

UNIVERSITÉ PAUL CEZANNE (AIX-MARSEILLE III)

N° attribué par la bibliothèque
2007AIX30051

Titre :

MÉTHODES PAR SOUS-ESPACES ET D'OPTIMISATION : APPLICATION AU TRAITEMENT D'ANTENNE, A L'ANALYSE D'IMAGES, ET AU TRAITEMENT DE DONNÉES TENSORIELLES

THÈSE
pour obtenir le grade de

DOCTEUR
DE L'UNIVERSITÉ PAUL CEZANNE
Faculté des Sciences et Techniques

Discipline : Optique, Image et Signal

Présentée et soutenue publiquement par :

Julien MAROT

Directeur de thèse : Pr. Salah BOURENNANE
École Doctorale : Physique et Sciences de la Matière

JURY;

Rapporteurs : Pr. Eric MOREAU Univ. de Toulon et du Var
Pr. Yide WANG Univ. de Nantes

Examinateurs : Pr. Jean-Pierre SESSAREGO Lab. de Mécanique et d'Acoustique - CNRS
M. Hamid AGHAJAN Stanford University, CA, USA
M. Jacques BLANC-TALON DGA, MRIS

Directeur de thèse : Pr. Salah BOURENNANE **Ecole Centrale Marseille, Institut Fresnel, Marseille**

Abstract

THIS thesis is devoted to subspace-based and optimization methods. The methods proposed in this dissertation are developed and adapted in three contexts : for applications in array processing, that is, for one-dimensional signal processing, for image understanding purposes, that is, for two-dimensional signal processing, and for tensor signal processing, that is, for multidimensional signal processing.

The first part of the manuscript deals with array processing methods and applications. Basic definitions which are relevant in array processing are presented. The interest of second order statistics is emphasized : within the measurement space spanned by the eigenvectors of a covariance matrix, one distinguishes signal subspace eigenvectors associated with the dominant eigenvalues, and noise subspace eigenvectors. High-resolution methods of array processing, such as MUSIC (MUltiple SIgnal Characterization) and TLS-ESPRIT (Total-Least-Squares Estimation of Signal Parameters via Rotational Invariance Techniques) methods, are based on the orthogonality between signal and noise subspaces. A novel optimization method is presented. It combines either gradient or DIRECT (DIviding RECTangles) algorithms with spline interpolation. An array processing application is considered : We propose a novel method for source localization in presence of phase distortions. Especially, a novel algorithm handles a distorted antenna composed of a large number of sensors, keeping a small computational load. The second part of the manuscript considers image understanding applications. We review SLIDE method which adapts array processing methods to straight line retrieval. We propose novel fast subspace-based methods for the estimation of straight line orientations and offsets. We consider several optimization methods for nearly straight distorted contour retrieval. Especially, we propose a novel optimization method which is fast and handles distortions with high curvature. Then, we consider star-shaped contour retrieval. For this, we adapt a virtual circular antenna to the processed image. The corresponding signal generation scheme yields linear phase signals from concentric circles. This enables the use of high-resolution methods which distinguish between possibly close radius values. We consider various contour shapes such as distorted circles or ellipses.

The third part of the manuscript is concerned with tensor signal processing. We remind the definitions concerning tensors, and review several multiway signal processing methods, which rely on data projection upon signal subspace : the lower rank- (K_1, \dots, K_N) truncation of the HOSVD, the rank- (K_1, \dots, K_N) tensor approximation, in particular its version based on fourth order cumulants, which handles the case of correlated Gaussian noise. We present a multiway version of Wiener filtering. Both rank- (K_1, \dots, K_N) tensor approximation and multiway Wiener filtering rely on an optimization algorithm named Alternating Least Squares (ALS) loop. We propose a nonorthogonal tensor flattening method to improve denoising results. For this we use the straight line detection method proposed in the first part.

Key words : Array, second order statistics, subspace-based method, high-resolution method, distorted wavefront, estimation, array processing, optimization, spline interpolation, image, contour, linear antenna, circular antenna, multilinear algebra, tensor decomposition, multiway array, lower-rank approximation, filtering.

Résumé

CETTE thèse est consacrée aux méthodes par sous-espaces et d'optimisation. Les méthodes proposées dans ce manuscrit sont développées et adaptées dans trois contextes : pour des applications en traitement d'antenne, c'est-à-dire pour le traitement de signaux unidimensionnels, pour l'analyse d'images, c'est-à-dire de signaux bidimensionnels, pour le traitement du signal tensoriel, c'est-à-dire de signaux multidimensionnels. La première partie de ce manuscrit traite de méthodes par sous-espaces et d'optimisation et de leurs applications en traitement d'antenne. Des définitions concernant un problème de traitement d'antenne sont présentées, l'intérêt de statistiques d'ordre deux est mis en valeur : dans l'espace des mesures couvert par les vecteurs propres d'une matrice de covariance, l'on distingue les vecteurs propres du sous espace signal associés aux valeurs propres dominantes, et les vecteurs propres du sous-espace bruit. Les méthodes haute résolution du traitement d'antenne, telles que MUSIC (MULTiple SIgnal Characterization) et TLS-ESPRIT (Total-Least-Squares Estimation of Signal Parameters via Rotational Invariance Techniques), sont fondées sur l'orthogonalité entre sous-espace bruit et sous-espace signal. On présente une nouvelle méthode d'optimisation combinant les méthodes du gradient ou DIRECT (DIviding RECTangles) et une méthode d'interpolation par splines. On propose une application pour la résolution d'un problème en traitement d'antenne : Une nouvelle méthode est proposée pour la localisation de sources en présence de distorsions de phase, tout en conservant une charge de calcul faible. Ceci permet de faire face au cas d'antennes distordues à grand nombre de capteurs.

La seconde partie de ce manuscrit concerne des applications en analyse d'images. Nous rappelons les principes de la méthode SLIDE (subspace-based LIne DEtection) qui adapte des méthodes de traitement d'antenne à l'estimation de contours rectilignes. Nous proposons des méthodes par sous-espaces rapides pour l'estimation de l'orientation et de l'offset de contours rectilignes. Nous présentons plusieurs méthodes d'optimisation pour l'estimation de contours approximativement rectilignes et distordus. En particulier, nous proposons une nouvelle méthode d'optimisation qui est rapide et permet d'estimer des distorsions de courbure importante. Nous proposons aussi d'estimer des contours étoilés. Pour cela, nous adaptions une antenne circulaire virtuelle à l'image à traiter. Le schéma de génération de signal correspondant conduit à des signaux à phase linéaire à partir de cercles concentriques. Cela permet d'appliquer des méthodes haute résolution du traitement d'antenne, et de distinguer des valeurs de rayon pouvant être très proches. Nous nous intéressons à des formes de contour variées telles que des cercles distordus, ou des ellipses.

La troisième partie du manuscrit concerne le traitement du signal tensoriel. Nous rappelons les principales définitions concernant les tenseurs, et donnons un état de l'art sur les méthodes de traitement du signal multidimensionnel, qui reposent sur la projection des données sur le sous-espace signal : la troncature de rang- (K_1, \dots, K_N) de la HOSVD, l'approximation de rang- (K_1, \dots, K_N) inférieur d'un tenseur, en particulier la version de cette méthode utilisant les cumulants d'ordre quatre, qui traite le cas d'un bruit Gaussien corrélé. Nous présentons

une version multidimensionnelle du filtrage de Wiener. L'approximation de rang- (K_1, \dots, K_N) inférieur d'un tenseur et le filtrage de Wiener multidimensionnel reposent sur un algorithme d'optimisation nommé moindres carrés alternés ou Alternating Least Squares (ALS). Nous proposons une procédure de déploiement de tenseur qui est nonorthogonale : le déploiement est effectué selon la direction des contours rectilignes dans l'image et pas dans la direction des lignes ou des colonnes. Ceci a pour but d'améliorer les résultats de débruitage. Pour estimer les directions de déploiement appropriées, nous utilisons la méthode d'estimation de contour présentée dans la première partie du manuscrit.

Mots clé : Antenne, statistiques d'ordre deux, méthodes par sous-espaces, méthodes haute résolution, front d'onde distordu, estimation, traitement d'antenne, optimisation, interpolation par splines, image, contour, antenne linéaire, antenne circulaire, algèbre multilinéaire, décomposition tensorielle, tableau multidimensionnel, approximation de rang inférieur, filtrage.

Acknowledgements

This PhD has been performed in the "Groupe Signaux Multidimensionnels" (GSM) team of the Fresnel Institute, in Marseille, France.

I am very grateful to PROFESSOR SALAH BOURENNANE, head of GSM team, and head of the research department in the Ecole Centrale de Marseille, for his supervision. Salah Bourennane managed to sustain my initiatives at all opportune moments, his unmatched enthusiasm and above all his unfailing trust made me feel confident about my work. His rigorous way of considering theoretical issues were very profitable. Salah Bourennane always incited me to concretize my ideas and initiatives by submitting articles for publication in journal or conference papers. Salah Bourennane's meticulous readings of the articles that we jointly published greatly improved their quality. Moreover he permitted me to participate in several international conferences and thereby to meet researchers and share the knowledge and experience I acquired during my PhD.

I was honored by all members of the jury who examined this dissertation :

- M. J.-P. Sessarego,
- M. Y. Wang,
- M. H. Aghajan,
- M. E. Moreau,
- M. J. Blanc-Talon.

It was a privilege for me to be inserted in GSM group. During my three years of PhD, I have constantly been working with talented people. Especially, I thank all PhD students who maintained a pleasant atmosphere. In particular, I thank Damien, Cyril, Jean-Michel, Simon, William, Damien, Alexis and of course Nadine. I wish to express my sincere appreciation to all researchers of GSM group, in particular Stéphane Derrode, Thierry Gaidon, Mireille Guillaume, Mouloud Adel. They provided me with valuable advice and practical help, concerning my research investigations or for teaching purposes.

Special gratitude is also due to my friends from sport training who participated with me in numerous lessons or competitions, that is, my swimming coach Didier, cyclists Roger, Robert, Daniel, Eric, Christian, triathletes Ludovic and Jean-Jacques, swimmers Eric, Pierre, Christelle, Adeline and Julie. Thanks for their support to my family, to Francois and Anne, and to Albert and Lluis for their constant welcoming in my home country Andorra. Thanks to my parents and grandparents for their support.

Table des matières

Abstract	iii
Résumé	v
Acknowledgements	vii
Introduction	1
I Principles of array processing and optimization, estimation of directions-of-arrival in the case of a distorted antenna with large number of sensors	7
1 Principles of array processing and optimization methods	9
1.1 Basics of array processing	9
1.1.1 Definitions	10
1.1.2 Signal model	10
1.2 Second order statistics : cross-spectral matrix and distinction between signal subspace and noise subspace	11
1.2.1 Cross-spectral matrix	12
1.2.2 Distinction between signal and noise subspaces	13
1.2.3 Orthogonality between signal and noise subspaces	14
1.3 High-resolution methods	15
1.3.1 MUSIC method	15
1.3.2 TLS-ESPRIT method	16
1.3.3 Propagator method	17
1.3.4 Propagator method applied to signal	18
1.3.5 Balance on high-resolution methods	19

1.4 Optimization method : combination of gradient or DIRECT algorithm with spline interpolation	19
1.5 Conclusion of chapter 1	21
2 Phase distortion estimation by DIRECT and spline interpolation algorithms	23
2.1 Introduction	23
2.2 Problem statement	24
2.3 Retrieval and cancellation of phase distortions	24
2.3.1 Phase shift retrieval	24
2.3.2 Cancel the phase shifts in the received signals	25
2.3.3 Proposed algorithm	26
2.4 Simulation results	26
2.5 Conclusion of chapter 2	28
II Array processing and optimization methods applied to image understanding	29
3 Straight line and nearly straight contour retrieval, by high-resolution methods of array processing and optimization methods	31
3.1 Introduction	32
3.2 Data model, signal generation out of the image data	33
3.3 Straight contour estimation	35
3.3.1 Angle estimation, overview of the SLIDE method	35
3.3.2 Angle estimation by Propagator method	36
3.3.3 Offset estimation	37
3.3.3.1 Extension of the Hough Transform	37
3.3.3.2 Proposed method : MFBLP	38
3.4 Straight line estimation : results obtained	40
3.4.1 Straight line estimation by SLIDE method, and MFBLP	40
3.4.2 Close-valued offset estimation	42
3.4.3 Computational complexity of the proposed method for offset estimation and Hough transform	43
3.4.4 Balance on straight contour retrieval	44
3.5 Region-based contour retrieval	44
3.5.1 Proposed method	45
3.5.2 Application of the proposed method to pectoral muscle segmentation .	46

3.5.3	Balance on region-based segmentation	48
3.6	Estimation of non rectilinear contours in an image as an inverse problem : principles of the proposed optimization method and gradient algorithm	48
3.6.1	Retrieval of a general phase model	48
3.6.2	Initialization of the proposed algorithm	49
3.6.3	Distorted curve : proposed algorithm	49
3.6.4	Convergence of the gradient method	51
3.6.5	Summary of the proposed algorithm	52
3.6.6	Numerical complexity of the method	52
3.7	Generalization of distorted contour estimation to the estimation of several curves	53
3.8	Simulations on non rectilinear contours	54
3.9	Estimation of non rectilinear contours in an image by means of Propagator and optimization methods	59
3.9.1	Formulation of a phase model	59
3.9.2	Propagator method for phase shift estimation	59
3.10	Propagator applied to the covariance matrix associated with gradient algorithm : Experimental results, computational times	60
3.10.1	Real-world images	60
3.10.2	Statistical results	62
3.11	Distorted contour retrieval by the combination of gradient and spline interpolation	63
3.11.1	Formulation of a signal model	63
3.11.2	Optimization algorithm : combination of gradient and spline interpolation	64
3.12	Distorted contour retrieval by gradient algorithm combined with spline interpolation : results and discussion	65
3.12.1	Hand-made images	66
3.12.2	Distorted contour retrieval in real-world grey level images	70
3.13	Conclusion of chapter 3	71
4	Subspace-Based and DIRECT algorithms for distorted circular contour estimation	73
4.1	Introduction	74
4.2	Problem statement and signal generation	75
4.2.1	Problem statement	75
4.2.2	Virtual signal generation	76
4.3	Proposed method for radius and center estimation	78
4.4	Optimization method for the estimation of nearly circular contours	80
4.4.1	Proposed optimization method	81

4.4.2	Summary of the proposed method	81
4.5	Results and discussion	82
4.5.1	Hand-made images	82
4.5.2	Real-world images	89
4.6	Conclusion of chapter 4	92
III	Tensor approach of image processing, nonorthogonal tensor flattening via main direction detection	93
5	Lower-rank tensor approximation and multiway filtering	95
5.1	Introduction	96
5.2	Tensor representation and properties	98
5.3	Tensor filtering problem formulation	99
5.3.1	Channel-by-channel SVD-based filtering	100
5.3.2	Tensor filtering based on multimode PCA	101
5.3.2.1	White decorrelated Gaussian noise and second order statistics based method	101
5.3.2.2	Correlated Gaussian noise and higher-order statistics based method	102
5.3.3	Multiway Wiener filtering	104
5.4	Simulation results	106
5.4.1	Performance criterion	106
5.4.2	Denoising of color images	107
5.4.2.1	Denoising of a color image impaired by additive Gaussian noise	107
5.4.2.2	HOSVD- (K_1, K_2, K_3) , rank- (K_1, K_2, K_3) approximation based on second order and higher-order statistics, applied to an image impaired by an additive correlated Gaussian noise . . .	111
5.4.3	Denoising of multispectral images	113
5.4.4	Statistical performances	114
5.4.5	Filtering of a multicomponent seismic type signal	115
5.4.5.1	Filtering of a multicomponent seismic type signal impaired by an additive white Gaussian noise	115
5.4.5.2	Filtering of a multicomponent seismic type signal impaired by an additive correlated Gaussian noise	117
5.5	Nonorthogonal tensor flattening for hyperspectral image filtering	119
5.5.1	Influence of n th-mode ranks on SNR	120
5.5.2	Main directions of tensors and $MWFR$ algorithm	121

5.5.3	Real-world data : HYDICE HSI	122
5.5.3.1	Noise level in HYDICE images	122
5.5.3.2	HSI02 image	123
5.5.4	Balance about nonorthogonal tensor flattening	123
5.6	Conclusion of chapter 5	124
	Concluding remarks	125
	A Appendix A : mathematical considerations about multiway filtering	129
A.1	Appendix : Alternating Least Squares algorithms	129
A.1.1	ALS algorithm in Rank- (K_1, \dots, K_N) approximation - TUCKALS3 algorithm	129
A.1.2	Fourth order cumulant slice matrix-based multimode PCA-based filtering	130
A.1.3	ALS algorithm in multiway Wiener filtering	131
A.2	Appendix : n th-mode Wiener filter analytical expression	132
A.3	Appendix : Assumptions and related expression of the n th-mode Wiener filter	134
	B Appendix B : Synthèse du manuscrit	139
B.1	Introduction générale	139
B.2	Corps du manuscrit	142
B.2.1	Résumé du chapitre 1 : Principes des méthodes de traitement d'antenne et d'optimisation	142
B.2.1.1	Bases de traitement d'antenne	143
B.2.1.2	Statistiques d'ordre deux : matrice interspectrale et distinction entre sous-espace signal et sous-espace bruit	144
B.2.1.3	Méthodes haute résolution	145
B.2.1.4	Méthode MUSIC	145
B.2.1.5	Méthode TLS-ESPRIT	145
B.2.1.6	Méthode du Propagateur	145
B.2.1.7	Algorithme d'optimisation associé à une méthode d'interpolation par splines	146
B.2.1.8	Conclusion du chapitre 1	146
B.2.2	Résumé du chapitre 2 : Estimation de distorsions de phase par DIRECT associé à l'interpolation par splines	146
B.2.2.1	Position du problème	147
B.2.2.2	Résultats : Caractérisation de trois sources	147
B.2.2.3	Conclusion du chapitre 2	147

B.2.3 Résumé du chapitre 3 : Estimation de contours rectilignes et approximativement rectilignes	148
B.2.3.1 Modèle de données, génération de signal à partir de l'image .	149
B.2.3.2 Estimation de contours distordus	149
B.2.3.3 Quelques résultats : estimation de contours rectilignes et approximativement rectilignes	150
B.2.3.4 Conclusion du chapitre 3	152
B.2.4 Résumé du chapitre 4 : Estimation de contours circulaires et approximativement circulaires	152
B.2.4.1 Position du problème et génération de signal	153
B.2.4.2 Estimation de cercles multiples avec des centres et des rayons différents	155
B.2.4.3 Quelques résultats	155
B.2.4.4 Conclusion du chapitre 4	158
B.2.5 Résumé du chapitre 5 : Approximation de tenseur de rang inférieur et filtrage multimodal	159
B.2.5.1 Les données tensorielles	159
B.2.5.2 Quelques résultats de débruitage de données tensorielles . .	160
B.2.5.3 Conclusion du chapitre 5	161
B.3 Conclusion générale	161
Liste des figures	165
Liste des tableaux	171
Bibliographie	173

Introduction

DATA processing and analysis have been developed and encouraged by the progress in numerical computing, video, and acquisition systems such as sensors. A set of sensors forms an antenna, or array. When an array processing problem is considered, one generally assumes that signals received on each sensor are centered around one frequency. So, for each sensor, one chooses the signal Fourier transform component corresponding to this frequency of interest. One obtains then a unidimensional signal : one component per sensor. An antenna is used for source localization, and gets various shapes. Circular antennas are used in telecommunications, linear antennas are used for instance for object detection. In particular, antennas with large number of sensors have been used to improve localisation results. Object detection and localization is of great interest for civilian and military applications, for the detection of mines, pipe-lines, cables. Solving an array processing problem consists in determining the origin of waves emitted by possibly close sources. Important fundings were devoted to array processing techniques to solve difficult cases such as the case of sources with close directions-of-arrival, correlated sources, and the case where correlated noise impairs the data.

The approach adopted during this thesis comes from the following remark : Array processing methods have been widely improved, and could profit to image processing and multidimensional signal processing fields. It appeared then of great interest, during this thesis, to set analogies between array processing methods, which are unidimensional signal methods, and image processing problems. Now, images are either bidimensional images, when binary or grey level images are considered, either multidimensional when color or multispectral images are considered. We considered the problem of contour estimation in binary or grey level images. A first issue is to transcript the image content into a signal. For this, a particular signal generation method consists of adapting an antenna to the processed image. This antenna is either linear or circular, depending on the shape of the expected object. To apply array processing methods to multidimensional data processing, and in particular to data denoising, it is possible to choose another formalism. We chose, to enable the application of algebraic methods of array processing to such a problem, to consider each vector of a multidimensional entity as a signal realization. We provide in the remainder of this introduction some details concerning the issues of processings applied to signals, images, or multidimensional data.

Data are bidimensional when image acquisition is performed. Image processing consists in producing new images which are in some way more desirable. Recovering from blur or noise impairment is required in the case of noisy or moving sensors. The purpose of image understanding is to extract desired information from two-dimensional images. Locating and measurement of features and patterns is a typical image understanding task. Such a task is easily performed by the human brain which possesses the ability to extract pertinent information and interpret a small number of parameters from images. The output of image understanding

is either a number of parameter values or a description of the objects present in the image. Numerous phenomena depend on several physical variables such as those associated with space (three dimensions), time, wavelength, color or wave polarisation. In this context data acquisition yields multidimensional data. To record, study and extract useful information from these phenomena, several multidimensional or multicomponent sensors have appeared for the last fifty years. The data provided by multicomponent sensors are called multiway data. Multiway data are modeled by arrays, and each array entry is associated with a physical parameter, such as time, space or wavelength. Such an array is denominated tensor [92, 38], and each dimension of a tensor is called *n*th-mode ; the number of *n*th-modes is called the order of the tensor.

Data are sampled such that a signal is a set of samples, an image is a set of pixels, and a tensor is a multiway array. Therefore, signal, image and tensor processing are performed by algebraic methods. This type of data processing relies, in general, on second order [133, 114, 90, 5, 3] or higher-order statistics [80, 89, 108] computation. The most common processing which is applied to a covariance matrix is the Singular Value Decomposition (SVD [43, 51]), or the eigendecomposition. Efficient techniques for extracting the desired parameters from the array data are based on the so-called signal subspace concept [115, 114, 90, 3]. All eigenvectors of the covariance matrix of the collected signals form the measurement space. The key observation while computing second order or higher-order statistics is that a low-dimensional subspace permits to extract useful information, in particular expected parameters [5, 3]. The knowledge, or the estimation by statistical criteria [132], of the useful signal subspace dimension permits to partition the measurement space into a signal subspace that contains the desired information, and a noise subspace, that contains all undesired contributions.

Optimization methods mean to retrieve a set of unknown parameters, by minimizing a criterion, generally squared distance between data model and measurements. They distinguish in two categories : global and local methods. Global methods retrieve the global minimum of a function, they are not supposed to converge on a local minimum, at the expense of a generally high computational cost. For instance, DIviding RECTangles (DIRECT) method [64], or the genetic algorithms [50] are global optimization methods. Local methods are generally faster than global methods. A classical local method for the resolution of nonlinear and non-constrained optimization problems is the gradient or Newton [48] method, which can be accelerated as a variable step gradient method.

There is great interest in combining subspace-based and optimization methods to obtain an optimal estimation of either a set of parameters or expected features from signal, image or tensor data. In array processing, subspace-based Propagator method was associated with an optimization method to improve direction-of-arrival (DOA) estimation in the presence of phase distortion [90]. In the field of image processing, DIRECT method [64] was adapted to image registration, especially for medical images [130]. In the field of tensor processing, an Alternating Least Squares (ALS) algorithm that yields a particular tensor decomposition was proposed in [76, 77]. An ALS algorithm was also proposed in [96] to solve a tensor data filtering problem, for a denoising application.

The purpose of this manuscript is to propose and associate several methods, either subspace-based ones, or optimization ones, for data processing and (or) understanding, in the fields of array processing for one-dimensional signals, image processing for two-dimensional signals, and tensor processing for multidimensional signals. The background and proposed improvements, in array processing, image processing, and tensor processing respectively are as follows :

- In an array processing problem, several signal realizations are usually available, they are collected upon the array at several instants. The signal subspace dimension is equal to the number of emitting sources. In practice, if the number of sources is not *a priori* known, this number can be estimated from the data by statistical criteria such as Minimum Description Length (MDL) [132]. In this framework we introduce novel optimization algorithms in an array processing problem concerning distorted antennas with large number of sensors.
- Two major types of image problems are processing and understanding. Many applications have been developed and encouraged thanks to revolutionary trends in computer and video technologies. The purpose of image understanding is to extract desired information from an image or a set of images. Image processing is concerned with transforming images to produce new images which exhibit a better quality, with respect to any criterion. Some image processing applications are, among others, deblurring and denoising (image restoration), improving the appearance (image enhancement), and reducing the storage requirements.

An original approach of contour estimation consists in placing the processed image in a wave propagation context. More precisely, a contour is considered as a frozen wavefront, and the image background is considered as a propagation medium [5]. However, in this framework, when one image is processed, there are no time-dependent signals. So the question arises as how a sample covariance matrix can be formed. In [3, 5], Aghajan proposed a specific formalism that permits to transcript the content of an image into several signal realizations. This enables the application of subspace-based methods [114]. In this frame we adapt array processing and optimization algorithms to an understanding problem, especially to retrieve linear contours, nearly linear distorted contours, circular contours, and nearly circular distorted contours.

- In a tensor processing problem, it is possible to extend the subspace approach, with some prior assumption about the dimension of the signal subspace [92]. The projection of the processed data on the signal subspace along each mode yields efficient denoising [96]. Recent studies have shown that denoising results are improved when the multiway structure of the data is taken into account, and multilinear algebra is applied [41, 96, 97]. The subspace-based methods which are applied in the core of these algorithms require the availability of signal realizations. With a stationarity and ergodicity assumption, each column-vector of a processed tensor is considered as a signal realization, and second or higher-order statistics are computed from these signal realizations. Then, assuming that the n th-mode rank is smaller than the n th-mode dimension, whatever the n th-mode, a signal subspace and a noise subspace can be distinguished from this data space. In this frame we present the last advances in multiway tensor processing relying on subspace-based and optimization methods, and prove the interest of straight line retrieval for the improvement of tensor signal processing.

This manuscript falls into five chapters :

Chapter 1 states the problem of source localization in array processing, to introduce the reader with the physical elements that compose this framework, and with the array processing signal model. The interest of second order statistics is presented, in particular we emphasize the distinction between signal and noise subspaces in the set of eigenvectors of a covariance matrix computed from signal realizations. The principles of the high-resolution methods MU-

SIC (MUltiple SIgnal Characterization) and TLS-ESPRIT (Total-Least-Squares Estimation of Signal Parameters via Rotational Invariance Techniques) are explained. We present an optimization method that will be adapted for several purposes in the manuscript.

Chapter 2 solves a source localization problem in a particular case, namely when wavefronts are impinging on a distorted antenna composed of a large number of sensors. We show that it is necessary to cancel the phase errors resulting from the antenna distortions, to resolve all possibly close directions-of-arrival. We propose to estimate phase distortions by a robust optimization algorithm that combines DIRECT (DIviding RECTangles) algorithm and spline interpolation.

Chapter 3 is devoted first to the estimation of rectilinear and nearly rectilinear contours in images by high-resolution methods. In the case of rectilinear contours it has been shown that it is possible to transpose this image processing problem to an array processing problem. The existing straight line characterization method called Subspace-based Line Detection (SLIDE) leads to models with orientations and offsets of straight lines as the desired parameters. We propose several fast high-resolution methods to estimate straight line parameters. The signal generation process devoted to straight line retrieval is retained for the case of nearly rectilinear distorted contour estimation. This issue is handled for the first time thanks to an inverse problem formulation and a phase model fitting. We propose several optimization algorithms to retrieve the contour distortions, from the phase of the generated signal. The whole proposed contour estimation algorithm works blindly, as the optimization method is initialized by the estimated straight line parameters.

Chapter 4 adapts array processing methods to circular and nearly circular contour estimation. We show how to adapt subspace-based high-resolution methods of array processing to the estimation of several radii while extending the circle estimation to retrieve circular-like distorted contours. Especially, we develop and validate a new model for virtual signal generation by simulating a circular antenna. A variable speed propagation scheme toward circular antenna yields a linear phase signal. Therefore, a high-resolution method provides the possibly close radii. Either the gradient method or the more robust combination of DIRECT (DIviding RECTangles) and spline interpolation can extend this method for free-form object segmentation. The retrieval of multiple non concentric circles and rotated ellipses is also considered. To evaluate the performance of the proposed nearly linear and nearly circular contour retrieval methods, we compare them with a least-squares method, Hough transform and GVF (Gradient Vector Flow). We apply the proposed method to hand-made images while considering some real-world ones.

Chapter 5 presents some recent filtering methods based on the lower-rank tensor approximation approach for denoising tensor signals. In this approach, multicomponent data are represented by tensors, that is, multiway arrays. The classical channel-by-channel SVD-based filtering method is overviewed. This method processes successively all slice matrices of the considered tensor. In particular, and contrary to channel-by-channel SVD-based filtering, multiway filtering methods include a flattening step, that is, the reorganization of the tensor data as a bidimensional entity. This permits to consider the processed tensor as a whole, instead of splitting the data. We present the lower rank- (K_1, \dots, K_N) truncation of the HOSVD, which performs a multimode Principal Component Analysis (PCA). Then, a refined version of lower rank- (K_1, \dots, K_N) truncation of the HOSVD is emphasized, namely the lower rank- (K_1, \dots, K_N) tensor approximation. This method is initialized by the truncation of HOSVD, and includes an optimization stage, that is, an Alternating Least Squares (ALS) loop. The criterion which is thereby minimized is the mean squared difference between estimated and

processed tensors, for a set of lower ranks. Then, two recently developed tensor filtering methods are overviewed. The first method consists of an improvement of the lower rank- (K_1, \dots, K_N) tensor approximation in the case of an additive correlated Gaussian noise. This improvement is especially done thanks to the fourth order cumulant slice matrix. The second method consists of an extension of Wiener filtering for tensor data. The criterion which is minimized by this multiway Wiener filtering is the squared difference between estimated and expected tensors, for a set of lower ranks. The performance and comparative results between all these tensor filtering methods are presented for the cases of noise reduction in color images, multispectral images, and multicomponent seismic data. **Chapter 5** also proves the interest of straight line retrieval methods in multiway filtering. In particular, the straight line retrieval methods proposed in **Chapter 3** retrieve major orientations which are possibly present in some bidimensional slice matrices of the data tensor. Then, we prove that performing flattening along these major orientations improves multiway tensor denoising methods. Appendix A concerns mathematical details about the presented methods. Appendix B provides a summary of the manuscript in french.

First part

Principles of array processing and optimization, estimation of directions-of-arrival in the case of a distorted antenna with large number of sensors

Principles of array processing and optimization methods

In this chapter, we state an array processing problem and describe some high-resolution methods of array processing. We also propose a novel optimization method. The purpose of section 1.1 is to introduce the reader with the definition of the elements that compose an array processing problem, with the signal model that is derived as a function of the parameters of the considered array processing issue, such as the number of sensors and sources, the directions-of-arrival (DOA). In section 1.2, we introduce the reader with second order statistics, in particular with the estimation of a covariance matrix. We distinguish between signal and noise subspaces, and emphasize the orthogonality between signal and noise subspaces. In section 1.3.1, we detail the MUSIC (MULTiple SIgnal Characterization) method, which is a parametric method based on the orthogonality hypothesis between a signal subspace vector model and noise subspace vectors. In section 1.3.2, we present briefly TLS-ESPRIT (Total-Least-Squares - Estimation of Signal Parameters via Rotational Invariance Techniques) method which is based on the array partitioning into two identical sub-arrays. These methods solve close-valued directions-of-arrival. In section 1.4, we present an optimization method that will be adapted further, to solve several problems in array processing and image understanding.

1.1 Basics of array processing

In several application fields such as acoustics, geophysics, astronomy, telecommunications and medical imagery, signals provided by an array of sensors are used to characterize one or several emitting sources. Array processing considers the issue of time and space sampled signals, which are collected by an array of sensors. Collected signals contain information about the expected sources. Beamforming [75] is the first developed method for source localization. Its principle is to compute the energy received in all directions via rotating the antenna. This method is a low resolution one, because its spatial resolution depends on the width of the reception diagram, which depends on the number of sensors of the antenna. This prevents this method from distinguishing between close sources. The so-called high-resolution methods, such as MULTiple SIgnal Characterization (MUSIC) have been developed to cope with close sources. These methods are based on the orthogonality between signal and noise subspaces.

1.1.1 Definitions

The objectives of array processing are to estimate the characteristics (direction-of-arrival, amplitude) of emitting sources. The measured signal depends on the array, the medium, the sources :

- The reception array is composed of N identical aligned sensors. Sensors receive signals emitted by independent sources.
- The medium contains sources, located far enough from the array for the received waves to be planar. The medium is assumed linear, homogeneous, and isotropic.
- The number of sources is smaller than the number of sensors.

These elements are drawn on Fig. 1.1.

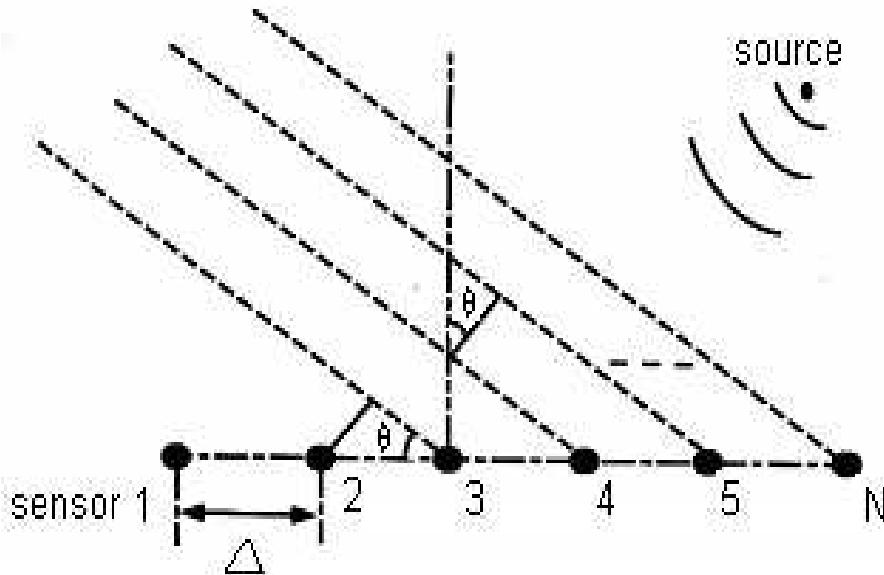


Figure 1.1 — Elements of an array processing problem. Δ is the inter-sensor distance, θ is the direction-of-arrival.

1.1.2 Signal model

The array is composed of N sensors indexed by $i = 1, \dots, N$.

Fixing to zero the phase of the first sensor signal (indexed by 1) and denoting by s the emitted signal amplitude, the signal received by the first sensor is $z(1) = s$.

For a source placed at azimuth θ , the reception delay td between two sensors for the plane wave that is associated with this source is : $td = \Delta * \sin(\theta)/c$ where Δ is the inter-sensor distance and c is the wave celerity. This duration yields phase shift value $\varphi = 2 * \pi * f * td$ where f is the source emission frequency.

