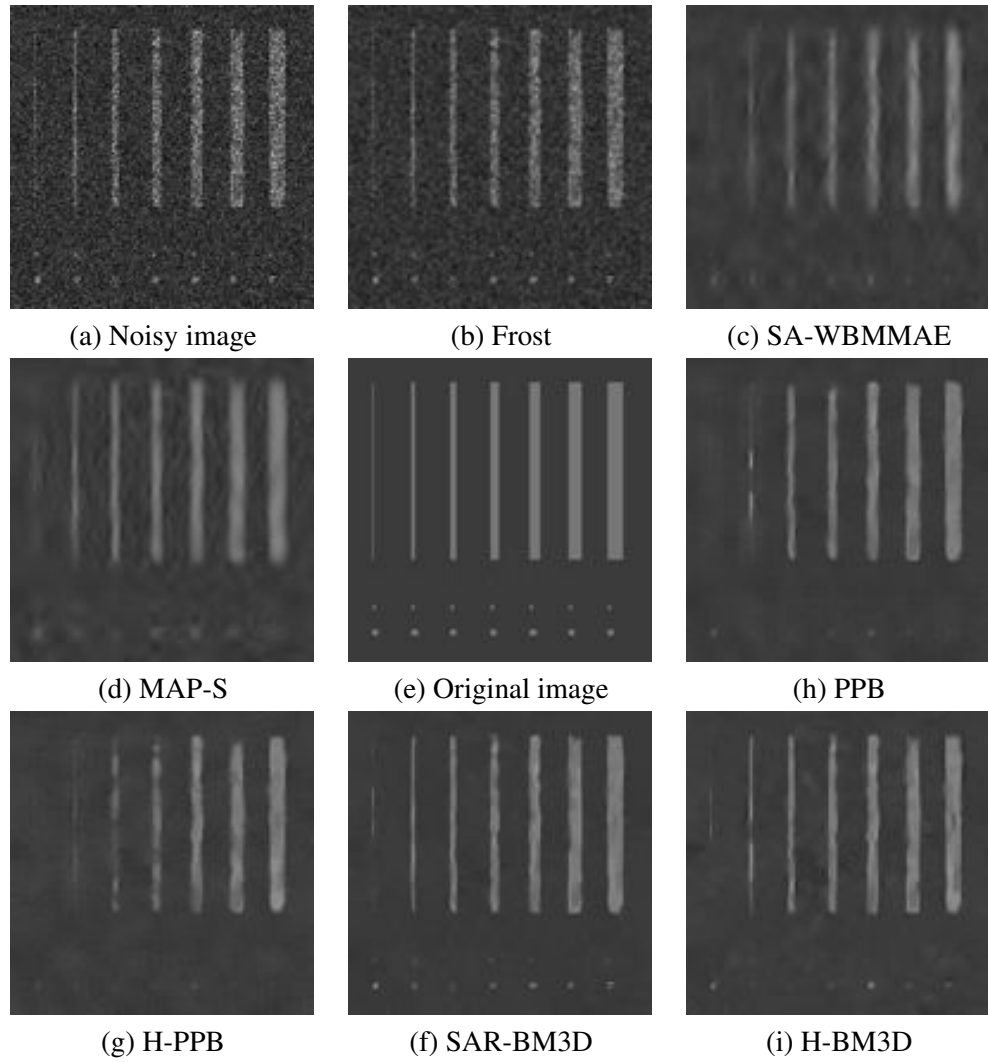


Figure 6.4: Filtered images with the various techniques for target.

also some information of interest. Fig. 6.8 shows the enhanced ratio images corresponding to the denoised images of Fig. 6.7. The Frost ratio image presents visible traces of the man-made structures, denouncing an unwanted smoothing of sharp boundaries. Similar traces, although weaker, are also present for SA-WBMAE and MAP-S. The PPB ratio image, differently from SAR-BM3D, exhibits different patterns depending on the different areas of the scene, though not marked by linear structures, showing again a dependence on the original SAR image.

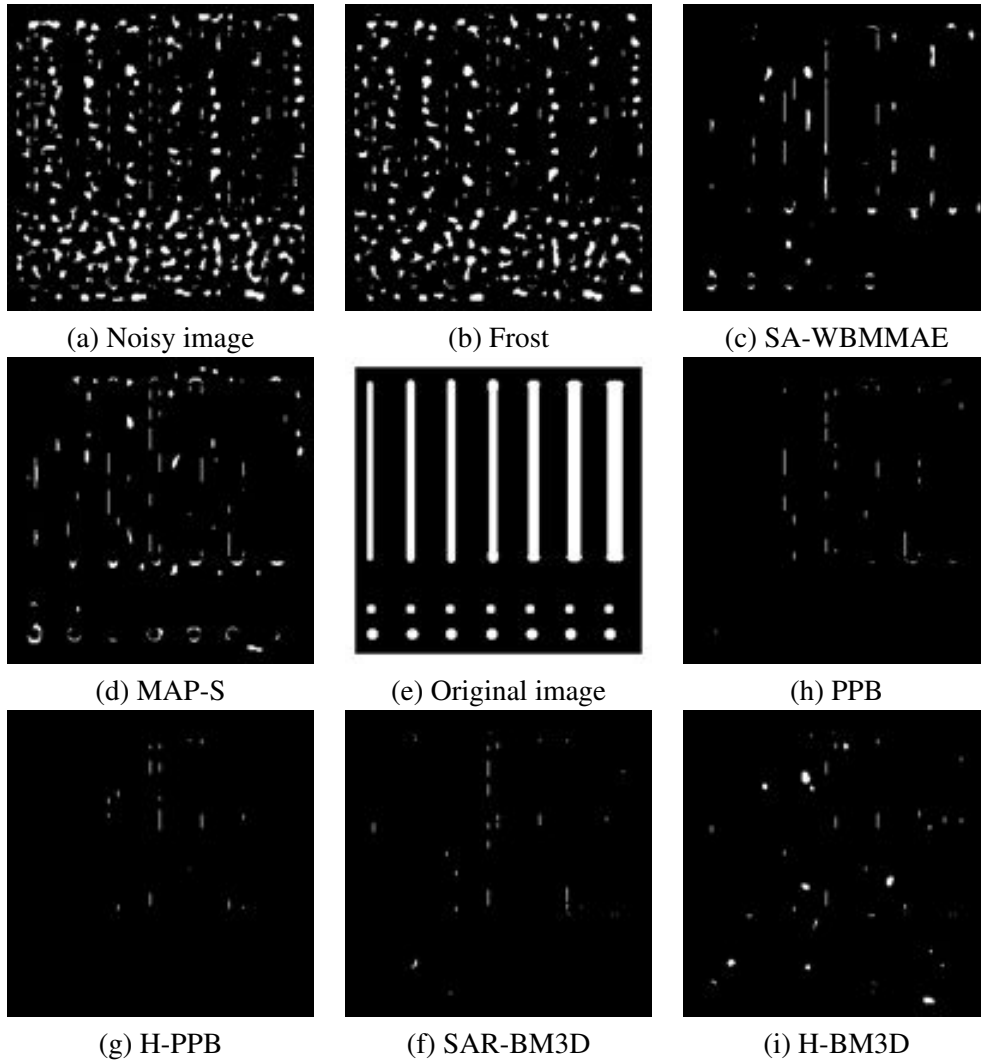


Figure 6.5: False alarms (FA) images with the various techniques for target.

6.6 Conclusion and future work

In this chapter we have proposed a novel and very promising algorithm for SAR image despeckling. We have drawn concepts and tools from the non-local filtering approach, adapting them to the specificities of SAR imagery. Major innovations regard the choice of the similarity measure, which takes into account the probabilistic noise distribution of speckle, and the wavelet shrinkage in the 3D domain, which is derived from the additive signal-dependent model following a linear minimum-MSE estimation approach.

Results on simulated speckled images are quite satisfactory, with a consistent PSNR gain of 1-2 dB over the best reference algorithms to date. Similar improve-

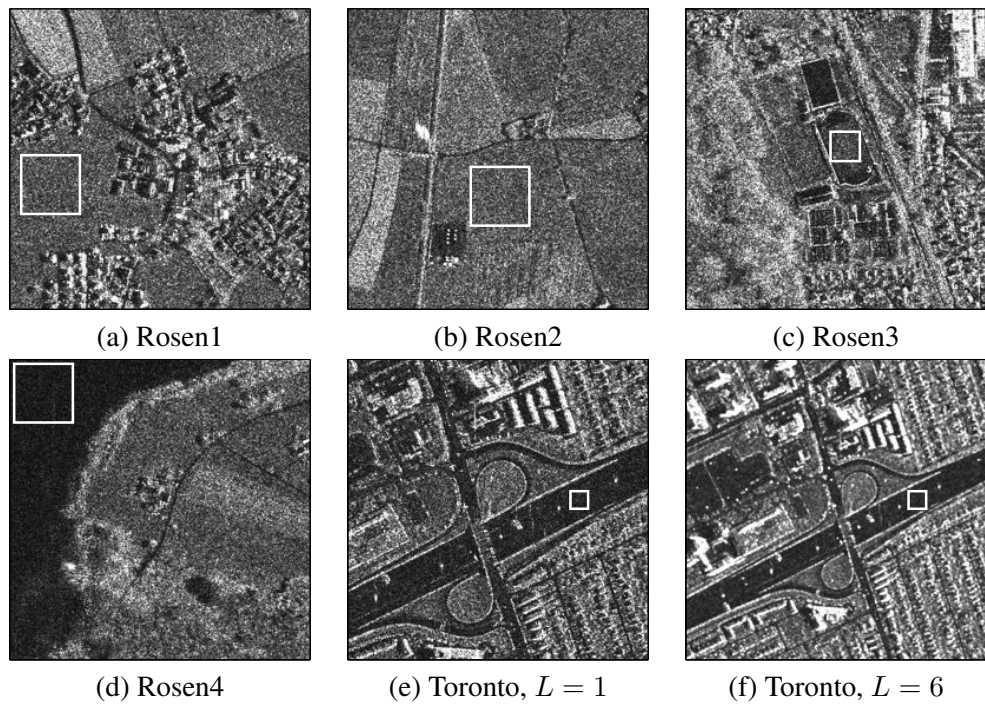
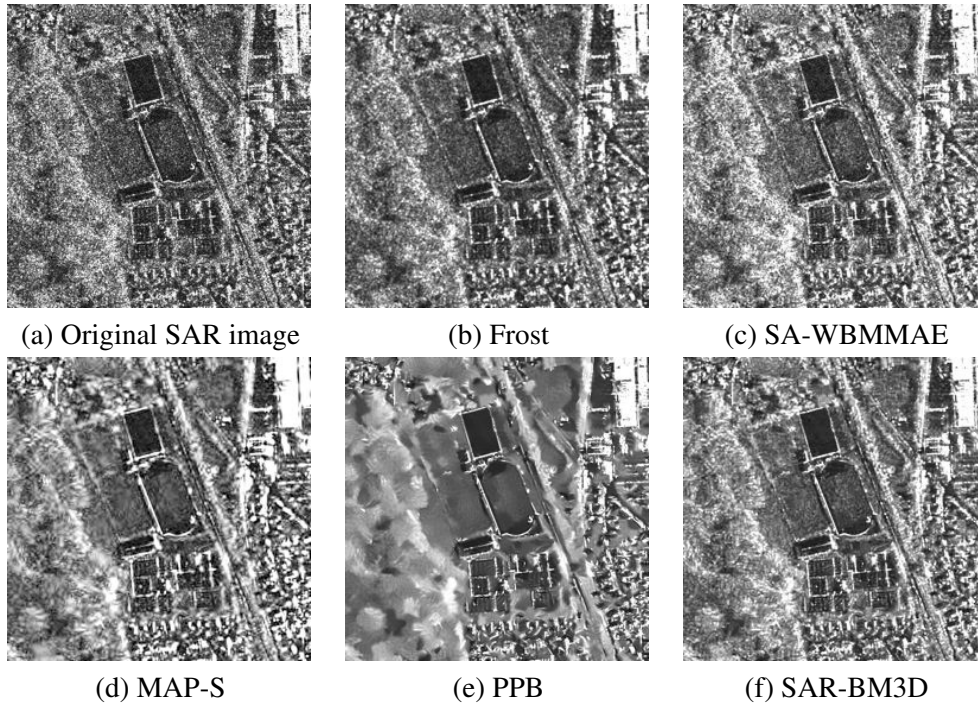


Figure 6.6: Test SAR-X images (©Infoterra GmbH) with selected areas for ENL computation (white rectangle).