A sensor with index i receives signal :

$$z(i) = s \cdot \exp(j(i - 1)\varphi), \quad i = 1, \dots, N. \quad (1.1)$$

When d sources are considered, in an additive noise environment, the received signal is expressed as :

$$\mathbf{z} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \exp(j\varphi_1) & \exp(j\varphi_2) & \dots & \exp(j\varphi_d) \\ \dots & \dots & \dots & \dots \\ \exp(j(N-1)*\varphi_1) & \exp(j(N-1)*\varphi_2) & \dots & \exp(j(N-1)*\varphi_d) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \dots \\ s_d \end{bmatrix} + \begin{bmatrix} n(1) \\ n(2) \\ \dots \\ n(N) \end{bmatrix} \quad (1.2)$$

where $\mathbf{z} = [z(1), \dots, z(N)]^T$, T denoting transpose, is a vector containing all signal components, $[s_1, \dots, s_d]^T$ is a vector containing all source amplitudes, and $[n(1), \dots, n(N)]^T$ is a vector containing all noise components. Defining $a_i(\varphi_k) = \exp(j(i-1)*\varphi_k)$, each component $z(i)$ of \mathbf{z} is expressed as :

$$z(i) = \sum_{k=1}^d a_i(\varphi_k) s_k + n(i), \quad i = 1, \dots, N. \quad (1.3)$$

We obtain a model for the signal which is received by an array of N sensors, when d sources are expected.

We set $\mathbf{a}(\varphi_k) = [a_1(\varphi_k), a_2(\varphi_k), \dots, a_N(\varphi_k)]^T$.

Signals can be expressed in a vector form :

$$\mathbf{z} = \mathbf{A}(\Phi) \cdot \mathbf{s} + \mathbf{n}, \quad (1.4)$$

where $\mathbf{s} = [s_1, \dots, s_d]^T$ is a source amplitude vector, $\mathbf{n} = [n(1), \dots, n(N)]^T$, and $\mathbf{A}(\Phi) = [\mathbf{a}(\varphi_1), \dots, \mathbf{a}(\varphi_d)]$ is an $N \times d$ Vandermonde matrix, called transfer matrix or directional matrix.

Considering a set of K realizations $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_K$ of received signals, a set $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K$ of source amplitudes, a set of noise realizations $\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K$, and by storing each realization as a column vector of matrices \mathbf{Z} , \mathbf{S} , \mathbf{N} respectively, we obtain in matrix form :

$$\mathbf{Z} = \mathbf{A}(\Phi) \mathbf{S} + \mathbf{N} \quad (1.5)$$

Equation (1.5) contains the whole information provided by the signals received on the array of sensors.

1.2 Second order statistics : cross-spectral matrix and distinction between signal subspace and noise subspace

From the signals received on the array of sensors, it is of great interest to exploit second order statistics. In particular, the eigendecomposition of the cross-spectral matrix yields a distinction between signal and noise subspaces.

1.2.1 Cross-spectral matrix

We consider a set of K realizations $\{\mathbf{z}_l, l = 1, \dots, K\}$. A cross-spectral matrix is a covariance matrix which is computed in the Fourier space. It can be expressed as follows, for the signal realizations quoted above :

$$\mathbf{R}_{zz} = \mathbb{E} [\mathbf{z}_l \mathbf{z}_l^H] \quad (1.6)$$

where H denotes transpose conjugate. An estimate of the covariance matrix is given by :

$$\mathbf{R}_{zz} = \frac{1}{K} \sum_{l=1}^K \mathbf{z}_l \mathbf{z}_l^H, \quad (1.7)$$

or, by considering the forward and backward (indicated by subscript r) versions of the signal realizations :

$$\mathbf{R}_{zz} = \frac{1}{2K} \sum_{l=1}^K \mathbf{z}_l \mathbf{z}_l^H + \mathbf{z}_l^r \mathbf{z}_l^{rH} \quad (1.8)$$

where \mathbf{z}_l^r denotes the l^{th} backward sub-vector : $[\mathbf{z}_l^r = z_l(N), z_l(N-1), \dots, z_l(1)]$.

Matrix form of Eq. (1.7) is the following :

$$\mathbf{R}_{zz} = \frac{1}{K} \mathbf{Z} \mathbf{Z}^H \quad (1.9)$$

The signal is assumed independent from noise ; realizations of noise vector \mathbf{n}_l are uncorrelated, so the noise covariance matrix is diagonal. All noise realizations have identical variance σ^2 , so the noise realization covariance matrix is $\sigma^2 \mathbf{I}$.

Useful signal part is assumed independent from noise part of the data, so the covariance matrix of the received signals is the summation of the covariance matrix of the useful signal part of the data and the covariance matrix of noise realizations. By using Eq. (1.4) in Eq. (1.7), we get :

$$\mathbf{R}_{zz} = \mathbf{A}(\Phi) \mathbf{R}_{ss} \mathbf{A}^H(\Phi) + \sigma^2 \mathbf{I} \quad (1.10)$$

with

$$\mathbf{R}_{ss} = \mathbb{E} [\mathbf{s}_l \mathbf{s}_l^H], \quad (1.11)$$

and

$$\mathbb{E} [\mathbf{n}_l \mathbf{n}_l^H] = \sigma^2 \mathbf{I} \quad (1.12)$$

Using backward and forward versions of the signals yields a better estimation of the covariance matrix [133].

The eigen-decomposition of the covariance matrix is used in general to characterize sources by subspace (signal and noise subspaces)-based methods [115, 107, 114]. The covariance matrix eigendecomposition is expressed as follows :

$$\mathbf{R}_{zz} = \sum_{i=1}^N \lambda_i \mathbf{e}_i \mathbf{e}_i^H \quad (1.13)$$

where $\{\lambda_i, i = 1, \dots, N\}$ are the eigenvalues of the covariance matrix, and $\{\mathbf{e}_i, i = 1, \dots, N\}$ are the eigenvectors of the covariance matrix.

1.2.2 Distinction between signal and noise subspaces

The eigenvectors of the covariance matrix span the measurement space. Within the measurement space, we distinguish between a signal and a noise subspace. When the dimension of the signal subspace is d , that is, when there are d sources in the considered problem, this distinction between subspaces is expressed by :

$$\mathbf{R}_{zz} = \sum_{i=1}^d \lambda_i \mathbf{e}_i \mathbf{e}_i^H + \sum_{i=d+1}^N \lambda_i \mathbf{e}_i \mathbf{e}_i^H = \mathbf{E}_s \mathbf{\Lambda}_s \mathbf{E}_s^H + \mathbf{E}_n \mathbf{\Lambda}_n \mathbf{E}_n^H \quad (1.14)$$

where \mathbf{E}_s contains the useful signal eigenvectors and \mathbf{E}_n contains the noise eigenvectors. $\mathbf{\Lambda}_s$ is a diagonal matrix that contains the d largest eigenvalues, $\mathbf{\Lambda}_n$ is a diagonal matrix that contains the eigenvalues associated with the noise subspace.

One can prove that the d largest eigenvalues are associated with the signal subspace. For this, we consider the case where noise is null. Then one can express a realization of the received signal as :

$$\mathbf{z} = \mathbf{A}(\Phi) \cdot \mathbf{s} \quad (1.15)$$

The covariance matrix is then :

$$\mathbf{R}_{zz} = \mathbf{A}(\Phi) \mathbf{R}_{ss} \mathbf{A}^H(\Phi) \quad (1.16)$$

We assume that sources are uncorrelated. Therefore, matrix \mathbf{R}_{ss} is diagonal and expressed as :

$\mathbf{R}_{ss} = \text{diag}\{\alpha_1, \alpha_2, \dots, \alpha_d\}$ where $\alpha_1, \alpha_2, \dots, \alpha_d$ are the powers of sources $1, 2, \dots, d$ respectively.

Matrix \mathbf{R}_{ss} has rank d , matrices $\mathbf{A}(\Phi)$ and $\mathbf{A}^H(\Phi)$ have rank d . Therefore matrix \mathbf{R}_{zz} resulting from the product of these three matrices has rank d when noise is null. Then the covariance matrix can be expressed as :

$$\mathbf{R}_{zz} = \sum_{i=1}^d \gamma_i \mathbf{e}_i \mathbf{e}_i^H \quad (1.17)$$

Equation (1.17) shows that matrix \mathbf{R}_{zz} exhibits d non-zero eigenvalues. These eigenvalues are $\{\gamma_i, i = 1, \dots, d\}$. As \mathbf{R}_{zz} is positive definite, all γ_i values are strictly positive. We notice that if sources are correlated, matrix \mathbf{R}_{zz} may not have rank d when noise is null. Then we use spatial smoothing to decorrelate the sources. In the case where noise is not null, and assuming that noise is decorrelated and white, the covariance matrix is expressed as :

$$\mathbf{R}_{zz} = \sum_{i=1}^d \gamma_i \mathbf{e}_i \mathbf{e}_i^H + \sigma^2 \mathbf{I} \quad (1.18)$$

Completely correlated sources

Two sources are completely correlated if they are related by a time shift. The source vector is then :

$\mathbf{s}(t) = [s(t), s(t - \tau_0)]^T$ where τ_0 is the time shift between two sources. In the frequency domain, the source vector is expressed as : $\mathbf{s} = [s, s \exp(-j2\pi f_0 \tau)]^T$.

The covariance matrix of the useful signal part is expressed as : $\mathbf{A}(\Phi) \mathbf{R}_{ss} \mathbf{A}^H(\Phi)$ with

$$\mathbf{R}_{ss} = \begin{bmatrix} |s|^2 & |s|^2 \exp(j2\pi f_0\tau) \\ |s|^2 \exp(-j2\pi f_0\tau) & |s|^2 \end{bmatrix} \quad (1.19)$$

The determinant of matrix \mathbf{R}_{ss} is null, so there exists only one non-zero eigenvalue. One cannot distinguish between the two sources although two sources are actually expected.

In this case we use a spatial smoothing procedure : A sub-array of length M is translated along the array. The new directional matrix is denoted by $\mathbf{A}_M(\Phi)$ and is composed of M rows and d columns. The M rows of $\mathbf{A}_M(\Phi)$ are equal to the M first rows of $\mathbf{A}(\Phi)$. The signal corresponding to the position of the sub-array of index l is :

$$\mathbf{z}_l = \mathbf{A}_M(\Phi)\mathbf{D}^{l-1}\mathbf{s} + \mathbf{n}_l \quad (1.20)$$

where $\mathbf{D} = \text{diag}\{\zeta_1, \dots, \zeta_d\}$, with $\zeta_k = \exp(j\varphi_k)$, $\forall k = 1, \dots, d$. By considering the noise vector components : $\mathbf{n} = [n(1), \dots, n(N)]^T$ one can express vector $\mathbf{n}_l = [n(l), \dots, n(l+M-1)]^T$.

The covariance matrices of the signal realizations \mathbf{z}_l , $l = 1, \dots, K$, received on each sub-array are expressed as :

$$\begin{aligned} & \mathbf{A}_M(\Phi)\mathbf{R}_{ss}\mathbf{A}_M^H(\Phi) + \sigma^2\mathbf{I} \\ & \vdots \\ & \mathbf{A}_M(\Phi)\mathbf{D}^{l-1}\mathbf{R}_{ss}(\mathbf{D}^H)^{l-1}\mathbf{A}_M^H(\Phi) + \sigma^2\mathbf{I} \\ & \vdots \\ & \mathbf{A}_M(\Phi)\mathbf{D}^{K-1}\mathbf{R}_{ss}(\mathbf{D}^H)^{K-1}\mathbf{A}_M^H(\Phi) + \sigma^2\mathbf{I} \end{aligned}$$

A covariance matrix with rank K is obtained by computing the mean of the covariance matrices calculated from the signals received on each sub-array.

$$\mathbf{R}_{zz} = \frac{1}{K} \sum_{l=1}^K (\mathbf{A}_M(\Phi)\mathbf{D}^{l-1}\mathbf{R}_{ss}(\mathbf{D}^H)^{l-1}\mathbf{A}_M^H(\Phi) + \sigma^2\mathbf{I}) \quad (1.21)$$

$$\mathbf{R}_{zz} = \mathbf{A}_M(\Phi) \frac{1}{K} \left(\sum_{l=1}^K \mathbf{D}^{l-1} \mathbf{R}_{ss} (\mathbf{D}^H)^{l-1} \right) \mathbf{A}_M^H(\Phi) + \sigma^2 \mathbf{I} \quad (1.22)$$

This summation of K different matrices of rank 1 leads to a rank K matrix.

Turning to the case of non correlated sources, from Eq. (1.18) we notice that the d largest eigenvalues of the covariance matrix are expressed as $\{\lambda_i = \gamma_i + \sigma^2, i = 1, \dots, d\}$. Indeed $\{\gamma_i, i = 1, \dots, d\}$ are strictly positive, then $\gamma_i + \sigma^2 > \sigma^2$, $\forall i = 1, \dots, d$. The $N - d$ smallest eigenvalues of the covariance matrix are $\{\lambda_i = \sigma^2, i = d + 1, \dots, N\}$. We define all eigenvectors associated with the d largest eigenvalues of the covariance matrix as the vectors of the signal subspace and the eigenvectors associated with the $N - d$ smallest eigenvalues of the covariance matrix as the vectors of the noise subspace.

1.2.3 Orthogonality between signal and noise subspaces

Referring to Eq. (1.9) and to the expression of the covariance matrix, we note that the covariance matrix is hermitian, that is, it is equal to its transpose conjugate. Therefore,

its eigenvalues are positive or null, and its eigenvectors are orthonormal. Thus, each noise subspace vector is orthogonal to the signal subspace vectors. We draw the same conclusion in this way :

In the case where noise is decorrelated from the signal, and noise is uncorrelated and white, the covariance matrix can be expressed as :

$$\mathbf{R}_{zz} = \mathbf{A}(\Phi)\mathbf{R}_{ss}\mathbf{A}^H(\Phi) + \sigma^2\mathbf{I} \quad (1.23)$$

Let \mathbf{e}_i be an eigenvector of \mathbf{R}_{zz} . $\mathbf{A}(\Phi)\mathbf{R}_{ss}\mathbf{A}^H(\Phi)\mathbf{e}_i = (\lambda_i - \sigma^2)\mathbf{e}_i$.

We have shown that if $\lambda_i = \sigma^2$, \mathbf{e}_i can only be a vector that belongs to the noise subspace. Therefore, if \mathbf{e}_i is a noise subspace vector, $\mathbf{A}(\Phi)\mathbf{R}_{ss}\mathbf{A}^H(\Phi)\mathbf{e}_i = \mathbf{0}$ then $\mathbf{A}^H(\Phi)\mathbf{e}_i = \mathbf{0}$. This proves that a noise subspace vector is orthogonal to the subspace that is spanned by the columns of $\mathbf{A}(\Phi)$, that is, the vectors that span the signal subspace.

1.3 High-resolution methods

In this section we present three high-resolution methods. The main property of these methods is that they distinguish between close direction-of-arrival values.

1.3.1 MUSIC method

We have shown in section 1.2 that the d largest eigenvalues of the cross-spectral matrix are associated with the signal subspace vectors, and that the noise subspace vectors are orthogonal to the signal subspace vectors.

MUSIC (MULTiple SIgnal Characterization) method selects the eigenvectors that are not associated with the dominant eigenvalues, that is, the noise subspace eigenvectors. MUSIC method takes into account the fact that noise subspace vectors are orthogonal to the signal subspace vectors. MUSIC is a parametric method : it calculates the scalar product between noise subspace vectors and a signal subspace vector model, which includes the direction-of-arrival parameter value.

We build a model vector which is convenient for all directional vectors :

$$\mathbf{a}(\varphi) = [1, \exp(j\varphi), \exp(j2\varphi), \dots, \exp(j(N-1)*\varphi)]^T$$

Thus, the scalar product $\mathbf{E}_n^H \mathbf{a}(\varphi)$ is null for all parameter values φ which are equal to the source directions-of-arrival :

$$\mathbf{E}_n^H \mathbf{a}(\varphi) = \mathbf{0} \text{ if } \varphi = \varphi_k, \quad (1.24)$$

for $k = 1, 2, \dots, d$.

We remind that $\varphi = 2 * \pi * f * \Delta * \sin(\theta)/c$. In practice, we draw the "pseudospectrum" $f(\varphi)$ of MUSIC for values of θ between $-\pi$ and π .

$$f(\varphi) = \frac{1}{|\mathbf{E}_n^H \mathbf{a}(\varphi)|^2} \quad (1.25)$$

The maximum arguments of this function are the values of θ which are the directions-of-arrival of the $d = 3$ sources (see Fig. 1.2). Fig. 1.2(a) presents the pseudospectrum obtained with classical Beamforming method [75], Fig. 1.2(b) presents the pseudospectrum obtained

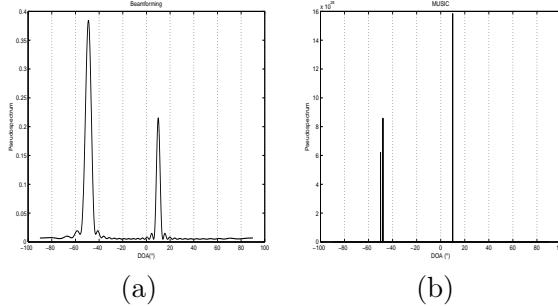


Figure 1.2 — Pseudospectra : a)Beamforming ; b)MUSIC

with high-resolution MUSIC method. MUSIC resolves two close-valued directions-of-arrival (-50° and -48°), whereas Beamforming does not.

In the next subsection we present another high-resolution method, that avoids pseudospectrum maxima research.

1.3.2 TLS-ESPRIT method

MUSIC method studies the directional matrix by means of the matrix that contains the eigenvectors which are associated with the noise subspace \mathbf{E}_n . TLS-ESPRIT (Total-Least-Squares Estimation of Signal Parameters via Rotational Invariance Techniques) method yields the directions-of-arrival by working on the eigenvectors which are associated with the signal subspace \mathbf{E}_s .

ESPRIT method is applied in the particular case where the array is composed of two identical sub-arrays [114]. TLS-ESPRIT is an extended version of ESPRIT. Its principle is to split the array into two sub-arrays. The two sub-arrays are shifted by value Δ . Directions-of-arrival depend only on the propagation delay between the two sub-arrays, not on the propagation delay between two array sensors. Although originally, in the ESPRIT algorithm, no specific geometry is assumed for each sub-array [114], we consider the case where the sub-arrays are identical. We consider a size N array, from which we build signal realizations of length M . In this way we can perform spatial smoothing while building the covariance matrix. The shift parameter Δ between the sub-arrays can be set as δ , the shift between two sensors. This yields two arrays of length $M - 1$, that partially overlap.

We define \mathbf{E}_1 as the sub-matrix of \mathbf{E}_s composed of rows 1 to $M - 1$, and \mathbf{E}_2 as the sub-matrix of \mathbf{E}_s composed of rows 2 to M . There exists a unitary operator that relates the directional matrices associated with each sub-array. This relationship is used in TLS-ESPRIT method. Let us form the eigendecomposition of $2d \times 2d$ matrix :

$$\begin{bmatrix} \mathbf{E}_1^H \\ \mathbf{E}_2^H \end{bmatrix} \begin{bmatrix} \mathbf{E}_1 & \mathbf{E}_2 \end{bmatrix} = \mathbf{F} \boldsymbol{\Lambda} \mathbf{F}^H.$$

Matrix \mathbf{F} is partitioned into sub-matrices of size $d \times d$,

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} \\ \mathbf{F}_{21} & \mathbf{F}_{22} \end{bmatrix}$$

By setting as β_k , $k = 1, 2, \dots, d$, the eigenvalues of matrix $-\mathbf{F}_{12}\mathbf{F}_{22}^{-1}$, the directions-of-arrival are given by [114] :

$$\hat{\theta}_k = \sin^{-1} \left[\frac{c}{(2\pi f \Delta)} \operatorname{Im} \left(\ln \left(\frac{\beta_k}{|\beta_k|} \right) \right) \right], \quad k = 1, \dots, d \quad (1.26)$$

where Im denotes imaginary part. Therefore, we obtained the estimated values $\hat{\theta}_k$ of the source directions-of-arrival.

In this framework, the covariance matrix eigenvalues lead to an estimation of the value of d , the number of sources. This estimate is the minimum argument of the "Minimum Description Length" statistical criterion :

$$MDL(k) = -\log \left(\frac{GM(k)}{AM(k)} \right)^{(M-k)P} + \frac{k}{2}(2M-k)\log(P) \quad (1.27)$$

where $GM(k)$ and $AM(k)$ are respectively the geometric and arithmetic means of the $(M-k)$ smallest eigenvalues, given by :

$$GM(k) = \prod_{i=k+1}^M \lambda_i^{\frac{1}{M-k}} \quad (1.28)$$

and

$$AM(k) = \frac{1}{M-k} \sum_{i=k+1}^M \lambda_i \quad (1.29)$$

TLS-ESPRIT method, as well as MUSIC method, relies on the covariance matrix eigendecomposition. In the next subsection, we present a method which does not rely on eigendecomposition, namely the propagator.

1.3.3 Propagator method

The main property of Propagator method is that it does not rely on the eigendecomposition of the covariance matrix. From the data vector $\mathbf{z} = [z(1), \dots, z(N)]^T$ we perform spatial smoothing while estimating the covariance matrix : we build K vectors $\mathbf{z}_l = [z(l), \dots, z(M+l-1)]^T$, $l = 1, \dots, K$ of length M with $d < M \leq N-d+1$. The transfer matrix $\mathbf{A}_M(\Phi)$ has size $M \times d$. Propagator exploits the linear dependence between the lines of transfer matrix $\mathbf{A}_M(\Phi)$ (see Eq. (1.4)) : $\mathbf{A}_M(\Phi)$ has d independent columns and $M > d$ lines. With Propagator method [90], we estimate the phase shift values $\{\varphi_k, k = 1, \dots, d\}$. Propagator method [90] relies on the partition of matrix $\mathbf{A}_M(\Phi)$:

$$\mathbf{A}_M^H(\Phi) = [\mathbf{A}_1^H \quad | \quad \mathbf{A}_2^H] \quad (1.30)$$

\mathbf{A}_1 is a $(d \times d)$ matrix and \mathbf{A}_2 is a $(M-d) \times d$ matrix. Matrix $\mathbf{A}_M(\Phi)$ has d columns and then its rank is up to d . If we suppose that the rows (or columns) of \mathbf{A}_1 are linearly independent, there exists a linear relationship between matrix \mathbf{A}_1 and matrix \mathbf{A}_2 :

$$\mathbf{A}_2 = \boldsymbol{\Pi}^H \mathbf{A}_1, \quad (1.31)$$

where $\boldsymbol{\Pi}$ is a matrix of size $(d \times (M - d))$.

Defining as the "propagator operator" an $(M \times (M - d))$ matrix \mathbf{Q} such that :

$$\mathbf{Q}^H = [\boldsymbol{\Pi}^H \quad | \quad -\mathbf{I}] \quad (1.32)$$

where \mathbf{I} is the $((M - d) \times (M - d))$ identity matrix, we get :

$$\mathbf{Q}^H \mathbf{A}_M(\Phi) = \boldsymbol{\Pi}^H \mathbf{A}_1 - \mathbf{A}_2 = \mathbf{0}. \quad (1.33)$$

The operator $\boldsymbol{\Pi}$ has to be estimated in order to build the propagator matrix \mathbf{Q} . Let \mathbf{R}_{zz} be the covariance matrix of signals $\{\mathbf{z}_l, l = 1, \dots, K\}$. We partition the covariance matrix of the received signals as follows :

$$\mathbf{R}_{zz} = [\mathbf{G} \quad | \quad \mathbf{H}] \quad (1.34)$$

where \mathbf{G} is of size $M \times (M - d)$. Matrix $\boldsymbol{\Pi}$ is obtained from \mathbf{G} and \mathbf{H} by minimizing the Frobenius norm of $(\mathbf{H} - \mathbf{G} \boldsymbol{\Pi})$, which results in [90, 23] :

$$\boldsymbol{\Pi} = [\mathbf{G}^H \mathbf{G}]^{-1} \mathbf{G}^H \mathbf{H}. \quad (1.35)$$

The expected values for φ are such that they lead to the d strongest local maxima of function f defined as : $f(\varphi) = (|\mathbf{Q}^H \mathbf{a}(\varphi)|^2)^{-1}$ for values of θ between $-\pi$ and π . Propagator then yields an estimate of the directions-of-arrival, by partitioning the covariance matrix, and without performing its eigendecomposition, contrary to more commonly used methods such as MUSIC or TLS-ESPRIT. We propose in the following a version of this method which is even faster, namely the Propagator applied to the generated signals.

1.3.4 Propagator method applied to signal

This version differs from the original Propagator method that partitions the covariance matrix in the sense that operator $\boldsymbol{\Pi}$ is estimated by partitioning the data vector \mathbf{z} . We first select the M first components of vector \mathbf{z} . This yields a vector which has same length as the signal realizations which are employed to compute the covariance matrix in the methods that were previously presented. Length M vector \mathbf{z} is partitioned as follows :

$$\mathbf{z} = [\mathbf{z}_1^T \mathbf{z}_2^T]^T \quad (1.36)$$

where \mathbf{z}_1 is a length d vector and \mathbf{z}_2 is a length $M - d$ vector. Using Eq. (3.8) and Eq. (1.30), we obtain the following expression for signal \mathbf{z} as a function of $\mathbf{A}_1(\Phi)$:

$$\mathbf{z} = \begin{bmatrix} \mathbf{A}_1(\Phi)\mathbf{s} + \mathbf{n}_1 \\ \vdots \\ \mathbf{A}_2(\Phi)\mathbf{s} + \mathbf{n}_2 \end{bmatrix}$$

Using Eq. (1.31), we obtain :

$$\mathbf{z} = \begin{bmatrix} \mathbf{A}_1(\Phi)\mathbf{s} + \mathbf{n}_1 \\ \vdots \\ \mathbf{A}_2(\Phi)\mathbf{s} + \mathbf{n}_2 \\ \hline \mathbf{\Pi}^H \mathbf{A}_1(\Phi)\mathbf{s} + \mathbf{n}_2 \end{bmatrix} \quad (1.37)$$

where \mathbf{n}_1 (respectively \mathbf{n}_2) is a length d (respectively $M - d$) vector.

To obtain a stable estimate for $\boldsymbol{\Pi}^H$, we reformulate the problem as a regularized unconstrained minimization problem, so that the proposed solution for $\boldsymbol{\Pi}^H$ minimizes [56] :

$$g(\boldsymbol{\Pi}) = |\mathbf{z}_2 - \boldsymbol{\Pi}^H \mathbf{z}_1|^2 + \varsigma |\boldsymbol{\Pi}^H|^2 \quad (1.38)$$

where ς is a Lagrange multiplier [56]. First term in Eq. (1.38) ensures that the estimated solution has small residuals, while the second favors "well-behaved solutions" [56].

$g(\boldsymbol{\Pi})$ is minimum when its derivative with respect to $\boldsymbol{\Pi}$ is null. The optimal solution is :

$$\boldsymbol{\Pi}^H = \mathbf{z}_2 \mathbf{z}_1^H (\mathbf{z}_1 \mathbf{z}_1^H + \varsigma \mathbf{I})^{-1} \quad (1.39)$$

where \mathbf{I} is the identity matrix.

Matrix \mathbf{Q} is computed from matrix $\boldsymbol{\Pi}$ as in Eq. (1.32).

From Eq. (1.39), we notice that, if $\varsigma > 0$, matrix $(\mathbf{z}_1 \mathbf{z}_1^H + \varsigma \mathbf{I})$ is invertible. Regularization constant ς then turns the problem into a well-conditionned one. By avoiding the estimation of a covariance matrix, we expect a faster algorithm.

We notice that $\boldsymbol{\Pi}^H$ is a rank 1 matrix and cannot yield a correct estimate of several parameter values.

To cope with it, we adapt a spatial smoothing procedure, using all signal realizations \mathbf{z}_l , $l = 1, \dots, K$. We use the signals \mathbf{z}_{l1} , \mathbf{z}_{l2} , $l = 1, \dots, K$ that result from the partition in a length d signal and a length $M - d$ signal of each signal realization.

$$\boldsymbol{\Pi}^H = \frac{1}{K} \sum_{l=1}^K (\mathbf{z}_{l2} \mathbf{z}_{l1}^H (\mathbf{z}_{l1} \mathbf{z}_{l1}^H + \varsigma \mathbf{I})^{-1}) \quad (1.40)$$

In Eq. (1.40), $\boldsymbol{\Pi}^H$ is a rank $K > d$ matrix and may yield d parameter values.

1.3.5 Balance on high-resolution methods

The main advantage of TLS-ESPRIT method is that it avoids the research procedure of the maxima of a pseudospectrum, inherent in MUSIC method, and it yields the parameter estimates in terms of eigenvalues [114]. Then TLS-ESPRIT method exhibits a low computational load compared to MUSIC algorithm. Propagator method either estimates the covariance matrix without performing eigendecomposition, or works directly on a single signal realization. This yields a fast algorithm. High-resolution performances will be exemplified in next chapters of this manuscript. In the next section, we present an optimization method which will be adapted in parts 1 and 2 of the manuscript for array processing and image understanding applications.

1.4 Optimization method : combination of gradient or DIRECT algorithm with spline interpolation

We propose here an optimization method that aims at fitting a data vector by a prior model. This optimization method finds a set of N unknown parameters, starting from a set of N initialization values. The components of the complex-valued data vector are denoted by $z_{input}(1), \dots, z_{input}(N)$ and stored in vector $\mathbf{z}_{input} = [z_{input}(1), \dots, z_{input}(N)]^T$. The unknown values are denoted by $\rho(1), \dots, \rho(N)$, and stored in vector $\rho = [\rho(1), \dots, \rho(N)]^T$. The components of the model vector which is supposed to fit the data are denoted by $z_\rho(1), \dots, z_\rho(N)$ and stored in vector $\mathbf{z}_\rho = [z_\rho(1), \dots, z_\rho(N)]^T$. The initialization values are denoted by $\rho_0(1), \dots, \rho_0(N)$ and stored in vector $\rho_0 = [\rho_0(1), \dots, \rho_0(N)]^T$. The proposed optimization method minimizes a criterion denoted by J , which is expressed as a function of all

unknown values $\rho(1), \dots, \rho(N)$ and depends on the processed data. The proposed optimization algorithm is iterative and recursive, that is, a coordinate vector ρ_{l+1} computed at step $l + 1$ of the algorithm depends on coordinate vector ρ_l computed at step l . At step l of the recursive procedure, a coordinate vector ρ_l is computed from criterion J defined by :

$$J(\rho_l) = \|\mathbf{z}_{\text{input}} - \mathbf{z}_{\rho_l}\|^2 \quad (1.41)$$

where $\|\cdot\|$ represents the norm induced by the usual scalar product of \mathbb{C}^N .

To minimize the criterion of previous equation, we can use gradient method, with either constant or variable step size. The vectors of the series are obtained by the relation :

$$\forall l \in \mathbb{N} : \rho_{l+1} = \rho_l - \lambda \nabla(J(\rho_l)), \quad (1.42)$$

where λ is the step for the descent. The recurrence loop is $\rho_l \rightarrow \mathbf{z}_{\text{estimated for } \rho_l} \rightarrow J(\rho_l)$. We stop when the gradient becomes lower than a threshold.

A variable step gradient method is such that the step λ depends on the iteration index l . For instance we can choose $\lambda_{l+1} = 1.05\lambda_l$ for instance.

Another optimization method that may be used to minimize the criterion of previous equation is DIRECT (DIviding RECTangles). The main property of DIRECT is that it obtains the global minimum of a function. DIRECT does not require the knowledge of the criterion to be minimized. Few parameters are needed in the optimization procedure. DIRECT normalizes the research space and obtains hypercubes from this research space. Then DIRECT evaluates the solution which is at the center of each hypercube. Hypercubes are divided into smaller cubes : The smaller the evaluation at the center of a cube, the smaller the cube. In this way, the algorithm favors the regions of the initial research space where the evaluations are small. The division process is performed recursively as is shown in Fig. 1.3. The selection of optimal blocks is based on a compromise between the size and the evaluation at the center of the block. Thus, DIRECT selects small blocks that provide an interesting (small) evaluation in the sense of the criterion which is minimized, and DIRECT selects large blocks which are less interesting in the sense of the minimized criterion. The main drawback of DIRECT algorithm is that it exhibits an elevated computational load.

A more elaborated optimization algorithm consists in combining either gradient or DIRECT algorithms with spline interpolation. The main purpose of this method is to reduce the computational load of DIRECT method and to obtain a solution vector that exhibits some continuity properties.

Let O be an integer smaller than N . A cubic spline f interpolating on the partition $\{y(1), \dots, y(O)\}$ of $\{1, \dots, N\}$ that we call "node points", to the elements $\rho(1), \dots, \rho(N)$, is a function for which $f(y(o)) = \rho(y(o))$ for $o = 1, \dots, O$. It is a piecewise polynomial function that consists of $O - 1$ cubic (third order) polynomials f_o defined on the interval $[\rho(y(o)), \rho(y(o + 1))]$. Furthermore, each f_o is joined at $y(o)$, for $o = 2, \dots, O - 1$, so that $\rho'(y(o)) = f'(y(o))$ and $\rho''(y(o)) = f''(y(o))$ are continuous. The o^{th} polynomial curve, f_o , is defined over the fixed interval $[y(o), y(o + 1)]$ and is a third order polynomial. Function f_o is the one that minimizes the integral E of Eq. (1.43).

$$E = \int_{u=y(o)}^{u=y(o+1)} \frac{f''(u)^2}{(1 + f'(u)^2)^{5/2}} du \quad (1.43)$$

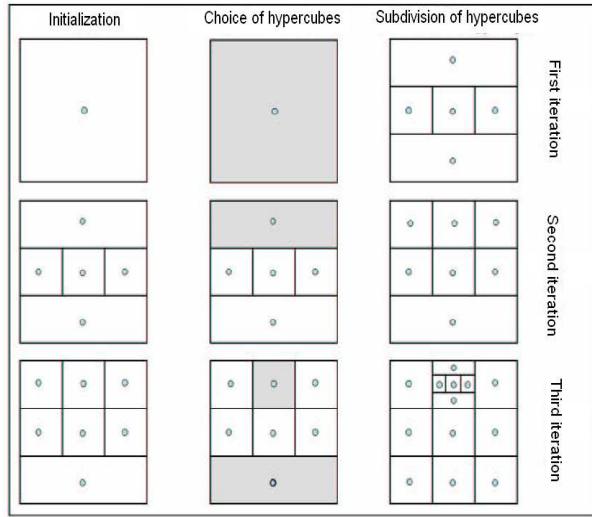


Figure 1.3 — Principles of DIRECT method.

Then, interpolation provides an approximate value of N elements starting from O elements. We propose to reduce the number of unknowns retrieved by DIRECT or gradient algorithm, thanks to interpolation by cubic spline polynomials [134] at each iteration of DIRECT or gradient. In particular the computational load of DIRECT increases drastically with the number of unknown parameters, so using interpolation accelerates drastically DIRECT algorithm. At each step of the recursive algorithm, vector ρ_l , which contains N components, is computed by interpolating between $O < N$ "node" values that are retrieved by DIRECT. Then, we propose spline interpolation to obtain the N components of $\hat{\rho}$. The more interpolation nodes, the more precise the estimation, but the slower the algorithm. The proposed algorithm can be generalized into a multiresolution one : the proposed multiresolution algorithm consists of several steps. At each step, gradient or DIRECT algorithm combined with spline interpolation is run. Consecutive steps are associated with an increasing number of nodes. Hence an improved resolution in the course of the procedure.

The performances of the novel proposed optimization method that combines either gradient or DIRECT algorithms with spline interpolation will be illustrated in chapter 2, and in part II of the manuscript.

1.5 Conclusion of chapter 1

In this chapter were introduced the basic concepts concerning array processing. We emphasized second order statistics and the distinction between signal subspace and noise subspace. Then we gave a short presentation of two high-resolution methods. These methods resolve close-valued directions-of-arrival in a noisy environment. Especially, TLS-ESPRIT method exhibits a low computational load compared to MUSIC method. We also present a novel optimization method based on either gradient or DIRECT method, that will be adapted in several applications in parts 1 and 2 of the manuscript. In the next chapter, we will study a concrete application case in array processing : DOA estimation in the case of phase distortions.