	Rosen1	Rosen2	Rosen3	Rosen4	Toronto L1	Toronto L6
<i>Noisy</i>	0.95	0.98	0.90	0.94	0.91	7.43
<i>Frost</i>	3.84	4.08	3.20	4.03	3.61	12.42
<i>SA-WBMAE</i>	3.08	3.05	2.36	2.29	2.79	9.63
<i>MAP-S</i>	14.55	18.56	5.73	18.56	6.70	12.86
<i>PPB</i>	43.01	47.04	19.79	51.24	66.59	15.35
<i>SAR-BM3D</i>	6.75	8.03	4.84	7.78	7.76	11.99

Table 6.4: ENL for real SAR images.**Figure 6.7:** Filtered images with the various techniques for Rosen3.

ments are observed in a simple automatic detection task on a synthetic image. Experiments on actual SAR images are likewise encouraging, as the proposed technique seems to have a better capacity to preserve relevant details while smoothing homogeneous areas.

There are several aspects of the proposed algorithm that can be improved. In particular, the speckle statistics of actual SAR images, especially at high-resolution, often deviate from the simplified model used in this work, as well as in most of the literature. Adapting the algorithm to the case of correlated speckle will be certainly the object of future work.

Another sore point is the lack of objective quality measure for SAR images which weakens all experimental analyses. We are therefore set to carry out a more thorough and reliable analysis of SAR despeckling algorithms, based on *ad hoc*

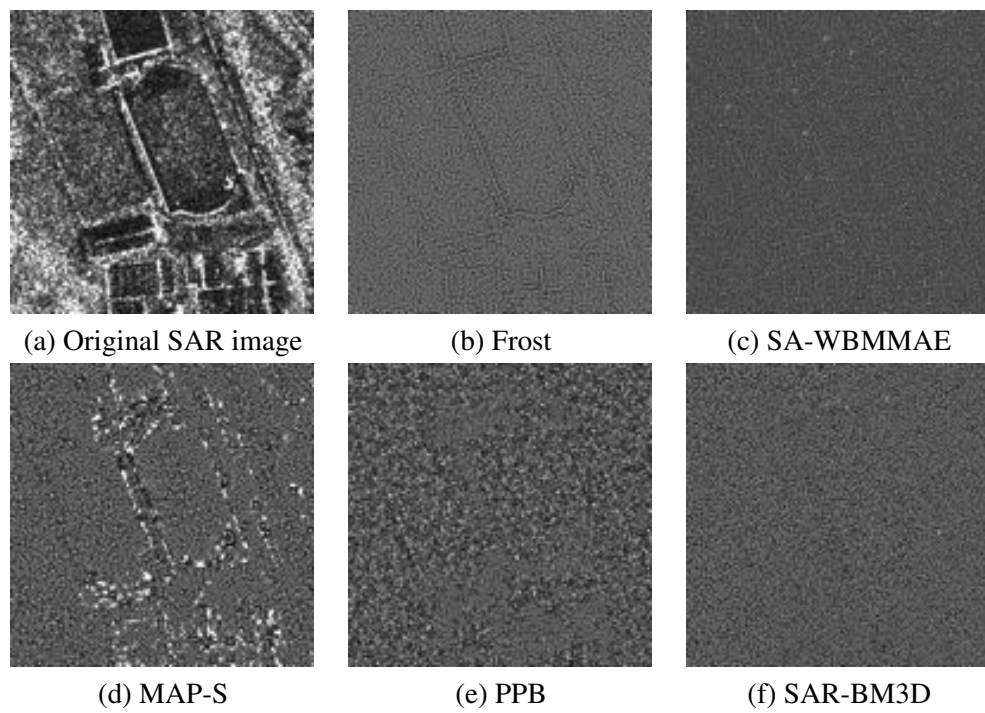


Figure 6.8: A zoom of enhanced ratio between the noisy and denoised images for the various techniques.

simulated SAR images and indicators, as well as on the measurable performance of algorithms and human interpreters on a large set of denoised images.

Chapter 7

Digital Image Inpainting

7.1 Introduction

Image inpainting is an increasingly popular area of image restoration research. The goal of inpainting is to reconstruct missing regions within an image in such a way that it is visually plausible to an observer. The application field of such a technique moves from the restoration of damaged paintings and/or photographs to the removal/replacement of selected objects.

The origins of inpainting date back to many centuries ago. Indeed, medieval works began to be restored since the beginning of the Renaissance, and the motivations that drove the restoration were manifold, ranging from simple reporting to the splendor of old paintings (Figure 7.1), the frescoes, or more generally any work, to repair or fill damaged areas in a way which does not become aware of handling an observer unaware of the features of the original (or deteriorated) version.

With the advent of photography, the need to adjust an image in a manner not detectable spread naturally from paintings to photos, the objectives of the restoration itself, such as the elimination of deterioration due to the time (*e.g.* staining) or poor conservation (*e.g.* cracks or burns), joined other objectives, such as the elimination of entire objects or people in the photo. This capability has often made of



Figure 7.1: The Last Supper (Italian: Il Cenacolo or L'Ultima Cena), Leonardo Da Vinci, 15th century.



Figure 7.2: An early example of inpainting. From “The commissar vanishes”, D. King, 1997.

inpainting a useful technique for purposes of forgery also politically motivated: for example, Stalin used it to remove from the documents evidence of political figures become awkward (see Figure 7.2).

Pictures often capture special moments in the life of an individual, or depict important people to whom one was particularly attached. The photo itself, for decades, was the only way to preserve these memories imprinted on some support. Therefore, it is clear how great is the need to keep them in the best way and always in good conditions. But if time and wear had already made their contribution in the deterioration, here comes the need for a technique to restore these old photographs (Figure 7.3).

The advent of digital photography brought a great improvement to the conservation of photographs. In fact, they are no longer subject to the ravages of time and bad storage. It's in digital photography that inpainting has a large scope: from the restoration of old photos digitized, often characterized by “old” faults to digital photo retouching, such as the adjustment of the famous “Red Eye”, the elimination of overlapping captions and text to images (Figure 7.4), *e.g.*, the date that digital cameras often imprint on pictures.

The inpainting, going beyond the simple need of editing, is also used in the transmission of compressed images [RSB03]. In fact, for example, in the transmission of images in JPEG format, it can happen that not all blocks containing information are transmitted correctly, and then there is the need to reconstruct the lost block, or you can choose to not send all blocks for higher transmission rates, downstream and reconstruct the missing blocks, again through inpainting techniques. The inpainting also extends to the video signal [KAPB05, PSB07]. In fact it could happen that some frames of a film can be damaged or even lost. These portions can be reconstructed through the use of special techniques for inpainting, who obtained such information from the frames immediately preceding and following the lost or damaged. Always remaining in the field of video, it could happen that during a shoot in front of the camera an object of the set, like the end of a microphone or an operator “careless”. In this case reshoot the scene could cost too much



Figure 7.3: Application of inpainting to restoration of an old photography.



Figure 7.4: An example of text removing. From Bertalmio [[BSCB00](#)].



Figure 7.5: Application to cinema post production.

time and money while using inpainting can easily remove these intruders from the scene (Figure 7.5).

7.2 State of the art

Algorithms for digital image inpainting are essentially of two kind:

1. **variational methods based on partial differential equation (PDE).** In this kind of methods the holes in images are filled by propagating linear structures (called isophotes) into the target region via a diffusion process. These algorithms are good at filling-in small/narrow gaps in piecewise smooth images (like cartoon images). However, the diffusion process introduces blur, which becomes noticeable when filling larger regions. Furthermore, textures cannot be restored;
2. **exemplar-based texture synthesis.** These methods rely on the Efros-Leung texture synthesis algorithm [EL99]. They are cheap and effective in replicating consistent textures. Moreover, no blur or other degradations affect the inpainted images. However, they present difficulty to preserve linear structures and composite textures in real world scenes.

7.2.1 PDE based algorithms

The reconstruction of missing or damaged portions of images is an ancient practice, used extensively in the restoration of works of art. Recently, some models have been proposed for digital inpainting based on the solution of partial differential equations (PDE).

Bertalmio *et al* [BSCB00] were the pioneers of the algorithms of digital image inpainting model based on partial differential equations (PDE). A mask-defined specific portions of the input image to be retouched. The algorithm treats the input image as three separate channels (Red Green & Blue). For each channel, the algorithm fills the areas to be retouched by propagating information from the edges of the masked region along the contour, these isophote. The isophote directions are obtained by calculating for each pixel, a gradient vector discretized along the edge (this gives the direction of large spatial changes), and rotating the resulting vector by $\pi/2$. This is to propagate information while preserving the edges. A two-dimensional Laplacian [GV97] is used to estimate the local variation in the linearity of the image and that change is propagated along the direction of the isophote. At every step of the inpainting process, the algorithm makes a series of iterations of spreading to rebuild the area to be retouched. Anisotropic diffusion is used to preserve the edges in the target region.

Inspired by the work of Bertalmio *et al* [BSCB00], Chan and Shen proposed two algorithms for image inpainting. The Total Variational inpainting model (TV) [CS02], which employs a simple anisotropic diffusion based on the contrast of the isophote. This model was designed for inpainting small regions, and while it works well in removing the noise, does not connect the broken edges (single lines embedded in a uniform background). The model Curvature Diffusion Driver (CCD) [CS01] extends the algorithm to take into account TV geometrical information of isophote defining the strength of the diffusion process, and allowing you to proceed with this type of inpainting in larger areas, but even able to connect some lines cut, the resulting interpolated segments appear blurred.

Although the PDE-based algorithms have the potential to systematically preserve the edges, their structure leads them to be too expensive from a computational point of view, for this reason were chosen faster algorithms that can adapt well to the kind of problem, providing results comparable with those obtained by PDE-based algorithms. In general, although some information about the context of the area to be retouched are available, one can hope to have a plausible reproduction, as close to the original, but not perfect. Therefore, any strategy that is able to reproduce the inpainting region selected with reasonable success for a large class of images, the region to be rebuilt must be locally small. Over this region becomes smaller, and simpler diffusion models can be used with good performance.

7.2.2 Exemplar-based texture synthesis

The definition of texture from a technical point of view is not as simple as his visual perception. This could be described as a repetition, according to different rules, of basic elements, called texels, which in turn may consist of multiple pixels. In computer graphics, a texel (contraction of texture element) is the smallest graphical element in a two-dimensional texture mapping. This definition, however, is not the only possible because you can also define a texture based on its spectral or statistical properties. For example, as regards the statistical properties, a texture can be well represented by a linear stationary stochastic process.

The synthesis of texture has a variety of applications in computer vision, graphics and image processing. A major motivation for the synthesis of texture comes from the texture mapping. Often, the photographs available may be too small to cover all surfaces with texture. In this situation, simply cover the desired area with "tiles" texture introduce unacceptable artifacts in the form of repetition and discontinuity effect in the joints. The synthesis of texture solves this problem by generating texture. The problem of texture synthesis can be formulated as follows: given a finite sample of a certain texture, the goal is to generate another sample of that texture of different sizes (obviously larger). To formulate the problem correctly, you also need to assume that the sample is large enough to capture somehow the stationarity of the texture, and it is known, at least an approximate scale of the texel [EL99].

The textures are traditionally classified as regular (consisting of repeated texels) or stochastic (without explicit texel). However, almost all of the real texture lie in intermediate positions between these two extremes, and can be defined by an individual model. As opposed to texture synthesis, the problem of inpainting is, rather than the generation of additional samples of a texture from a seed, to fill the area selected by the mask so that the viewer become unaware of the manipulation. We must therefore try to take into account the degree of repetitiveness and the stationarity of the texture to fill.

The approach of Zhu *et al* [ZWM98] model as a texture using MRF and Gibbs sampling for synthesis. Unfortunately, Gibbs sampling is notoriously slow, and it is not possible to tell when it converges. De Bonet [DB97] uses an approach based on a multi-resolution filter, in which a patch of texture on a scale more "fine" is conditioned by corresponding patch in a more "raw" scale. The algorithm works by taking the input sample texture and making random, in order to preserve these "inter-scale" dependencies. This method can successfully synthesize a wide range of textures, although the parameter of randomness seems to offer good performances only on texture with largely stochastic behavior. Another drawback of this method is the manner in which images are generated texture larger than the original. The sample input texture is simply replicated to the desired size, implicitly assuming that all textures are replicable such as "tiles", which is clearly not correct. The work of Efros and Leung [EL99] provides a new and simple way of looking at the problem of texture synthesis by growing a texture with a pixel at

a time starting from an initial seed. The color of the pixel data is determined by analyzing square patch of the sample texture that are similar to the patch on the texture to be generated. A patch in the random sample is selected by trying to satisfy the criterion of similarity. The similarity is measured by a particular rule (sum of squared differences) on the RGB colors weighted by a Gaussian kernel. Given the slowness of the original algorithm of Efros and Leung [EL99] and subsequent extensions have sought to improve its performance, in particular the work of Wei and Levoy [WL00], which uses a raster scan ordering "to transform the pixel noise in the desired texture. The work of Ashikhmin [Ash01] instead, improve the technique of Wei and Levoy introducing two new contributions to the original paper: first, the algorithm improves the synthesis of natural textures, which often failed in the algorithm of Wei and raised. It also introduces a new image called *target*. The final synthesis combines pixels from both the sample and the target, producing effects such as writing letters on textures.

7.3 Proposed algorithm and results

Image inpainting can be viewed as a denoising task, since noise can be any unwanted signal such as, in image context, objects, text, scratches etc... Recent inpainting techniques have focused on the concept of exemplar-based synthesis [WR06, CPT04]. In these techniques, a best match sample from the source region is found and copied directly into the target region. However, as images usually contain redundant contents, several samples could be used to estimate missing information. We adapted our algorithm based on the k NN framework to image inpainting. The apriori information about the inpainting zone is given by a noise mask M . For each pixel in the mask we search for the k -nearest neighbors patches within a search zone¹, then we replace the pixel in the mask with the center pixel of a patch chosen within the k candidates. Another way is to replace the pixel in the mask with an average of center pixels. However, we found that this solution leads to more flattened zones (due to averaging) in the inpainted image. The pseudocode in Figure 7.3 resumes the algorithm fundamental steps.

In all our experiments we set the patch radius r equal to 2, i.e., a patch of 5×5 pixel. The search window radius w was set equal to 15, and the neighboring order k equal to 5. Figure 7.7 shows the restoration of an old photo of the sixteenth president of the United States of America Abraham Lincoln taken by Alexander Gardner on February 5, 1865 few months before his assassination. In shooting the image, Gardner used a large glass negative, which broke before it could be processed. Nevertheless, he managed to make one print (some have interpreted the crack running through the image as a portent of Lincoln's impending assassination). Figure 7.8 shows the results of the proposed method in application to filling-in small regions. Inpainted images look natural and no processing is noticeable. In Figure 7.9 large objects have been removed from pictures. In these

¹We exclude patches which contain pixel of the noise mask

Inpainting Pseudo code

For each pixel $s \in M$:

- Let z_s be the patch of radius r formed by the pixel color x_s and the neighborhood $y_s = \{x_t, t \in C_s\}$ where C_s is the neighborhood of radius r of the pixel s ;
- Let $A(s)$ be the search area of radius w centered at s
- For each pixel $t \in A(s)$

$$\rho(s, t) \leftarrow |z_s - z_t| \quad (7.1)$$

- Select randomly one $z_{\tilde{t}}$ among the k nearest patches z_t
- Then

$$x_s \leftarrow x_{\tilde{t}} \quad (7.2)$$

Figure 7.6: Pseudocode for the proposed inpainting algorithm.

experiments some processing becomes noticeable to an attentive observer.

7.4 Conclusion and perspectives

In this chapter a simple inpainting algorithm has been derived from the general denoising framework proposed in this thesis. The key idea rely on the self-similarity commonly present in images: certain image patches tend to repeat over and over, with minor modifications, throughout the scene. This property allows to recover the missing information within the inpainting mask by looking for similar patches which carry the same neighborhood information and copying the remaining color information while reducing the zone to be inpainted. Results on several images are encouraging. Inpainted areas seem very natural and they are not noticeable to an observer unaware of the original image. A further modification is to adopt a priority filling scheme as in [CPT04] in order to better preserve image structures. Furthermore, allowing an overlapping of inpainted patches an aggregation procedure can be adopted, improving the estimation of inpainted regions.



Figure 7.7: Photo of Abraham Lincoln taken by Alexander Gardner on February 5, 1865. Original image (left) and inpainted image (right).

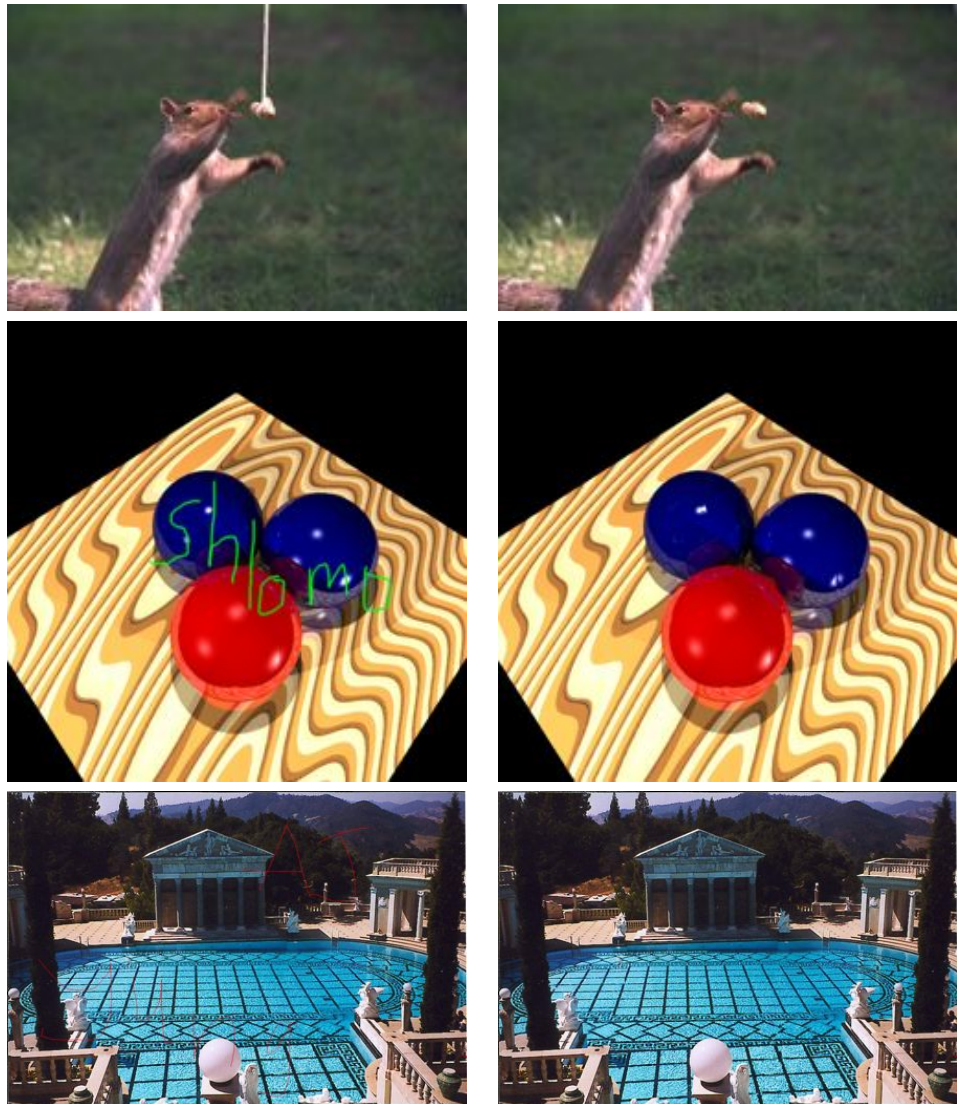


Figure 7.8: Results on inpainting 2



Figure 7.9: Application to object removal.

Conclusions

The work of this thesis has concerned the study and development of new information theory-based algorithms for image restoration. In particular, three major problems have been treated in this work: deconvolution, denoising and inpainting.

The first part of this thesis deals with deconvolution. The deconvolution problem is set in the context of variational methods. Given the observation, the solution minimizes a functional or energy of the residual, i.e. the difference between the observation and the solution. We chose as energy functional the differential entropy of the residual image, which turns out to be robust with respect to outliers. Furthermore, no model is imposed to the noise which can be Gaussian, Uniform, a mixture, etc. Results show the robustness of the algorithm with respect to different noise distributions. We showed that the residual entropy is lower bounded by the noise entropy. We also showed that the minimization algorithm can be interpreted as the process which use the information carried by the residual to recover the original image, until there is no more information, i.e., the residual entropy reaches the lower bound.

The second part of this work deals with the denoising problem. The patch-based methods for texture preserving denoising have been taken into account, with application to the domain of digital photography and remotely sensed SAR images. We proposed a full patch denoising algorithm PCkNN which tries to minimize the entropy of patches which was first proposed in [AW06]. However, the use of entropy has been objectively motivated since Information Theory allows to provide a lower bound for this quantity. Starting from this Information Theoretic context we showed that entropy minimization leads to non-local filtering approaches, giving also a variational interpretation of such methods.

Experimental results show that this technique better preserves texture and the natural appearing of the images. Indeed the residual noise is quite natural and does not exhibit spurious patterns, thus leading to a denoised image with a very natural appearance while other algorithms create flattened regions. The residual noise is a relevant method for quality assessment. Indeed if the reconstruction error spectrum is mostly located into the high frequencies it is not noticeable to the Human Visual System which acts, at first sight, as a low-pass filter.

The PCkNN has also be adapted to Color Filter Array (CFA) RAW images. In modern digital cameras the light is acquired by collecting for each pixel only one color information. Therefore, to have a complete RGB image a further step,

called demosaicing, is needed. Image denoising is still an hot topic also for high quality sensors especially in low-light exposure images. A naive approach treats each color channel separately focusing only on intra-channel correlation. We went one step further adapting the PCkNN algorithm to deal with CFA images exploiting the inter-channel dependencies by adapting the block-matching step. Results show the good compromise between noise reduction and edge/texture preservation.

We also developed a SAR oriented version of the BM3D algorithm which represents the state-of-the-art of AWGN denoising techniques. We have drawn concepts and tools from the non-local filtering approach, adapting them to the specificities of SAR imagery. Major innovations regard the choice of the similarity measure, which takes into account the probabilistic noise distribution of speckle, and the wavelet shrinkage in the 3D domain, which is derived from the additive signal-dependent model following a linear minimum-MSE estimation approach. Results on simulated speckled images are quite satisfactory, with a consistent PSNR gain of 1-2 dB over the best reference algorithms to date. Similar improvements are observed in a simple automatic detection task on a synthetic image. Experiments on actual SAR images are likewise encouraging, as the proposed technique seems to have a better capacity to preserve relevant details while smoothing homogeneous areas. Experimental results show this algorithm outperforms all of state-of-the-art despeckling algorithms.