Phase distortion estimation by DIRECT and spline interpolation algorithms

An important cause of performance loss in source localization in underwater acoustics is that towed flexible antennas deviate from the assumed rectilinear shape. In this work the localization of sources in presence of phase errors is studied. Cancellation of phase errors is necessary to solve the source localization problem. A previous work led to interesting results for antennas composed of a few sensors. We propose here a novel algorithm which is adapted to the antennas composed of a large number of sensors, keeping a small computational load. Our method is based on an orthogonality property between signal and noise subspaces and a novel optimization method : the robust DIRECT (DIviding RECTangles) algorithm accelerated by spline interpolation [85]. The performances of the proposed method are illustrated by applying it to the characterization of three sources.

Section 2.1 emphasizes the interest of this work, with respect to existing methods. Section 2.2 presents the problem statement. The method proposed is described in Section 2.3, where we estimate phase distortions by an optimization method. In Section 2.4, we evaluate the performances of the proposed method by an example with three wavefronts and a statistical study. Concluding remarks about this chapter are provided in section 2.5.

This work has been published in

J. Marot and S. Bourennane, "Phase Distortion Estimation by DIRECT and spline interpolation algorithms", IEEE Signal Processing Letters, Vol. 14, no. 7, pp : 461-464, July 2007.

2.1 Introduction

Finding the parameters of wavefronts impinging on a distorted antenna has been the purpose of various previous studies [100, 63, 91]. In this chapter, we consider a distorted antenna and wavefronts that are distorted because of the inhomogeneity of the propagation medium [63]. Previous works considered antennas with a relatively low number of sensors [63, 91, 115]. Contrary to these works, we consider antennas composed of a large number of sensors. The proposed method must still lead to small computational times, to permit an easy real time implementation. The proposed method is particularly useful when the number of sensors

is much greater than the number of sources, which is usually the case in practice [91].

2.2 Problem statement

Fig. 2.1 represents distorted wavefronts impinging on a distorted antenna (sensors 1, 2, ..., N). A limitation of the high-resolution methods based on the estimation of a covariance matrix is their inability to perform satisfactorily when a received wavefront is different from the assumed linear phase wavefront model. In [63] a statistically efficient weighted signal subspace fitting algorithm is proposed, that exhibits low root mean square error values. We consider the case where the antenna is composed of many sensors. In order to keep small computational times, we propose an accelerated version of DIRECT algorithm [64], that gives valuable results in image processing [27].

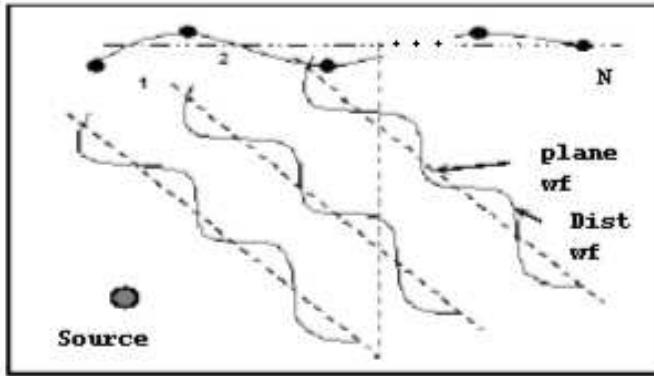


Figure 2.1 — Distorted wavefronts received on a distorted antenna composed of a large number of sensors.

We consider a d spacing linear array of N identical sensors which receive the narrow-band signals from P sources, centered at frequency f_0 , in the presence of an additive noise. Using vector notations, the Fourier transforms of the outputs of the array can be written as [100, 63, 91, 115] : $\mathbf{z}(f_0) = \mathbf{A}(\phi)\mathbf{s}(f_0) + \mathbf{n}(f_0)$, where the $N \times P$ matrix $\mathbf{A}(\phi) = [\mathbf{a}(\phi_1), \mathbf{a}(\phi_2), \dots, \mathbf{a}(\phi_P)]$ is the transfer matrix of the source sensor-array system [115]. The p^{th} column of $\mathbf{A}(\phi)$, called directional vector of the p^{th} source, is $\mathbf{a}(\phi_p) = [e^{-j\varphi_{p1}}, e^{-j\varphi_{p2}}, \dots, e^{-j\varphi_{pN}}]^T$, where :

$$\varphi_{pi} = \varphi_{pi}^{lin} + \Delta\varphi_{pi} \quad (2.1)$$

and $\varphi_{pi}^{lin} = 2\pi f_0 \frac{d}{c} (i - 1) \sin(\theta_p)$. θ_p is the direction-of-arrival of the p^{th} source, c is the wave speed, and $\Delta\varphi_{pi}$ is an additive phase shift value, also called distortion, that can be due to the deformation of the array and to the perturbation of the impinging wavefronts. The vector containing all phase values for source p is denoted by : $\phi_p = [\varphi_{p1}, \varphi_{p2}, \dots, \varphi_{pN}]^T$.

$\mathbf{z}(f_0) = [z_1(f_0), z_2(f_0), \dots, z_N(f_0)]^T$, $\mathbf{s}(f_0) = [s_1(f_0), s_2(f_0), \dots, s_P(f_0)]^T$ and $\mathbf{n}(f_0) = [n_1(f_0), n_2(f_0), \dots, n_N(f_0)]^T$ are the Fourier transforms of the array outputs, the source signals and noise vectors, respectively. We consider that K snapshots are available :

$\{\mathbf{z}_1(f_0), \mathbf{z}_2(f_0), \dots, \mathbf{z}_K(f_0)\}$. The high-resolution method proposed in [115] uses the eigendecomposition of the cross-spectral matrix of the received signals : $\boldsymbol{\Gamma}_{zz}(f_0) = \frac{1}{K} \sum_{l=1}^K \mathbf{z}_l(f_0) \mathbf{z}_l^H(f_0)$, where H denotes transpose conjugate. By denoting $\mathbf{V}(f_0)$ the matrix containing the vectors of the noise subspace [115] associated with the $(N-P)$ smallest eigenvalues, we get [115] : $\mathbf{V}(f_0)^H \mathbf{A}(\phi) = \mathbf{0}$. In the rest of the chapter the frequency f_0 is omitted. This orthogonality property is used in MUSIC algorithm [115]. The inverse of $|\mathbf{V}(f_0)^H \mathbf{a}(\phi)|_F$, where $|\cdot|_F$ denotes Frobenius norm, is called pseudospectrum. Firstly, using MUSIC algorithm, initial values of directions-of-arrival are estimated. Secondly, we calculate for each initial direction-of-arrival $\hat{\theta}_{0p}$, $p = 1, \dots, P_0$ ($P_0 \leq P$) a vector of phase values $\hat{\phi}_p^0 = [\hat{\varphi}_{p1}^{lin}, \hat{\varphi}_{p2}^{lin}, \dots, \hat{\varphi}_{pN}^{lin}]^T$, where :

$$\hat{\varphi}_{pi}^{lin} = 2\pi f_0 \frac{d}{c} (i-1) \sin(\hat{\theta}_{0p}) \quad (2.2)$$

We use DIRECT algorithm initialized by vector ϕ_p^0 , in order to calculate the expected phase values :

$$\phi_p = \text{argmin}(|\mathbf{V}^H \mathbf{a}(\phi_p)|_F), \quad (2.3)$$

DIRECT performs global optimization [64]. The computational load of DIRECT algorithm grows rapidly when the number of sensors, or equivalently the number of unknown phase values, increases. At any iteration m , we accelerate DIRECT algorithm by reducing the number of retrieved unknowns and then we propose spline interpolation [27] to obtain the N phase values of $\hat{\phi}_p^m$: we interpolate a subset of values of $\hat{\phi}_p^m$, which are retrieved by DIRECT algorithm. The more interpolation nodes, the more accurate the estimation. See 1.4 for details about this optimization method. The series of vectors $\hat{\phi}_p^m$ converges when m tends to infinity [64], towards a vector $\widehat{\phi}_p$ such that $|\mathbf{V}^H \mathbf{a}(\widehat{\phi}_p)|_F$ is minimum.

2.3.2 Cancel the phase shifts in the received signals

Then for each wavefront the optimization method is run anew and initialized by the approximate value of the corresponding direction-of-arrival. For each source p , phase distortions of the received signals are canceled to obtain signals $\mathbf{z}_{l,processed}^p$, $l = 1, \dots, K$ that suit perfectly the method based on the orthogonality between signal and noise subspaces [115]. For one wavefront, phase distortions do not depend only on the physical shape of the antenna but also on the orientation of the wavefront. At this point an approximate value of several directions-of-arrival is available. We get the vector of estimated phase distortions :

$$\widehat{\Delta\phi}_p = [\widehat{\Delta\varphi}_{p1}, \widehat{\Delta\varphi}_{p2}, \dots, \widehat{\Delta\varphi}_{pN}]^T \quad (2.4)$$

corresponding to the current wavefront using Eq. (2.1). We can cancel the phase distortions and compute the signals to be processed defined by :

$$\mathbf{z}_{l,processed}^p = \mathbf{D}^p(\widehat{\Delta\phi}_p) \mathbf{z}_l, \quad l = 1, \dots, K \quad (2.5)$$

where

$$\mathbf{D}^p(\widehat{\Delta\phi}_p) = \text{diag}[e^{j\widehat{\Delta\varphi}_{p1}}, e^{j\widehat{\Delta\varphi}_{p2}}, \dots, e^{j\widehat{\Delta\varphi}_{pN}}] \quad (2.6)$$

Then the signals $\mathbf{z}_{l,processed}^p, l = 1, \dots, K$ are employed in the high-resolution method. This permits to distinguish several close directions-of-arrival. This process is repeated iteratively for each direction-of-arrival : At each iteration it , for one direction-of-arrival index p , the following criterion must hold to pursue the algorithm : $|\hat{\theta}_p^{it} - \hat{\theta}_p^{it-1}| > \epsilon$, *a priori* fixed threshold. That is, we consider we reached the accurate estimate of the p^{th} direction-of-arrival when the estimated value does not vary from an iteration to another.

2.3.3 Proposed algorithm

In the following we summarize the proposed algorithm to improve MUSIC algorithm in the presence of phase distortions :

1. Estimate grossly initial values of several directions-of-arrival $\hat{\theta}_{0p}, p = 1, \dots, P_0$, ($P_0 \leq P$) with MUSIC [115].
2. For each $\hat{\theta}_{0p}$ repeat the following process, until the estimate of the direction-of-arrival does not vary from an iteration to another :
 - (a) Estimate $\widehat{\phi}_p$ by DIRECT algorithm associated with spline interpolation : Retrieve the phase shifts between the vector $\widehat{\phi}_p^0$ of phase values corresponding to a plane wavefront (see Eq. (2.2)), and a vector of phase values described by Eq. (2.1) ;
 - (b) Cancel the phase shifts in the received signal realizations (see Eq. (2.5)) ;
 - (c) apply the MUSIC method [115] to the signal realizations obtained by Eq. (2.5), only in a reduced angle interval around the first estimation $\hat{\theta}_{0p}$ of θ_p . Owing to phase cancellation, several close sources may be resolved inside the considered interval.
3. Regroup the P_0 pseudospectra calculated around the P_0 initial directions-of-arrival of the high-resolution method. Each maximum argument is the direction-of-arrival of one source.

2.4 Simulation results

An antenna of $N = 100$ equi-spaced sensors with inter-element spacing $d = \frac{c}{32f_0}$ is used. Such an inter-element spacing avoids any phase indetermination for any angle. Time signals collected upon each sensor are 5 second and 1000 sample long. The additive noise is independent from the signals and with diagonal cross-spectral matrix $\mathbf{\Gamma}_n = \sigma^2 \mathbf{I}$, \mathbf{I} being Identity matrix. The signal-to-noise ratio (*SNR*) is defined as the ratio of the power of each source signal to the average power of the noise. We consider the configuration with $P = 3$ sources impinging on the array at $\theta_1 = -5.0^\circ$, $\theta_2 = 8.0^\circ$, and $\theta_3 = 18.0^\circ$. The distribution of the time delay of the received signal at the sensor i is given, for $(i = 1, \dots, N)$, $(p = 1, \dots, P)$, by :

$$\tau_{pi} = \frac{d}{c}((i-1)\sin(\theta_p) + \alpha_p \sin\left(\frac{2\pi(i-1)}{T_p}\right) + t_i), \quad (2.7)$$

where $[t_1, \dots, t_N]^T$ is Gaussian vector with zero-mean used as perturbation, α_p is a distortion amplitude factor, T_p is a distortion period factor. Fig. 2.2 shows the three impinging wavefronts, including the phase shifts provoked by the distortions of the antenna. Fig. 2.3 shows the pseudospectra obtained with MUSIC, for an experiment with $SNR = 16 \text{ dB}$. Source localization is improved when we estimate the shape of the antenna with 25 iterations of DIRECT in

our optimization method and when we perform phase cancellation : see Fig. 2.3(c), compared with that obtained without phase cancellation (see Fig. 2.3(a)), and compared with the case where only 10 iterations of DIRECT are run (see Fig. 2.3(b)). When phase cancellation is not done, the pseudospectrum of MUSIC exhibits only two maxima. The eigendecomposition of the cross-spectral matrix of signal realizations leads to two dominant eigenvalues. Therefore we assume that the dimension of the signal subspace is $P_0 = 2$. The values obtained for the directions-of-arrival are $\hat{\theta}_{01} = -4.0^\circ$ and $\hat{\theta}_{02} = 16.5^\circ$. Then we initialize our recursive procedure with $\hat{\theta}_{01}$, and $\hat{\theta}_{02}$. When our algorithm is applied and phase cancellation is done for each direction-of-arrival, the values obtained for the directions-of-arrival are $\hat{\theta}_1 = -4.9^\circ$, $\hat{\theta}_2 = 8.2^\circ$, and $\hat{\theta}_3 = 17.6^\circ$. One supplementary source is resolved in place of the single source with direction $\hat{\theta}_{02}$, thanks to phase cancellation. For one iteration of the procedure of estimation of the phase distortions, computational time is 25 sec. for our optimization method with 10 spline interpolation nodes and 25 iterations of DIRECT, on a 3.0 Ghz pentium 4 processor.

We propose a statistical study, considering

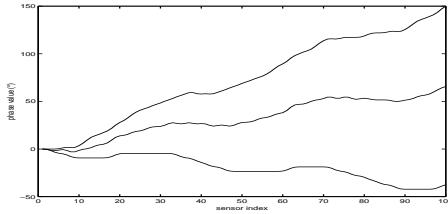


Figure 2.2 — Evolution of the phase of the three impinging wavefronts as a function of the sensor index.

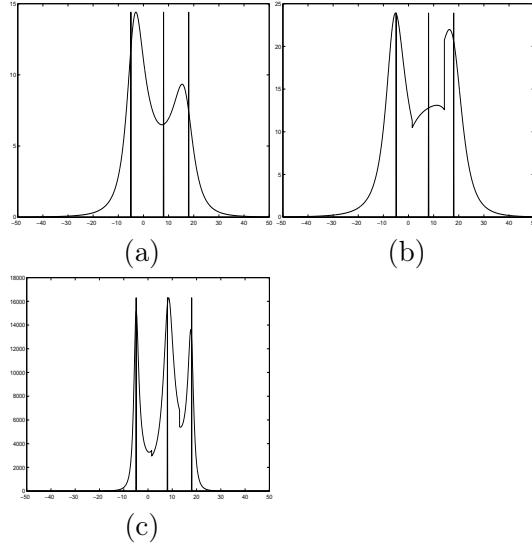


Figure 2.3 — Pseudospectra of MUSIC, represented in the interval $[-50^\circ, 50^\circ]$, superimposed to the expected values (vertical lines) : (a) without cancellation of phase distortions ; (b) and (c) with the estimation with 10 (resp. 25) iterations of DIRECT and cancellation of the phase distortions.

the same wavefront parameters as above, performing 1000 trials. Mean error over the values of $\theta_1, \theta_2, \theta_3$ is :

$ME = \frac{1}{3000} \sum_{p=1}^3 (\sum_{j=1}^{1000} (\hat{\theta}_{pj} - \theta_p))$, where j indexes the trials and $\hat{\theta}_{pj}$ is the estimation obtained at the j^{th} trial for the p^{th} direction-of-arrival. Root mean square error is defined by :
 $RMSE = \sqrt{\frac{1}{3000} \sum_{p=1}^3 \sum_{j=1}^{1000} (\hat{\theta}_{pj} - \theta_p)^2}$.

The statistical results obtained are always acceptable (see Table 2.1).

SNR (dB)	30	34	38	42	46
ME ($^{\circ}$)	0.13	0.08	0.06	0.04	0.02
$RMSE$ ($^{\circ}$)	0.09	0.06	0.03	0.01	0.005

Tableau 2.1 — ME and $RMSE$ values over angle estimation for several SNR values.

2.5 Conclusion of chapter 2

A novel algorithm for the characterization of distorted wavefronts and the improvement of resolution in the presence of phase distortions is proposed. We focused on the case of antennas with many sensors. To keep low computational times, we proposed a version of "DIRECT" algorithm accelerated by spline interpolation to retrieve distorted wavefronts impinging on a distorted antenna. We have shown that, taking into account phase distortions and canceling them, the estimation of the directions-of-arrival of several wavefronts is possible and accurate. In the next chapter, we show that array processing methods can be transposed to an image processing problem. In particular, we show that subspace-based methods of array processing yield close-valued feature parameters.

Second part

Array processing and optimization
methods applied to image
understanding

Straight line and nearly straight contour retrieval, by high-resolution methods of array processing and optimization methods

This chapter is devoted to the estimation of rectilinear and distorted contours in images by High Resolution and optimization methods. In the case of rectilinear contours it has been shown that it is possible to transpose this image processing problem to an array processing problem. The existing straight line characterization method called Subspace-based Line Detection (SLIDE) leads to models with orientations and offsets of straight lines as the desired parameters. In this part, we propose several high-resolution methods which are adapted to the retrieval of the orientations of straight lines. In particular, we propose a high-resolution method named Propagator which does not rely on the eigendecomposition of the covariance matrix, and which is then faster than other high-resolution methods such as MUSIC or TLS-ESPRIT. We adapt several optimization algorithms : we illustrate the performances of gradient algorithm with images which are noise-free or only slightly impaired. Then we adapt a more robust optimization method, that combines DIRECT (DIviding RECTangles) with spline interpolation. DIRECT is a global optimization method and is then robust to noise impairment, and spline interpolation yields a continuous contour with continuity and curvature constraints. This method will be exemplified in particular on real-world or noisy images. This chapter is organized as follows :

In section 3.1, we present an overview of the last advances in contour retrieval methods, and mention the SLIDE method that was proposed to cope with the retrieval of straight lines with close-valued orientations in a noisy environment. In section 3.2, we state the model which is adopted for a processed image, and we remind the signal generation scheme which is used in SLIDE method to generate signals out of the image data. In section 3.3, we review TLS-ESPRIT method adapted to the estimation of the orientations of straight lines, we propose to adapt Propagator method, which either relies on the partition of the covariance matrix and not on its eigendecomposition, or works directly on the generated signal. Propagator type

methods are faster than TLS-ESPRIT method. We also propose a novel method for offset estimation, which is based on a high-resolution method and is faster than the extension of the Hough transform. In section 3.4, we present some results obtained on hand-made and real-world images with the proposed straight line estimation methods. In section 3.5, we propose an automatic threshold procedure : a squared criterion compares the signal generated from the threshold image with an image containing a straight line that fits an expected contour. The threshold image that minimizes the chosen criterion contains a selected region aside the fitting straight line. In section 3.6, we present the principles of the proposed optimization method for distorted contour retrieval, and we adapt the gradient optimization algorithm : we solve an inverse problem to retrieve contour distortions from the phase of the generated signal. In section 3.7, we propose a generalization of the proposed method to the retrieval of several contours. In section 3.8, we present some simulation results obtained with gradient algorithm. In section 3.9, we propose to associate gradient method with Propagator to retrieve several distorted contours. Section 3.10 presents the results obtained with this particular method. Section 3.11 presents a robust optimization method which associates gradient with spline interpolation, which copes with noisy images. In section 3.12 we present the results obtained by the combination of gradient with spline interpolation, especially on real-world images, and compared with Gradient Vector Flow (GVF).

This work has been published in :

S. Bourennane and J. Marot, "Estimation of straight line offsets by a high-resolution method", IEE Vision Image and Signal Processing, vol. 153, pp. 224-229, 2006., in

S. Bourennane and J. Marot, "Contour estimation by array processing methods", Applied signal processing, article ID 95634, 15 pages, 2006., in

J. Marot and S. Bourennane, Propagator method for an application to contour estimation, Pattern recognition letters, vol. 28, no. 12, pp. 1556-1562, sept. 2007., in

S. Bourennane and J. Marot, "Line parameters estimation by array processing methods", IEEE ICASSP'05, Vol. 4, pp. 965-968, Mar. 2005, and in

S. Bourennane and J. Marot, "Optimization and interpolation for distorted contour estimation", IEEE ICASSP'06, Vol. 2, pp. 717-20, April 2006.

3.1 Introduction

Shape description is an important goal of computational vision and image processing. Giving the characteristics of lines or distorted contours is faced in robotic way screening, measuring of wafer track width in microelectronics, aerial image analysis, vehicle trajectory and particle detection. Distorted contour retrieval is also encountered in medical imaging. To solve this problem, active contour or "snakes" segmentation was proposed [69, 141, 103, 105, 15, 66, 88, 137, 135, 34, 29]. The idea behind active contour segmentation methods is to evolve a parametric curve in the image domain. The curve evolution is described by a partial differential equation that drives the active contour to a minimum of a functional. Approaches to active contour segmentation can be either region or edge-based. Among region-based methods, level set methods in particular, introduced in [103] perform texture segmentation. The key idea of level set methods is to represent the evolving curve with an implicit function which takes positive values inside an image domain with some texture properties, negative values outside this image domain, and null values on the boundaries of this domain. Level set methods allow for automatic change of topology, such as merging

and breaking [105, 15, 66]. Level set segmentation results have been improved by taking into account the noise model [88]. This strong noise environment, but intuitively assume that gray level distributions follow different probability density functions, depending on the region of the image. Edge-based segmentation schemes aim at finding out the transitions between uniform areas, rather than directly identifying them [69, 137, 135]. In particular, edge-based segmentation schemes retrieve contours that cross a uniform background. Such algorithms fall into two steps : they firstly compute an edge strength map of the scene, and finally extract the local maxima of this map. Edge-based segmentation schemes have improved, considering robustness to noise and sensitivity to initialization [137, 135]. They provide valuable results, especially with concave contours. Some active contour methods were combined with spline type interpolation to reduce the number of control points in the image [29]. This increases the robustness to noise and computational load. In particular, [111] uses smoothing splines in the B-spline interpolation approach of [125]. The balloon models [136] are motivated by the desire to drive the snake automatically to a good position, thereby decreasing the dependence on the initial conditions. Balloons introduce an additional force term which pushes the contour out (or in) along its normal. Snake or balloon models make use of information along the boundary and require good initial estimates to yield correct convergence. In [137] the proposed "Gradient Vector Flow" (GVF) method which is proposed uses a blurring kernel to extend the capture range, which relaxes the conditions of good initialization. GVF provides valuable results, but is prone to shortcomings : contours with high curvature may be skipped unless an elevated computational load is devoted. Concerning straight lines in particular, in [71] the extension of the Hough transform retrieves roughly aligned points main direction. This method gives a good resolution even with noisy images, but restrictions remain in its use. Its resolution depends on the quantization step, and the computational load for the bi-dimensional search of the maxima is elevated. Least-squares fit of straight lines seeks to minimize the summation of the squared error-of-fit with respect to measures [47, 36]. This method is sensitive to outliers. An original approach consists in adapting high-resolution methods [114, 90, 5] for straight line segmentation [5, 6, 55, 72, 54, 26, 25, 2, 24]. In particular, SLIDE (Subspace-based LIne DEtection) algorithm associates a linear antenna with the processed image [5, 6].

3.2 Data model, signal generation out of the image data

Let I be the recorded image (see Fig. 3.1 (a), from [4]). We consider that I is compound of d straight lines and an additive uniformly distributed noise. Moreover, in this model, image I is supposed to contain only pixel values '1' or '0' [5]. Pixels '1', which form the straight lines, are called "useful pixels", whereas '0' pixels are associated with the background. The image size is $N \times C$: it contains N lines and C columns. Each straight line within an image is associated with an offset x_0 on the X axis and an angle θ , between this straight line and the line of equation $x = x_0$ (Fig. 3.1(b)).

It is possible to establish an analogy between source localization [107, 133] in array processing and line detection in image processing. For this purpose some signals are generated out of the image data [3] : we create artificially, out of the N lines of the image-matrix, N inputs to a linear array composed of N equidistant sensors located along the image side. A formalism

adopted in [5] allows signal generation, by the following computation :

$$z(i) = \sum_{k=1}^C I(i, k) \exp(-j\mu k), \quad i = 1, \dots, N \quad (3.1)$$

where $\{I(i, k); i \in \{1, \dots, N\}; k \in \{1, \dots, C\}\}$ denote the image pixels, and where μ is a propagation parameter [7, 3] that can be constant or variable : we can consider a constant or variable parameter propagation scheme. We focus in the following on the case where a binary image is considered. The contours are composed of 1-valued pixels also called "edge pixels", whereas 0-valued pixels compose the background. When d lines are present in the image, there are d non zero pixels on the i^{th} line of the image-matrix, localized on the columns x_1, \dots, x_d respectively. The signal received by the sensor in front of the i^{th} line, when no noise is present in the image, is written [4] :

$$z(i) = \sum_{k=1}^d \exp(-j\mu x_k(i)), \quad i = 1, \dots, N \quad (3.2)$$

First we consider the case of only one line with angle θ and offset x_0 , as it is shown on Fig. 3.1 b). Supposing that the width, along the X axis, of each line is equal to one pixel, the horizontal coordinate of a straight line pixel in front of the i^{th} sensor is :

$$x(i) = x_0 - (i - 1) \tan(\theta) \quad (3.3)$$

Hence the signal received on the i^{th} sensor is written :

$$\begin{aligned} z(i) &= \exp(-j\mu x(i)) \\ z(i) &= \exp(-j\mu x_0) \exp(j\mu(i - 1) \tan(\theta)) \end{aligned} \quad (3.4)$$

In this expression we took into account the possible values of each pixel '1' or '0'. In the presence of d different straight lines in the image and an additive noise, the signal received on the sensor i is :

$$z(i) = \sum_{k=1}^d \exp(j\mu(i - 1) \tan(\theta_k)) \exp(-j\mu x_{0k}) + n(i), \quad (3.5)$$

where $n(i)$ is the noise on the i^{th} line, that can be due to several noise pixels. As a consequence of the presence of noisy non zero pixels, the linear variation of the phase in Eq. (3.5) is no longer verified.

Defining :

$$a_i(\theta_k) = \exp(j\mu(i - 1) \tan(\theta_k)), \quad s_k = \exp(-j\mu x_{0k}), \quad (3.6)$$

Equation (3.5) becomes

$$z(i) = \sum_{k=1}^d a_i(\theta_k) s_k + n(i), \quad i = 1, \dots, N. \quad (3.7)$$

Equation (3.15) gives the signal model that will be employed in the following, and that fully characterizes the d lines within the noisy image. Grouping all terms in a single vector, Eq. (3.7) becomes :

$$\mathbf{z} = \mathbf{A}(\theta) \mathbf{s} + \mathbf{n} \quad (3.8)$$

with $\mathbf{A}(\theta) = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_d)]$ where $\mathbf{a}(\theta_k) = [a_1(\theta_k), a_2(\theta_k), \dots, a_N(\theta_k)]^T$, with $a_i(\theta_k) = \exp(j\mu(i - 1) \tan(\theta_k))$, $i = 1, \dots, N$, superscript T denoting transpose.

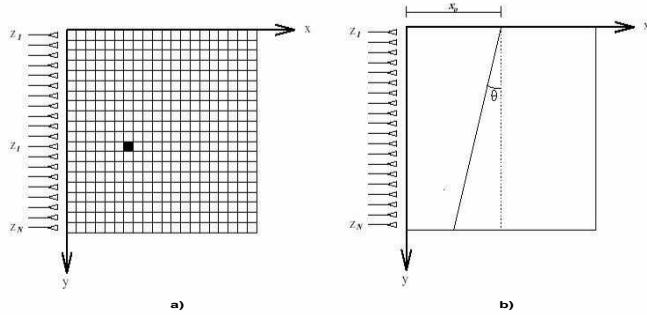


Figure 3.1 — The image model (see [4]) : (a) The image-matrix provided with the coordinate system and the rectilinear array of N equidistant sensors. (b) A straight line characterized by its angle θ and its offset x_0 .

3.3 Straight contour estimation

In this section, we review an existing method for the estimation of the orientation of straight lines. SLIDE (Subspace-based LIne DEtection) uses a particular signal generation method, and high-resolution TLS-ESPRIT method. Starting from SLIDE signal generation scheme, we adapt faster methods, namely Propagator applied to the generated signal and Propagator applied to the covariance matrix. We also propose a high-resolution method for the estimation of close-valued offsets, which is faster than the extension of the Hough transform.

3.3.1 Angle estimation, overview of the SLIDE method

The method for angle estimation falls into two parts : the estimation of a covariance matrix and the application of a total least squares criterion.

Numerous works have been developed in the frame of the research of a reliable estimator of the covariance matrix when the duration of the signal is very short or the number of realizations is small. This situation is often encountered, for instance, with seismic signals. To cope with it, numerous frequency and/or spatial means are computed to replace the temporal mean. In this study the covariance matrix is estimated by using the spatial mean [55]. From the observation vector we build K vectors of length M with $d < M \leq N - d + 1$. In order to maximize the number of sub-vectors we choose $K = N + 1 - M$. By grouping the whole sub-vectors obtained in matrix form, we obtain

$$\mathbf{Z}_K = [\mathbf{z}_1, \dots, \mathbf{z}_K] \quad (3.9)$$

where

$$\mathbf{z}_l = \mathbf{A}_M(\theta)\mathbf{s}_l + \mathbf{n}_l, \quad l = 1, \dots, K. \quad (3.10)$$

$\mathbf{A}_M(\theta) = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_d)]$ is a Vandermonde type matrix of size $M \times d$. The signal is supposed to be independent from the noise ; the components of noise vector \mathbf{n}_l are supposed to be uncorrelated, and to have identical variance. The covariance matrix can be estimated from the observation sub-vectors as it is performed in [4]. Using the sub-vectors in the forward and the backward sense leads to a better estimation of the covariance matrix [133, 109]. The eigen-decomposition of the covariance matrix is, in general, used to characterize the sources

by subspace techniques in array processing. In the frame of image processing the aim is to estimate the angle θ of the d straight lines. Several high-resolution methods that solve this problem have been proposed in the literature [115, 107, 114]. SLIDE algorithm is applied to a particular case of an array consisting of two identical sub-arrays [7]. Through TLS-ESPRIT method, which is detailed in section 1.3.2, SLIDE leads to the following estimated angles [7] :

$$\hat{\theta}_k = \tan^{-1} \left[\frac{1}{(\mu * \Delta)} \operatorname{Im} \left(\ln \left(\frac{\beta_k}{|\beta_k|} \right) \right) \right], \quad (3.11)$$

where $\{\beta_k, k = 1, \dots, d\}$ are the eigenvalues of a diagonal unitary matrix that relates the measurements from the first sub-array to the measurements resulting from the second sub-array. Parameter μ is the propagation constant, and Δ is the distance between two sensors. TLS-ESPRIT yields an accurate estimation of the orientations of the expected straight lines. We propose in the following to adapt high-resolution methods that do not require eigendecomposition of the covariance matrix, and predict that they are faster than TLS-ESPRIT.

3.3.2 Angle estimation by Propagator method

The high-resolution method called "*Propagator*" [90, 23], which led good results in array processing, is used for the first time in the field of image processing. This fast high-resolution method estimates the orientation of straight lines, and we further adapt it to the retrieval of several distorted contours.

First, we consider one contour which is strictly a straight line, with angle θ and offset x_0 (see Fig. 3.1). The horizontal coordinate of a pixel in front of the i th sensor is :

$$x(i) = x_0 - (i - 1)\tan(\theta) \quad (3.12)$$

Hence the signal received on the i th sensor reads :

$$z(i) = \exp(-j\mu x(i)) = \exp(-j\mu x_0) \exp(j\mu(i - 1)\tan(\theta)). \quad (3.13)$$

If d straight lines are expected and additive noise is present, the signal received on sensor i reads :

$$z(i) = \sum_{k=1}^d \exp(j\mu(i - 1)\tan(\theta_k)) \exp(-j\mu x_{0k}) + n(i) \quad (3.14)$$

where $n(i)$ is the noise due to random edge pixels on the i th row.

Setting $a_i(\theta_k) = \exp(j\mu(i - 1)\tan(\theta_k))$, $s_k = \exp(-j\mu x_{0k})$, Eq. (3.14) becomes :

$$z(i) = \sum_{k=1}^d a_i(\theta_k) s_k + n(i). \quad (3.15)$$

From the data vector $\mathbf{z} = [z(1), \dots, z(N)]^T$ we build K vectors $\mathbf{z}_l = [z(l), \dots, z(M+l-1)]^T$, $l = 1, \dots, K$, of length M with $d < M \leq N - d + 1$. We define matrix $\mathbf{A}_M(\theta)$ as :

$$\mathbf{A}_M(\theta) = [\mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_d)] \quad (3.16)$$

where $\mathbf{a}(\theta_k) = [1, \zeta_k, \dots, \zeta_k^{M-1}]^T$, with $\zeta_k = \exp(j\mu \tan(\theta_k))$. With Propagator method [90] (see subsection 1.3.3), we estimate the orientations $\{\theta_k\}$ of the straight lines. The angle

values are such that they lead to the d strongest local maxima of function f defined as : $f(\theta) = (|\mathbf{Q}^H \mathbf{a}(\theta)|^2)^{-1}$ over the interval J_θ , defined by : $J_\theta =]-\tan^{-1}(\pi/\mu), \tan^{-1}(\pi/\mu)[$. Propagator yields an estimate of the straight line orientations, by partitioning the covariance matrix, and without performing its eigendecomposition, contrary to more commonly used methods such as MUSIC or TLS-ESPRIT. Another method that can be adapted to the estimation of straight line orientations is Propagator method, using directly the generated signals (refer to subsection 1.3.4).

3.3.3 Offset estimation

The aim of this part is to present two methods that lead to the estimation of the offsets of the straight lines, when their angle is known. The first one is the well-known "Extension of the Hough Transform" [117]. It is based on the projection of the image along the straight line angle values. The second proposed method remains in the frame of array processing : it employs a variable parameter propagation scheme [6, 7, 3] and uses a high-resolution method. This high-resolution "MFBBLP" method relies on the concept of forward and backward organisation of the data [55, 109, 138].