The last part of this thesis is devoted to the digital image inpainting, which is a growing research area. We adapted our denoising algorithm in order to fill selected areas in the image in an undetectable manner. Experimental results are encouraging. Indeed the algorithm despite its simplicity performs well in filling small regions. However, when the regions become very large some algorithm artifacts become noticeable.

Part IV

Appendix

Appendix A

Derivative of the residual entropy E

In this Section we tackle the derivative of the deconvolution energy $E(x)$ presented in Section 3.3.1. Let us remind that

$$\begin{aligned} E(x) &= |\Omega| H_{\text{A-L}}(r) \\ &= - \int_{\Omega} \log(p_x(r(u))) \, du . \end{aligned} \quad (\text{A.1})$$

is the continuous version of the Ahmad-Lin [AL76] entropy estimator ($H_{\text{A-L}}(r)$) and $p_x(\cdot)$ is the KDE of the residual PDF,

$$p_x : s \in \mathbb{R}^d \rightarrow \frac{1}{|\Omega|} \int_{\Omega} K_{\sigma}(s - r(u)) \, du, \quad (\text{A.2})$$

where K_{σ} is a smoothing kernel (from \mathbb{R}^d to \mathbb{R}) of bandwidth σ such that $K_{\sigma}(-r) = K_{\sigma}(r)$, $r \in \mathbb{R}^d$.

A.1 Definitions and notations

Let us first introduce a bit of notations and definitions used throughout this Section. With x, y, r and h we indicate vector-valued images, *i.e.*, functions

$$f : u \in \Omega \subset \mathbb{R}^2 \rightarrow (f_1(u) \, f_2(u) \, \dots \, f_d(u)) \in \mathbb{R}^d \quad (\text{A.3})$$

The PSF is $m : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that $m(-u) = m(u)$. Note that $m(\cdot)$ blurs each image component (or channel) in the same way, *i.e.*,

$$y(u) = (m \star x)(u) \quad (\text{A.4})$$

$$= \int_{\mathbb{R}^2} m(u-v) x(v) \, dv \quad (\text{A.5})$$

$$= \begin{pmatrix} \int_{\mathbb{R}^2} m(u-v) x_1(v) \, dv \\ \cdots \\ \int_{\mathbb{R}^2} m(u-v) x_d(v) \, dv \end{pmatrix}. \quad (\text{A.6})$$

The symbol \star denotes the convolution operator.

Note that in general images are non-continuous functions, *i.e.*, $x \notin C^0(\Omega)$, but they are bounded functions, $x \in L^\infty(\Omega)$, which is a normed vector space. Therefore,

$$E : x \in L^\infty(\Omega) \rightarrow E(x) \in \mathbb{R} \quad (\text{A.7})$$

is a non-linear functional of x and

$$P : x \in L^\infty(\Omega) \rightarrow p_x \in L(\Omega). \quad (\text{A.8})$$

In the following we proceed in the calculus of the Gateaux derivative.

A.2 Derivative of p

We first deal with the Gateaux derivative of the function P . We want to calculate

$$\lim_{\theta \rightarrow 0} \frac{P(x + \theta h) - P(x)}{\theta}. \quad (\text{A.9})$$

Let us consider the following notation

$$\begin{cases} q_x = |\Omega| p_x \\ \delta s = s - r(u) \\ \tilde{h} = m \star h \end{cases}. \quad (\text{A.10})$$

Then consider,

$$q_{x+\theta h}(s) - q_x(s) = \int_{\Omega} [K_\sigma(\delta s + \theta(m \star h)(u)) - K_\sigma(\delta s)] \, du \quad (\text{A.11})$$

$$= \int_{\Omega} \int_0^1 \frac{d}{d\tau} [K_\sigma(\delta s + \tau \theta \tilde{h}(u))] (\tau) \, d\tau \, du \quad (\text{A.12})$$

$$= \int_{\Omega} \int_0^1 \nabla K_\sigma(\delta s + \tau \theta \tilde{h}(u)) \cdot \theta \tilde{h}(u) \, d\tau \, du \quad (\text{A.13})$$

$$= \theta \int_{\Omega} \left(\int_0^1 \nabla K_\sigma(\delta s + \tau \theta \tilde{h}(u)) \, d\tau \right) \tilde{h}(u) \, du \quad (\text{A.14})$$

and take the limit for $\theta \rightarrow 0$

$$\lim_{\theta \rightarrow 0} \int_{\Omega} \left(\int_0^1 \nabla K_{\sigma}(\delta s + \tau \theta \tilde{h}(u)) \, d\tau \right) \tilde{h}(u) \, du \quad (\text{A.15})$$

$$= \int_{\Omega} \nabla K_{\sigma}(\delta s) \cdot (m \star h)(u) \, du \quad (\text{A.16})$$

$$= \int_{\Omega} \nabla K_{\sigma}(\delta s) \left(\int_{\mathbb{R}^2} m(u-v) h(v) \, dv \right) \, du \quad (\text{A.17})$$

$$= \int_{\Omega} \int_{\mathbb{R}^2} \nabla K_{\sigma}(\delta s) [m(u-v) h(v)] \, dv \, du \quad (\text{A.18})$$

$$= \int_{\Omega} \int_{\mathbb{R}^2} h(v) [m(u-v) \nabla K_{\sigma}(\delta s)] \, dv \, du \quad (\text{A.19})$$

$$= \int_{\mathbb{R}^2} \int_{\Omega} h(v) [m(u-v) \nabla K_{\sigma}(\delta s)] \, du \, dv \quad (\text{A.20})$$

$$= \int_{\mathbb{R}^2} h(v) \left(\int_{\Omega} m(u-v) \nabla K_{\sigma}(s-r(u)) \, du \right) \, dv \quad (\text{A.21})$$

$$= \int_{\mathbb{R}^2} h(v) \left(\int_{\Omega} m(v-u) \nabla K_{\sigma}(s-r(u)) \, du \right) \, dv \quad (\text{A.22})$$

$$= \int_{\mathbb{R}^2} h(v) [m \star_{\Omega} \nabla K_{\sigma}(s-r)](v) \, dv \quad (\text{A.23})$$

$$= \langle h, m \star_{\Omega} \nabla K_{\sigma}(s-r) \rangle. \quad (\text{A.24})$$

A.3 Derivative of E

$$E(x + \theta h) - E(x) = \quad (\text{A.25})$$

$$= - \int_D \left[\log(p_{x+\theta h}(r(u) - \theta \tilde{h}(u))) - \log(p_x(r(u))) \right] \, du \quad (\text{A.26})$$

$$= - \int_D \left(\int_0^1 \frac{d}{d\tau} [\log(p_{x+\tau \theta h}(r(u) - \tau \theta \tilde{h}(u)))](\tau) \, d\tau \right) \, du \quad (\text{A.27})$$

$$= - \int_D \left(\int_0^1 \frac{1}{p_{x+\tau \theta h}(r(u) - \tau \theta \tilde{h}(u))} \frac{d}{d\tau} [p_{x+\tau \theta h}(r(u) - \tau \theta \tilde{h}(u))](\tau) \, d\tau \right) \, du. \quad (\text{A.28})$$

Notations

$$\begin{cases} X = x + \tau \theta h \\ R = r - \tau \theta \tilde{h} \\ H = \theta h \\ \tilde{H} = \theta \tilde{h} \end{cases}. \quad (\text{A.29})$$

$$\frac{d}{d\tau} [p_{x+\tau\theta h}(r(u) - \tau\theta\tilde{h}(u))] (\tau) = \quad (\text{A.30})$$

$$= \lim_{\epsilon \rightarrow 0} \frac{p_{x+(\tau+\epsilon)\theta h}(r(u) - (\tau+\epsilon)\theta\tilde{h}(u)) - p_{x+\tau\theta h}(r(u) - \tau\theta\tilde{h}(u))}{\epsilon} \quad (\text{A.31})$$

$$= \lim_{\epsilon \rightarrow 0} \frac{p_{X+\epsilon H}(R(u) - \epsilon\tilde{H}(u)) - p_X(R(u))}{\epsilon} \quad (\text{A.32})$$

$$= \frac{d}{d\epsilon} [p_{X+\epsilon H}(R(u) - \epsilon\tilde{H}(u))] \Big|_{\epsilon=0} \quad (\text{A.33})$$

$$= \frac{d}{d\epsilon} [p_{X+\epsilon H}(R(u))] \Big|_{\epsilon=0} + \frac{d}{d\epsilon} [p_X(R(u) - \epsilon\tilde{H}(u))] \Big|_{\epsilon=0} \quad (\text{A.34})$$

$$= \frac{1}{|D|} \frac{d}{d\epsilon} [q_{X+\epsilon H}(R(u))] \Big|_{\epsilon=0} - \nabla p_X(R(u)) \cdot \tilde{H}(u) \quad (\text{A.35})$$

$$= \frac{1}{|D|} \langle \theta h, m \star_D \nabla K_\sigma(r(u) - r - \tau\theta\tilde{h}) \rangle - \nabla p_{x+\tau\theta h}(r(u) - \tau\theta\tilde{h}(u)) \cdot \theta\tilde{h}(u) . \quad (\text{A.36})$$

Therefore,

$$\begin{aligned} E(x + \theta h) - E(x) &= -\frac{\theta}{|D|} \int_D \left(\int_0^1 \frac{1}{p_{x+\tau\theta h}(r(u) - \tau\theta\tilde{h}(u))} \times \right. \\ &\quad \times \left[\langle h, m \star_D \nabla K_\sigma(r(u) - r - \tau\theta\tilde{h}) \rangle \right. \\ &\quad \left. \left. - |D| \nabla p_{x+\tau\theta h}(r(u) - \tau\theta\tilde{h}(u)) \cdot \tilde{h}(u) \right] d\tau \right) du \end{aligned} \quad (\text{A.37})$$

and

$$dE(x, h) = -\frac{1}{|D|} \int_D \frac{1}{p_x(r(u))} \left[\langle h, m \star_D \nabla K_\sigma(r(u) - r) \rangle - |D| \nabla p_x(r(u)) \cdot \tilde{h}(u) \right] du \quad (\text{A.38})$$

$$= \int_D \frac{\nabla p_x(r(u))}{p_x(r(u))} \cdot \tilde{h}(u) du - \frac{1}{|D|} \int_D \int_{\mathbb{R}^2} h(v) \cdot \frac{[m \star_D \nabla K_\sigma(r(u) - r)](v)}{p_x(r(u))} dv du \quad (\text{A.39})$$

$$= A - B . \quad (\text{A.40})$$

$$A = \int_D \frac{\nabla p_x(r(u))}{p_x(r(u))} \cdot \left(\int_{\mathbb{R}^2} m(u-v) h(v) \, dv \right) \, du \quad (\text{A.41})$$

$$= \int_D \int_{\mathbb{R}^2} h(v) \cdot \left[m(u-v) \frac{\nabla p_x(r(u))}{p_x(r(u))} \right] \, dv \, du \quad (\text{A.42})$$

$$= \int_{\mathbb{R}^2} h(v) \cdot \left(\int_D m(u-v) \frac{\nabla p_x(r(u))}{p_x(r(u))} \, du \right) \, dv \quad (\text{A.43})$$

$$= \int_{\mathbb{R}^2} h(v) \cdot \left[m \star_D \frac{\nabla p_x(r)}{p_x(r)} \right] (v) \, dv \quad (\text{A.44})$$

$$= \langle h, m \star_D \frac{\nabla p_x(r)}{p_x(r)} \rangle. \quad (\text{A.45})$$

$$B = \frac{1}{|D|} \int_{\mathbb{R}^2} \int_D h(v) \cdot \frac{[m \star_D \nabla K_\sigma(r(u) - r)](v)}{p_x(r(u))} \, du \, dv \quad (\text{A.46})$$

$$= \frac{1}{|D|} \int_{\mathbb{R}^2} h(v) \cdot \int_D \frac{[m \star_D \nabla K_\sigma(r(u) - r)](v)}{p_x(r(u))} \, du \, dv \quad (\text{A.47})$$

$$= \frac{1}{|D|} \langle h, \int_D \frac{m \star_D \nabla K_\sigma(r(u) - r)}{p_x(r(u))} \, du \rangle. \quad (\text{A.48})$$

Finally, the gradient of E associated with the inner product $\langle \cdot, \cdot \rangle$ is equal to

$$\nabla E(x) = m \star_D \frac{\nabla p_x(r)}{p_x(r)} - \frac{1}{|D|} \int_D \frac{m \star_D \nabla K_\sigma(r(u) - r)}{p_x(r(u))} \, du \quad (\text{A.49})$$

$$= m \star_D \frac{\nabla p_x(r)}{p_x(r)} - \frac{1}{|D|} m \star_D \int_D \frac{\nabla K_\sigma(r(u) - r)}{p_x(r(u))} \, du \quad (\text{A.50})$$

$$= m \star_D \left(\frac{\nabla p_x(r)}{p_x(r)} - \frac{1}{|D|} \int_D \frac{\nabla K_\sigma(r(u) - r)}{p_x(r(u))} \, du \right) \quad (\text{A.51})$$

Appendix B

Computation of the second order term $\chi(w)$ of the entropy derivation

To compute $k(w)$ we have to calculate the second term of (3.18), as the first one is simply proportional to the local mean shift vector.

$$\chi(w) = \frac{1}{|D|} \int_D \frac{\nabla K_\sigma(r(u) - r(w))}{p_x(r(u))} du. \quad (\text{B.1})$$

Analytic computation of eq.(B.1) is a difficult task in the general case. However, hypothesis and approximation on the residual pdf can be assumed to make it easier.