3.3.3.1 Extension of the Hough Transform

We consider the polar parametrization. We call "sinogram" the representation of the values taken by the Hough Transform for all considered values of polar coordinates θ and ρ . For a fixed θ value, the sinogram depends only on the ρ variable. The two polar coordinates can define a straight line. The distance $\{\rho_k\}$ between the origin and the straight line indexed by k is estimated by projecting the image along the orientation of polar coordinate θ_k and by retrieving :

$$\rho_k = \operatorname{argmax}_{-\sqrt{2}N \leq \rho \leq \sqrt{2}N} \sum_{i=1}^{i=N_p} c(\rho - x_i \cos \theta_k - y_i \sin \theta_k) \quad (3.17)$$

$$k = 1, \dots, d$$

where N is the size of the image, N_p is the number of useful pixels having components (x_i, y_i) , contained in the image and c is the real function defined for a given variable value r and a width parameter R by :

$$c(r) = \begin{cases} \cos\left(\frac{\pi}{2} \frac{r}{R}\right) & \text{if } |r| < R \\ 0 & \text{otherwise} \end{cases} \quad (3.18)$$

The offsets are obtained by the relation

$$\rho_k = x_{0_k} \cos \theta_k \quad (3.19)$$

This method has a good behavior in the presence of noise. In practice we will take $R = 3$ pixels. This parameter can be reduced in order to improve the estimation of the offsets [117]. The drawback of the method is its numerical cost. When the number of non zero pixels in the image is large, the summation in equation (3.17) contains a large number of terms. A major property of the Extension of the Hough Transform is that the case of several straight lines for a given angle value can be treated if several local maxima of the sinogram are selected. These maxima are obtained for values of ρ_k which are proportional to all offset values for a given orientation.

3.3.3.2 Proposed method : MFBLP

The variable speed propagation scheme method [6, 7] enables the estimation of the offsets with a lower computational load than the Extension of the Hough Transform. We associate to this specific signal generation scheme a high-resolution method called "MFBLP" (Modified Forward Backward Linear Prediction). In a previous work (see [55]), the concept of using forward-backward averaging led to effective results when it was applied to the SLIDE algorithm.

The basic idea in this method is to associate a propagation speed which is different for each line in the image. By setting artificially a propagation speed that linearly depends on the index of the lines in the matrix, we will be able to apply a frequency retrieval method to compute the offset values. When the first orientation value is considered, the signal received on sensor i ($i = 1, \dots, N$) is then :

$$z(i) = \sum_{k=1}^{d_1} \exp(-j\tau x_{0k}) \exp(j\tau(i-1)\tan(\theta_1)) + n(i) \quad (3.20)$$

d_1 is the number of lines with angle θ_1 . When τ varies linearly as a function of the line index the measure vector \mathbf{z} contains a modulated frequency term. Indeed we set $\tau = \alpha(i-1)$.

$$\begin{aligned} z(i) &= \\ &\sum_{k=1}^{d_1} \exp(-j\alpha(i-1)x_{0k}) \exp(j\alpha(i-1)^2\tan(\theta_1)) + n(i) \end{aligned} \quad (3.21)$$

This is a sum of d_1 signals that have a common quadratic phase term but different linear phase terms. The first treatment consists in obtaining an expression containing only linear terms. This goal is reached by dividing $z(i)$ by the non zero term $a_i(\theta_1) = \exp(j\alpha(i-1)^2\tan(\theta_1))$. We obtain then :

$$\begin{aligned} w(i) &= \sum_{k=1}^{d_1} \exp(-j\alpha(i-1)x_{0k}) + n'(i), \\ i &= 1, \dots, N. \end{aligned} \quad (3.22)$$

The resulting signal appears as a combination of d_1 sinusoids with frequencies :

$$f_k = \frac{\alpha x_{0k}}{2\pi}, \quad k = 1, \dots, d_1. \quad (3.23)$$

Consequently, the estimation of the offsets can be transposed to a frequency estimation problem. Estimation of frequencies from sources having the same amplitude was considered in [124]. In the following a high-resolution algorithm, initially introduced in spectral analysis [124], is proposed for the estimation of the offsets.

After adopting our signal model we adapt it to the spectral analysis method called modified forward backward linear prediction (MFBLP) [124] for estimating the offsets :

We consider d_k straight lines with given angle θ_k , and apply the MFBLP method. We consider d_k straight lines with given angle θ_k , and apply the MFBLP method, to the vector \mathbf{w} . For a convenient representation the components of \mathbf{w} will be written $[w_1, w_2, \dots, w_N]$.

1) For a N -data vector \mathbf{w} , form the matrix \mathbf{Q} of size $2 \cdot (N - L) \times L$ -where the subscript "*" indicates conjugate- :

$$\mathbf{Q} = \begin{bmatrix} w_L & w_{L-1} & \dots & w_1 \\ w_{L+1} & w_L & \dots & w_2 \\ w_{L+2} & w_{L+1} & \dots & w_3 \\ \vdots & \vdots & \ddots & \vdots \\ w_{N-1} & w_{N-2} & \dots & w_{N-L} \\ w_2^* & w_3^* & \dots & w_{L+1}^* \\ w_3^* & w_4^* & \dots & w_{L+2}^* \\ w_4^* & w_5^* & \dots & w_{L+3}^* \\ \vdots & \vdots & \ddots & \vdots \\ w_{N-L+1}^* & w_{N-L+2}^* & \dots & w_N^* \end{bmatrix}$$

Build the length $2 \cdot (N - L)$ vector :

$$\mathbf{h} = [w_{L+1}, w_{L+2}, \dots, w_N, w_1^*, w_2^*, \dots, w_{N-L}^*]^T$$

L is such that :

$$d_k \leq L \leq N - \frac{d_k}{2}$$

2) Calculate the singular value decomposition of \mathbf{Q} :

$$\mathbf{Q} = \mathbf{U}\Lambda\mathbf{V}^H$$

3) Form the matrix Σ by setting to 0 the $L - d_k$ smallest singular values contained in Λ .

$$\Sigma = \text{diag} \{ \lambda_1, \lambda_2, \dots, \lambda_{d_k}, 0, \dots, 0, 0, 0 \}$$

4) Form the vector \mathbf{g} from the following matrix computation :

$$\mathbf{g} = [g_1, g_2, \dots, g_L]^T = -\mathbf{V} * \Sigma' * \mathbf{U}^H \mathbf{h}$$

where Σ' is the pseudo-inverse of Σ .

5) Determine the roots of the polynomial function H , where

$$H(z) = 1 + g_1 z^{-1} + g_2 z^{-2} + \dots + g_L z^{-L}$$

6) d_k zeros of H are located on the unit circle. The complex arguments of these zeros are the frequency values ; according to equation (3.23) these frequency values are proportional to the offsets, the proportionality coefficient being $-\alpha$. The main advantage of this method comes from its low complexity. We will show further with some experiments on noisy images that the complexity of the variable parameter propagation scheme associated with MFBLP is much less than the complexity of the Extension of the Hough Transform as soon as the number of non zero pixels in the image is increased. This algorithm enables the characterization of straight lines with same angle and different offset.

3.4 Straight line estimation : results obtained

This section falls into three subsections dedicated to the efficiency of the presented straight line estimation methods. The first subsection concerns the estimation of rectilinear curves by SLIDE method and MFBLP. The second subsection emphasizes the ability of MFBLP to resolve close-valued offsets. The third subsection compares MFBLP and the extension of the Hough transform in terms of computational load.

3.4.1 Straight line estimation by SLIDE method, and MFBLP

As a first example, we propose an application of our method in the case of robotic vision. Fig. 3.2(a) is a photography taken by a camera and transmitted to the automatic command of a vehicle moving on the railway. This vehicle is used in particular for servicing of railways, *i.e.* for the replacement of the parallel crosspieces. The vehicle, when moving along the railway, determines first the position of the rails from the obtained picture. Then, the position of the nearest crosspiece is detected. It places itself over the detected crosspiece and the replacement of this one is performed by an auxiliary engine. The iterative replacement of the crosspieces is realized step by step. First, the position of the rails is determined. The array processing methods of "SLIDE" and variable propagation scheme associated with MFBLP are employed. The result of this determination is presented on Fig. 3.2 (b) Referring to the retrieved position of the rails, the vehicle decides about the correction to give to its progression. Once the rails are retrieved, the image is processed once again, the localization of the first crosspiece is performed and presented on Fig. 3.2(c) The crosspiece can be detected by changing the position of the antenna (this technique is described in [3]). The process is repeated and the crosspieces are retrieved iteratively. For this grey level image, the computational time which is required to retrieve the two rails by means of SLIDE algorithm, when MFBLP method is associated with the variable speed generation scheme, is the following : the estimation of the two orientation values, including signal generation and running TLS-ESPRIT method, needs 0.063 sec. The estimation of the offsets needs 1.1 sec. As a comparison, the Extension of the Hough Transform, employed with the *a priori* knowledge of the angles, needs 47 sec. to find the offsets of the two lines that fit the rails.

We exemplify now the estimation of convex object contours with polygonal geometry, as a second application case. Our method is employed to retrieve the characteristics of a polygon, namely the number of sides and the coordinates of the summits. For this purpose, the straight lines that fit the sides of the polygon are determined by the proposed method. The number and the parameters (angles, offsets) of the straight lines allow respectively to estimate the number of sides and the summits. The summits are fitted by considering the polygon as the smallest convex of the image, corresponding to the common intersection of the half-plans associated respectively with the support straight lines. Fig. 3.3(a) presents the case of a polygon included in a noisy image. This image contains 15% of randomly distributed noisy pixels. The straight lines that fit the sides are given on Fig. 3.3(b) and are determined in spite of the presence of noise in the image. For an image with more noisy pixels, a bias on the values of angle and offset can appear. It is difficult to obtain a valuable result from images with more than 20% of randomly distributed noisy pixels.

Fig. 3.4 presents the result obtained on an image containing a set of roughly aligned points. Like images containing dashed lines, this kind of images leads to generated signals that are

not continuous. Nevertheless the employed method manages to retrieve the main direction of the points of the image [24]. The image on Fig. 3.4(a) contains a set of points. Fig. 3.4(b) shows the result given by our line detection algorithm ; on Fig. 3.4(c) the superposition of the initial image and the result obtained shows that the overall orientation of these points is efficiently retrieved by the proposed method.

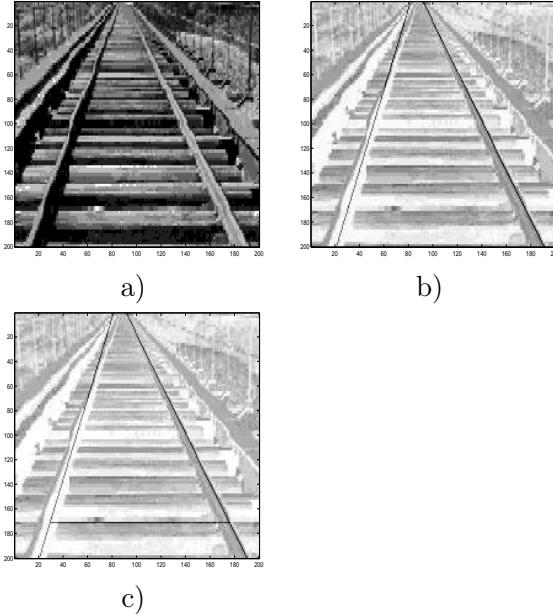


Figure 3.2 — (a) Image transmitted to the automatic command of a vehicle that is moving on a railway for the servicing of the railways. (b) Detection of the rails for the progress of the vehicle. (c) Localization of the first crosspiece that the vehicle has to replace. The process is iterated crosspiece after crosspiece : photography, detection of the rails and detection of the next crosspiece.

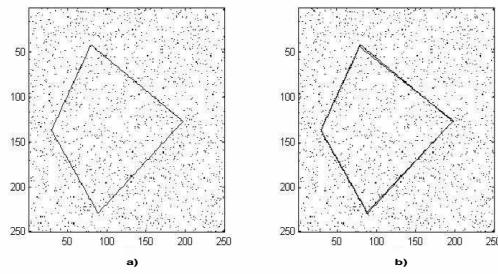


Figure 3.3 — (a) Noisy image containing a convex polygon the summits of which we aim at detecting. (b) Superposition of the segments that fit the sides of the polygon, estimated by our method, and the original image.

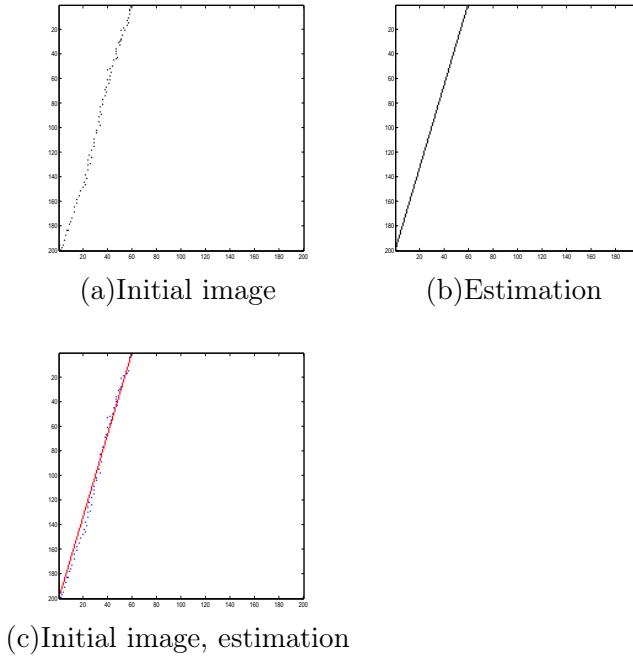


Figure 3.4 — The main direction of a set of points

3.4.2 Close-valued offset estimation

In this subsection we illustrate the abilities of MFBLP to distinguish between two close offset values. Processed image is provided in Fig. 3.5(a), result obtained by MFBLP is provided in Fig. 3.5(b), and result obtained by Hough transform is provided in Fig. 3.5(c).

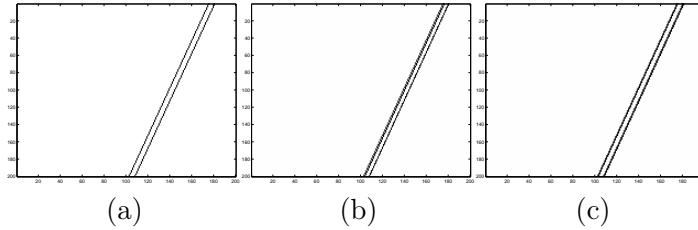


Figure 3.5 — (a) Two parallel curves : orientation estimation by Propagator and (b) Offset estimation by MFBLP, (c) Offset estimation by Hough transform.

The expected orientation value is 20° , the value estimated by propagator is 20.0° . The expected offset values are 175 and 180 pixels ; the values estimated by MFBLP are 175.9 and 179.8 ; the values estimated by Hough transform are 175.4 and 180.9. Note that both methods manage to retrieve the close offset values, with less than a one pixel bias.

For the estimation of parameter θ computational times are, excluding the time required for signal generation, as follows : the proposed method based on Propagator applied to the signal (see subsection 3.3.2) lasts 5.4×10^{-4} sec. Propagator method based on the covariance matrix

(see subsection 1.3.4) processing lasts 1.02×10^{-3} sec. A method that relies on the eigen-decomposition of a covariance matrix, such as TLS-ESPRIT method used in SLIDE algorithm (see subsection 3.3.1) lasts 1.08×10^{-3} sec. [5]. Propagator applied to the signal is then two times faster than the two compared methods, and Propagator method based on the covariance matrix is slightly faster than TLS-ESPRIT method.

We now consider a similar but more realistic case : an image with two nearly parallel distorted curves (see Fig. 3.6(a)). Constant speed signal generation scheme associated with a high-resolution retrieves the common overall orientation 20° of both curves and MFBLP yields the two close offset values, that is, 181 and 166 pixels (see Fig. 3.6(b)).

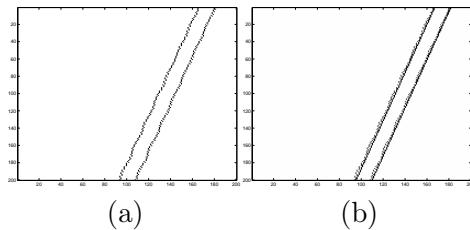


Figure 3.6 — (a) Two nearly parallel curves. (b) Orientation estimation by Propagator, offset estimation by MFBLP.

3.4.3 Computational complexity of the proposed method for offset estimation and Hough transform

In order to check the accuracy of our theoretical complexity determination, we measured the computational time for each method on images with two straight lines and different noise percentage, on the same computer. For images with two straight lines, 5, 8, 10, 20 and 25 percent of noisy pixels, we computed the ratio between the experimental values of the complexities of the extension of the Hough transform and the high-resolution based methods. Table 3.1 gives for these noisy images and the real-world grey level image of Fig. 3.2 the computational time of both methods, the experimental computational time ratio, the number N_p of edge pixels, the theoretical complexity ratio. We can draw the following conclusions on the complexities of both methods : the complexity of the high-resolution based method does not depend on the number of edge pixels and is more than ten times lower than the complexity of the extension of the Hough transform as long as the image contains more than 10% of edge pixels. When photographs (real grey level images) are treated, the computational time ratio between the two methods can be larger than 40 as for the case of Fig. 3.2, choosing our programming methods. The small difference between theoretical and experimental complexity values can be due to the approximation made when the eigen-decomposition was considered as dominating in the function "roots" of the MFBLP method. For the three highly noisy images, during the application of the extension of the Hough transform, the presence of uniformly distributed pixels provokes the appearance of a number of relative maxima which is larger than for the other images. This leads to a longer computational time for the sorting operation.

noise level	5%	8%	10%	20%	25%	grey level image
high-resolution : t_{HR} (s)	1.1	1.1	1.1	1.1	1.1	1.1
Hough Transform : t_{HT} (s)	6.1	10	15.6	29	40	47
Experimental ratio : $\frac{t_{HT}}{t_{HR}}$	5.5	9.1	14.2	26.4	36.4	42.7
Np	2099	3217	4167	7846	9947	13126
Theoretical ratio	6.9	10.5	13.7	25.7	32.5	42.8

Tableau 3.1 — Experimental and theoretical data concerning the complexity of the high-resolution based method and the extension of the Hough transform method.

Knowing that the complexity of the orientation estimation step is lower than the complexity of the offset estimation step, if we make the reasonable assumption that the microprocessors speed will rapidly increase, our high-resolution based set of methods could be employed for a real-time industrial image processing application.

3.4.4 Balance on straight contour retrieval

We proposed two methods for the estimation of orientations of straight lines in an image. Propagator applied to the generated signal is the fastest, and Propagator applied to the covariance matrix is still faster than TLS-ESPRIT method. Moreover, as we will show in the remainder of the chapter, Propagator applied to the covariance matrix can be extended to the estimation of distorted contours. In the remainder of the chapter, we use the methods devoted to straight line retrieval as an initialization for a region-based segmentation method, and for the estimation of any distorted contour.

3.5 Region-based contour retrieval

In this section we consider an application of straight line estimation methods to region-based segmentation by automatic threshold. Many studies have been devoted to pectoral muscle segmentation. Karssemeijer [67] and Ferrari [44] use the Hough transform, Kwok *et al.* [79] use an adaptive scheme based on "cliff detection" to delineate the pectoral muscle more accurately. Almost all these methods approximate the pectoral edge as a straight line and refine the straight line into a more accurate curve. This two-step process is also used in the approach presented in this section but the algorithms used are fundamentally different from existing methods.

The level set method, introduced by Osher, Fedkiw, and Sethian [102, 103] describes a curve in two spatial dimensions or a surface in three spatial dimensions by the zero-contour of a higher dimensional function, called the level set function.

In this section, we present a novel region based segmentation method, which relies on array processing methods. We extend the method presented above for straight contour retrieval to region segmentation in grey level images.

3.5.1 Proposed method

In this subsection, we propose a method for an automatic threshold of the images where an expected approximately linear distorted contour is present. The automatic method that we set is based on the comparison between the image where a threshold was applied and an image that contains only a straight line. The straight line fits an approximately linear contour and is retrieved by SLIDE method.

We propose to compare the signal generated from the binary image resulting from the threshold with the signal generated from the image that contains only the initialization straight line. We predict that this method will permit to obtain a binary image with a region of the input image which has nearly linear orientation.

Let \mathbf{z}_{input_t} be the signal generated from the threshold image by the following scheme :

$$z_{input_t}(l) = \sum_{m=1}^C I_t(l, m) \exp(-j\mu m), \quad l = 1, \dots, N \quad (3.24)$$

where j stands for $\sqrt{-1}$, I_t is the binary image resulting from the threshold operation, performed with the threshold value t , and μ is a propagation constant. The rows of the image are indexed by l , the columns of the image are indexed by m . N is the number of rows in the image, C is the number of columns in the image.

Let I_{init} be the image that contains only the initialization straight line provided by SLIDE algorithm. The components of the signal \mathbf{z}_{init} generated from the image that contains only the initialization straight line are obtained through the following computation :

$$z_{init}(l) = \sum_{m=1}^C I_{init}(l, m) \exp(-j\mu m), \quad l = 1, \dots, N \quad (3.25)$$

Let $\mathbf{X}_{init} = [X_{init}(1), \dots, X_{init}(N)]$ be the vector containing the coordinates of the pixels of the initialization straight line. If we take into account, for each value of l in Eq. (3.25), the relation $I_{init}(l, m) = 1$ only if $s = X_{init}(l)$, and $I_{init}(l, m) = 0$ otherwise, we get that the signal vector obtained from the image containing only the initialization straight line is defined by :

$$\mathbf{z}_{init} = [e^{-j\mu X_{init}(1)}, \dots, e^{-j\mu X_{init}(N)}] \quad (3.26)$$

Let $sett = \{t_k, k = 1, \dots, T\}$ be a set of threshold values. Values in $sett$ are all values between 1 and 254 for an image encoded with 8 bits. We set the following recursive procedure : At step k of the recursive procedure, a signal $\mathbf{z}_{input_{t_k}}$ is generated out of the image on which threshold t_k was applied, and we aim at minimizing the criterion J defined, for all threshold values t_k ($k = 1, \dots, T$) by :

$$J(t_k) = \left\| \mathbf{z}_{input_{t_k}} - \mathbf{z}_{init} \right\|, \quad (3.27)$$

where $\|\cdot\|$ represents the L_2 norm computed from the usual scalar product of \mathbb{C}^N . Minimizing the criterion of Eq. (3.27) is performed by means of the following algorithm :

Input : processed image encoded with 8 bits.
 $sett = \{1, 2, 3, \dots, 254\}$.
 Compute :

$$argmint = \underset{1 \leq k \leq T=254}{\operatorname{argmin}} (J(t_k)),$$

 Apply threshold $sett(argmint)$.
 Output : threshold image

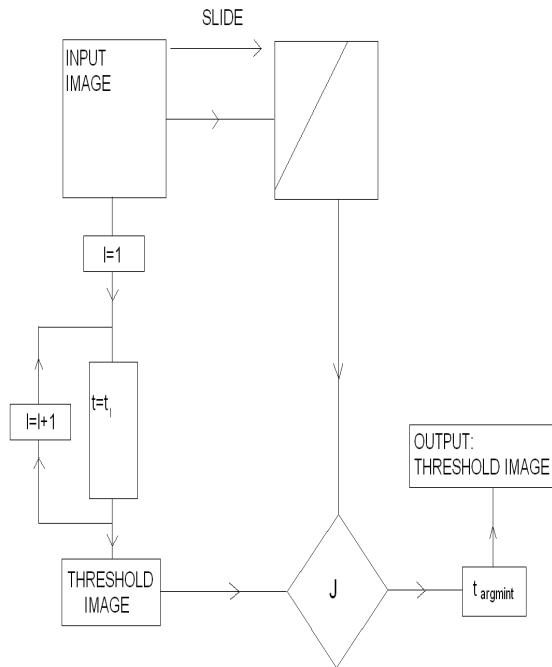


Figure 3.7 — Proposed method : region segmentation by global threshold.

Fig. 3.7 is an organization chart that presents the principles of our region segmentation by global threshold. The box at the top of the image, crossed by a straight line, represents the initialization image obtained directly from the input image. The box "J" in the center symbolises the computation of criterion J from the initialization image and the threshold image.

3.5.2 Application of the proposed method to pectoral muscle segmentation

Medical images are of crucial interest for surgeons to treat correctly their patients. An image processing algorithm can provide them a valuable help when they consider mammographic images : more precisely, some methods have been proposed to detect the boundary of the pectoral muscle. In this subsection, we adapt the proposed method to the detection of the boundary of the pectoral muscle in a mammographic image.

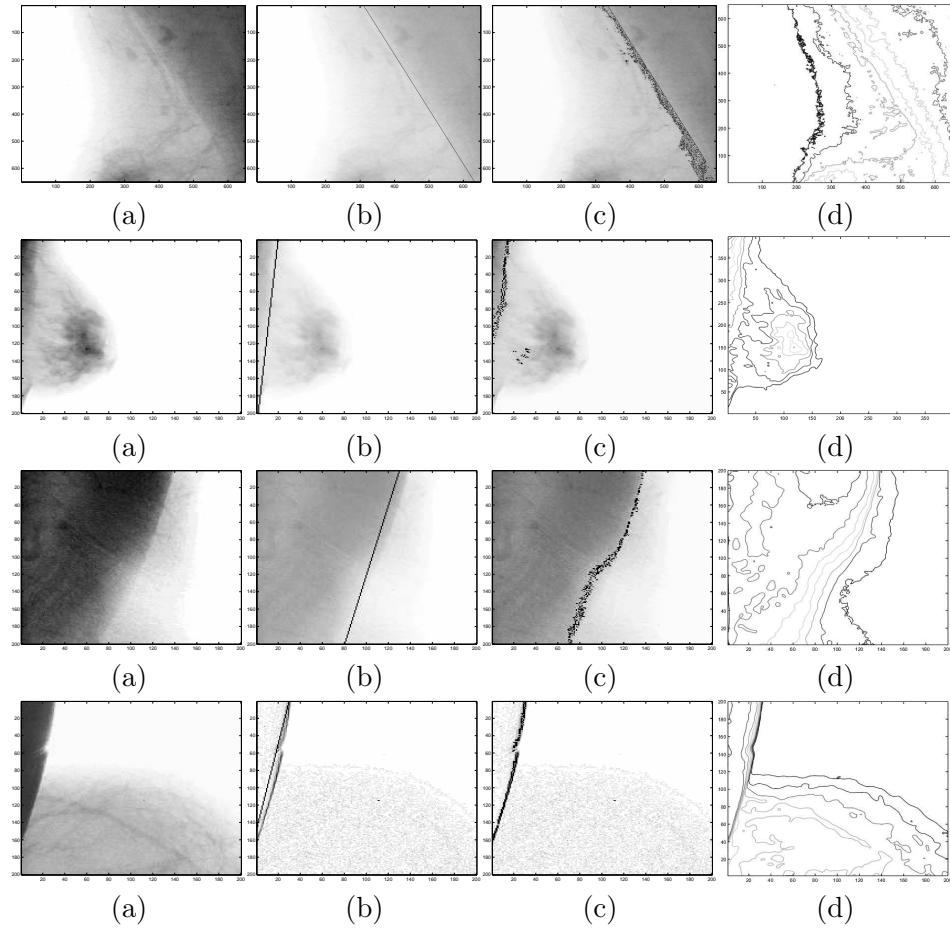


Figure 3.8 — Detection of the pectoral muscle : (a) Processed image (b) Superposition of the initialization image and the edge enhanced processed image (c) Superposition of the estimation by automatic threshold and the edge enhanced processed image (d) Result obtained with levelset method.

Fig. 3.8 presents the results obtained when our method is applied to mammographic images. The result obtained with our method is compared with the result obtained by a levelset method. Images (a) are the processed images. Images (b) present the initialization for our method, performed by SLIDE algorithm. Images (c) present the superposition of the estimation by automatic threshold and the processed image. Images (d) present the result obtained with the levelset method. For the last image (bottom of Fig. 3.8) a preprocessing, the gradient along the rows of the image, was applied to reduce the influence of the background. Images (b) show that the diffuse limit of the muscle is recovered. Our method manages to retrieve only this region of interest thanks to its ability to privilege diffuse boundaries which have an approximately linear shape. The levelset method does not take into account any prior information about the shape of the expected diffuse contour and then segments all regions in the image, whatever the shape of their boundary.

3.5.3 Balance on region-based segmentation

In this section, we propose a distorted contour retrieval method which is based on a threshold. The first step of this contour retrieval method consists in an initialization by means of a straight line. The second step consists in finding the binary image which results from a threshold and is the closest to the image that contains only the initialization straight line. Our criterion is based on a signal generated from the image containing the initialization straight line and a signal generated from the threshold image. The threshold image only contains a region which is selected from the processed image. This region is, in the original image, a smooth transition with overall straight orientation, between two homogeneous parts of the image.

In order to exemplify the efficiency of the proposed method, we consider an application to medical images : we have shown that our method is able to retrieve the limits of the pectoral muscle in a mammographic image. In the remainder of the chapter, we no longer consider region but contour retrieval.

3.6 Estimation of non rectilinear contours in an image as an inverse problem : principles of the proposed optimization method and gradient algorithm

In the previous sections we recalled a specific formalism for image representation, and presented an application of the high-resolution methods of array processing to the retrieval of straight lines in an image.

In this section, we keep the formalism retained for straight lines retrieval. The more general case of distorted contour estimation is proposed. As well as in the previous sections, this problem can be considered as an array processing problem in which a wave front has to be estimated. It is possible to make an analogy with a physical phenomenon that can be observed in wave physics. We suppose that the distorted curve actually contained in the image can be assimilated to a distorted wave front. Such a distorted front can be observed when the propagation medium is not isotropic. In order to estimate the wave front distortion, we propose to apply a recursive algorithm. In [23], a similar problem is solved, in the case of a plane wave received by a distorted antenna.

We propose a new method called ECAPMO (Estimation of Contours by Array Processing Methods and Optimization) for the estimation of continuous non rectilinear contours. It relies on the formulation of an inverse problem over the generated signals and the determination of phase fluctuations.

3.6.1 Retrieval of a general phase model

The ECAPMO method relies on the idea of a continuous phase model. We propose to extend the formalism proposed in [5], that sets the analogy between the phase model used in array processing and a contour in image processing. Instead of assuming that the phase model is known, *i.e.* that there exists a predefined model for the contour that we aim at retrieving, we create an artificial evolution of the wave front and of the corresponding received signal. By setting a recursive algorithm, we modify the phase of a current signal until it is equal to the

input signal generated out of the image. The proposed ECAPMO method leads to the phase parameters characterizing the distorted wave front.

In order to retrieve the characteristics of the wave front corresponding to the distorted curve, we can start from an initial signal corresponding to a plane wave front. We will modify recursively the components of a current signal until it becomes equal to the signal actually generated out of the image.

3.6.2 Initialization of the proposed algorithm

Our recursive optimization algorithm needs to be initialized. For this purpose, we choose as initialization parameters the phase values corresponding to a plane wavefront. Through the signal generation formalism that we adopted, this plane wavefront corresponds to a straight line in the image. Therefore, in order to initialize our recursive algorithm, we apply the SLIDE algorithm, which is supposed to return the straight line that fits the best the distorted contour which is present in the image. In this section we consider only the case where the estimated number d of curves is equal to one. The parameters angle and offset recovered by the straight line retrieval method are employed to build an initialization vector \mathbf{x}_0 , containing the position of the pixels of the initialization straight line :

$$\mathbf{x}_0 = [x_0, x_0 - \tan(\theta), \dots, x_0 - (N - 1) \tan(\theta)]^T$$

Fig. 3.9 presents a distorted curve, and an initialization straight line that fits this distorted curve.

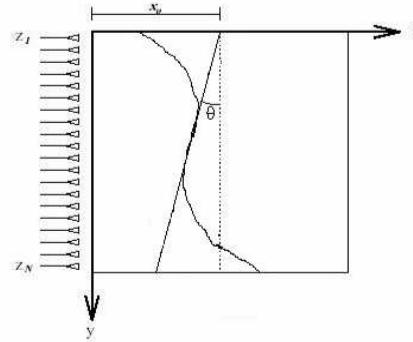


Figure 3.9 — A model for an image containing a distorted curve.

3.6.3 Distorted curve : proposed algorithm

We aim at determining the N unknowns $x(i)$, $i = 1, \dots, N$ of the image, forming a vector \mathbf{x}_{input} , each of them taken into account respectively at the i^{th} sensor :

$$z(i) = \exp(-j\mu x(i)), \quad \forall i = 1, \dots, N$$

The observation vector is

$$\mathbf{z}_{input} = [exp(j\varphi_1), \dots, exp(j\varphi_N)]^T \quad (3.28)$$

with $\varphi_i = -\mu x(i)$, representing the phase of sensor i . So we try to recreate the signal from which we ignore the N parameters. We start from the initialization vector \mathbf{x}_0 , characterizing a straight line that fits a locally rectilinear portion of the curve to be studied. Then, with k indexing the steps of this recursive algorithm, we aim at minimizing

$$J(\mathbf{x}_k) = \|\mathbf{z}_{input} - \mathbf{z}_{estimated \ for \ \mathbf{x}_k}\|^2 \quad (3.29)$$

where $\|\cdot\|$ represents the norm induced by the usual scalar product of \mathbb{C}^N . For this purpose we use gradient methods with fixed step type. The vectors of the series are obtained by the relation

$$\forall k \in \mathbb{N} : \mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \nabla(J(\mathbf{x}_k)) \quad (3.30)$$

where $0 < \lambda < 1$ is the step for the descent. The recurrence loop is

$$\mathbf{x}_k \rightarrow \mathbf{z}_{estimated \ for \ \mathbf{x}_k} \rightarrow J(\mathbf{x}_k)$$

The gradient is estimated using finite differences. We stop when the gradient becomes lower than a threshold.

At this point, by minimizing the function J , we find the components of vector \mathbf{x} leading to the signal \mathbf{z} which is the closest to the input signal. Nevertheless, by employing the criterion of minimum square error between signals, a phase indetermination over the input signal is remaining. Therefore, we propose an algorithm that aims at canceling the phase indetermination induced by the criterion that we chose. This algorithm is based upon the continuity of the phase of the signal which is actually generated out of the image. Moreover we prove that the series $(\mathbf{x}_k)_{k \in \mathbb{N}}$ actually converges towards a local minimum argument of the criterion J . By defining as \hat{x}_l the components of $\hat{\mathbf{x}}$ and starting from the following relation

$$\begin{aligned} \forall p \in \mathbb{Z}, \ exp(j\varphi_l) &= exp(-j\mu x_l) = exp(-j\mu \hat{x}_l) \\ &= exp(-j\mu(\hat{x}_l + \frac{2p\pi}{\mu})), \end{aligned}$$

we deduce that there exists an N -tuple of relative integers denoted by p_l such that $\mathbf{x}_{input} = \hat{\mathbf{x}} + \frac{2\pi}{\mu}[p_1, p_2, \dots, p_N]^T$.

This relation is equivalent to a shift which is proportional to $\frac{2\pi}{\mu}$ between x_l and \hat{x}_l for each line l of the image, or to a phase delay appearing on the signals obtained on the lines of the image-matrix.

The general formulation of the N -components vectors which minimize the cost function J are defined by

$$\begin{aligned} \text{Argmin } J &= \hat{\mathbf{x}} + \frac{2\pi}{\mu} \mathbb{Z}^N = \\ &\left(\hat{x}_1 + \frac{2\pi p_1}{\mu}, \dots, \hat{x}_N + \frac{2\pi p_N}{\mu} \right), \\ &(p_1, \dots, p_N) \in \mathbb{Z}^N \end{aligned}$$

The choice of the descent method towards such a minimum $\hat{\mathbf{x}}$ is such that

$$\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{y}} \{|\mathbf{y} - \mathbf{x}_0|\}$$

where $|\cdot|$ symbolises here the norm induced by the scalar product in \mathbb{R}^N , and $\mathbf{y} \in \operatorname{Argmin} J$. This implies that the main characteristics of $\hat{\mathbf{x}}$ are

1. to minimize the criterion J ,
2. to guarantee that its distance to \mathbf{x}_0 , i.e. $|\hat{\mathbf{x}} - \mathbf{x}_0|$ is minimum with respect to the distances of \mathbf{x}_0 to the other solutions \mathbf{x} of J .

The next step concerns the determination of the actual values of the vector \mathbf{x}_{input} . The uniqueness of the correct N -uplet for the reconstruction of the distorted wave requires to determine at least one of the components x_l of \mathbf{x} . At this stage of the method the choice of an initialization by a convenient straight line and the interest of the work presented about the determination of the curves step in. The hypotheses of curve continuity is exploited for this purpose. A reconstruction method (going successively forward and backward over the lines) of the curve is proposed starting from a fixed point. Before going further, we choose as an arbitrary point i_{max} , the maximum value of the set $\{i = 1, \dots, N\}$ such that $\hat{x}_i = x_0 - (i - 1) \tan \theta$, determined from the data \mathbf{x}_0 and $\hat{\mathbf{x}}$. We obtain

$$p_{i_{max}} = 0 \text{ and } x_{i_{max}} = \hat{x}_{i_{max}}$$

We set $\delta(\mu) = \operatorname{Max} \{|\hat{x}_l - \hat{x}_{l-1}|, l = 2, \dots, N\}$. For the forward part over the remaining lines of the image-matrix, for each line $l = i_{max}, \dots, N$, we determine successively $m_l \in \mathbb{N}$ such that $|\hat{x}_l - \hat{x}_{l-1} - \frac{2\pi}{\mu} \operatorname{sign}(\hat{x}_l - \hat{x}_{l-1})m_l| < \delta(\mu)$ and we set $p_{l-1} = \operatorname{sign}(\hat{x}_l - \hat{x}_{l-1})m_l$. For the descent method over the lines we start anew for $l = i_{max} + 1, \dots, N - 1$ by increasing the index l for the two relations above.

3.6.4 Convergence of the gradient method

For $k \rightarrow +\infty$ the series \mathbf{x}_k converges towards a vector $\hat{\mathbf{x}}$ such that :

$$\mathbf{z}_{input} = \mathbf{z}_{\hat{\mathbf{x}}}$$

That is to say $\hat{\mathbf{x}}$ is the argument of a local minimum of J contained in the neighborhood (in the sense of topology) of \mathbf{x}_0 . Indeed, let us denote for all $k \in \mathbb{N}$, $R_k = \lambda \|\nabla J(\mathbf{x}_k)\|$. The order one Taylor series of J over the R_k radius ball centered on \mathbf{x}_k allows to write

$$\begin{aligned} \forall \omega \in \mathbb{R}^N, |\omega| \leq R_k : \\ J(\mathbf{x}_k + \omega) = J(\mathbf{x}_k) + \langle \nabla J(\mathbf{x}_k), \omega \rangle + |\omega| \varepsilon(\omega) \end{aligned}$$

Thus for $\omega = -\lambda \nabla J(\mathbf{x}_k)$, with λ small enough so that ω have negligible norm, we obtain

$$J(\underbrace{\mathbf{x}_k - \lambda \nabla J(\mathbf{x}_k)}_{\mathbf{x}_{k+1}}) = J(\mathbf{x}_k) - \lambda \|\nabla J(\mathbf{x}_k)\|^2 \leq J(\mathbf{x}_k).$$

That is to say,

$$\forall k \geq 0, J(\mathbf{x}_{k+1}) \leq J(\mathbf{x}_k).$$

The series $(\mathbf{x}_k)_{k \in \mathbb{N}}$ induces the decrease of J with

$$\lim_{k \rightarrow +\infty} \nabla J(\mathbf{x}_k) = \mathbf{0}$$

This proves the convergence of the proposed optimization algorithm.

3.6.5 Summary of the proposed algorithm

An outline of the proposed distorted contour estimation method is given as follows :

1. Derive artificial signals using Equation (3.43) ;
2. Apply SLIDE algorithm : estimate line angle and offset that fit the best the distorted contour (see subsections 2.2 and 2.3) ;
3. Initialize the ECAPMO method using the straight line parameters obtained after applying the straight line retrieval method ;
4. Estimate the fluctuations of the position of the pixels around the initialization straight line, by using the gradient algorithm ;
5. Solve the phase indetermination problem, by using the hypothesis of continuity of the curve.

3.6.6 Numerical complexity of the method

We previously defined N_p as the number of non zero pixels, and d as the number of straight lines. With given values of these parameters and of the image size parameter N , the order of magnitude of the complexity of the angle estimation method is $N_p + N \cdot (\sqrt{N} + d)$ [7, 117]. As concerns the algorithm of offset estimation, let us remind that L is a parameter chosen close to N , and d_k is the number of parallel lines with a given orientation index k . In practice L is the integer part of $(N - \frac{dk}{2})$.

For signal generation, $7 \cdot N_p$ operations are needed to obtain the signal \mathbf{z} of Equation (3.20). For each of the d orientations found through constant parameter propagation, the signal \mathbf{w} of Equation (3.22) is obtained from the signal \mathbf{z} with $4 + 3 \cdot N$ operations. For the MFBLP method, we consider the case when one offset is expected for each orientation value. We chose for the parameter L the value $N - 1$. The procedure "roots" employed at step 5) in order to find the zeros of the polynomial function H is based on an eigen-decomposition of an $L \times L$ matrix. This eigen-decomposition dominates the other operations realized by the MFBLP algorithm in terms of complexity. Thus the complexity of this dominant step is L^3 or equivalently $(N - 1)^3$.

Therefore, the order of magnitude of the computational complexity of the offset determination algorithm is $7 \cdot N_p + d \cdot (4 + 3 \cdot N + (N - 1)^3)$.

The complexity dominating part of our algorithm for curve distortion estimation is the iterative algorithm. Let "Niter" be the number of iterations, necessary for the convergence of the algorithm.

We count the number of operations, neglecting the time required by the additions, and including one storage operation for the current value of the estimated vector \mathbf{x}_k of a current iteration k . Computing and storing the vector \mathbf{x}_0 from the parameters angle and offset given by the initialisation step, requires $N + 1$ operations. For each iteration k , including one storage operation for each computed value of $\nabla J(\mathbf{x}_k)$, we obtain the following results :

1. The numerical derivative $\nabla J(\mathbf{x}_k)$ requires the computation of two values of the function J , computed for vector \mathbf{x}_k and an incremented version of \mathbf{x}_k , and one division by the incremental vector. Then, $10 \cdot N$ operations are needed for the computation of the function J . The subtraction of two successive values of the function J and a division by the size

N incremental vector needs N operations. So, $2 \cdot (10 \cdot N) + N + 1$ operations are needed for the computation and storage of the derivative $\nabla J(\mathbf{x}_k)$.

2. N operations are needed for the multiplication of $\nabla J(\mathbf{x}_k)$ by λ , so the computation of \mathbf{x}_{k+1} from Equation (3.30) and the storage of \mathbf{x}_{k+1} needs $N + 1$ operations.

So $N + 1$ operations are needed for the computation of \mathbf{x}_0 , and $22 \cdot N + 2$ operations for the computations of all iterations. In total, $N + 1 + N_{iter} \cdot (22 \cdot N + 2)$ operations are needed. Some experimental results about the computational time required for the distorted curve retrieval method will be presented in the Section 5.

3.7 Generalization of distorted contour estimation to the estimation of several curves

In this section we consider the case where the estimated number of curves d is larger than one. We will suppose that each curve is composed of a single pixel per line in the image. Therefore the model for the input signal $\mathbf{z}_{input} = [z(1), z(2), \dots, z(N)]^T$ is :

$$\begin{aligned} \mathbf{z} &= \begin{bmatrix} z(1) \\ z(2) \\ \vdots \\ z(N) \end{bmatrix} = \sum_{k=1}^d \begin{bmatrix} \exp(j\varphi_{1k}) \\ \exp(j\varphi_{2k}) \\ \vdots \\ \exp(j\varphi_{Nk}) \end{bmatrix} + \begin{bmatrix} n(1) \\ n(2) \\ \vdots \\ n(N) \end{bmatrix} \\ &= \begin{bmatrix} \exp(j\varphi_{11}) & \dots & \exp(j\varphi_{1d}) \\ \exp(j\varphi_{21}) & \dots & \exp(j\varphi_{2d}) \\ \vdots & \ddots & \vdots \\ \exp(j\varphi_{N1}) & \dots & \exp(j\varphi_{Nd}) \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + \mathbf{n} \\ &= \mathbf{A}(\varphi) \mathbf{s} + \mathbf{n} \end{aligned}$$

where $\mathbf{A}(\varphi)$ is a matrix of size $N \times d$ taken as a model for the matrix of the directional vectors of the sources and \mathbf{s} is the vector of sources amplitudes, all equal to 1.

So the term $\varphi_{ik} \in]-\pi, \pi]$ represents the phase of the transfer function of the system source k and sensor i having an amplitude equal to 1. The source-vector related to source k is

$$\mathbf{a}(\varphi_k) = [\exp(j\varphi_{1k}), \exp(j\varphi_{2k}), \dots, \exp(j\varphi_{Nk})]^T \quad (3.31)$$

Referring to matrix notations, we obtain thus

$$\mathbf{A}(\varphi) = [\mathbf{a}(\varphi_1), \mathbf{a}(\varphi_2), \dots, \mathbf{a}(\varphi_d)] \quad (3.32)$$

So that we can define an application written \mathbf{z} , such that to every matrix variable

$$\varphi = [\varphi_1, \varphi_2, \dots, \varphi_d] = \begin{bmatrix} \varphi_{11} & \dots & \varphi_{1d} \\ \varphi_{21} & \dots & \varphi_{2d} \\ \vdots & \ddots & \vdots \\ \varphi_{N1} & \dots & \varphi_{Nd} \end{bmatrix} \quad (3.33)$$

we associate the vector $\mathbf{z}(\varphi)$ such that :

$$\mathbf{z}(\varphi) = \sum_{k=1}^d \begin{bmatrix} \exp(j\varphi_{1k}) \\ \exp(j\varphi_{2k}) \\ \vdots \\ \exp(j\varphi_{Nk}) \end{bmatrix} = \mathbf{A}(\varphi)\mathbf{s}. \quad (3.34)$$

To all $\mathbf{z}(\varphi)$, we associate a real positive value written $J \circ \mathbf{z}(\varphi)$. If we consider both J and \mathbf{z} as functions of a vector or matrix variable, \circ denotes composition between functions J and \mathbf{z} . The function which is obtained is, applied to variable φ , such that

$$J \circ \mathbf{z}(\varphi) = \|\mathbf{z}_{input} - \mathbf{z}(\varphi)\|^2 \quad (3.35)$$

From a numerical point of view we stack successively the columns φ_i of the matrix φ of size $N \times d$ in a vector ϕ of size $N \cdot d$ such that

$$\phi = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_d \end{bmatrix} \quad (3.36)$$

We set $\tilde{\mathbf{z}}(\phi) = \mathbf{z}(\varphi)$ and we replace the previous problem by

$$\text{Minimize } \tilde{J}(\phi) = \|\mathbf{z}_{input} - \tilde{\mathbf{z}}(\phi)\|^2 \quad (3.37)$$

We initialize ϕ taking as column-vectors φ_i the vectors of the d straight lines obtained by the method for the case of the rectilinear contours \mathbf{x}_{0k} , $k = 1, \dots, d$. We use afterwards the gradient methods in order to estimate a vector $\hat{\phi}$ minimizing \tilde{J} . In the case when $d = 1$, we find anew the work presented in section 3.6.

3.8 Simulations on non rectilinear contours

This part is dedicated to the method employed in order to retrieve distorted curves. Several examples of use of ECAPMO are presented. Fig. 3.10 presents a curve that we wish to determine. This distorted contour containing an almost straight section is a typical example of curve retrieved by the method ECAPMO. The different steps of the method are presented on Fig. 3.11.

On the example of Fig. 3.12, we chose a curve presenting some shift of the useful pixels of the curve at the beginning and at the end of its shape. The ECAPMO manages to return the shift values. On the example of Fig. 3.13 are presented the results obtained with an image containing a single curve. For this 200×200 image, with a 3.0 Ghz pentium processor, the initialization needs the following computational times : the estimation of the angle takes 0.047 sec., the estimation of the offset takes 0.66 sec. As a comparison, when the Extension of the Hough Transform [72] is employed for the estimation of both angle and offset of the initialization straight line, the computational time is 8.54 sec. The computational time required to run the iterative algorithm of the ECAPMO method is 0.80 sec., 1800 iterations were necessary while solving the inverse problem in order to obtain this result. Fig. 3.14 shows the results obtained

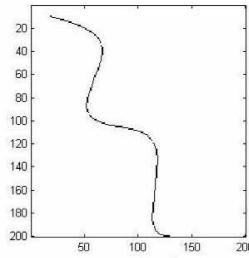


Figure 3.10 — An image the contour of which exemplifies the distorted curves our method can cope with.

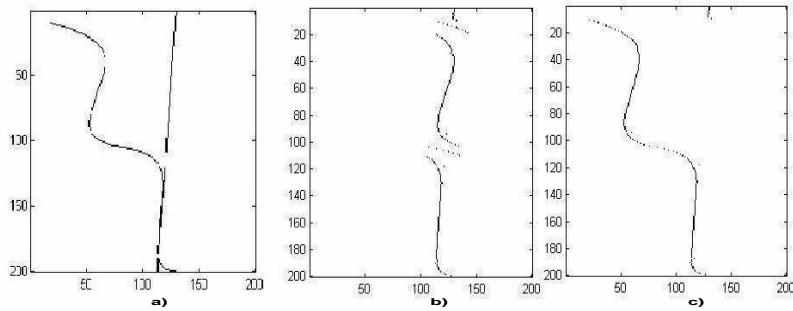


Figure 3.11 — (a)Initialization of the method (b)Determination of the vector \hat{x} (c)Junction of the different parts of the curve by determining the coefficients p_l

in the case of a noisy image. This image contains 10% of randomly distributed noisy pixels. The curve is still efficiently retrieved. On Fig. 3.15 appears a figure with two distorted curves. The specific method described in section 3.7 is employed in this case. It manages to retrieve the two curves. Fig. 3.16 shows that the method for distorted contour estimation copes with straight lines as well. The specific algorithm was applied in common to both distorted curve and straight line of the image. The slight bias on the offset value is canceled by the algorithm for distorted curves estimation.

Some practical situation was examined on Fig. 3.17. This image symbolises a vehicle and two road borders. The algorithm for the estimation of all contours is the following :

- The borders of the road are obtained through an initialization step : they are the two dominant directions in the image.
- Referring to the information obtained in step 1, the vehicle in the center is isolated. Then its contours are estimated.
- The algorithm dedicated to multiple-curve images is applied in order to estimate finely the borders of the road.

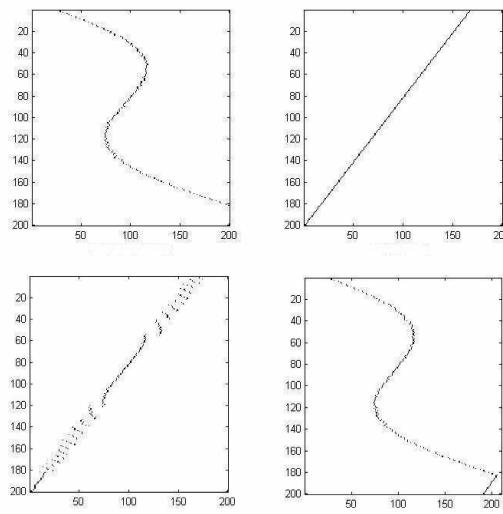


Figure 3.12 — An image and the steps of the method : Original image, initialization, estimation obtained before sticking sections together, final estimation. The method manages to return the shifts.

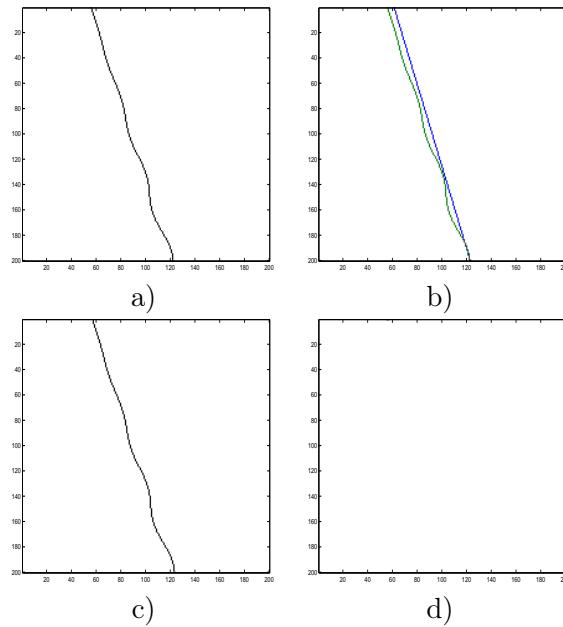


Figure 3.13 — The main results obtained by distorted contour estimation : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Difference between the initial image and the estimation

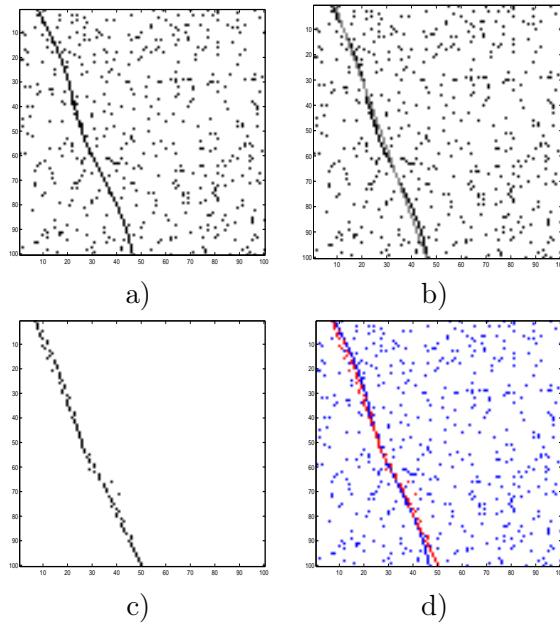


Figure 3.14 — Distorted contour estimation on a noisy image : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Superposition of the initial image and the estimation.

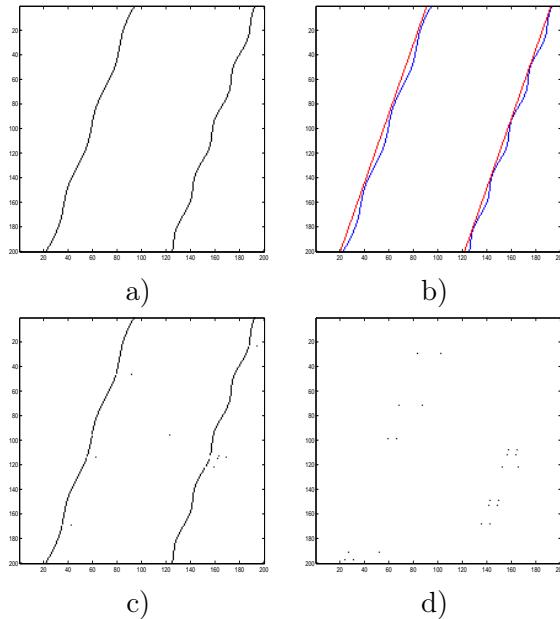


Figure 3.15 — The case of an image containing two distorted curves : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Difference between the initial image and the estimation

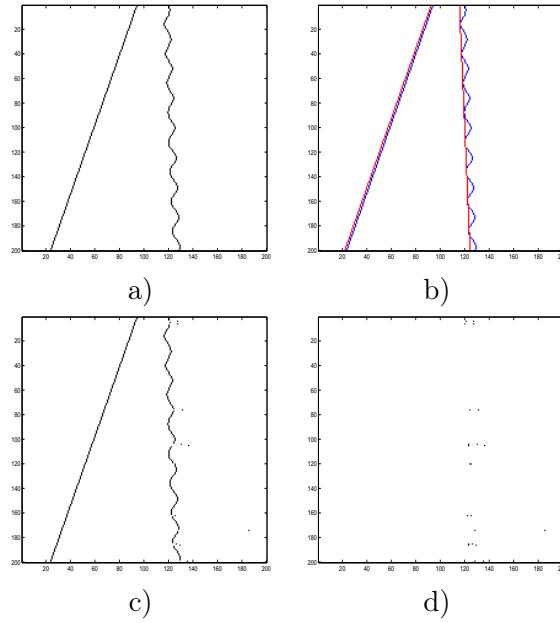


Figure 3.16 — Results obtained by the method for the estimation of several curves. One curve is a straight line : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Difference between the initial image and the estimation

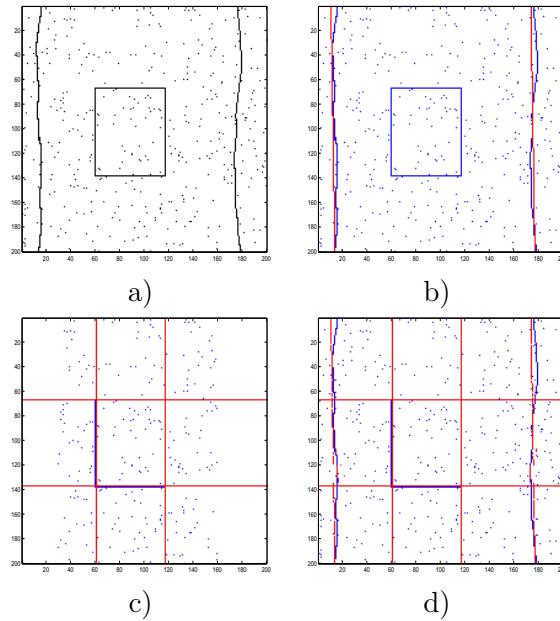


Figure 3.17 — A practical situation is simulated. A vehicle and the two sides of the road are retrieved : a)Initial image b)First estimation of the borders of the road c)Estimation of the sides of the vehicle d)Refined estimation of the borders of the road

3.9 Estimation of non rectilinear contours in an image by means of Propagator and optimization methods

3.9.1 Formulation of a phase model

The adopted approach for signal generation permits to obtain a general phase model when distorted contours are expected. Let us consider the generated signal \mathbf{z} . Each component of \mathbf{z} is as follows :

$$z(i) = \exp(-j\mu x(i)) = \exp(j\mu(i-1)\tan(\theta) - j\mu\Delta x(i))\exp(-j\mu x_0) \quad (3.38)$$

This expression contains, for one curve and the i th row of the image, the value $\Delta x(i)$ of the shift between the position of the pixel belonging to a straight line fitting the curve, and the pixel of the curve itself. Equation (3.38) is equivalent to : $z(i) = a_i(\theta)s$, where $a_i(\theta) = \exp(j\mu(i-1)\tan(\theta) - j\mu\Delta x(i))$ and $s = \exp(-j\mu x_0)$. It is possible to set together in a vector model the components $a_i(\theta)$ of all rows of the image. If several orientation values are considered, the vector model concerning the orientation k is : $\mathbf{a}(\theta_k) = [e^{-j\mu\Delta x(1)}, e^{j(\mu\tan(\theta_k)-\mu\Delta x(2))}, \dots, e^{j(\mu(N-1)\tan(\theta_k)-\mu\Delta x(N))}]^T$. The purpose of the next subsection is to estimate the values $\Delta x(1), \dots, \Delta x(N)$ of the pixel shifts.

3.9.2 Propagator method for phase shift estimation

Referring to Equation (1.32), matrix \mathbf{Q} has M lines and $M-d$ columns. Therefore, the vector that will be estimated will be of length M . We remind that the value of M can be chosen up to $M = N - d + 1 \leq N$. In practice, the images to be treated are not supposed to contain a large number of curves, so that M can be fixed to a value close to N . The technique that we use is the following : an initialization vector holding for the N rows of the image is computed. This initialization vector fits the distorted curve by a dominant straight line. Then, starting from this initialization vector, M phase values of the signals are computed. We proposed a method for this purpose in [28]. The last $N - M$ phases are supposed to differ from the $N - M$ phases of the initialization vector by a phase shift which is equal to the last computed phase shift. In the general case where several curves with parameters θ_k , $k = 1, \dots, d$ are present, we have to find the vector $\mathbf{a}(\theta_k)$, for $k = 1, \dots, d$, such that :

$$|\mathbf{Q}^H \mathbf{a}(\theta_k)|^2 = 0 \quad (3.39)$$

where $|.|$ denotes the L_2 norm. Let $\mathbf{a}(\theta_k)_o = [1, e^{j\mu\tan(\theta_k)}, \dots, e^{j\mu(N-1)\tan(\theta_k)}]^T$ be a vector obtained by using the initial estimate of the orientation of the straight lines fitting the k th contour. We use the conjugated gradient method, initialized by the vector $\mathbf{a}(\theta_k)_o$, to estimate $\mathbf{a}(\theta_k)$ by minimizing the criterion of Eq. (3.39). The sequence of vectors of the recurrence loop are obtained by the relation :

$$\forall q \in \mathbb{N} : \mathbf{a}(\theta_k)_{q+1} = \mathbf{a}(\theta_k)_q - 2\lambda \mathbf{Q} \mathbf{Q}^H \mathbf{a}(\theta_k)_q, \quad 0 \leq q \leq niter \quad (3.40)$$

where q indexes the elements of the sequence of the recurrence loop, $0 < \lambda < 1$ is the step size, $niter$ is the number of iterations. We stop the recursion when the criterion is under a fixed threshold.

From the complex argument of the components of vector $\mathbf{a}(\theta_k)$, we get the values of $\Delta x(i)$.

At this point, the values of θ_k for each curve indexed by k , each offset x_0 and $\Delta x(i)$ for $i = 1, \dots, M$, have been calculated. The last $N - M$ pixels are supposed to be straightly aligned along the orientation of the initialization straight line, from the M th pixel to the bottom of the image. Thus, the position of the pixels of each distorted curve are known at this point.

3.10 Propagator applied to the covariance matrix associated with gradient algorithm : Experimental results, computational times

3.10.1 Real-world images

This subsection is divided into two parts : one is devoted to straight line reconstruction, and the other concerns distorted contour retrieval.

When the procedure for straight lines retrieval is run, the values of parameters μ and α have to be chosen. As concerns parameter μ , [5] provides a study that gives the maximum value of an estimated orientation, with a value of μ equal to 1. This maximum value is 73° and is enough, considering that if the image is rotated by 90° , all orientation values present in the image can be computed. We applied such a rotation to the image of Fig. 2, in order to detect the crosspieces which are supposed to have an orientation of 90° . This is equivalent to place the antenna at the bottom side of the image. If μ is smaller, the maximum orientation value is higher, but it was empirically shown that the value 1 gives the best results. If μ is higher the maximum value of an estimated orientation is lower. That is why we chose to use value 1 for μ . As concerns α , it must be such that the value of α multiplied by the maximum offset value remains in an interval of length 2π . Indeed MFBLP method leads to the frequency value $-\alpha x_0$. This frequency value must be in the interval $[0, 2\pi[$ in order to avoid any phase indetermination. Therefore we can choose for instance the value 2.5×10^{-3} for an image containing 200 columns. As concerns parameter M , it can be chosen up to $N - d + 1$, where d is the number of expected contours. As the number of estimated phase shift values between the initialization straight line and the expected contour is equal to M , we decided to fix M to an elevated value, close to the number of rows in the image. The research step angle is 0.3° in interval J_θ , otherwise it is specified.

Gradient Vector Flow [137] is well-suited for a comparison with our method. Its popularity is due to its ability of attracting an active contour toward object boundary from a sufficiently large distance and its ability of moving the contour into object cavities. This enables an initialization by any contour -for instance a rectilinear one- whatever the curvature of the expected contour. GVF is based on a recursive optimization method. We may perform any number of iterations and thus control its computational load. In the following, we denote by ϵ the mean bias over the position of the pixels, computed over all pixels of the curve : $\epsilon = \frac{1}{N} \sum_{i=1}^N |\hat{x}(i) - x(i)|$, where $\hat{x}(i)$ is the estimation obtained for the position of the pixel of row i . Fig. 3.18(a) presents an aerial image containing a road. One side of the image has size $N = 470$. An edge enhancement and a threshold are applied (see Fig. 3.18(b)). When the proposed methods are applied, parameter M was chosen equal to 468 to maximize the number of pixel shift values which are actually estimated. One initialization straight line is obtained, which has the same overall orientation as the road. The angle value is -3.3° and the offset value

is 289 pixels (see Fig. 3.18(c)). After the initialization step for which $\mu = 1$, the propagator matrix is computed again with $\mu = 5 \times 10^{-3}$. This avoids any phase indetermination over the value of the pixel shifts. Our optimization method is run with $\lambda = 5 \times 10^{-4}$ and 400 iterations. Fig. 3.18(d) shows that the bias obtained with our method ($\epsilon = 0.2$) is due to some disruptions. When GVF is applied, it is initialized independently from our methods. As Fig. 3.18(e) shows, for this image the initialization contour must be close to the expected contour in order for GVF to converge. We perform 40 iterations for the computation of the edge image and 25 iterations for the deformation step. Parameter values are $\mu_{GVF} = 0.15$ (regularization parameter in the GVF formulation), $\alpha_{GVF} = 0.1$ (tension), $\beta_{GVF} = 0.001$ (rigidity). Fig. 3.18(f) shows that the mean pixel bias ($\epsilon = 0.6$) is due to a focalization on some noisy pixels. Fig. 3.19 presents the result obtained, with a photography of a river, with the same parameters except $N = 200$ and $M = 197$. Relation (3.39) holds independently for both orientations and corresponding pixel shifts. So our method detects the two borders of the river, whereas GVF cannot.

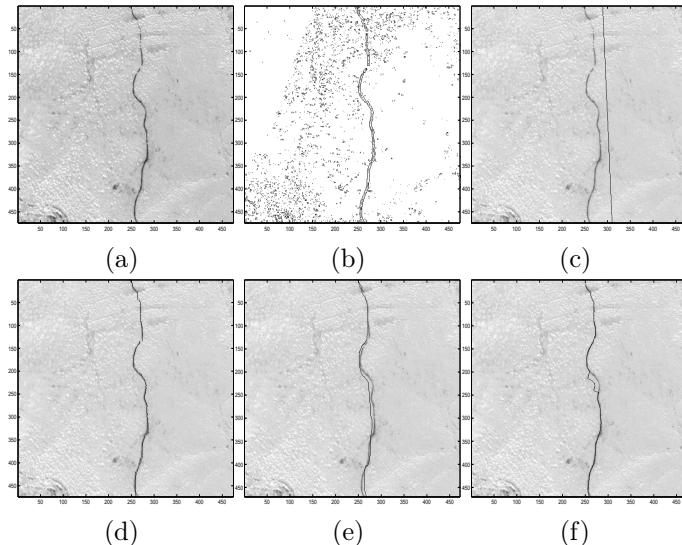


Figure 3.18 — Aerial image processing : (a) Initial image ; (b) Result of the edge detector. Proposed method : (c) Superposition of the initial image and the initialization straight line ; (d) Superposition of the initial image and the estimation. GVF method : (e) Initialization ; (f) Superposition of the initial image and the estimation.

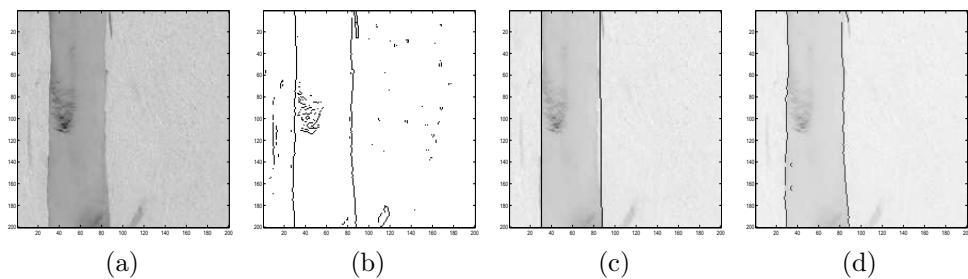


Figure 3.19 — Aerial image processing : (a) Initial image ; (b) Result of the edge detector. Proposed method : (c) Initialization ; (d) Superposition of the initial image and the estimation.

3.10.2 Statistical results

Here we first study the speeds of our methods for angle and offset estimation and of the extension of the Hough transform. In our experiments, we consider 200×200 images, containing one straight line, and impaired by an impulse noise : some percentage of the background pixels become edge pixels. We chose the noise precentage values : 0 ; 1 ; 2 ; 4 ; 10 ; and 15%. For all noise percentage values, angle estimation by Propagator method lasts 0.12 sec, and offset estimation by the proposed method based on the combination of a variable speed propagation scheme with MFBLP lasts 0.39 sec. The extension of the Hough transform lasts 0.40 ; 0.50 ; 1.2 ; 2.0 ; 5.6 ; and 9.6 sec. So our method for offset estimation is faster when noise percentage is larger than 1 %, which is generally the case for real-world images. The maximum ratio between computational times (24.6) is obtained with the highest noise percentage value. When Hough transform is used to estimate all angle and offset values, it lasts 8.6 ; 20.4 ; 30.7 ; 51.4 ; 105.5 ; and 152.0 sec. Running the set of proposed methods for both angle and offset estimation lasts 0.51 sec. Therefore computational time obtained by running Hough transform method is up to 300-fold higher.

Now, let us compare the robustness of our method and GVF to the amplitude of the distortions of a single curve. In order to simulate real-world conditions, the position of the edge pixels is given by the summation of two amplitude modulated sinusoids. We denote by *amp* and *per* the multiplicative factors that characterize the amplitude and period of the first sinusoid, which are five fold as high as the amplitude and period factors of the second sinusoid. The second sinusoid simulates a small amplitude high frequency perturbation. A unique straight line, which is obtained by our method for straight line retrieval, is used to initialize both methods. One hundred points regularly distributed along this straight line are chosen to initialize the Gradient Vector Flow. Parameters for GVF and for our initialization methods are the same as in subsection 3.10.1. We choose $M = 199$ in order to maximize the adequation between the processed data and the image. Our optimization method is run with $\lambda = 5 \times 10^{-4}$ and 500 iterations. The number of iterations for each method is chosen such that the computational time is the same for our method and for GVF. For all images, the proposed method for angle estimation lasts 0.11 sec., our method for offset estimation lasts 0.39 sec. For the retrieval of the pixel shifts GVF needs 24.3 sec. whereas our method needs 21.3 sec. The first criterion that is used to measure the accuracy of the results obtained is the mean value of the mean bias ϵ . For Tr trials, mean error ME is defined by : $ME = \frac{1}{Tr} \sum_{j=1}^{Tr} |\epsilon_j|$, where j indexes the trials and ϵ_j is the mean bias obtained at the j th trial. Standard deviation Std is defined by : $Std = \sqrt{\frac{1}{Tr} \sum_{j=1}^{Tr} (\epsilon_j - ME)^2}$.

We first illustrate in Fig. 3.20 the results obtained by both methods on one curve with amplitude parameter 3 and period parameter 1.5. The result images (see Figs. 3.20(c) and (d)) show that the mean pixel bias obtained with our method is lower.

The statistical results presented now are obtained with similar curves, having several couples (*amp*, *per*) of amplitude and period values given in Table 3.2. We perform $Tr = 1000$ trials. At each trial, amplitude and period factors are multiplied by a random number following a normal law with mean 1 and standard deviation 0.01. Statistical results for the proposed method and GVF method respectively, for each couple of amplitude and period factors are (in pixels) as presented in Table 1.

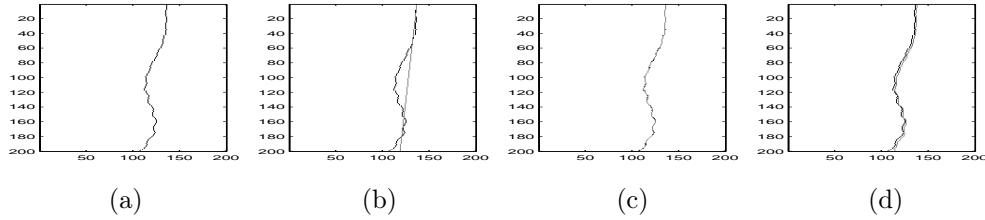


Figure 3.20 — Hand-made image, common initialization : (a) Example of hand-made image ; (b) Result of the initialization ; (c) Superposition initial image and result obtained by the proposed method ; (d) Superposition initial image and result obtained by GVF.