B.1 Gaussian Residual PDF

As long as the noise is gaussian the residual pdf can be assumed to be gaussian with mean $\bar{\mu}$ and standard deviation $\bar{\sigma}$,

$$p_x(\alpha) = \frac{1}{(2\pi\bar{\sigma}^2)^{d/2}} \exp -\frac{\|\alpha - \bar{\mu}\|^2}{2\bar{\sigma}^2}. \quad (\text{B.2})$$

B.1.1 Gaussian Kernel

Assuming a symmetric Gaussian Kernel $K_\sigma(x)$, with mean μ and standard deviation σ .

$$K_\sigma(x) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp \left\{ -\frac{\|x\|^2}{2\sigma^2} \right\}, \quad (\text{B.3})$$

$$\Rightarrow \nabla K_\sigma(x) = -\frac{x}{(2\pi\sigma^2)^{d/2}\sigma^2} \exp \left\{ -\frac{\|x\|^2}{2\sigma^2} \right\}. \quad (\text{B.4})$$

Thus by substituting

$$\begin{aligned}\chi(w) &= -\frac{1}{|D|} \int_D \frac{\frac{r(u)-r(w)}{(2\pi)^{d/2}\sigma^{d+2}} \exp -\frac{\|r(u)-r(w)\|^2}{2\sigma^2}}{\frac{1}{(2\pi)^{d/2}\bar{\sigma}^d} \exp -\frac{\|r(u)-\bar{\mu}\|^2}{2\bar{\sigma}^2}} du \\ &= -\frac{1}{|D|} \frac{\bar{\sigma}^d}{\sigma^{d+2}} \int_D (r(u) - r(w)) \exp -\frac{1}{2} \left[\frac{\|r(u) - r(w)\|^2}{\sigma^2} - \frac{\|r(u) - \bar{\mu}\|^2}{\bar{\sigma}^2} \right] du\end{aligned}\quad (\text{B.5})$$

with some algebra we obtain

$$\chi(w) = K \int_D (r(u) - r(w)) \exp \left\{ -\frac{1}{2\sigma^2\bar{\sigma}^2} \langle r(u), \bar{\sigma}^2(r(u) - 2r(w)) - \sigma^2(r(u) - 2\bar{\mu}) \rangle \right\} du. \quad (\text{B.6})$$

with

$$K = -\frac{1}{|D|} \frac{\bar{\sigma}^d}{\sigma^{d+2}} \exp -\left\{ \frac{\|r(w)\|^2}{2\sigma^2} - \frac{\bar{\mu}^2}{2\bar{\sigma}^2} \right\}. \quad (\text{B.7})$$

If $\sigma = \bar{\sigma}$ and $\bar{\mu} = 0$, simplifications are allowed:

$$\chi(w) = -\frac{1}{|D|\sigma^2} \exp -\frac{\|r(w)\|^2}{2\sigma^2} \int_D [r(u) - r(w)] \exp \frac{\langle r(u), r(w) \rangle}{\sigma^2} du. \quad (\text{B.8})$$

B.1.2 Epanechnikov Kernel

Let us now suppose as kernel

$$K_\sigma(x) = \begin{cases} \frac{d+2}{2} \frac{\Gamma(d/2+1)}{\pi^{d/2}\sigma^d} \left(1 - \left\|\frac{x}{\sigma}\right\|^2\right), & \|x\| \leq \sigma \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.9})$$

$$\Rightarrow \nabla K_\sigma(x) = \begin{cases} -(d+2) \frac{\Gamma(d/2+1)}{\pi^{d/2}\sigma^d} \frac{x}{\sigma^2}, & \|x\| < \sigma \\ 0, & \|x\| > \sigma \end{cases} \quad (\text{B.10})$$

Thus

$$\chi(w) = -\frac{d+2}{|D|\sigma^2} \left(\frac{\bar{\sigma}}{\sqrt{2}\sigma} \right)^d \Gamma(d/2+1) \int_D [r(u) - r(w)] \exp \frac{\|r(u) - \bar{\mu}\|^2}{2\bar{\sigma}^2} \text{rect} \frac{\|r(u) - r(w)\|}{2\sigma} du. \quad (\text{B.11})$$

B.2 Histogram Residual pdf

In Kernel Estimation Methods (KDE), as Parzen method, the value of the pdf at the estimation point α is, roughly speaking, a superposition of the sample occurrences in a neighborhood of α , related to the bandwidth of the kernel. However if the

kernel has a narrow bandwidth (asymptotically a δ -function), only samples very close to the actual estimation point α will contribute to the pdf.

Under this assumption the residual pdf is approximatively

$$p_x(\alpha) \approx \frac{N(\alpha)}{|D|}, \quad (\text{B.12})$$

where $N(\alpha)$ is the number of samples such $r(w) = \alpha$. Note that, this approximation does not make any assumption on the underlying residual pdf. $\chi(w)$ is the sample mean of a function of the random variable R , i.e.,

$$\frac{1}{|D|} \int_D \frac{\nabla K_\sigma(r(u) - r(w))}{p_x(r(u))} du \approx \int_{\text{supp}R} \nabla K_\sigma(\alpha - r(w)) d\alpha. \quad (\text{B.13})$$

This is function of $r(w)$, if $r(w) = 0$ and the support of R is symmetric, the value of $\chi(w)$ is 0 as long as it is an integral of an even function. By means of this considerations, we could expect a negligible value of $\chi(w)$ if $r(w)$ is small, and higher values near the boundary of the support of R .

In grayscale images ($d = 1$), the residual $R \in [m, M]$ and the integral can be easily analytically calculated.

B.2.1 Gaussian Kernel

We consider a Gaussian Narrowband Kernel, so:

$$\nabla K_\sigma(x) = -\frac{x}{\sqrt{2\pi}\sigma^3} \exp -\frac{x^2}{2\sigma^2}. \quad (\text{B.14})$$

Thus

$$\chi(w) = \int_m^M -\frac{\alpha - r(w)}{\sqrt{2\pi}\sigma^3} \exp -\frac{(\alpha - r(w))^2}{2\sigma^2} d\alpha. \quad (\text{B.15})$$

By letting $\alpha - r(w) = \lambda$ and solving the integral, we obtain

$$\chi(w) = \frac{1}{\sqrt{2\pi}\sigma} \left\{ \exp \left(-\frac{(M - r(w))^2}{2\sigma^2} \right) - \exp \left(-\frac{(m - r(w))^2}{2\sigma^2} \right) \right\}. \quad (\text{B.16})$$

Actually, pointing out three values

$$r(w) = m \Rightarrow \chi(w) = -\frac{1}{\sqrt{2\pi}\sigma} \left(1 - \exp -\frac{(M - m)^2}{2\sigma^2} \right) < 0, \quad (\text{B.17})$$

$$r(w) = 0 \Rightarrow \chi(w) = 0, \text{ if } M = |m| \quad (\text{B.18})$$

$$r(w) = M \Rightarrow \chi(w) = \frac{1}{\sqrt{2\pi}\sigma} \left(1 - \exp -\frac{(M - m)^2}{2\sigma^2} \right) > 0. \quad (\text{B.19})$$

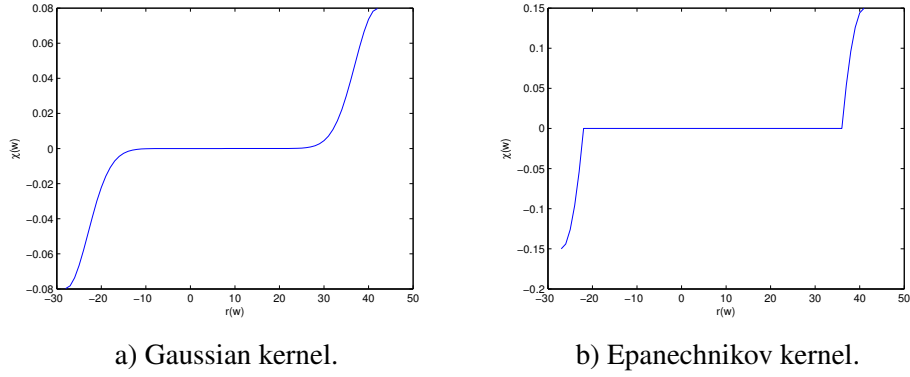


Figure B.1: $\chi(w)$ for a Gaussian and Epanechnikov Kernel, both with $\sigma = \text{std}(R)$

B.2.2 Epanechnikov Kernel

$$\nabla K_\sigma(x) = \begin{cases} -\frac{3}{2\sigma^3} x, & |x| < \sigma \\ 0, & |x| > \sigma \end{cases} \quad (\text{B.20})$$

Thus

$$\chi(w) = \int_m^M -\frac{3}{2\sigma^3}(\alpha - r(w)) \text{rect}\left[\frac{|\alpha - r(w)|}{2\sigma}\right] d\alpha. \quad (\text{B.21})$$

Here, we suppose $\sigma < \min\{|m|, M\}$, a situation always verified in practice. Thus

$$\chi(w) = \int_{\max\{m, r(w)-\sigma\}}^{\min\{M, r(w)+\sigma\}} -\frac{3}{2\sigma^3}(\alpha - r(w)) d\alpha = -\frac{3}{4\sigma^3}(\alpha - r(w))^2 \Big|_{\max\{m, r(w)-\sigma\}}^{\min\{M, r(w)+\sigma\}} \quad (\text{B.22})$$

So $\chi(w)$ takes the following expression

$$\chi(w) = \begin{cases} -\frac{3}{4\sigma^3} [\sigma^2 - (m - r(w))^2], & m < r(w) < m + \sigma \\ 0, & m + \sigma < r(w) < M - \sigma \\ -\frac{3}{4\sigma^3} [(M - r(w))^2 - \sigma^2], & M - \sigma < r(w) < M \end{cases} \quad (\text{B.23})$$

Appendix C

Derivative of Conditional entropy

In this section the calculus of the derivative of the conditional entropy of Eq. (4.12) is performed. Let us remind that

$$h(X|Y = y_i) \approx -\frac{1}{|T|} \sum_{t_j \in T} \log p(x_j|y_i) \quad (\text{C.1})$$

and

$$p(\alpha|y_i) = \frac{1}{|T_{y_i}|} \sum_{t_m \in T_{y_i}} K(\alpha - x_m), \quad (\text{C.2})$$

where T_{y_i} is the set of index pixels which have the same neighborhood y_i . Thus we have

$$h(X|Y = y_i) = -\frac{1}{|T|} \sum_{t_j \in T} \log \left[\frac{1}{|T_{y_i}|} \sum_{t_m \in T_{y_i}} K(x_j - x_m) \right]. \quad (\text{C.3})$$