	(amp, per)	(0.5, 1.0)	(1, 1.1)	(1.5, 1.2)	(2, 1.3)	(2.5, 1.4)	(3, 1.5)
Propag.	<i>ME</i>	$4.59 \cdot 10^{-1}$	$4.70 \cdot 10^{-2}$	$4.85 \cdot 10^{-2}$	$4.98 \cdot 10^{-2}$	$5.06 \cdot 10^{-2}$	$5.11 \cdot 10^{-2}$
	<i>Std</i>	$1.90 \cdot 10^{-3}$	$2.12 \cdot 10^{-3}$	$8.42 \cdot 10^{-3}$	$3.23 \cdot 10^{-2}$	$5.24 \cdot 10^{-2}$	$8.07 \cdot 10^{-2}$
GVF	<i>ME</i>	1.56	2.09	2.65	3.66	4.17	4.70
	<i>Std</i>	$2.04 \cdot 10^{-3}$	$2.19 \cdot 10^{-3}$	$1.26 \cdot 10^{-2}$	$4.33 \cdot 10^{-2}$	$6.13 \cdot 10^{-2}$	$9.54 \cdot 10^{-2}$

Tableau 3.2 — *ME* and *Std* values (in pixel) obtained with the proposed method and GVF versus amplitude and period values.

ME values obtained with our method are less than 1. GVF method leads to *ME* values which are 3-fold as high as the error obtained with our method, for all amplitude factor values. The standard deviation *Std* values obtained with GVF are at least 1.2-fold as high as the values obtained with our method. This can be due to a dependence of GVF on its multiple parameters which is higher than the dependence of our method on its own parameters.

3.11 Distorted contour retrieval by the combination of gradient and spline interpolation

A method for distorted contour estimation has been proposed in [24]. This algorithm is based on array processing and optimization methods. DIRECT algorithm [64] is an optimization method which retrieves a function global minimum. It was employed in image processing but leads to an elevated computational load when the number of unknown values is elevated. In this chapter, the number of unknown values to be retrieved is equal to the number of rows in the image, thus rather elevated. Therefore, we use the variable step Gradient method. The principle of the proposed method is to minimize the squared difference between the signal generated out of the processed image, \mathbf{z}_{input} , and a model signal which is recursively modified. Gradient method is not a global optimization method, but we combine it with spline interpolation which yields a continuous contour.

3.11.1 Formulation of a signal model

The adopted signal generation method yields a general phase model when distorted contours are expected. Let us consider the generated signal \mathbf{z} . Each component of \mathbf{z} is as

follows :

$$\begin{aligned} z(i) &= \exp(-j\mu x(i)) = \\ &= \exp(j\mu(i-1)\tan(\theta) - j\mu\Delta x(i))\exp(-j\mu x_0) \end{aligned} \quad (3.41)$$

This expression contains, for one curve and the i th row of the image, the pixel shift value $\Delta x(i)$ between the pixel of the straight line fitting the curve, and the pixel of the curve itself. Eq. (3.41) is equivalent to : $z(i) = a_i(\theta)s$, where $a_i(\theta) = \exp(j\mu(i-1)\tan(\theta) - j\mu\Delta x(i))$ and $s = \exp(-j\mu x_0)$. It is possible to set together in a vector model the components $a_i(\theta)$ of all rows of the image. If several orientation values are considered, the components of the vector model $\mathbf{a}(\theta_k)$ concerning the orientation k are expressed as : $e^{-j\mu\Delta x(1)}, e^{j(\mu\tan(\theta_k)-\mu\Delta x(2))}, \dots, e^{j(\mu(N-1)\tan(\theta_k)-\mu\Delta x(N))}$. The purpose of the next subsection is to estimate the values $\Delta x(1), \dots, \Delta x(N)$ of the pixel shifts.

3.11.2 Optimization algorithm : combination of gradient and spline interpolation

We initialize the procedure with a signal vector corresponding to the initialization straight line [24]. We remind that $\mathbf{X}_0 = [X_0(1), \dots, X_0(N)]^T$ is the vector containing the coordinates of the pixels of the initialization straight line. Let $\mathbf{X}_{input} = [X_{input}(1), \dots, X_{input}(N)]^T$ be the vector containing the distorted curve pixel coordinates. The signal vector generated out of an image that would contain only the initialization straight line is defined by :

$$\mathbf{z}_0 = [e^{-j\mu X_0(1)}, \dots, e^{-j\mu X_0(N)}]^T \quad (3.42)$$

The signal vector generated out of the processed image is defined by :

$$\mathbf{z}_{input} = [e^{-j\mu X_{input}(1)}, \dots, e^{-j\mu X_{input}(N)}]^T + \mathbf{n} \quad (3.43)$$

Where \mathbf{n} is the noise vector that appears in the noisy case. At step l of the recursive procedure, a coordinate vector \mathbf{X}_l is computed so that it minimizes the criterion J defined by :

$$J(\mathbf{X}_l) = |\mathbf{z}_{input} - \mathbf{z}_{\mathbf{X}_l}|^2 \quad (3.44)$$

where

$$\mathbf{z}_{\mathbf{X}_l} = [e^{-j\mu X_l(1)}, \dots, e^{-j\mu X_l(N)}]^T \quad (3.45)$$

The series of vectors \mathbf{X}_l converges when l tends to infinity, towards a vector $\widehat{\mathbf{X}}$ so that $|\mathbf{z}_{input} - \mathbf{z}_{\widehat{\mathbf{X}}}|^2$ is minimum. That is, $\widehat{\mathbf{X}}$ is the minimum argument of J . To find this minimum, a variable step gradient method can be employed [25], which is summarized as follows :

We start from the initialization vector \mathbf{X}_0 . Then the vectors of the series are obtained by the relation :

$$\forall k \in \mathbb{N} : \quad \mathbf{X}_{l+1} = \mathbf{X}_l - \lambda_k \nabla(J(\mathbf{X}_l)), \quad (3.46)$$

where λ_k is the step for the descent, updated at each iteration. The recurrence loop is $\mathbf{X}_l \rightarrow \mathbf{z}_{estimated \ for \ \mathbf{x}_l} \rightarrow J(\mathbf{X}_l)$. We stop when the gradient becomes lower than a threshold.

A more elaborated optimization algorithm consists in combining gradient with spline interpolation. See section 1.4 for a detailed explanation about this method.

The proposed algorithm can be generalized into a multi-step one : At each step, gradient algorithm combined with spline interpolation is run. Consecutive steps are associated with an increasing number of nodes (see Fig. 3.21). After the last step, a few iterations of the gradient algorithm are run, to cancel the residual pixel shifts.

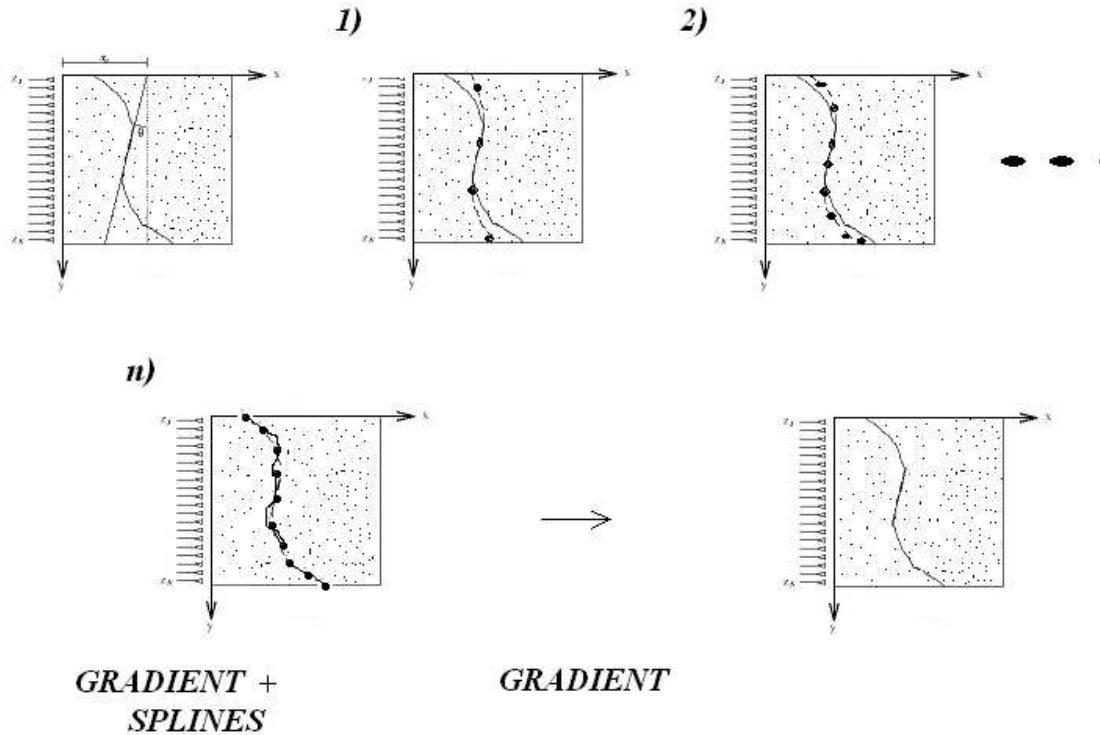


Figure 3.21 — Combination of gradient algorithm and spline interpolation : increasing number of nodes permits an increased resolution.

3.12 Distorted contour retrieval by gradient algorithm combined with spline interpolation : results and discussion

Firstly, we present results obtained on hand-made images : we consider an image with two nearly parallel distorted curves, then a statistical study performed on binary images which contain a curve with variable distortion amplitude. Secondly, we present typical results obtained with real-world grey level images. The proposed method is compared with GVF, which is a snake type algorithm [137]. GVF can be compared with the proposed optimization method for the following reasons :

- GVF is a contour approach method, which enables an initialization by a rectilinear contour, and which is commonly initialized by a continuous contour ;

- GVF is rather insensitive to initialization, that is, the initialization contour can be far from the expected contour, and have any shape. Thus it can be chosen as a straight line, as for the proposed optimization method ;
- GVF retrieves both convex and concave contours.

GVF is based on a local approach, and develops an external force for active contours. It relies on the local characteristics of the processed image and provides a continuous contour. The active contour evolves along an edge map. In the case of close boundaries, the blurring that enables convergence may fuse parts of the contour. Therefore, we predict that the results obtained by GVF method can differ sensibly from the results obtained with the proposed method. Part 3.12.1 presents statistical results obtained by GVF and the proposed method when they are initialized by the proposed straight line estimation method. The efficiency of both methods is measured thanks to the criterion $ME_{\hat{\mathbf{X}}}$, which is the mean error over the coordinates of the curve pixels, with respect to the hand-drawn contour : $ME_{\hat{\mathbf{X}}} = \frac{1}{N} \sum_{i=1}^N |\hat{\mathbf{X}}(i) - \mathbf{X}_{HD}(i)|$, where $\mathbf{X}_{HD}(i)$ is the i^{th} component of vector \mathbf{X}_{HD} containing the coordinates of the hand-drawn contour. The squared processed images have $N = 200$ rows.

When the procedure for straight line retrieval is run, the values of parameters μ and α have to be chosen. Concerning parameter μ , [5] provides a study that gives the maximum value of an estimated orientation, with a value of μ equal to 1. This maximum value is 73° and is enough, considering that if the image is rotated by 90° , all orientation values present in the image can be computed. This is equivalent to place the antenna at the bottom side of the image [3]. If μ is smaller, the maximum orientation value is higher, but it was empirically shown that the value 1 gives the best results. If μ is higher the maximum value of an estimated orientation is lower. That is why we chose to use value 1 for μ . Concerning α , it must be such that the value of α multiplied by the maximum offset value remains in an interval of length 2π . Indeed MF-BLP method leads to the frequency value $-\alpha x_0$. This frequency value must be in the interval $[0, 2\pi[$ in order to avoid any phase indetermination. Therefore we can choose for instance the value 2.5×10^{-3} for an image containing 200 columns. Concerning parameter M , it must be in the interval $[d, N - d + 1]$, where d is the number of expected contours. As advised in [5] the simulation of several signal realizations is applied, we choose $M \simeq \sqrt{N+1}$. Precisely we choose $M = 15$. We apply the proposed optimization method including the initialization. Gradient algorithm is applied to the signals generated upon the virtual antenna. When the signal generation procedure is applied before the optimization procedure, the constant propagation parameter value is $\mu = 0.5 \cdot 10^{-3}$. This avoids any phase indetermination problem [24]. The descent step parameter is updated at each iteration : initial descent step parameter is $\lambda_0 = 0.05$, the step variation is such that $\lambda_{k+1} = 1.05\lambda_k$, and 500 iterations are performed. The number O of interpolation nodes is increased from 4 to 32.

GVF method is run with the following parameter values [135] : For the computation of the edge map : 200 iterations for the computation of the gradient ; $\mu_{GVF} = 0.15$ (regularization coefficient) ; for the snakes deformation : 200 initialization points and 100 iterations. $\alpha_{GVF} = 0.02$ for non-noisy images or $\alpha_{GVF} = 0.2$ for noisy images (tension) ; $\beta_{GVF} = 0.03$ (rigidity) ; $\gamma_{GVF} = 1$ (regularization coefficient) ; $\kappa_{GVF} = 0.8$ (gradient strength coefficient).

3.12.1 Hand-made images

We first consider an image with two nearly parallel distorted curves (see Fig. 3.22(a)). Propagator retrieves the common overall orientation 20° of both curves and MFBP yields

the two close offset values, that is, 181 and 166 (see Fig. 3.22(b)). The proposed optimization method estimates the distortions of both curves (see Fig. 3.22(c) and (d)).

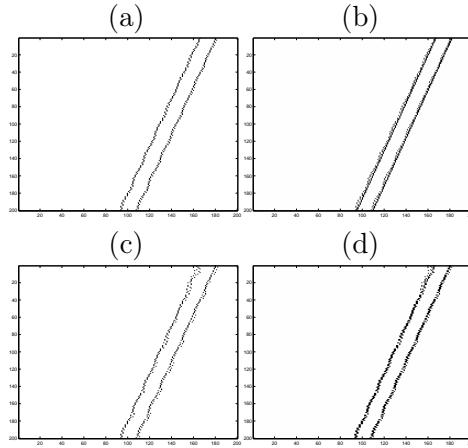


Figure 3.22 — (a) Two nearly parallel curves. (b) Orientation estimation by Propagator, offset estimation by MFBLP. Distortion retrieval by the proposed optimization method : (c) estimation, (d) superposition estimation and processed.

In the next experiment, the proposed optimization method and GVF [137] are applied to images containing a distorted contour. [135] performs a statistical study on images containing a regular shape and demonstrate that a standard GVF method and their generalized version of GVF give satisfactory results. We aim at demonstrating the robustness of the proposed method compared to GVF, when the considered distortions are irregular and with narrow and deep concavities.

To measure quantitatively the abilities of the proposed methods to retrieve any pixel shift value between initialization contour and expected contour, we propose a statistical study : we consider 15 images, with different values for the distortion amplitude. The original non-noisy images have pixel values 1 (expected contour) or 0 (background). They are impaired by Gaussian noise having mean m_{No} and standard deviation σ_{No} equal respectively to 20% and 1% of the value of a pixel that belongs to the expected contour. Noise is added to various percentage values : 0%, 1%, 2%, 4%, 6%. For each noise percentage and distortion amplitude value, 100 trials are performed, with a different noise realization for each trial. To assess the performances of the proposed method, we define the mean error ME (resp. the error standard deviation Std) :

$$ME = \frac{1}{100} (\sum_{j=1}^{100} ME_{\hat{\mathbf{x}}_j}) \text{ (respectively } Std = \sqrt{\frac{1}{100} (\sum_{j=1}^{100} (ME_{\hat{\mathbf{x}}_j} - ME)^2)} \text{)},$$

where j indexes the trials and $ME_{\hat{\mathbf{x}}_j}$ is the mean error over all pixels of the contour, obtained at the j^{th} trial, either when the proposed method or GVF is used.

The least and most distorted contours are the ones of Figs. 3.23(a) and (b). The pixel coordinates of the least and most distorted contours, and their estimation by the proposed method and by GVF are drawn on Figs. 3.23(a) and (b).

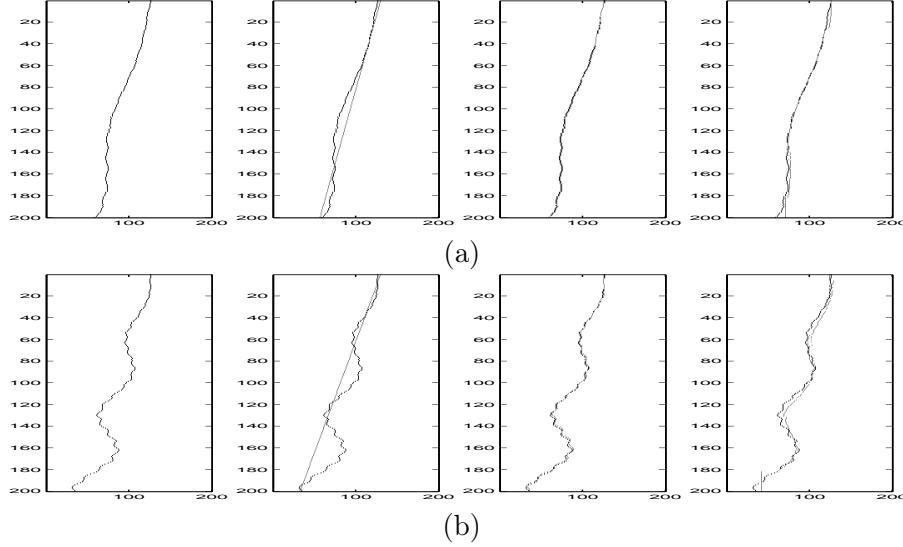


Figure 3.23 — (a) Least distorted contour : initialization, results obtained (b) Most distorted contour : initialization, results obtained by the proposed method and GVF respectively.

ME values (pixels)												
n. dist.	0%		1%		2%		4%		6%		ME _d	
	(A)	(B)	(A)	(B)								
(2 ; 0.53)	0.01	1.54	0.80	1.55	1.28	1.56	1.70	1.60	1.93	1.61	1.14	1.57
(4 ; 0.89)	0.02	1.59	0.82	1.59	1.31	1.59	1.73	1.63	1.93	1.64	1.16	1.60
(5 ; 1.15)	0.02	1.80	0.83	1.84	1.31	1.85	1.82	1.86	1.98	1.87	1.19	1.84
(6 ; 1.62)	0.02	1.82	0.84	1.84	1.31	1.84	1.84	1.91	1.99	1.92	1.20	1.86
(7 ; 1.91)	0.02	1.98	0.84	1.99	1.32	1.99	1.89	2.01	2.09	2.10	1.23	2.01
(9 ; 2.26)	0.03	2.29	0.85	2.29	1.33	2.30	1.90	2.31	2.26	2.37	1.27	2.31
(11 ; 2.79)	0.03	2.46	0.86	2.47	1.36	2.48	1.91	2.49	2.31	2.50	1.29	2.48
(14 ; 3.15)	0.03	2.47	0.86	2.49	1.37	2.51	1.92	2.62	2.32	2.66	1.30	2.55
(14 ; 3.29)	0.03	2.82	0.87	2.82	1.38	2.83	2.01	2.84	2.36	2.85	1.33	2.83
(16 ; 3.91)	0.03	2.83	0.90	2.84	1.42	2.85	2.01	2.88	2.41	2.89	1.35	2.85
(18 ; 4.41)	0.04	2.85	0.91	2.88	1.43	2.89	2.10	2.89	2.56	2.91	1.41	2.88
(21 ; 4.89)	0.04	2.93	0.92	2.96	1.47	2.97	2.23	2.98	2.67	2.99	1.47	2.97
(23 ; 5.14)	0.05	3.15	0.97	3.17	1.55	3.18	2.31	3.20	2.96	3.31	1.57	3.20
(21 ; 5.21)	0.05	3.21	0.98	3.23	1.58	3.24	2.56	3.25	3.26	3.35	1.69	3.25
(23 ; 5.63)	0.05	3.39	0.99	3.40	1.73	3.41	2.58	3.42	3.47	3.43	1.76	3.41
ME _n	0.03	2.48	0.88	2.49	1.41	2.50	2.03	2.53	2.43	2.56	1.36	2.51

Tableau 3.3 — ME values (in pixel) obtained with the proposed method (A) and with GVF (B), versus distortion (maximum amplitude; standard deviation) -dist.- and noise percentage values -n..

Table 3.3 (respectively 3.4) presents the *ME* (respectively *Std*) values for all images and noise percentage values. The first column of Table 3.3 or 3.4 indicates the couple (maximum distortion amplitude; standard deviation of the distortions), for all images. Table 3.3 shows

		Std values (pixels)											
n. dist.		0%		1%		2%		4%		6%		Std_d	
		(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)	(A)	(B)
(2 ; 0.53)	0.00	0.03	0.05	0.04	0.13	0.07	0.10	0.04	0.10	0.10	0.08	0.06	
(4 ; 0.89)	0.00	0.05	0.07	0.05	0.13	0.07	0.11	0.14	0.11	0.11	0.08	0.08	
(5 ; 1.15)	0.00	0.07	0.08	0.05	0.14	0.09	0.12	0.19	0.16	0.14	0.10	0.11	
(6 ; 1.62)	0.00	0.09	0.09	0.06	0.14	0.09	0.16	0.21	0.17	0.14	0.11	0.12	
(7 ; 1.91)	0.00	0.10	0.10	0.07	0.17	0.10	0.16	0.23	0.18	0.15	0.12	0.13	
(9 ; 2.26)	0.00	0.11	0.12	0.07	0.18	0.11	0.18	0.25	0.24	0.16	0.14	0.14	
(11 ; 2.79)	0.00	0.11	0.13	0.08	0.19	0.12	0.21	0.25	0.25	0.17	0.16	0.15	
(14 ; 3.15)	0.00	0.12	0.13	0.08	0.19	0.12	0.23	0.26	0.25	0.21	0.16	0.16	
(14 ; 3.29)	0.00	0.12	0.14	0.10	0.20	0.13	0.24	0.27	0.28	0.23	0.17	0.17	
(16 ; 3.91)	0.00	0.13	0.14	0.11	0.20	0.16	0.26	0.28	0.28	0.26	0.18	0.19	
(18 ; 4.41)	0.00	0.13	0.15	0.12	0.21	0.17	0.27	0.29	0.29	0.29	0.18	0.20	
(21 ; 4.89)	0.00	0.14	0.16	0.12	0.21	0.24	0.28	0.29	0.30	0.30	0.19	0.22	
(23 ; 5.14)	0.01	0.15	0.17	0.13	0.24	0.25	0.28	0.32	0.31	0.31	0.20	0.23	
(21 ; 5.21)	0.01	0.15	0.17	0.14	0.25	0.27	0.28	0.33	0.32	0.32	0.21	0.24	
(23 ; 5.63)	0.02	0.15	0.17	0.14	0.25	0.28	0.28	0.36	0.34	0.32	0.21	0.25	
Std_n	0.00	0.11	0.12	0.09	0.19	0.15	0.21	0.25	0.24	0.21	0.15	0.16	

Tableau 3.4 — Std values (in pixel) obtained with the proposed method (A) and with GVF (B), versus distortion (maximum amplitude; standard deviation) -dist.- and noise percentage values -n..

that mean error values are between 0.01 pixel and 3.47 pixels for the proposed method and between 1.5 pixel and 3.4 pixels for GVF. Mean values over noise percentage and distortion amplitude ME_n and ME_d (respectively Std_n and Std_d) show that the accuracy (respectively the robustness) of the proposed method is not sensitive to distortion amplitude, and that the accuracy and robustness of GVF are not sensitive to noise but sensitive to distortion amplitude for the parameter values which are adopted. When noise percentage is 0%, the expected contour perfectly matches the model adopted for our optimization method, then it leads to a perfect reconstruction, whatever the amplitude of the distortions. In a noisy environment, outliers provoke unexpected phase fluctuations in the generated signal, hence some pixel bias. Pixel bias obtained with GVF can be explained as follows : to adopt the same conditions for GVF and the proposed method the active contour is initialized by a straight line. The pixels of the initialization straight line may be far from the expected contour pixels for some of the rows of the image. In these conditions, for GVF method to converge, it is necessary to apply a gaussian blurring kernel to the processed image, to extend the capture range of GVF. The vector field components point toward each object boundary. If the capture range is extended, GVF field vectors may neutralize each other. That is why the active contour cannot progress into the sections of high curvature. The computational times required to retrieve any considered distorted curve, on a 3.0 Ghz PC running Windows, are the following : For the estimation of parameter θ propagator applied to the signal lasts 5.4×10^{-4} sec. Propagator method based on the covariance matrix processing lasts 1.02×10^{-3} sec. A method that relies on the eigen-decomposition of a covariance matrix, such as TLS-ESPRIT method used in

SLIDE algorithm lasts 1.08×10^{-3} sec. [5]. Propagator applied to the signal is then two times faster than the two compared methods. The proposed method for offset estimation using MFBLP last 0.592 sec. The proposed optimization method, for which 30 interpolation nodes and 3000 iterations are necessary, lasts 26.5 sec. GVF method lasts 28.1 sec.

3.12.2 Distorted contour retrieval in real-world grey level images

The proposed optimization method is initialized by means of the straight line detection method presented in subsection 3.3.1. When grey level images are considered, an edge enhancing procedure attenuates the contribution of the continuous background, which eases the procedure of signal generation [7]. Figs. 3.24, 3.25 and 3.26 present the results obtained on several images acquired in the context of aerial imaging and of an industrial application -default detection in tissue. Initialization contour and final result are superimposed to the processed image. Column (a) contains the processed image, column (b) contains the initialization by the proposed method, column (c) contains the result obtained with the proposed method, column (d) contains the initialization for GVF algorithm, column (e) contains the result obtained with GVF.

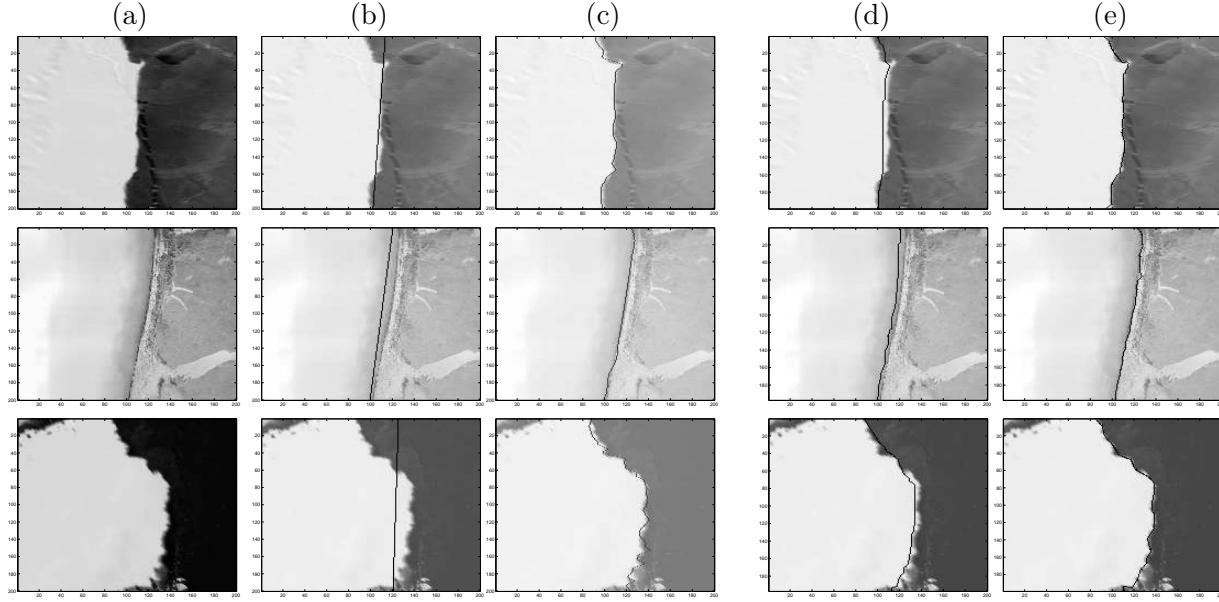


Figure 3.24 — Aerial image analysis : coast detection. (a) processed image, (b) initialization, (c) result with the proposed method, (d) initialization for GVF, (e) result with GVF.

when the active contour performs well, namely when it is initialized close to the expected contour, and when it converges, its result is close to the result obtained with the proposed

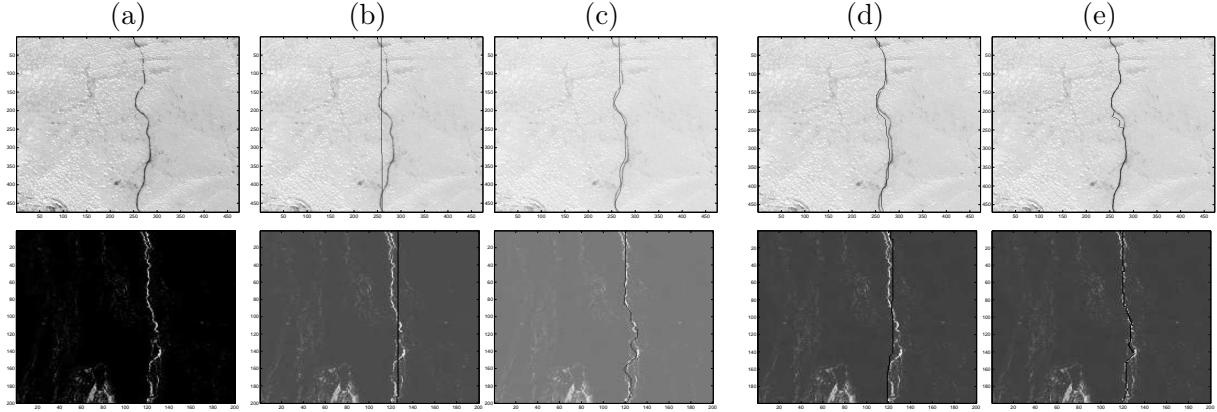


Figure 3.25 — Aerial image analysis : road detection. (a) processed image, (b) initialization, (c) result with the proposed method, (d) initialization for GVF, (e) result with GVF.

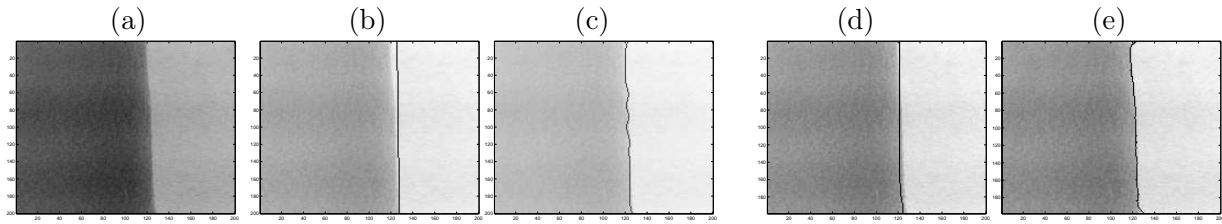


Figure 3.26 — Industrial application : default detection in tissue. (a) processed image, (b) initialization, (c) result with the proposed method, (d) initialization for GVF, (e) result with GVF.

method. We noticed that it is necessary to place the initialization contour very close to the expected contour to obtain acceptable results with the GVF method.

3.13 Conclusion of chapter 3

In this chapter, we considered an image understanding problem : the retrieval of straight lines and nearly linear distorted contours. For this, we transposed subspace-based methods of array processing and optimization methods. To transcript the content of the image into a onedimensionnal signal, we adopted either a constant parameter or a variable parameter propagation scheme. These propagation schemes lead to signal models with orientation or offsets of straight lines as expected frequencies. These parameters are possibly close-valued, and we consider a noisy environment. Now, high-resolution methods of array processing are supposed to cope with these constraints. Therefore, we adapted TLS-ESPRIT and Propagator methods to the estimation of straight line orientations. In particular, the version of Propagator that avoids the computation of a covariance matrix exhibits low computational load. We also adapted high-resolution MFBLP method to the estimation of possibly close straight line offsets.

We proposed an optimization method for distorted contour retrieval, and we adapted gradient optimization algorithm, to retrieve phase distortions from the signal generated out of the image. Phase distortions are proportional to the distortions between an expected curve and

a fitting straight line. This procedure is generalized to the retrieval of several contours by an association of Propagator with gradient method. We also propose a novel optimization method which is still fast, and yields a continuous contour. We compared the results obtained with this method with those obtained with gradient vector flow. Especially, the proposed method retrieves contours with high curvature, and leads to good results when real-world images are processed. These methods rely on signal generation upon a linear antenna. In the next chapter, we consider contours which are no longer linear or nearly linear but circular or nearly circular. For this, we adapt a circular antenna to the processed image.

Subspace-Based and DIRECT algorithms for distorted circular contour estimation

Circular features are commonly sought in digital image processing. SLIDE (subspace-based LIne DEtection) method proposed to estimate the center and the radius of a single circle. In this chapter, we introduce a novel method for estimating several radii while extending the circle estimation to retrieve circular-like distorted contours. Especially, we develop and validate a new model for virtual signal generation by simulating a circular antenna.

This circular antenna yields linear phase signals which suit high-resolution methods which were applied to straight or nearly straight contour estimation. Therefore, the same high-resolution and optimization methods as those used for straight or nearly straight contour estimation can be adapted to the retrieval of circular or nearly circular contour retrieval.

The circle center is estimated by the SLIDE method. A variable speed propagation scheme toward the circular antenna yields a linear phase signal. Therefore, a high-resolution method provides the radius. Either the gradient method or the more robust combination of DIRECT (DIviding RECTangles) and spline interpolation can extend this method for free form object segmentation. The retrieval of multiple non concentric circles and rotated ellipses is also considered. To evaluate the performance of the proposed methods, we compare them with a least-squares method, Hough transform and GVF (Gradient Vector Flow). We apply the proposed method to hand-made images while considering some real-world images.

This chapter is organized as follows. In Section 4.1, we review the existing methods for contour retrieval, and emphasize the interest of array processing methods applied to image understanding. In Section 4.2, the problem of circle retrieval is highlighted, while explaining how to generate a signal from the image, upon a circular antenna. In Section 4.3, a signal model is derived. We will show that a linear phase signal is obtained when circles are expected and either variable [7] or constant propagation parameter is adopted for signal generation. By using the MDL (Minimum Description Length) criterion [132, 5], the number of concentric features is detected ; then a high-resolution method [114] estimates the possibly close radius values of the expected concentric circles. Then, the proposed approach is generalized to retrieve several circles or ellipses with different centers and radii. In Section 4.4, the retrieval of circular contours is extended to any closed contour fitted by a circle. Optimization methods are investigated for this purpose. The fast variable step gradient method and the robust

DIRECT (DIviding RECTangles) method [64] are accelerated using spline interpolation. In Section 4.5, we discuss the results obtained by the proposed approach when it is applied to hand-made and real-world images. The proposed radius estimation method is compared with least-squares fitting and Hough transform applied to radius estimation [122]. Our optimization method is compared with GVF [137], in particular concerning the robustness to noise and contour curvature. Concluding remarks about the methods presented in this chapter are provided in section 4.6.

This work has been published in :

Julien Marot and Salah Bourennane, "Subspace-Based and DIRECT Algorithms for Distorted Circular Contour Estimation", IEEE trans. on Image Processing, vol. 16, no. 9, pp. 2369-2378, september 2007, in

J. Marot and S. Bourennane, "Array processing and fast Optimization Algorithms for Distorted Circular Contour Retrieval", Advances on signal processing, article ID 57354, 13 pages, july 2007, and in

J. Marot and S. Bourennane, "Array processing approach for object segmentation in images", IEEE ICASSP'07, Vol. 1, pp. 621-24, April 2007.