By taking the derivative of (C.3) with respect to the current pixel x_i , we obtain

$$\frac{\partial h(X|Y = y_i)}{\partial x_i} = -\frac{1}{T} \sum_{t_j \in T} \frac{1}{p(x_j|y_i)} \frac{1}{T_{y_i}} \sum_{t_m \in T_{y_i}} \frac{\partial K(x_j - x_m)}{\partial x_i}. \quad (\text{C.4})$$

Note that the PDF, being estimated from the data, is also a function of the N data points x_m . Thus, as x_i changes, $p(x_i|y_i)$ changes both from its argument and from the behavior of $p(\cdot)$ itself. The last term in (C.4) is equal to

$$\frac{\partial K(x_j - x_m)}{\partial x_i} = \begin{cases} -\nabla K(x_j - x_m) \delta_{m-i} & j \neq i \\ (1 - \delta_{m-i}) \nabla K(x_j - x_m) & j = i \end{cases} \quad (\text{C.5})$$

where

$$\delta_{m-i} = \begin{cases} 1 & m = i \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.6})$$

Thus by substituting (C.5) into (C.4) we have

$$\begin{aligned} \frac{\partial h(X|Y=y_i)}{\partial x_i} &= -\frac{1}{|T|} \frac{\nabla p(x_i|y_i)}{p(x_i|y_i)} \\ &\quad + \frac{1}{|T|} \frac{1}{|T_{y_i}|} \sum_{t_j \in T} \frac{\nabla K(x_j - x_i)}{p(x_j|y_i)}. \end{aligned} \quad (\text{C.7})$$

By multiplying numerator and denominator of the first term in (C.7) with $p(y_i)$

$$\frac{\nabla p(x_i|y_i)}{p(x_i|y_i)} \cdot \frac{p(y_i)}{p(y_i)} = \frac{\nabla p(z_i)}{p(z_i)} \cdot \frac{\partial z_i}{\partial x_i}, \quad (\text{C.8})$$

where the projection operator $\partial z_i / \partial x_i$ is because we change from x_i to z_i . Finally, the derivative of (4.12) is

$$\frac{\partial h(X|Y=y_i)}{\partial x_i} = -\frac{1}{|T|} \frac{\nabla p(z_i)}{p(z_i)} \frac{\partial z_i}{\partial x_i} + \chi(x_i), \quad (\text{C.9})$$

with

$$\chi(x_i) = \frac{1}{|T|} \frac{1}{|T_{y_i}|} \sum_{t_j \in T} \frac{\nabla K(x_j - x_i)}{p(x_j|y_i)}. \quad (\text{C.10})$$

The term $\chi(\cdot)$ of Eq. (C.10) can be expressed in a closed form when the kernel $K(\cdot)$ has a narrow window size. Indeed, in this case, only samples very close to each estimation point will contribute to the pdf. Under this assumption the conditional pdf is $p(s|y_i) \approx N_s / |T_{y_i}|$, where N_s is the number of pixels equal to s . Thus by substituting we have

$$\begin{aligned} \chi(x_i) &\approx \frac{1}{|T|} \sum_{t_j \in T} \frac{\nabla K(x_j - x_i)}{N_{x_j}} \\ &\approx \frac{1}{|T|} \int_{X(T)} \nabla K(\alpha - x_i) d\alpha, \end{aligned} \quad (\text{C.11})$$

where $X(T)$ is the set of the image values.

Since $\nabla K(\cdot)$ is an odd function, $\chi(\cdot)$ is zero if x_i is such that the support of $\nabla K(\alpha - x_i)$ is contained by the support of $X(\cdot)$ and assumes nonzero values in a ring near the boundary of the latter. Therefore, we observe that $\chi(x_i)$ is negligible for almost every value assumed by x_i .

Appendix D

PCkNN: comparative tables

In this appendix we report for completeness the PSNR and SSIM tables for all experiments of Chapter 3. We considered several images commonly used in the AWGN denoising and shown in Figure D.1. Noisy images are obtained by adding a simulated white Gaussian noise with standard deviation σ of 5, 10, 15, 20 and 25. We recall here that we compared PCkNN with three state-of-the-art denoising algorithms: the UINTA algorithm [AW06], the Non Local Means (NLmeans) [BCM05], and the Block Matching 3D (BM3D) algorithm [DFKE07a], all briefly described in Section 4.2. For all these algorithms, the free parameters are set as described in Section 4.4 and reported for PCkNN in Table D.1.

In Tab. D.2 to D.5, we report results for the whole database of images for noise $\sigma = 5, 10, 15, 20$ and 25. The best two PSNR for each case is put in boldface for the sake of clarity. Also, in Figure D.2 PSNR and SSIM curves averaged over the 13 test images are plotted.

patch radius	7
search radius	15
# of neighbors	20
filtering parameter σ_w	σ
PCA dimension	50

Table D.1: Parameters setting for the PCkNN algorithm



Figure D.1: Image dataset used for algorithm comparison

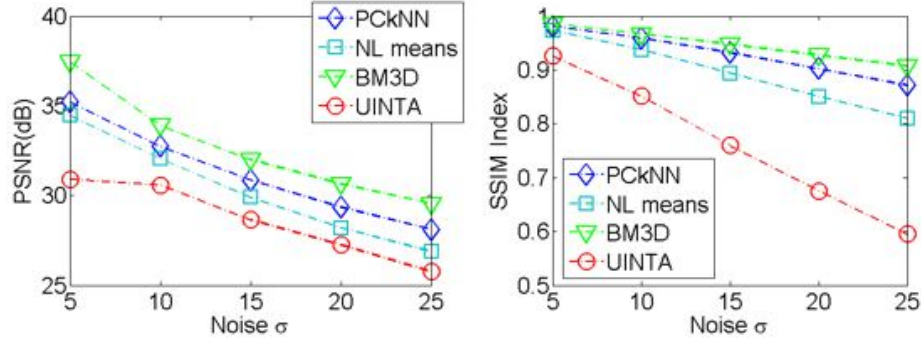


Figure D.2: Averaged PSNR and SSIM measures for the different algorithms.

↓ Image / σ →		5	10	15	20	25
Airplane	PCKNN	37.42	34.67	32.50	30.80	29.37
	NLmeans	36.43	34.13	32.04	30.29	28.74
	UINTA	27.84	31.64	29.48	27.67	26.13
	BM3D	39.35	35.96	34.01	32.63	31.42
Barbara	PCKNN	35.81	33.17	31.33	29.84	28.61
	NLmeans	35.75	33.39	31.02	29.03	27.36
	UINTA	34.19	32.56	29.83	27.68	25.98
	BM3D	38.28	34.92	33.03	31.64	30.44
Boat	PCKNN	35.37	32.77	30.80	29.17	27.94
	NLmeans	34.67	31.94	29.70	27.98	26.62
	UINTA	34.87	32.00	29.53	27.60	25.99
	BM3D	37.27	33.89	32.07	30.75	29.67
Couple	PCKNN	35.48	32.85	30.79	29.14	27.84
	NLmeans	34.55	31.92	29.35	27.35	25.93
	UINTA	34.85	31.93	29.408	27.45	25.79
	BM3D	37.49	34.00	32.04	30.63	29.48
Elaine	PCKNN	35.37	33.25	32.07	31.12	30.24
	NLmeans	34.36	32.31	31.09	30.03	29.04
	UINTA	35.09	32.58	29.97	27.92	26.23
	BM3D	36.75	33.36	32.16	31.43	30.79
Fingerprint	PCKNN	33.74	31.07	29.10	27.53	26.14
	NLmeans	33.07	30.89	28.42	26.53	25.09
	UINTA	32.94	30.67	28.49	26.64	25.18
	BM3D	36.49	32.44	30.28	28.79	27.62

Table D.2: PSNR (dB) results for the test images.

\downarrow Image/ $\sigma \rightarrow$		5	10	15	20	25
Goldhill	PCKNN	35.55	32.70	30.81	29.49	28.45
	NLmeans	34.83	31.73	29.58	28.01	26.83
	UINTA	34.54	32.24	29.67	27.61	26.00
	BM3D	37.17	33.65	31.89	30.69	29.71
Lena	PCKNN	37.62	35.16	33.27	31.74	30.46
	NLmeans	36.98	34.45	32.26	30.58	29.28
	UINTA	31.75	33.12	30.29	28.10	26.36
	BM3D	38.70	35.89	34.25	33.02	31.98
Man	PCKNN	34.36	31.58	29.41	27.78	26.59
	NLmeans	33.72	30.87	28.57	26.89	25.58
	UINTA	32.86	31.50	29.18	27.31	25.71
	BM3D	37.20	33.07	30.73	29.12	27.87
Mandrill	PCKNN	30.36	28.55	26.87	25.35	23.96
	NLmeans	29.01	27.71	25.78	23.93	22.71
	UINTA	23.11	27.43	26.69	25.37	24.17
	BM3D	35.24	30.57	28.17	26.60	25.42
Peppers	PCKNN	36.24	34.05	32.31	30.85	29.58
	NLmeans	35.70	33.72	31.89	30.24	28.78
	UINTA	31.16	31.93	29.80	27.84	26.15
	BM3D	37.62	34.91	33.40	32.18	31.08
Stream	PCKNN	31.40	29.28	27.38	25.77	24.71
	NLmeans	30.34	28.26	26.04	24.57	23.55
	UINTA	30.57	29.05	27.50	26.06	24.74
	BM3D	35.73	31.12	28.69	27.11	25.97
Zelda	PCKNN	39.08	36.43	34.46	32.81	31.40
	NLmeans	38.49	35.37	32.93	31.15	29.75
	UINTA	17.98	20.98	22.37	26.87	26.35
	BM3D	39.41	36.90	35.23	33.84	32.58

Table D.3: Tab. D.2 (continued). PSNR (dB) results for the test images.

\downarrow Image / $\sigma \rightarrow$		5	10	15	20	25
Airplane	PCKNN	0.982	0.964	0.943	0.923	0.901
	NLmeans	0.973	0.949	0.925	0.902	0.879
	UINTA	0.934	0.839	0.727	0.623	0.535
	BM3D	0.984	0.970	0.956	0.943	0.931
Barbara	PCKNN	0.985	0.969	0.953	0.933	0.912
	NLmeans	0.982	0.962	0.934	0.901	0.866
	UINTA	0.952	0.888	0.803	0.715	0.639
	BM3D	0.989	0.977	0.964	0.952	0.937
Boat	PCKNN	0.982	0.955	0.923	0.888	0.856
	NLmeans	0.973	0.925	0.873	0.825	0.784
	UINTA	0.925	0.850	0.755	0.663	0.580
	BM3D	0.986	0.967	0.946	0.924	0.903
Couple	PCKNN	0.983	0.959	0.930	0.898	0.865
	NLmeans	0.972	0.929	0.876	0.819	0.766
	UINTA	0.935	0.863	0.772	0.683	0.596
	BM3D	0.986	0.967	0.947	0.926	0.905
Elaine	PCKNN	0.969	0.947	0.929	0.911	0.893
	NLmeans	0.963	0.930	0.903	0.877	0.853
	UINTA	0.905	0.815	0.708	0.606	0.515
	BM3D	0.979	0.954	0.935	0.920	0.905
Fingerprint	PCKNN	0.994	0.987	0.976	0.959	0.936
	NLmeans	0.993	0.983	0.958	0.926	0.890
	UINTA	0.971	0.952	0.923	0.889	0.850
	BM3D	0.997	0.990	0.982	0.971	0.959

Table D.4: SSIM ($\in [0, 1]$) results for the test images.