4.1 Introduction

Circular features are commonly sought in digital image processing. Circle fitting is suitable in several domains such as quality inspection for food industry, mechanical parts [81] and particle trajectories [37, 65]. Circle fitting has been applied in microwave engineering [68] and ball detection for robotic vision systems [35]. Nearly circular features are sought in character recognition. [36, 45] proposed a template-based system for online character recognition, where the representative templates are automatically counted. These templates are viewed as different writing styles for any character. Ordinary or total least-squares methods for circle fitting seek to minimize the squares sum of error-of-fit with respect to measures [16, 47, 20, 101]. Using geometric fitting [16], error distances are defined with the orthogonal, or shortest, distances from given points to the geometric feature to be fitted. In [20], a least-squares fitting approach is proposed. It is based on the hypothesis that a set of circular arcs extracted from the image is related to a set of circles contained in a model by translation, rotation and scaling. [20] presents an analytical solution based on least-squares fitting to obtain an optimal geometric transformation for the alignment of the circular arcs with circles. For a single circular arc, this method is easy to implement. For multiple arcs with different center, this method relies on the strong hypothesis that all model circles should be transformed by the same translation, rotation and scaling operations to fit the arcs in the image. One of the major limitations of most of least-squares approaches is their sensitivity to outliers [20]. [9, 10] focused on planar spline interpolation between control-points by quadratic rational Bézier curves, to retrieve circular arcs. Compared to [20], this method does not take into account the prior knowledge of models or templates. It is based on interpolation between control-points. Other methods support circle retrieval, even in a noisy environment. The generalized Hough transform (GHT) provides estimation of the circle center coordinates when their radius is known [61, 19]. Its main drawback is the computational load, although fast versions have been proposed [122]. Contour-based snakes methods [69], such as Gradient Vector Flow (GVF), were largely used [137, 135], to retrieve concavities and weak edges with blurred boundaries. GVF limitations can be observed when the expected contour exhibits a high curvature. Levelset type methods [15, 66]

enhance blindly all contours in images. Levelset does not provide explicitly the characteristics of contours with particular predefined parameters. Array processing methods [3, 5] have been proposed to find some characteristics of circular contours. The formalism proposed by Aghajan [3] detects circular or elliptic contours. A propagation phenomenon and the impinging of a wavefront upon the antenna are simulated through a variable speed propagation scheme. The center vertical and horizontal coordinates are estimated by placing the antenna successively at the top and left sides of the image. [3] proposed a radius estimation method, which relies on an approximation. This method can obtain a sub-image from the initial image, in such a way the top left corner is centered on the circle center, that was previously estimated. In [3], Aghajan showed that a truncation of the Taylor series of the generated signal leads to the value of the radius of the expected circle.

In this chapter, we propose a new approach for radius estimation, and for the retrieval of distortion between any nearly circular contour and a circle. The approach uses a circular antenna. Especially, we adapt an optimization method to retrieve a nearly circular contour. We consider concentric and non concentric features, either nearly circular or nearly oval. In [25, 27], we adopted a similar strategy for approximately rectilinear distorted contours. In [25, 27], a uniform linear array was used. It was demonstrated that array processing methods applied to contour detection are robust to noise impairment and yield fast algorithms. As it will be illustrated later, the proposed approach based on circular antenna exhibits the same robustness and speed properties. Another advantage of the proposed method over the Hough transform method is that the estimates in our approach are inherently continuous, whereas in the Hough transform method, the resolution is limited by the bin size chosen for the parameter research interval. The proposed optimization approach retrieves any pixel shift value, whatever the curvature of the contour.

4.2 Problem statement and signal generation

4.2.1 Problem statement

We highlight the problem of radius estimation, and of the estimation of the distortions between a closed contour and a circle that fits with this contour. We propose a circular antenna that enables a particular signal generation [87, 84]. We emphasize the phase of the generated signals. The generated signals fit classical array processing and optimization methods, contrary to signals derived by the existing signal generation methods [3]. Fig. 4.1(a) presents a binary digital image I as a square matrix of dimensions $N \times N$. Each element represents an image pixel. An object in the image is made of edge pixels with value 1, over a background of zero-valued pixels. The object is fitted by a circle with radius value r and center coordinates (l_c, m_c) . Fig. 4.1(b) shows a sub-image extracted from the original image, so that its top left corner is the center of the circle. This sub-image is associated with a set of polar coordinates (ρ, θ) . Each pixel of the expected contour in the sub-image has the following coordinates : $r + \Delta\rho, \theta$. $\Delta\rho$ is the shift between the contour pixel and the circle one that fits the contour and which has the same coordinate θ . We seek for star-shaped contours, that is, contours described by the relation $\rho = f(\theta)$, where f is any function that maps $[0, 2\pi]$ to \mathbb{R}_+ .

A traditional circle fitting method is the generalized Hough transform (GHT) [19, 122]. We employ a variant of Hough transform that estimates the radii of concentric circles when their

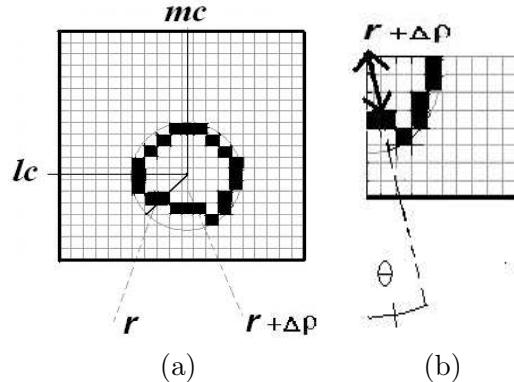


Figure 4.1 — (a) Circular-like contour; (b) Bottom right quarter of the contour and pixel coordinates in the polar system (ρ, θ) having its origin on the center of the circle. r is the radius of the circle. $\Delta\rho$ is the value of the shift between a pixel of the contour and the pixel of the circle having same coordinate θ .

center is known. Its basis is to count the number of pixels which are located on a circle for all possible radius values. The drawback of Hough transform is its elevated computational load. From this emerges the need for a faster radius estimation. In [5, 8, 3], Aghajan *et al.* replaced the Hough transform by the much faster SLIDE (subspace-based LIne DEtection) algorithm to retrieve straight lines. Existing algorithms [25, 27] adapt an optimization method to signals generated upon a linear antenna composed of one sensor per row. When the expected contours consist of one pixel per row, only one unknown parameter of the optimization problem is included in one component of the generated signal [25, 27].

Contours which are approximately circular are supposed to be made of more than one pixel per row for some of the rows of the image and more than one pixel per column for some columns of the image. Several pixels may lead to only one signal component. A linear antenna does not lead to a linear phase signal when a circular contour is present in the image.

4.2.2 Virtual signal generation

We set an analogy between the estimation of a circular contour in an image and the estimation of a wavefront in array processing. The basic idea is to obtain a linear phase signal from an image containing a quarter of circle. To achieve this, we use a circular antenna. The phase of the signals which are virtually generated on the antenna is constant or varies linearly as a function of the sensor index. A quarter of circle with radius r and a circular antenna are represented on Fig. 4.2. The antenna is a quarter of circle centered on the top left corner, and crossing the bottom right corner of the sub-image. Such an antenna is adapted to the sub-images containing each quarter of the expected contour (see Fig. 4.2). In practice, the extracted sub-image is possibly rotated so that its top left corner is the estimated center. The antenna has radius R_a so that $R_a = \sqrt{2}N_s$ where N_s is the number of rows or columns in the sub-image. When we consider the sub-image which includes the right bottom part of the expected contour, the following relation holds : $N_s = \max(N - l_c, N - m_c)$ where l_c and m_c are the vertical and horizontal coordinates of the center of the expected contour in a cartesian set centered on the top left corner of the whole processed image (see Fig. 4.1). Coordinates l_c and m_c are estimated by the method proposed in [3], or the one that is detailed later in this

chapter.

Signal generation scheme upon a circular antenna is the following : the directions adopted for signal generation are from the top left corner of the sub-image to the corresponding sensor. The antenna is composed of S sensors, so there are S signal components.

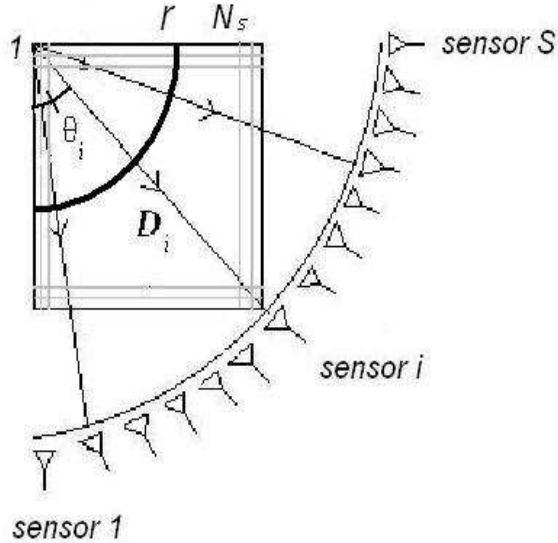


Figure 4.2 — Sub-image, associated with a circular array composed of S sensors.

Let us consider D_i , the line that makes an angle θ_i with the vertical axis and crosses the top left corner of the sub-image. The i^{th} component ($i = 1, \dots, S$) of the signal \mathbf{z} generated out of the image reads :

$$z(i) = \sum_{\substack{l,m=1 \\ (l,m) \in D_i}}^{l,m=N_s} I(l,m) \exp(-j\mu\sqrt{l^2 + m^2}), \quad (4.1)$$

The integer l (resp. m) indexes the lines (resp. the columns) of the image. j stands for $\sqrt{-1}$. μ is the propagation parameter [7]. Each sensor indexed by i is associated with a line D_i having an orientation $\theta_i = \frac{(i-1)\cdot\pi/2}{S}$. In Eq. (4.1), the term $(l, m) \in D_i$ means that only the image pixels that belong to D_i are considered for the generation of the i^{th} signal component. Satisfying the constraint $(l, m) \in D_i$, that is, choosing the pixels that belong to the line with orientation θ_i , is done in two steps : let $setl$ be the set of indexes along the vertical axis, and $setm$ the set of indexes along the horizontal axis. If θ_i is less than or equal to $\pi/4$, $setl = [1 : N_s]$ and $setm = \lfloor [1 : N_s] \cdot \tan(\theta_i) \rfloor$. if θ_i is greater than $\pi/4$, $setm = [1 : N_s]$ and $setl = \lfloor [1 : N_s] \cdot \tan(\pi/2 - \theta_i) \rfloor$. Symbol $\lfloor \cdot \rfloor$ means integer part. The minimum number of sensors that permits a perfect characterization of any possibly distorted contour is the number of pixels that would be virtually aligned on a circle quarter having radius $\sqrt{2}N_s$. Therefore, the minimum number S of sensors is $\sqrt{2}N_s$.

4.3 Proposed method for radius and center estimation

Estimation of multiple concentric circles

Most often, there exists more than one circle for one center. We demonstrate how several possibly close radius values can be estimated using a high-resolution method. We propose an estimation method for the number d of concentric circles, and each radius value. For this purpose, we employ a variable speed propagation scheme [7]. We set $\mu = \alpha(i - 1)$, for each sensor indexed by $i = 1, \dots, S$. From Eq. (4.1), the signal received on each sensor is :

$$z(i) = \sum_{k=1}^d \exp(-j\alpha(i-1)r_k) + n(i), \quad i = 1, \dots, S \quad (4.2)$$

where $r_k, k = 1, \dots, d$ are the values of the radius of each circle, and $n(i)$ is a noise term due to outliers. All components $z(i)$ compose the observation vector \mathbf{z} . TLS-ESPRIT (Total least-squares-ESTimation of Parameters via Rotational Invariance Techniques) algorithm requires the estimation of the covariance matrix of several snapshots. There is no time-dependent signals. So the question arises as how a sample covariance matrix can be formed. This can be done as follows [5] : From the observation vector we build K sub-vectors of length M with $d < M \leq S - d + 1$: $\mathbf{z}_l = [z(l), \dots, z(l + M - 1)]^T$, $l = 1, \dots, K$. To maximize the number of snapshots [25], the first component of a snapshot is the second component of the previous snapshot. This improves the estimation of the covariance matrix that is performed in TLS-ESPRIT algorithm. We obtain then $K = S + 1 - M$ snapshots. Grouping all sub-vectors obtained in matrix form, we get $\mathbf{Z}_K = [\mathbf{z}_1, \dots, \mathbf{z}_K]$, where

$$\mathbf{z}_l = \mathbf{A}_M \mathbf{s} + \mathbf{n}_l, \quad l = 1, \dots, K. \quad (4.3)$$

$\mathbf{A}_M = [\mathbf{a}(r_1), \dots, \mathbf{a}(r_d)]$ is a Vandermonde type matrix of size $M \times d$: the i^{th} component of $\mathbf{a}(r_k)$ is $\exp(-j\alpha(i-1)r_k)$. \mathbf{s} is a length d vector equal to $[1, 1, \dots, 1]^T$ -superscript T denotes transpose- and $\mathbf{n}_l = [n(l), \dots, n(l + M - 1)]^T$.

The signal model of Eq. (4.3) suits TLS-ESPRIT method, a subspace-based method that requires the dimension of the signal subspace, that is, in this problem, the number of concentric circles. MDL criterion estimates the dimension of the signal subspace [5] from the eigenvalues of the covariance matrix. TLS-ESPRIT is applied on the measurements collected from two overlapping sub-arrays, and falls into two parts : the covariance matrix estimation and the minimization of a total-least-squares criterion. The radius values are obtained as [114] :

$$\hat{r}_k = \frac{-1}{\alpha} \mathcal{Im}(\ln(\frac{\lambda_k}{|\lambda_k|})), \quad k = 1, \dots, d \quad (4.4)$$

where \mathcal{Im} denotes imaginary part, $\{\lambda_k, k = 1, \dots, d\}$ are the eigenvalues of a diagonal unitary matrix. It relates the measurements from the first sub-array with the measurements resulting from the second sub-array.

Estimation of multiple circles with different centers and radii

Usually, an image contains several circles which are possibly not concentric and have different radii (see Fig. 4.3). To apply the proposed method, the center coordinates for each

feature are required. To estimate these coordinates, we generate a signal with constant propagation parameter upon the image left and top sides. More details regarding signal generation upon a linear antenna can be found in [5]. The l^{th} signal component, generated from the l^{th} row, reads : $z_{lin}(l) = \sum_{m=1}^N I(l, m) \exp(-j\mu m)$, where μ is the propagation parameter [5]. The non-zero sections of the signals, as seen at the left and top sides of the image (see Fig. 4.3), indicate the presence of features. Each non-zero section width in the left (respectively the top) side signal gives the height (respectively the width) of the corresponding expected feature. The middle of each non-zero section in the left (respectively the top) side signal yields an approximate value of the center l_c (respectively m_c) coordinate of each feature. The feature height, width and center coordinates permit to select image regions. Each region contains a contour or a set of concentric contours. The proposed method based on circular antenna detects the number of concentric features for each estimated center, in each image region. It yields the values of the fitting circles radii. Then, an optimization method can be used to refine the estimation of each contour. Such method applied to the signals generated on our circular antenna is presented in the next section.

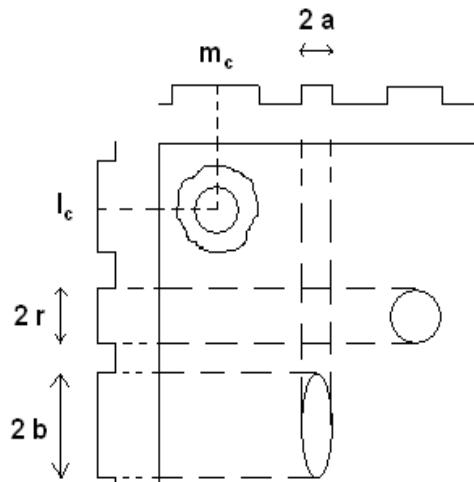


Figure 4.3 — Model for an image containing several nearly circular or elliptic features. r is the circle radius, a and b are the axial parameters of the ellipse.

Single circle estimation

Using a circular antenna, a constant parameter propagation scheme yields as well to the radius of a single circle. Signal components are $z(i) = \exp(-j\mu r) + n(i)$, $i = 1, \dots, S$. The estimated value \hat{r} of r is : $\hat{r} = \frac{1}{\mu} \cos^{-1}(\bar{z})$, with $\bar{z} = \mathcal{R}\text{e}(\frac{1}{S} \sum_{i=1}^S z(i))$, where \cos^{-1} denotes inverse cosine function and $\mathcal{R}\text{e}$ denotes real part. In this algorithm, noise contribution is lowered by averaging. In the next section, we use the fitting circle or ellipse provided by the method presented above to retrieve any distorted contour.

Numerical complexity of the circle fitting method

In the general case the image contains outlier pixels and several concentric circles. First, as concerns the estimation of the coordinates of the center [3] : it is performed by signal generation upon a linear antenna located on one horizontal and then one vertical side of the image, followed by TLS-ESPRIT method. This antenna contains N sensors, each corresponding to one row or column. The estimation of the coordinate of the center requires the following operations and computational complexity, for each coordinate along horizontal and vertical axes [3] :

- Variable speed propagation scheme upon a linear antenna aside the image : N^2 operations [5] ;
- Application of TLS-ESPRIT to the covariance matrix of the generated signals : for the estimation, and respectively the fast eigendecomposition, of the covariance matrix in TLS-ESPRIT method [117, 7] : $N \cdot M$, and respectively M^2 ;

We choose $M = \sqrt{N}$, as recommended in [7]. The computational complexity for center retrieval is then $N^2 + N \cdot (\sqrt{N} + 1)$.

As concerns the estimation of the radius values, we remind that d is the number of concentric circles and the dimension of the signal subspace in the covariance matrix in TLS-ESPRIT method. The computational complexity of the steps of our method for radius estimation is :

- For signal generation [117, 5] : the number of sensors multiplied by the number of pixels that are crossed by each line \mathbf{D}_i , that is : $S \cdot N_{subimage}$ or equivalently $S \cdot N$;
- For the estimation, and respectively the fast eigendecomposition, of the covariance matrix in TLS-ESPRIT method [117, 7] : $S \cdot M$, and respectively $d \cdot M^2$;

We choose $M = \sqrt{S}$, as recommended in [7]. The computational complexity of the angle estimation method is then $S \cdot N + S \cdot (\sqrt{S} + d)$. In practice, the order of magnitude of S is N , and the computational load of the proposed method for center and radius estimation is N^2 .

As concerns the generalized Hough transform we discretize the ρ axis to the minimum required number of values, that is, $\sqrt{2} \cdot N$ for the computation of the accumulator. Also, the θ axis for counting the edge pixels is discretized to $\sqrt{2} \cdot N$ values ($\sqrt{2} \cdot N$ is the minimum number of orientations that permits to characterize any contour in the image, see subsection 4.2.2). In these conditions the order of magnitude of the computational load of the generalized Hough transform, for the estimation of the center and the radius of the circles, is N^3 [122]. To conclude, the computational complexity of the proposed method is N^2 , as compared to N^3 for the generalized Hough transform. The same order of magnitude of computational loads were obtained in [5] when SLIDE algorithm was compared with the Hough transform for straight line retrieval.

4.4 Optimization method for the estimation of nearly circular contours

The optimization methods proposed in [25, 27] estimate the pixel shift values between a straight line and a nearly linear contour. These methods rely on the assumption that one component of the generated signal is associated with only one unknown pixel shift value in one image row (or column). The proposed circular antenna adapts optimization methods to retrieve distorted star-shaped contours.

4.4.1 Proposed optimization method

We employ a circular antenna to retrieve the shift values between an initialization circle and the expected contour. We process successively each quarter of the circle, and retrieve the distortions between one quarter of the initialization circle and the expected contour part that is located in the same quarter of the image. As an example, in Fig. 4.1, the right bottom quarter of the considered image is represented in Fig. 4.1(b). An optimization method inspired from [25] is defined as follows :

A contour in the considered sub-image is described with a set of polar coordinates by : $\{\rho(i), \theta(i), i = 1, \dots, S\}$. We estimate the S unknowns $\rho(i)$ that characterize the contour, forming a vector :

$$\rho = [\rho(1), \rho(2), \dots, \rho(S)]^T, \quad (4.5)$$

The basic idea is to consider that ρ can be written as : $\rho = [r + \Delta\rho(1), r + \Delta\rho(2), \dots, r + \Delta\rho(S)]^T$ (see Fig. 4.1), where r is the radius of a circle that fits the expected contour. The optimization method that we adapt to this problem estimates $\{\Delta\rho(i), i = 1, \dots, S\}$, that is, the shift values between the initialization circle and the expected contour.

If we take into account the position given by ρ of all edge pixels in Eq. (4.1), the components of signal \mathbf{z} generated out of the image containing the expected contour read :

$$z(i) = \exp(-j\mu\rho(i)), \quad \forall i = 1, \dots, S \quad (4.6)$$

We try to recreate the signal \mathbf{z} , whose components are defined by Eq. (4.6), and in which we ignore the S parameters. We start with an initialization vector ρ_0 , characterizing a quarter of circle that fits the expected distorted contour in the considered sub-image. The S components of ρ_0 are equal to r , the radius value that was previously estimated : $\rho_0 = [r, r, \dots, r]^T$. Then, with k indexing this recursive algorithm steps, we aim to minimize :

$$J(\rho_k) = \|\mathbf{z} - \mathbf{z}_{\text{estimated for } \rho_k}\|^2 \quad (4.7)$$

where $\|\cdot\|$ represents the norm induced by the usual scalar product of \mathbb{C}^S . The components of $\mathbf{z}_{\text{estimated for } \rho_k}$ are defined similarly to the components of \mathbf{z} (see Eq. (4.6)) as a function of the components of ρ_k , and the components of ρ_k are obtained from the components of ρ_0 by adding a shift : $\rho_k = [r + \Delta\rho_k(1), r + \Delta\rho_k(2), \dots, r + \Delta\rho_k(S)]^T$. To estimate ρ , we use either the variable step gradient method or the more robust combination of DIRECT and spline interpolation methods, which is detailed in section 1.4. We denote by $\hat{\rho}$ the vector including all the estimated values $\rho_k(i)$, $i = 1, \dots, S$

4.4.2 Summary of the proposed method

The proposed method for distorted contour estimation is summarized as follows :

- Generation of a signal with constant parameter, on a linear antenna placed at the top of and aside the image ;
- Estimation of the centers of the circles or ellipses that fit the expected contours ;
- Variable speed propagation scheme upon the proposed circular antenna : Estimation of the number of circles by MDL criterion, estimation of the radius of each circle fitting any expected contour (see Eqs. (4.1) and (4.2)) or the axial parameters of the ellipse ;

- Estimation of the radial distortions, in a polar system, between any expected contour and the circle or ellipse that fits this contour. Either the gradient method or the combination of DIRECT and spline interpolation may be used to minimize the criterion J of Eq. (4.7).

4.5 Results and discussion

We compare the proposed methods with a least-squares one, Hough transform and GVF. The efficiency of the proposed methods is measured by the mean error ME_ρ over the coordinates of the pixels of the contour. For the four quarters of an image, the coordinates of the pixels of the contour are contained in the vector ρ defined in Eq. (4.5), and their estimates are contained in vector $\hat{\rho}$. $ME_\rho = \frac{1}{S} \sum_{i=1}^S |\hat{\rho}(i) - \rho(i)|$ where $|\cdot|$ means absolute value. The proposed circle fitting method is applied to images having $N = 200$ columns and rows. The number of sensors for each image quarter is $S = 400$, larger than the minimum acceptable value (see subsection 4.2.2). The algorithms for center and radius estimation are run using a propagation parameter set at $\alpha = 1.35 \times 10^{-2}$. This optimal value is experimentally defined in [7] for almost the same number of sensors. In [7], a value for the length of each sub-array was empirically found to provide optimal results, namely $M = \sqrt{S} = 20$. Other experiments confirmed these values [25, 27]. When a constant parameter signal generation is performed, $\mu = 5 \times 10^{-3}$. This value of μ avoids phase indetermination [25, 27].

4.5.1 Hand-made images

We first consider the case where one circle is expected in a noisy image. Fig. 4.4 exemplifies the results obtained with a least-squares fitting method [47], and the proposed circle fitting method.

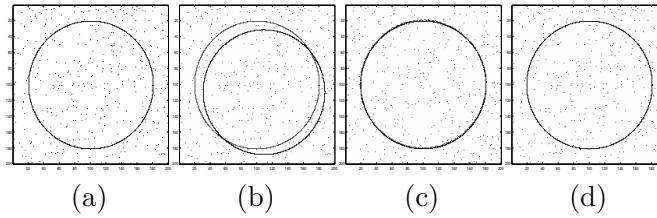


Figure 4.4 — (a) Processed (b) Superposition processed and result obtained after applying a least-squares method for circle fitting : $ME_\rho = 4.5$ pixels, (c) Superposition processed and result obtained after applying the Hough transform : $ME_\rho = 0.7$ pixel, (d) Superposition processed and result obtained after applying the proposed method : $ME_\rho = 0.3$ pixel.

Center coordinates are $(100, 100)$, the radius value is 80 pixels. When the image is not noisy, there is no bias over the estimated parameters, for all methods. Fig. 4.4 (a) contains 1% of noisy pixels with value 1. We performed 100 trials with the same circle and noise parameters, and different noise realization for each trial. Least-squares fitting (see Fig. 4.4 (b)) provides a 7 pixel bias, and a 9 pixel bias respectively, over the two center coordinates of the center, and a 1.3 pixel bias over the radius of the circle. Generalized Hough transform, with prior knowledge of the radius, provides center coordinates $(100, 100)$. With the prior knowledge of these center coordinate values, Hough transform provides a 0.2 pixel bias over

the radius value (see Fig. 4.4 (c)). The proposed method leads to a 0.2 and 0.3 pixel bias over the coordinates of the center, and a 0.2 pixel bias over the radius value (see Fig. 4.4 (d)). It is more robust to noise than the least-squares method, when a single circle in an impaired image is considered. These results validate the remarks provided in [20], where the sensitivity of least-squares-fit of circle to outliers is underlined.

Fig. 4.4 (c) exemplifies the ability of Hough transform to handle the cases of noisy images. We compared quantitatively and statistically the proposed method with the Hough transform method when a single circle is expected, in a noisy image context. The non-impaired image contains a circle with radius 80 pixels. We performed 100 trials with different noise realizations, with 2% of noisy pixels in the image, and obtained mean error values over the radius which are respectively 1.05 pixels when variable parameter propagation scheme is performed (see Eq. (4.4)) ; 1.20 when constant parameter propagation scheme is performed and $\hat{r} = \frac{1}{\mu} \cos^{-1}(\bar{z})$; and 1.15 pixels when Hough transform is applied.

Like the proposed method, Hough transform is supposed to retrieve the radius of two close concentric circles. We considered the case of an image containing two concentric circles. The expected radius values are 85 and 90 pixels (see Fig. 4.5a)) thus differing by only 6%. When the proposed method is applied, MDL criterion detects two circles, the estimated radius values are 85.1 and 89.9 pixels, and in these conditions the required computational time is 0.359 sec. on a 3.0 Ghz Pentium 4 PC running under Windows. The same computer and software are used throughout all experiments. When Hough transform is run, ρ axis is quantized so that the step between two values is 0.3 pixel, and θ axis is quantized to S values. Estimated radius values are 84.7 and 90.3 pixels, and the required computational time is 0.51 sec. When the least-squares fitting method [101] is used the estimated radius values are 85.4 and 90.3 pixels (see Fig. 4.5b)). The required computational time is 0.95 sec.

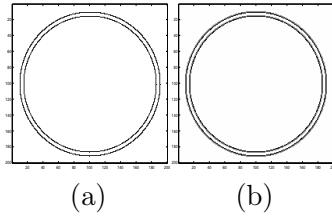


Figure 4.5 — (a) Processed, (b) Result (superimposed), with the proposed method for radius estimation or equivalently with Hough transform or a least-squares fitting method : $ME_\rho = 0.1$ (resp. 0.3 for Hough transform and 0.4 for the least-squares fitting method).

In the next experiment, we study the sensitivity of the proposed optimization methods to initialization. Gradient algorithm minimizes the criterion of Eq. (4.7). We have been confident that gradient algorithm gives good results for images containing continuous contours and low-valued noise pixels. The descent step parameter is updated at each iteration : initial descent step parameter is $\beta_0 = 0.05$, the step variation is so that $\beta_{k+1} = 1.05\beta_k$. We consider an image impaired by 5% of noisy pixels with value 0.1, containing one circle with center coordinates (100, 100) and radius 80. We choose an initialization circle with center coordinates (70, 70). Hough transform yields, as could be expected, to a biased radius value, more precisely 122.1 pixels. Radius value 20 pixels is used for the initialization. The result circle, obtained after 350 iterations of gradient algorithm, has center coordinates (100, 100) and ra-

dius 80 : all pixel shifts are canceled. This case is not easily handled by GVF. To converge in a few seconds, GVF method has to be initialized with a contour which is close to the expected one. Furthermore, the gradient strength has to be set high enough when the edge map of the image is derived [137]. Conversely, the proposed signal generation process transcribes the content of all pixels on one direction D_i (see Fig. 4.2) and thus takes necessarily any feature into account. In the example of Fig. 4.6(a) we consider a distorted and disrupted contour in

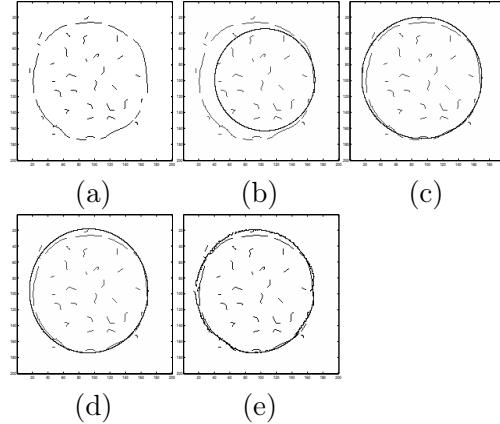


Figure 4.6 — Biased center and radius estimation, proposed optimization algorithm using DIRECT and spline interpolation. (a) Processed (b) Initialisation by the least-squares method, (c) Initialisation by the Hough transform, (d) Initialisation by the proposed method, (e) Superposition processed and final result, $ME_\rho = 0.9$ pixel.

a noisy environment. We have been confident that spline interpolation yields a continuous result contour, and that DIRECT handles the case where noisy pixels have the same value as the contour pixels. Fig. 4.6(b) shows the result obtained with least-squares fitting. Fig. 4.6(b) shows the result obtained with the Hough transform, used with the center coordinates provided by the proposed method -signal generation upon linear antenna-. Fig. 4.6(d) shows the result obtained with the proposed method for circle fitting. Note that there is a slight bias on the estimation of the center coordinates and the radius. We use this result to initialize the optimization method. Ten iterations of DIRECT are run, with 6 nodes for spline interpolation. Fig. 4.6(e) shows that in spite of this bias, the proposed optimization method yields a continuous result contour, without focusing on noisy pixels. We now perform a statistical study on the proposed optimization algorithm using variable step gradient method. The proposed optimization method and gradient vector flow [137] are applied to images containing a distorted circle. [135] presents a statistical study on noisy images containing a harmonic shape. A standard GVF method and a generalized version of GVF give satisfactory results when noise percentage values up to 10% are employed. Our goal is to demonstrate the sensitivity of the proposed method compared to GVF, when the considered distortions are no longer harmonic but irregular. Values of the parameters for GVF method [135] are the following. For the computation of the edge map : 100 iterations ; $\mu_{GVF} = 0.09$ (regularization coefficient) ; for the snakes deformation : 100 initialization points and 50 iterations ; $\alpha_{GVF} = 0.2$ (tension) ; $\beta_{GVF} = 0.03$ (rigidity) ; $\gamma_{GVF} = 1$ (regularization coefficient) ; $\kappa_{GVF} = 0.8$ (gradient strength coefficient). We study the robustness of the proposed method and GVF to both noise impairment and curvature. Statistical results presented below are obtained with 15 images containing a different distorted circle, with variable distortion amplitude. The maximum distortion amplitude and the standard deviation of the distortions is different for each image.

The least and most distorted contours are drawn on Figs. 4.7(a) and (b) respectively. Fig. 4.7 also presents the results obtained with the proposed method. Random noise is added to various percentage values of the image pixels : 0, 1, 2, 4, 6, and 8%.

The original non-noisy images have pixel values 1 (expected contour) or 0 (background). They are impaired with Gaussian noise having mean m_{No} and standard deviation σ_{No} equal respectively to 20% and 1% of the value of a pixel that belongs to the expected contour. For each image quarter, computational times are respectively 0.28 sec. for signal generation, 0.15 sec. for running variable step gradient algorithm -50 iterations are performed- ; and 12 sec. for running GVF algorithm. Furthermore, the proposed method is five times faster than GVF. For each noise percentage and each contour, 100 trials are performed, with a different noise realization for each trial. To assess the performance of the proposed method, we define the mean error ME :

$$ME = \frac{1}{100} \sum_{j=1}^{100} ME_{\rho_j}, \text{ where } j \text{ indexes the trials and } ME_{\rho_j} \text{ is the mean error over all pixels of the contour, obtained at the } j^{\text{th}} \text{ trial.}$$

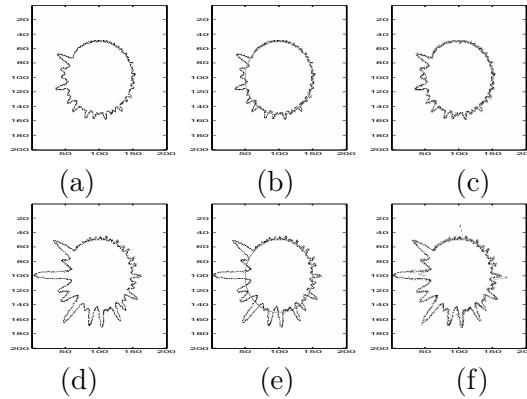


Figure 4.7 — Examples of processed images containing the less (a) and the most (d) distorted circles, initialization (b,e) and estimation using the proposed method (c,f). $ME_p = 1.4$ pixel and 2.7 pixels.

Table 4.1 presents the ME values for all maximum distortion amplitude and noise percentage values. Table 4.1 presents the ME values for all images and noise percentage values. The first column of Table 4.1 indicates the couple (maximum distortion amplitude ; standard deviation of the distortions), for all images. Table 4.1 shows that mean error values are between 1.44 and 3.43 pixels for the proposed method and between 1.40 and 4.33 pixels for GVF. The mean values ME_d of the ME values obtained for each distortion amplitude demonstrate that GVF is limited by high curvature values. These mean values increase rapidly when the maximum distortion amplitude increases. The mean values ME_n of the ME values obtained for each noise percentage highlight that the proposed method outperforms GVF for all chosen noise percentage values. The errors obtained with GVF are the consequence of its inability to progress into the furthest sections of some concavities, while the proposed method retrieves the pixel shifts whatever their values are. The increase in pixel bias come from unexpected fluctuations : noise corrupts the phase of the generated signals.

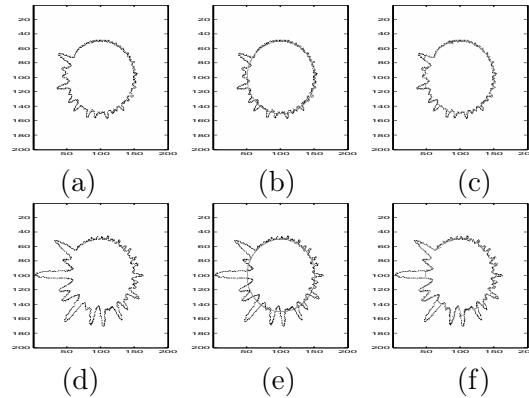


Figure 4.8 — Examples of processed images containing the less (a) and the most (d) distorted circles, initialization (b,e) and estimation using GVF algorithm (c,f). $ME_\rho = 1.4$ pixel and 4.1 pixels.