\downarrow Image / $\sigma \rightarrow$		5	10	15	20	25
Goldhill	PCKNN	0.979	0.944	0.906	0.872	0.841
	NLmeans	0.967	0.911	0.856	0.805	0.757
	UINTA	0.928	0.846	0.747	0.648	0.563
	BM3D	0.984	0.957	0.931	0.907	0.884
Lena	PCKNN	0.982	0.964	0.946	0.926	0.904
	NLmeans	0.975	0.948	0.920	0.893	0.868
	UINTA	0.928	0.837	0.726	0.617	0.524
	BM3D	0.984	0.969	0.955	0.940	0.925
Man	PCKNN	0.978	0.943	0.899	0.855	0.817
	NLmeans	0.969	0.911	0.848	0.793	0.744
	UINTA	0.929	0.854	0.763	0.674	0.592
	BM3D	0.983	0.953	0.920	0.888	0.856
Mandrill	PCKNN	0.977	0.948	0.909	0.859	0.797
	NLmeans	0.964	0.917	0.847	0.766	0.691
	UINTA	0.913	0.862	0.802	0.741	0.677
	BM3D	0.987	0.963	0.934	0.904	0.872
Peppers	PCKNN	0.978	0.959	0.940	0.920	0.899
	NLmeans	0.970	0.947	0.921	0.895	0.870
	UINTA	0.922	0.828	0.720	0.616	0.523
	BM3D	0.981	0.963	0.948	0.933	0.918
Stream	PCKNN	0.983	0.957	0.915	0.857	0.807
	NLmeans	0.974	0.923	0.843	0.769	0.705
	UINTA	0.906	0.860	0.801	0.735	0.667
	BM3D	0.990	0.969	0.941	0.908	0.874
Zelda	PCKNN	0.985	0.968	0.948	0.927	0.905
	NLmeans	0.980	0.953	0.921	0.890	0.862
	UINTA	0.875	0.770	0.645	0.567	0.483
	BM3D	0.985	0.971	0.955	0.939	0.922

Table D.5: Tab. D.4 (continued). SSIM ($\in [0, 1]$) results for the test images.

Bibliography

- [AA76] H. H. Arsenault and G. April, *Properties of speckle integrated with a finite aperture and logarithmically transformed*, J. Opt. Soc. Am. **66** (1976), no. 11, 1160–1163. [81](#)
- [ADB08] C. V. Angelino, E. Debreuve, and M. Barlaud, *Image restoration using a knn-variant of the mean-shift*, IEEE ICIP (San Diego, CA, USA), October 2008. [47](#)
- [AGPP09] Jan Aelterman, Bart Goossens, Aleksandra Pizurica, and Wilfried Philips, *Suppression of correlated noise*, Recent Advances in Signal Processing (Ashraf A Zaher, ed.), InTech, 2009, pp. 211–236. [xiii](#), [70](#), [71](#)
- [AL76] I. A. Ahmad and P. Lin, *A nonparametric estimation of the entropy for absolutely continuous distributions*, IEEE Transactions On Information Theory (1976), 372–375. [10](#), [25](#), [45](#), [119](#)
- [AL84] H. H. Arsenault and M. Levesque, *Combined homomorphic and local-statistics processing for restoration of images degraded by signal- dependent noise*, Appl. Opt. **23** (1984), no. 6, 845–850. [81](#)
- [Ash01] Michael Ashikhmin, *Synthesizing natural textures*, In ACM Symposium on Interactive 3D Graphics, 2001, pp. 217–226. [109](#)
- [ATB03] A. Achim, P. Tsakalides, and A. Bezarianos, *Sar image denoising via bayesian wavelet shrinkage based on heavy-tailed modeling*, IEEE Transactions on Geoscience and Remote Sensing **41** (2003), no. 8, 1773–1784. [82](#)
- [AW06] S. P. Awate and R. T. Whitaker, *Unsupervised, information-theoretic, adaptive image filtering for image restoration*, IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006), no. 3, 364–376. [39](#), [40](#), [43](#), [47](#), [49](#), [115](#), [131](#)
- [BA93] M. J. Black and P. Anandan, *A framework for the robust estimation of optical flow*, International Conference on Computer Vision, 1993. [24](#)

-
- [BAS07] M.I.H. Bhuiyan, M.O. Ahmad, and M.N.S. Swamy, *Spatially adaptive wavelet based method using the cauchy prior for denoising the sar images*, IEEE Transactions on Circuits and Systems for Video Technology **17** (2007), no. 4, 500–507. [82](#), [90](#)
 - [BCJ05] A. Buades, B. Coll, and J.M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul. **4** (2005), no. 2, 490–530. [41](#), [42](#), [83](#)
 - [BCM05] A. Buades, B. Coll, and J.M. Morel, *A non-local algorithm for image denoising*, IEEE International Conference on Computer Vision and Pattern Recognition (Washington, DC, USA), 2005, pp. 60–65. [39](#), [40](#), [47](#), [48](#), [52](#), [131](#)
 - [BKB07] J. Boulanger, C. Kervrann, and P. Bouthemy, *Space-time adaptation for patch-based image sequence restoration*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007), 1096–1102. [40](#)
 - [BSCB00] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester, *Image inpainting*, Proceedings of the 27th annual conference on Computer graphics and interactive techniques (New York, NY, USA), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424. [xiv](#), [105](#), [107](#)
 - [Che95] Yizong Cheng, *Mean shift, mode seeking, and clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995), no. 8, 790–799. [11](#)
 - [CHKB08] P. Coupé, P. Hellier, C. Kervrann, and C. Barillot, *Bayesian non local means-based speckle filtering*, 5th IEEE International Symposium on Biomedical Imaging, May 2008, pp. 1291–1294. [83](#)
 - [CIdSZ08] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian, *On the local behavior of spaces of natural images*, Int. J. Comput. Vision **76** (2008), 1–12. [40](#)
 - [CLF09] C. Deledalle, L. Denis, and F. Tupin, *Iterative weighted maximum likelihood denoising with probabilistic patch-based weights*, IEEE Transactions on Image Processing **18** (2009), no. 12, 2661–2672. [83](#), [86](#), [87](#), [90](#)
 - [CM02] D. Comaniciu and P. Meer, *Mean shift: A robust approach toward feature space analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002), 603–619. [11](#), [12](#), [47](#)
 - [Com03] D. Comaniciu, *An algorithm for data-driven bandwidth selection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003), 281–288. [28](#)

-
- [CPT04] A. Criminisi, P. Pérez, and K. Toyama, *Region filling and object removal by exemplar-based image inpainting*, IEEE Transactions on Image Processing **13** (2004), 1200–1212. [40](#), [109](#), [110](#)
- [CS01] Tony F. Chan and Jianhong Shen, *Non-texture inpainting by curvature-driven diffusions (cdd)*, J. Visual Comm. Image Rep **12** (2001), 436–449. [107](#)
- [CS02] ———, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math **62** (2002), 1019–1043. [107](#)
- [CT91] T. Cover and J. Thomas, *Elements of information theory*, first ed., Wiley-Interscience, 1991. [24](#), [25](#), [44](#)
- [DB97] Jeremy S. De Bonet, *Multiresolution sampling procedure for analysis and synthesis of texture images*, Proceedings of the 24th annual conference on Computer graphics and interactive techniques (New York, NY, USA), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 361–368. [108](#)
- [DFKE07a] K. Dabov, A. Foi, V. K., and K. Egiazarian, *Image denoising by sparse 3d transform-domain collaborative filtering*, IEEE Transactions Image Processing **16** (2007), 2007. [39](#), [40](#), [48](#), [52](#), [131](#)
- [DFKE07b] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse 3D transform-domain collaborative filtering*, IEEE Transactions on Image Processing **16** (2007), no. 8, 2080–2095. [xii](#), [41](#), [42](#), [43](#), [83](#), [89](#), [90](#)
- [Don95] D.J. Donoho, *De-noising by soft-thresholding*, IEEE Transactions on Information Theory **41** (1995), no. 3, 613–627. [41](#), [84](#)
- [DPCL04] M. Dai, C. Peng, A. K. Chan, and D. Loguinov, *Bayesian wavelet shrinkage with edge detection for sar image despeckling*, IEEE Transactions on Geoscience and Remote Sensing **42** (2004), no. 8, 1642–1648. [83](#)
- [DT74] Yu. G. Dmitriev and F. P. Tarasenko, *On the estimation of functionals of the probability density and its derivatives*, Theory of Probability and its Applications **18** (1974), no. 3, 628–633. [10](#)
- [EL99] Alexei Efros and Thomas Leung, *Texture synthesis by non-parametric sampling*, In International Conference on Computer Vision, 1999, pp. 1033–1038. [106](#), [108](#), [109](#)
- [Ela06] M. Elad, *Why simple shrinkage is still relevant for redundant representations?*, IEEE Transactions on Information Theory **52** (2006), no. 12, 5559–5569. [85](#)

-
- [Epa69] V.A. Epanechnikov, *Nonparametric estimation of a multivariate probability density*, Theory Prob. Appl. **14** (1969), 153–158. [13](#)
 - [FA02] F.Argenti and A.Alparone, *Speckle removal from sar images in the undecimated wavelet domain*, IEEE Transactions on Geoscience and Remote Sensing **40** (2002), 2363–2374. [83](#), [85](#), [88](#), [90](#)
 - [FH75] K. Fukunaga and L. D. Hostetler, *The estimation of the gradient of a density function, with applications in pattern recognition*, IEEE Transactions On Information Theory **21** (1975), 32–40. [11](#), [14](#), [26](#), [27](#), [47](#)
 - [FSSH82] V. S. Frost, J. A. Stiles, K. S. Shanmugan, and J. C. Holtzman, *A model for radar images and its application to adaptive digital filtering of multiplicative noise*, IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-4** (1982), no. 2, 157–166. [82](#), [90](#)
 - [FTA06] F.Argenti, T.Bianchi, and A.Alparone, *Multiresolution map despeckling of sar images based on locally adaptive generalized gaussian pdf modeling*, IEEE Transactions on Image Processing **15** (2006), no. 11, 3385–3399. [83](#), [90](#)
 - [FTA08] ———, *Segmentation-based map despeckling of sar images in the undecimated wavelet domain*, IEEE Transactions on Geoscience and Remote Sensing **46** (2008), no. 9, 2728–2742. [83](#), [90](#)
 - [FTKE08] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian, *Practical poissonian-gaussian noise modeling and fitting for single-image raw-data*, IEEE Transactions on Image Processing **17** (2008), no. 10, 1737–1754. [69](#)
 - [Gao10] G. Gao, *Statistical modeling of sar images: a survey*, Sensors (2010), 775–794. [82](#)
 - [GJ97] L. Gagnon and A. Jouan, *Speckle filtering of sar images: a comparative study between complex wavelet-based and standard filters*, Proc. SPIE, vol. 3169, 1997, pp. 80–91. [82](#)
 - [GLMI05] M. N. Goria, N. N. Leonenko, V. V. Mergel, and P. Inverardi, *A new class of random vector entropy estimators and its applications in testing statistical hypotheses*, Journal of Nonparametric Statistics **17** (2005), 277–297. [11](#)
 - [GN04] Michael D. Grossberg and Shree K. Nayar, *Modeling the space of camera response functions*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), no. 10, 1272–1282. [70](#)

-
- [GSB97] S.P. Ghael, A.M. Sayeed, and R.G. Baraniuk, *Improved wavelet denoising via empirical wiener filtering*, Proc. SPIE, vol. 5, july 1997, pp. 389–399. [41](#), [42](#), [88](#), [89](#)
- [GV97] Jonas Gomes and Luiz Velho, *Image processing for computer graphics*, 1st ed., Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997. [107](#)
- [GW02] R. C. Gonzalez and R. E. Woods, *Digital image processing*, second ed., Prentice Hall, 2002. [21](#)
- [Hal83] P. Hall, *On near neighbor estimates of a multivariate density*, Journal of Multivariate Analysis **12** (1983), 24–39. [14](#)
- [HJL09] H.Zhong, J.Xu, and L.Jiao, *Classification based nonlocal means despeckling for sar image*, proc. of SPIE, vol. 7495, October 2009. [83](#)
- [HJM⁺94] H.Guo, J.E.Odegard, M.Lang, R.A.Gopinath, L.W.Selesnick, and C.S.Burrus, *Wavelet based speckle reduction with application to sar based atd/r*, Proceedings of IEEE International Conference on Image Processing, vol. 1, 1994, pp. 75–79. [82](#)
- [HK94] Glenn E. Healey and Raghava Kondepudy, *Radiometric ccd camera calibration and noise estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994), no. 3, 267–276. [xiii](#), [66](#), [67](#), [68](#), [69](#)
- [HLF02] H.Xie, L.E.Pierce, and F.T.Ulaby, *Despeckling sar images using a low complexity wavelet denoising process*, Proceedings of IEEE International Geoscience and Remote Sensing Symposium, vol. 1, November 2002, pp. 321–324. [83](#), [88](#)
- [HM99] J. Huang and D. Mumford, *Statistics of natural images and models*, IEEE International Conference on Computer Vision and Pattern Recognition, 1999, pp. 541–547. [40](#)
- [HNO06] P. C. Hansen, J. G. Nagy, and D. P. O’Leary, *Deblurring images. matrices, spectra and filtering*, first ed., SIAM, 2006. [27](#), [28](#)
- [JFWJ03] F. Jin, P. Fieguth, L. Winger, and E. Jernigan, *Adaptive wiener filtering of noisy images and image sequences*, Proceedings of IEEE International Conference on Image Processing, vol. 3, September 2003, pp. 349–352. [88](#)
- [Joe89] Harry Joe, *Estimation of entropy and other functionals of a multivariate density*, Annals of the Institute of Statistical Mathematics **41** (1989), no. 4, 683–697. [10](#)

-
- [KAPB05] G. Sapiro K. A. Patwardhan and M. Bertalmio, *Video inpainting of occluding and occluded objects*, Image Processing, 2005. ICIIP 2005. IEEE International Conference on, vol. 2, 2005, pp. 69–72. [104](#)
- [Kay93] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*, Prentice Hall, Englewood Cliffs, NJ, 1993. [87](#)
- [Kaz03] M. Kazubek, *Wavelet domain image denoising by thresholding and wiener filtering*, IEEE Signal Processing Letters **10** (2003), 324–326. [88](#)
- [KF99] A. H. Kam and W. J. Fitzgerald, *General unsupervised multiscale segmentation of images*, Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers, vol. 1, 24–27 Oct. 1999, pp. 63–67. [11](#)
- [KFEA10] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, *From local kernel to nonlocal multiple-model image denoising*, International Journal of Computer Vision **86** (2010), no. 1, 1–30. [43](#), [83](#)
- [KL87] L. Kozachenko and N. N. Leonenko, *On statistical estimation of entropy of random vector*, Problems Infor. Transmiss. **23** (1987), 95–101. [10](#)
- [KSSC85] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel, *Adaptive noise smoothing filter for images with signal-dependent noise*, IEEE Transactions on Pattern Analysis and Machine Intelligence **7** (1985), 165–177. [82](#)
- [KSSC87] ———, *Adaptive restoration of images with speckle*, IEEE Transactions on Acoustics, Speech, and Signal Processing **35** (1987), 373–383. [82](#)
- [Lee80] J. S. Lee, *Digital image enhancement and noise filtering by use of local statistics*, IEEE Transactions on Pattern Analysis and Machine Intelligence **2** (1980), 165–168. [81](#), [82](#)
- [Lee81] ———, *Refined filtering of image noise using local statistics*, Comput. Graph. Image Process. **15** (1981), 380–389. [81](#), [82](#)
- [LFG06] Jan Lukas, Jessica Fridrich, and Miroslav Goljan, *Digital camera identification from sensor pattern noise*, IEEE Transactions on Information Forensics and Security **1** (2006), no. 2, 205–214. [70](#)
- [LGO96] M. Lang, H. Guo, and J. E. Odegard, *Noise reduction using undecimated discrete wavelet transform*, IEEE Signal Processing Letters **3** (1996), no. 1, 10–12. [85](#)

-
- [LJDaA94] J. S. Lee, L. Jurkevich, P. Dewaele, and P. Wambacq and A. Oosterlinck, *Speckle filtering of synthetic aperture radar images: A review*, *Remote Sensing Reviews* **8** (1994), no. 4, 313–340. [91](#), [92](#)
- [LK06] Qiming Luo and T. M. Khoshgoftaar, *Unsupervised multiscale color image segmentation based on mdl principle*, *IEEE Transactions on Image Processing* **15** (2006), no. 9, 2755–2761. [11](#)
- [LNTH90] A. Lopes, E. Nezry, R. Touzi, and H. Laur, *Maximum a posteriori speckle filtering and first order texture models in sar images*, *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, vol. 3, May 1990, pp. 2409–2412. [82](#)
- [LPM03] A. B. Lee, K. S. Pedersen, and D. Mumford, *The nonlinear statistics of high-contrast patches in natural images*, *Int. J. Comput. Vision* **54** (2003), no. 1-3, 83–103. [40](#)
- [LQ65] D. O. Loftsgaarden and C. P. Quesenberry, *A nonparametric estimate of a multivariate density function*, *Ann. Math. Stat.* **36** (1965), 1049–1051. [13](#)
- [LSK⁺08] Ce Liu, Richard Szeliski, Sing Bing Kang, C. Lawrence Zitnick, and William T. Freeman, *Automatic estimation and removal of noise from a single image*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30** (2008), no. 2, 299–314. [70](#)
- [LTN90] A. Lopes, R. Touzi, and E. Nezry, *Adaptive speckle filters and scene heterogeneity*, *IEEE Transactions on Geoscience and Remote Sensing* **28** (1990), 992–1000. [82](#)
- [Luc74] L. B. Lucy, *An iterative technic for the rectification of observed distribution*, *The Astronomical Journal* **79** (1974), no. 6, 745–754. [27](#), [28](#)
- [Mal98] S. Mallat, *A wavelet tour of signal processing*, Academic, Boston, MA, 1998. [41](#)
- [MKR99] M. K. Mihcak, I. Kozintsev, and K. Ramchandran, *Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising*, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (Phoenix, AZ, USA), March 1999, pp. 3253–3256. [88](#)
- [ML07] Y. Matsushita and S. Lin, *A probabilistic intensity similarity measure based on noise distributions*, *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8. [85](#)

-
- [MR79] Y. Mack and M. Rosenblatt, *Multivariate k -nearest neighbor density estimates*, Journal of Multivariate Analysis **9** (1979), 1–15. [14](#)
- [Nak05] Junichi Nakamura, *Image sensor and signal processing for digital still camera*, CRC Press (Imprint of Taylor & Francis Group), august 2005. [xii](#), [66](#)
- [NK07] Mattias Nilsson and W. Bastiaan Kleijn, *On the estimation of differential entropy from data located on embedded manifolds*, IEEE Transactions on Information Theory **53** (2007), no. 7, 2330–2341. [xi](#), [15](#), [16](#)
- [NLT91] E. Nezry, A. Lopes, and R. Touzi, *Detection of structural and textural features for sar image filtering*, Proceedings of IEEE International Geoscience and Remote Sensing Symposium, vol. 3, 1991, pp. 2169–2172. [82](#)
- [OE99] R. Öktem and K. Egizarian, *Transform domain algorithm for reducing effect of film-grain noise in image compression*, Electronic Letters **35** (1999), no. 21, 1830–1831. [83](#), [88](#)
- [OEW08] Jeff Orchard, Mehran Ebrahimi, and Alexander Wong, *Efficient nonlocal-means denoising using the svd.*, ICIP’08, 2008, pp. 1732–1735. [62](#)
- [OQ04] C. Oliver and S. Quegan, *Understanding synthetic aperture radar images*, SciTech, 2004. [81](#), [86](#), [95](#)
- [PSB07] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, *Video inpainting under constrained camera motion*, IEEE Transactions On Image Processing **16** (2007), no. 2, 545–553. [104](#)
- [Ric72] W. H. Richardson, *Bayesian-based iterative method of image restoration*, J. Opt. Soc. Am. **62** (1972), no. 1, 55–59. [27](#), [28](#)
- [RSB03] S. Rane, Guillermo Sapiro, and Marcelo Bertalmío, *Structure and texture filling-in of missing image blocks in wireless transmission and compression applications*, IEEE Transactions on Image Processing **12** (2003), no. 3, 296–303. [104](#)
- [Sco92] D. Scott, *Multivariate density estimation: Theory, practice, and visualization*, second ed., Wiley, 1992. [7](#), [13](#)
- [SE04] S. Solbø and T. Eltoft, *Homomorphic wavelet-based statistical despeckling of sar images*, IEEE Transactions on Geoscience and Remote Sensing **42** (2004), no. 4, 711–721. [82](#)

-
- [SGJ01] S.Foucher, G.B.Bénié, and J.M.Boucher, *Multiscale map filtering of sar images*, IEEE Transactions on Image Processing **10** (2001), 49–60. [83](#)
- [Sil86] B. W. Silverman, *Density estimation for statistics and data analysis*, Chapman and Hall/CRC, April 1986. [7](#)
- [SPH09] W.F. Sun, Y.H. Peng, and W.L. Hwang, *Modified similarity metric for non-local means algorithm*, Electronics Letters **45** (2009), no. 25, 1307–1309. [59](#)
- [Tas09] Tolga Tasdizen, *Principal neighborhood dictionaries for nonlocal means image denoising*, IEEE Transactions Image Processing **18** (2009), 2649–2660. [48](#), [62](#)
- [TLB88] R. Touzi, A. Lopes, and P. Bousquet, *A statistical and geometrical edge detector for sar images*, IEEE Transactions on Geoscience and Remote Sensing **26** (1988), no. 6, 764–773. [82](#)
- [Tou02] R. Touzi, *A review of speckle filtering in the context of estimation theory*, IEEE Transactions on Geoscience and Remote Sensing **40** (2002), no. 11, 2392–2404. [82](#)
- [TS92] G.R. Terrell and D.W. Scott, *Variable kernel density estimation*, Annals of Statistics **20** (1992), 1236–1265. [6](#), [14](#)
- [WBSS04] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Transactions on Image Processing **13** (2004), 600–612. [28](#), [52](#)
- [Wie49] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*, Technology Press and John Wiley & Sons, Inc., New York, 1949. [23](#), [27](#)
- [WJ95] M. P. Wand and M. Jones, *Kernel smoothing*, Chapman and Hall, 1995. [6](#), [12](#)
- [WL00] Li-Yi Wei and Marc Levoy, *Fast texture synthesis using tree-structured vector quantization*, Proceedings of the 27th annual conference on Computer graphics and interactive techniques (New York, NY, USA), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 479–488. [109](#)
- [WR06] Jiying Wu and Qiuqi Ruan, *Object removal by cross isophotes exemplar-based inpainting*, ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (Washington, DC, USA), IEEE Computer Society, 2006, pp. 810–813. [109](#)

- [XPU02] H. Xie, L.E. Pierce, and F.T. Ulaby, *Statistical properties of logarithmically transformed speckle*, IEEE Transactions on Geoscience and Remote Sensing **40** (2002), no. 3, 721–727. [86](#), [89](#), [91](#)
- [YC86] P. F. Yan and C. H. Chen, *An algorithm for filtering multiplicative noise in wide range*, Traitement du Sig. **3** (1986), no. 2, 91–96. [81](#)
- [ZBW05] L. Zhang, P. Bao, and X. Wu, *Multiscale lmmse-based image denoising with optimal wavelet selection*, IEEE Transactions on Circuits and Systems for Video Technology **15** (2005), no. 4, 469–481. [85](#)
- [ZWM98] Song Chun Zhu, Yingnian Wu, and David Mumford, *Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling*, Int. J. Comput. Vision **27** (1998), 107–126. [108](#)