ME values (pixels)														
n. dist.	0%		1%		2%		4%		6%		8%		Mean ME_d	
(9.28 ; 1.66)	(A) 1.44	(B) 1.40	(A) 1.83	(B) 1.42	(A) 2.00	(B) 1.43	(A) 2.05	(B) 1.44	(A) 2.08	(B) 1.45	(A) 2.10	(B) 1.47	(A) 1.91	(B) 1.43
(11.84 ; 1.97)	(A) 1.51	(B) 1.62	(A) 1.96	(B) 1.64	(A) 2.06	(B) 1.66	(A) 2.07	(B) 1.68	(A) 2.12	(B) 1.69	(A) 2.23	(B) 1.70	(A) 1.99	(B) 1.66
(14.68 ; 2.42)	(A) 1.68	(B) 1.67	(A) 1.98	(B) 1.69	(A) 2.09	(B) 1.72	(A) 2.10	(B) 1.72	(A) 2.16	(B) 1.73	(A) 2.27	(B) 1.74	(A) 2.04	(B) 1.71
(17.76 ; 2.93)	(A) 1.70	(B) 1.78	(A) 2.00	(B) 1.85	(A) 2.11	(B) 1.86	(A) 2.12	(B) 1.87	(A) 2.18	(B) 1.88	(A) 2.28	(B) 1.89	(A) 2.06	(B) 1.85
(18.36 ; 3.46)	(A) 1.72	(B) 2.07	(A) 2.02	(B) 2.08	(A) 2.13	(B) 2.08	(A) 2.14	(B) 2.08	(A) 2.19	(B) 2.11	(A) 2.30	(B) 2.12	(A) 2.08	(B) 2.09
(21.05 ; 3.07)	(A) 1.81	(B) 2.10	(A) 2.17	(B) 2.10	(A) 2.24	(B) 2.10	(A) 2.25	(B) 2.11	(A) 2.26	(B) 2.12	(A) 2.32	(B) 2.15	(A) 2.17	(B) 2.11
(21.55 ; 3.62)	(A) 1.82	(B) 2.28	(A) 2.18	(B) 2.29	(A) 2.26	(B) 2.31	(A) 2.27	(B) 2.32	(A) 2.32	(B) 2.33	(A) 2.38	(B) 2.34	(A) 2.20	(B) 2.31
(24.88 ; 4.25)	(A) 1.84	(B) 2.59	(A) 2.19	(B) 2.61	(A) 2.30	(B) 2.62	(A) 2.32	(B) 2.63	(A) 2.55	(B) 2.64	(A) 2.56	(B) 2.65	(A) 2.29	(B) 2.62
(28.34 ; 5.01)	(A) 2.02	(B) 2.69	(A) 2.29	(B) 2.71	(A) 2.46	(B) 2.72	(A) 2.67	(B) 2.74	(A) 2.77	(B) 2.75	(A) 2.79	(B) 2.76	(A) 2.50	(B) 2.72
(31.85 ; 5.21)	(A) 2.18	(B) 3.04	(A) 2.37	(B) 3.11	(A) 2.52	(B) 3.12	(A) 2.85	(B) 3.14	(A) 2.89	(B) 3.18	(A) 2.90	(B) 3.19	(A) 2.61	(B) 3.13
(35.39 ; 5.86)	(A) 2.19	(B) 3.12	(A) 2.47	(B) 3.16	(A) 2.60	(B) 3.19	(A) 2.89	(B) 3.21	(A) 2.90	(B) 3.34	(A) 2.96	(B) 3.41	(A) 2.66	(B) 3.23
(38.90 ; 6.61)	(A) 2.30	(B) 3.43	(A) 2.51	(B) 3.51	(A) 2.62	(B) 3.55	(A) 2.95	(B) 3.59	(A) 3.00	(B) 3.61	(A) 3.01	(B) 3.62	(A) 2.73	(B) 3.55
(42.33 ; 7.42)	(A) 2.42	(B) 3.69	(A) 2.63	(B) 3.71	(A) 2.67	(B) 3.82	(A) 3.00	(B) 3.83	(A) 3.13	(B) 3.84	(A) 3.19	(B) 3.85	(A) 2.84	(B) 3.79
(45.63 ; 8.28)	(A) 2.43	(B) 4.05	(A) 2.89	(B) 4.06	(A) 2.96	(B) 4.07	(A) 3.14	(B) 4.08	(A) 3.20	(B) 4.09	(A) 3.34	(B) 4.09	(A) 2.99	(B) 4.07
(48.75 ; 9.22)	(A) 2.77	(B) 4.21	(A) 3.11	(B) 4.22	(A) 3.16	(B) 4.23	(A) 3.30	(B) 4.27	(A) 3.38	(B) 4.28	(A) 3.43	(B) 4.33	(A) 3.19	(B) 4.25
Mean ME_n	1.98	2.64	2.30	2.67	2.41	2.69	2.54	2.71	2.60	2.73	2.67	2.75	2.41	2.69

Tableau 4.1 — ME values (in pixel) obtained with the proposed method (a) and with GVF (b), versus distortion (maximum amplitude; standard deviation) -dist.-, and noise percentage -n.-.

Multiple circles with different centers and radii

We retrieve nearly circular features which may be concentric or not, and which have different radii. GHT handles the case of non-concentric circles with same radius : it provides a good estimation of each circle center, when the radius of the expected circles is known. We use the proposed method for the characterization of several contours (see Section 4.3), as we

presented it in [86]. We perform constant parameter propagation scheme. Starting from the estimation of the center of each feature, the method proposed in Section 4.3 estimates the number of concentric features and the approximate radius value of each feature.

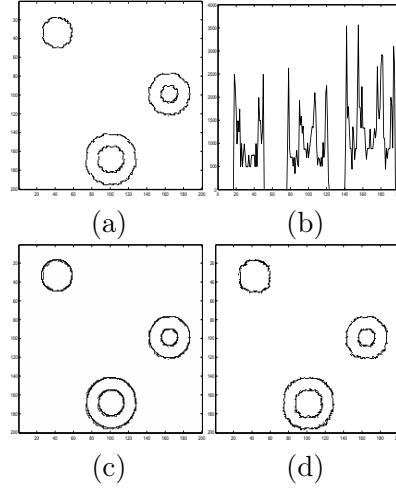


Figure 4.9 — Multiple circles : (a) Processed (b) Signal generated upon the left side of the image (c) Initialization (d) Result with gradient method.

Figs. 4.9 (a, b, c, d) show an image with multiple circles, the signal generated upon the left side of the image, the initialization and the final result obtained by the proposed methods. Starting from the modulus of the generated signal, we apply hard threshold. Then we take the derivative of the binary signal obtained. The points with non-zero derivative indicate successively the beginning and end of each non-zero section of generated signal. All circles in the image have different radius, and the proposed method retrieves efficiently the number of concentric circles -through MDL criterion- and the parameters of all circles. Distortions are retrieved by the proposed optimization method. In spite of a bias (3, 2) pixels over estimation of the two center coordinates of the bottom circles couple, the proposed optimization method retrieves the expected circles.

Two circles with different centers and same radius

We consider the ambiguous case where an image contains two circles with center coordinates (l_{c1}, m_{c1}) and (l_{c2}, m_{c2}) , and same radius r . Exchanging either horizontal or vertical coordinates of the centers does not change the modulus of the generated signals. The phase of the generated signals overcomes this ambiguity. The signal component at row l_{ci} , $i = 1, 2$ indexing circles 1 and 2 (see Fig. 4.1), reads : $z_{lin}(l_{ci}) = \sum_{m=1}^N I(l, m) \exp(-j\mu m) = e^{-j\mu(m_{ci}-r)} + e^{-j\mu(m_{ci}+r)} = z_{lin}(l_{ci}) = 2 e^{-j\mu m_{ci}} \cos(\mu r)$. Therefore, the center horizontal coordinate is obtained by :

$$m_{ci} = \frac{-1}{\mu} \operatorname{Im}(\ln(\frac{z_{lin}(l_{ci})}{2 \cos(\mu r)})), \quad i = 1, 2 \quad (4.8)$$

We consider images of Fig. 4.10, where circle radius is 15 pixels. Circles are only slightly distorted, which leads to the same length for all signal non-zero sections. For first (respectively second) processed image (see Figs. 4.10(a) and (b)), the first circle center coordinates are

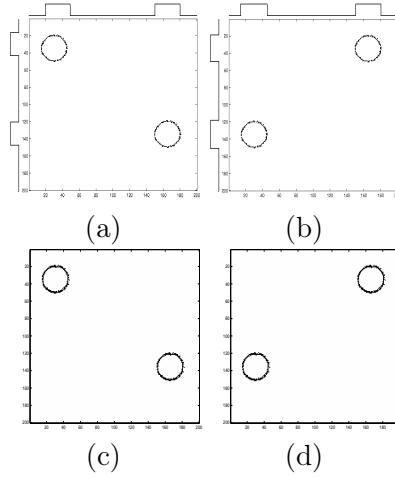


Figure 4.10 — Two circles with different centers and same radius. (a) and (b) Processed images, modulus of the generated signals ; (c) and (d) Superposition processed and result with gradient method.

$(l_{c1}, m_{c1}) = (30, 35)$, the second circle center coordinates are $(l_{c2}, m_{c2}) = (135, 165)$ (respectively $(30, 165)$ and $(135, 35)$ for the second image). For both images, the estimation obtained from the modulus of the generated signals (see aside and on top of the each image in Fig. 4.10) are, for the horizontal coordinates : 165, 35 ; for the vertical coordinates : 135, 30. The phase of the generated signals associates the horizontal coordinates obtained from the signal modulus, with the vertical coordinates, obtained from the phase (see Eq. (4.8)) : $l_{c1} = 30$ pixels yields $m_{c1} = 35.1$ (respectively 165.2 for the second image), and $l_{c2} = 135$ pixels yields $m_{c2} = 35.1$. Thus each circle is retrieved. Thus by studying the phase of the generated signals, we distinguish between the first image and second one. Running the proposed optimization method yields the results of Figs. 4.10(c) and (d).

Rotated ellipse retrieval

Let us consider the case of a vertical ellipse or a rotated ellipse (see Fig. 4.11 (a) and (c)). An ellipse is characterized by two axial parameters instead of a constant radius. For the case where an ellipse is expected, the method proposed in [3] leads to two axial parameters and the ellipse center. The axes of the ellipse may not be horizontal and vertical (see Fig. 4.11 (c)). In this case, we propose the following procedure. We consider the signals generated on the linear antenna placed at the top and the left side, and more precisely the non-zero section length of the signals. The image is rotated until these lengths are the most different from each other. The ellipse of Fig. 4.11 (c) has center coordinates $(100, 100)$, and axial parameters 80 and 25 pixels. Its inclination is 47° . We perform rotations of the image containing the ellipse, with a 1° step between each rotation, and test the length of the non-zero sections of the generated signals after each rotation. The estimated value of the ellipse inclination is 47° and the corresponding estimated axial parameter values are 80 and 25 pixels. When the ellipse is distorted, the proposed optimization method is logically applied.

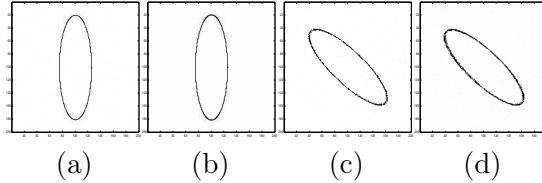


Figure 4.11 — Ellipse (and respectively rotated ellipse) fitting : (a) and (c) Processed ; (b) and (d) Superposition processed and result obtained after applying the proposed method for ellipse fitting : $ME_\rho = 0.7$ (respectively $ME_\rho = 0.6$) pixel.

4.5.2 Real-world images

The proposed methods summarized in subsection 4.4.2 are applied on real-world images (see Figs. 4.12 (1a), (2a), (3a)). Real-world images are supposed to be more difficult to process than hand-made ones. This is due to the disruptions in the expected contours. Therefore, we use the combination of the robust DIRECT method and spline interpolation to estimate distortions. First, we compute the mean of the image three color components. We apply a Canny edge-enhancing operator. The expected contours are supposed to be centered in the middle of the image. The centers of the initialization circles are taken as the center of the image, that is, pixel (100, 100). The initialization circles number and radius are obtained as follows. We apply variable speed propagation scheme on the proposed circular antenna. Then TLS-ESPRIT method yields the radius values. MDL criterion provides the number of expected nearly circular contours. Together, DIRECT combined with spline interpolation are fast enough to be compared with gradient vector flow, if a small number of nodes is chosen for the interpolation. The first processed image concerns pie calibration. Fig. 4.12(1a) gives the original color image. Fig. 4.12(1b) gives the result of Canny edge enhancement, Fig. 4.12(1c) gives the initialization circle superimposed to the processed image. Fig. 4.12(1d) shows that gradient provides a contour which is not continuous and whose pixels go aside the pixels of the expected contour. When gradient method is employed the mean error value ME_ρ is 2.0 pixels. Fig. 4.12(1e) gives the result using GVF. Mean error value ME_ρ is 1.6 pixel. Fig. 4.12(1f) gives the result obtained by DIRECT combined with spline interpolation. When this robust optimization method is used, the mean error value ME_ρ is 0.6 pixel. Parameters used to run DIRECT and spline interpolation are 6 interpolation nodes and 5 iterations for DIRECT. For this image the required computational time is 26 sec. Then we consider biometrics (iris fitting, see Figs. 4.12 (2)) and checking mechanical tool size (see Fig. 4.12 (3)).

Table 4.2 gives the mean error values ME_ρ , for all images. It shows that for all images, DIRECT combined with spline interpolation gives the best result in terms of mean error.

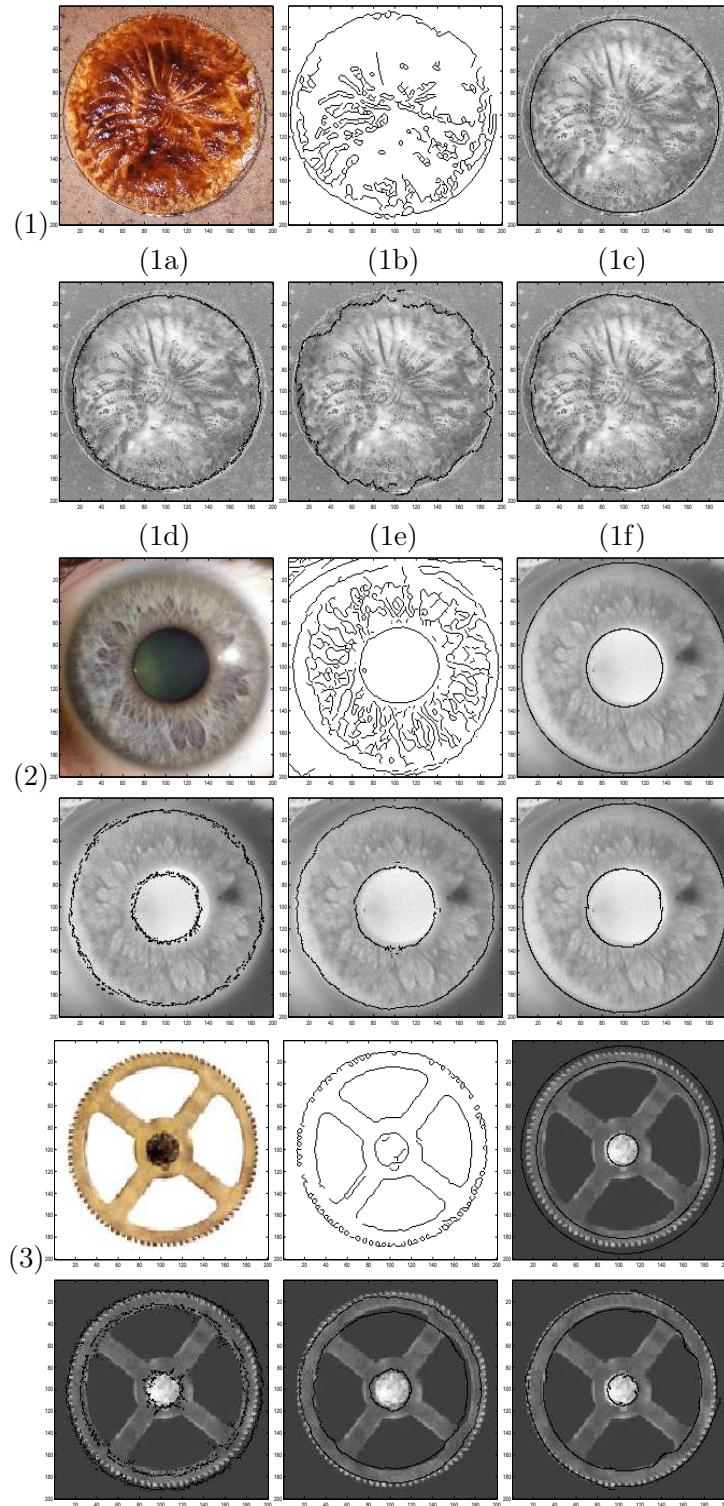


Figure 4.12 — (a) Processed image, (b) result of Canny operator (c) Initialisation, (d) Result obtained with gradient method, (e) Result obtained with GVF, (f) Result obtained with DIRECT combined with spline interpolation.

image	(1)	(2)	(3)
Gradient : ME_ρ	0.9	0.8	0.5
GVF : ME_ρ	1.8	0.7	0.5
DIRECT, spline : ME_ρ	0.8	0.4	0.2

Tableau 4.2 — Real-world images : ME values (in pixel).

Distorted circle fitting : medical application

We now consider biomedical applications. Fig. 4.13 concerns medical images : fibroglanular disc localization in a mammographic image. Fig. 4.14 illustrates the localization of the foveal avascular zone in a digital retinal angiogram. Figs. 4.13(a) and 4.14(a) give the processed image. Figs. 4.13(b) and 4.14(b) give the initialization circle superimposed to the processed image. Figs. 4.13(c) and 4.14(c) give the result obtained with Gradient method. Part of the small bias remaining after the initialization is cancelled. Computational times are respectively 2.1 sec. for the initialization and 16.1 sec. for the optimization method.

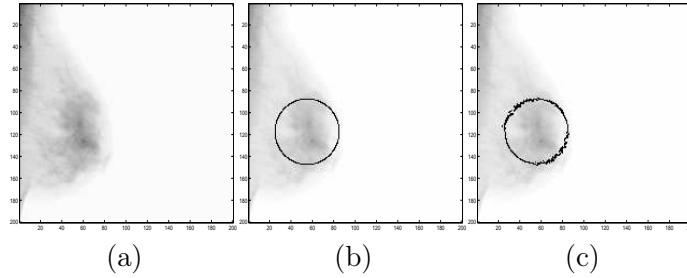


Figure 4.13 — (a) Processed image, (b) Initialisation, (c) Result obtained with the proposed optimization method.

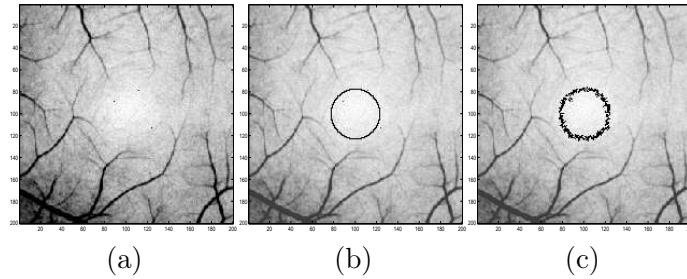


Figure 4.14 — (a) Processed image, (b) Initialisation, (c) Result obtained with the proposed optimization method.

4.6 Conclusion of chapter 4

This chapter investigates the estimation of distorted circular contours in images by means of array processing and optimization methods. We demonstrate the effectiveness and efficiency of a circular antenna for the generation of linear phase signals out of images containing circular contours. This enables the use of high-resolution methods and optimization algorithms in the estimation of distorted circles in images. A variable speed propagation scheme and MDL criterion estimate the number and radii of concentric circles. Using a circular antenna, a constant parameter propagation scheme and optimization methods extend circle estimation to nearly circular star-shaped contour retrieval. Gradient algorithm, or DIRECT algorithm combined with spline interpolation, retrieve the pixel shifts between an initialization circle and the expected contour. We generalize the procedure to the estimation of multiple circles with different centers and radii, and rotated ellipses. We applied the proposed methods to hand-made and real-world images and compared the results obtained with those from least-squares fitting, Hough transform and gradient vector flow. The retrieval of one circle in a noisy image exemplified the robustness to noise of our approach. By considering concentric circles, we proved the ability of our approach to distinguish between two close concentric circles. With an experiment concerning a disrupted contour in a noisy image, we showed that DIRECT combined with spline interpolation yields a continuous result in difficult conditions. The proposed signal generation and our optimization method using gradient outperforms gradient vector flow when contours with high curvature are retrieved. The generalization of our approach to multiple non concentric contours and ellipses was illustrated. The robust optimization method that combines DIRECT with spline interpolation was successfully applied to real-world images and compared with gradient vector flow, leading to low pixel bias. The results of the experiments show that the proposed method is fast and promising for feature retrieval. Further work could consist in retrieving occluded circles and ellipses.

Third part

Tensor approach of image processing,
nonorthogonal tensor flattening via
main direction detection

Lower-rank tensor approximation and multiway filtering

This chapter presents some recent filtering methods based on the lower-rank tensor approximation approach for denoising tensor signals. In this approach, multicomponent data are represented by tensors, that is, multiway arrays, and the presented tensor filtering methods rely on multilinear algebra. The chapter is organized as follows : In section 5.1, we present application fields of the presented methods, and provide definitions of tensor decomposition models. Section 5.2 presents the tensor data and a short overview of its main properties. Section 5.3 introduces the tensor formulation of the classical noise-removal problem as well as some new tensor filtering notations. Firstly, we explain how the channel-by-channel SVD-based method processes successively each component of the data tensor. Secondly, we consider two methods that take into account the relationships between each component of the considered tensor. These two methods are based on the n th-mode signal subspace. The first method for signal tensor estimation is based on multimode PCA achieved by rank- (K_1, \dots, K_N) approximation. The second method is a new tensor version of Wiener filtering. Section 5.4 presents some comparative results where the overviewed multiway filtering methods are applied to noise reduction in color images, denoising of multispectral images, and denoising of multicomponent seismic waves. Section 5.5 presents a nonorthogonal tensor flattening method, that aims at improving the result of tensor data denoising when main directions are expected in the processed tensor. We consider an application to hyperspectral imagery. Finally, section 5.6 concludes the chapter.

Appendices A.1, A.2, and A.3 concern mathematical details about the Alternating Least Squares (ALS) optimization algorithms used in the presented multiway algorithms, and some details about multiway Wiener filtering.

This work has been accepted in :

D. Muti, S. Boureinne and J. Marot, "Lower-Rank Tensor Approximation and Multiway Filtering", SIAM Journal on Matrix Analysis and Applications (SIMAX), accepted september 12, 2007

5.1 Introduction

Tensor data modeling and tensor analysis have been improved and used in several application fields such as quantum physics, economy, chemometrics, psychology, data analysis, etc. Nevertheless, only recent studies focus their interest on tensor methods in signal processing applications. Tensor formulation in signal processing has received great attention since the recent development of multicomponent sensors, especially in imagery (color or multispectral images, video, etc.) and seismic fields (antenna of sensors recording waves with polarization properties). Indeed, the digital data obtained from these sensors are fundamentally higher-order tensor objects, that is, multiway arrays whose elements are accessed via more than two indexes. Each index is associated with a dimension of the tensor generally called "*n*th-mode" [40, 41, 76, 77].

For the last decades, the classical algebraic processing methods have been specifically developed for vector and matrix representations. They are usually based on the covariance matrix, the cross-spectral matrix, or more recently, on the higher-order statistics. Their overall aim is classically to determine a subspace associated with the signal or the parameters to estimate. They mainly rely on three algebraic tools such as :

- (1) The Singular Value Decomposition (SVD) [51], which is used in Principal Component Analysis (PCA).
- (2) Penrose-Moore matrix inversion [51].
- (3) The matrix lower rank approximation, which, according to Eckart-Young theorem [43], can be achieved thanks to a simple SVD truncation.

These methods have proven to be very efficient in several applications.

When dealing with multicomponent data represented as tensors, the classical processing techniques consist in rearranging or splitting the data set into matrices or vectors in order for the previously quoted classical algebraic processing methods to be applicable. The original data structure is then built anew, after processing.

In order to keep the data tensor as a whole entity, new signal processing methods have been proposed [93, 94, 95]. Hence, instead of adapting the data tensor to the classical matrix-based algebraic techniques (by rearrangement or splitting), these new methods propose to adapt their processing to the tensor structure of the multicomponent data. This new approach implicitly implies the use of multilinear algebra and mathematical tools that extend the SVD to tensors.

Two main tensor decomposition methods that generalize the matrix SVD have been initially developed in order to achieve a multimode Principal Component Analysis and recently used in tensor signal processing. They rely on two models, which are the TUCKER3 model and the PARAFAC model.

(1) The TUCKER3 model [77, 123] adopted in higher-order SVD (HOSVD) [11, 40] and in lower rank- (K_1, \dots, K_N) tensor approximation [38, 41, 121]. We denote by HOSVD- (K_1, \dots, K_N) the truncation of HOSVD, performed with ranks (K_1, \dots, K_N) , in modes $1, \dots, N$, respectively. This model has recently been used as multimode PCA, in seismics for wave separation based on a subspace method, in image processing for face recognition and expression analysis [127, 131], and noise filtering of color images [94].

(2) The PARAFAC model, and the CANDECOMP model, developed in [57] and [31] respectively. In [78] the link was set between CANDECOMP and PARAFAC models. The

CANDECOMP / PARAFAC model, referred to as the CP model [70], has recently been applied to food industry [30], array processing [119], and telecommunications [120].

These two decomposition methods differ in the tensor rank definition on which they are based. The HOSVD- (K_1, \dots, K_N) and the rank- (K_1, \dots, K_N) approximation rely on the n th-mode rank definition, that is, the rank of the tensor n th-mode flattening matrix [40, 41]. The rank- (K_1, \dots, K_N) approximation [41] relies on an optimization algorithm which is initialized by the HOSVD- (K_1, \dots, K_N) [40]. The rank- (K_1, \dots, K_N) approximation improves the approximation obtained with the HOSVD- (K_1, \dots, K_N) . It relies on the determination of the signal subspace in every n th-mode of the data tensor and copes with additive white Gaussian noise. The rank- (K_1, \dots, K_N) approximation provides the best approximation in the sense of least Frobenius norm of the difference between estimated and expected tensors. Nevertheless it assumes a non-correlated Gaussian noise. In order to face the case of correlated Gaussian noise, a variant of rank- (K_1, \dots, K_N) approximation, based on fourth order cumulants, was proposed [97]. Indeed, as it is proven in [89], the fourth order cumulants of a Gaussian variable are null.

A tensor framework was employed by [39] in order to express the solution to the linear ICA problem which employs fourth order cumulants. The multilinear ICA (N-mode ICA) model [128, 129], which was developed for face recognition, encodes the fourth-order cumulants for each of the n th-mode flattening matrices of the tensor.

The CP model relies on a canonical decomposition of a tensor into a summation of rank-one tensors and on the extension of the classical matrix rank. Details on the tensor ranks and orthogonal tensor decomposition can be found in [59, 74].

When the TUCKER3 model and the PARAFAC model are associated with an ALS loop, they are known respectively as TUCKALS3 algorithm [77, 76], and PARAFAC ALS algorithm [78, 57]. Many recent studies have been conducted to improve the convergence of these algorithms [41, 73, 140, 118].

The goal of this chapter is to present an overview of the principal results concerning this new approach of data tensor filtering. More details on the algorithms presented in this survey can be found in [98, 93, 94, 96, 97]. These algorithms are analogous to Multilinear ICA, but were developed independently for image filtering. The presented algorithms are based on a signal subspace approach, so they are efficient when the noise components are uncorrelated, the signal and the additive noise are uncorrelated, and when some rows or columns of the image are redundant. In this case it is possible to distinguish between a signal subspace and a noise subspace, as for the traditional SVD-based filtering and Wiener filtering algorithms. Wiener filtering requires prior knowledge on the expected noise-free signal or image. However, multi-way filtering methods provide the following advantage over traditional filtering methods : by apprehending a multiway data set as a whole entity, they take into account the dependence between modes thanks to ALS algorithms. The goal of the chapter is also to present some simulations and comparative results concerning color images and multicomponent seismic signal filtering.

The following notations are used in the rest of the chapter : scalars are denoted by italic lowercase roman, like a ; vectors by boldface lowercase roman, like \mathbf{a} ; matrices by boldface uppercase roman, like \mathbf{A} ; tensors by uppercase calligraphic, like \mathcal{A} . We distinguish a random vector, like \mathbf{a} , from one of its realizations, by using a supplementary index, like \mathbf{a}_i .

5.2 Tensor representation and properties

We define a tensor of order N as a multidimensional array whose entries are accessed via N indexes. A tensor is denoted by $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, where each element is denoted by $a_{i_1 \dots i_N}$, and \mathbb{R} is the real manifold. Each dimension of a tensor is called n th-mode, where n refers to the n^{th} index. Fig. 5.1 shows how a color image can be represented by a third order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1 is the number of rows, I_2 is the number of columns, and I_3 is the number of color channels. In the case of a color image, we have $I_3 = 3$. Let us define $E^{(n)}$

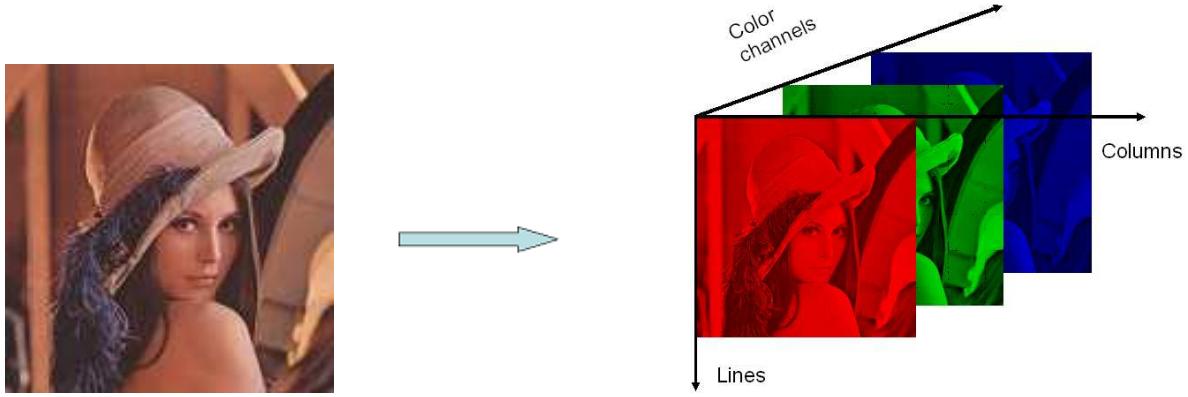


Figure 5.1 — ‘Lena’ standard color image and its tensor representation.

as the n th-mode vector space of dimension I_n , associated with the n th-mode of tensor \mathcal{A} . By definition, $E^{(n)}$ is generated by the column vectors of the n th-mode flattening matrix. The n th-mode flattening matrix \mathbf{A}_n of tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is defined as a matrix from $\mathbb{R}^{I_n \times M_n}$, where :

$$M_n = I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1}. \quad (5.1)$$

For example, when we consider a third-order tensor, the definition of the matrix flattening involves the dimensions I_1 , I_2 , I_3 in a backward cyclic way [17, 40, 70]. When dealing with a 1st-mode flattening of dimensionality $I_1 \times (I_2 I_3)$, we formally assume that the index i_2 varies more slowly than i_3 . For all $n = 1$ to 3 , \mathbf{A}_n columns are the I_n -dimensional vectors obtained from \mathcal{A} by varying the index i_n from 1 to I_n and keeping the other indexes fixed. These vectors are called the n th-mode vectors of tensor \mathcal{A} . An illustration of the 2nd-mode flattening of a color image is presented in Fig. 5.2.

In the following, we use the operator “ \times_n ” as the “ n th-mode product”, that generalizes the matrix product to tensors. Given $\mathcal{A} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, the n th-mode product between tensor \mathcal{A} and matrix \mathbf{U} leads to the tensor $\mathcal{B} = \mathcal{A} \times_n \mathbf{U}$, which is a tensor of $\mathbb{R}^{I_1 \times \dots \times I_{n-1} \times J_n \times I_{n+1} \times \dots \times I_N}$, whose entries are given by :

$$b_{i_1 \dots i_{n-1} j_n i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 \dots i_{n-1} i_n i_{n+1} \dots i_N} u_{j_n i_n}. \quad (5.2)$$

Next section presents the recent filtering methods for tensor data.

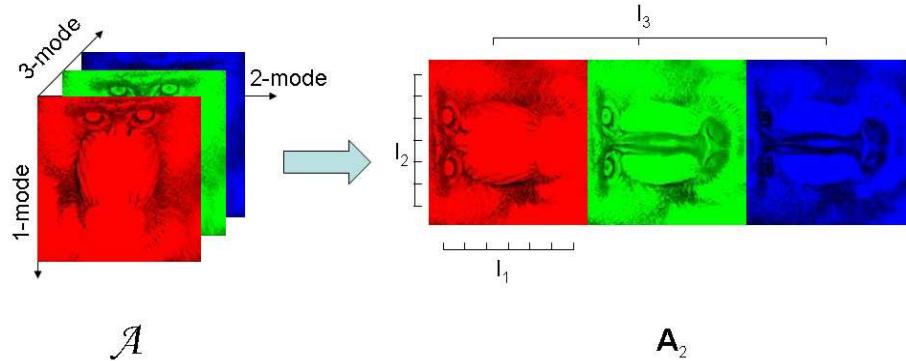


Figure 5.2 — 2nd-mode flattening of tensor $\mathcal{A} : \mathbf{A}_2$.

5.3 Tensor filtering problem formulation

The tensor data extend the classical vector data. The measurement of a multidimensional and multiway signal \mathcal{X} by multicomponent sensors with additive noise \mathcal{N} , results in a data tensor \mathcal{R} such that :

$$\mathcal{R} = \mathcal{X} + \mathcal{N}. \quad (5.3)$$

\mathcal{R} , \mathcal{X} and \mathcal{N} are tensors of order N from $\mathbb{R}^{I_1 \times \dots \times I_N}$. Tensors \mathcal{N} and \mathcal{X} represent noise and signal parts of the data respectively. The goal of this study is to estimate the expected signal \mathcal{X} thanks to a multidimensional filtering of the data [93, 94, 96, 97] :

$$\hat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \mathbf{H}^{(2)} \times_3 \dots \times_N \mathbf{H}^{(N)}, \quad (5.4)$$

From a signal processing point of view, the n th-mode product is a n th-mode filtering of data tensor \mathcal{R} by n th-mode filter $\mathbf{H}^{(n)}$. Consequently, for all $n = 1$ to N , $\mathbf{H}^{(n)}$ is the n th-mode filter applied to the n th-mode of the data tensor \mathcal{R} .

In this chapter we assume that the noise \mathcal{N} is independent from the signal \mathcal{X} , and that the n th-mode rank K_n is smaller than the n th-mode dimension I_n ($K_n < I_n$, for all $n = 1$ to N). Then it is possible to extend the classical subspace approach to tensors by assuming that, whatever the n th-mode, the vector space $E^{(n)}$ is the direct sum of two orthogonal subspaces, namely $E_1^{(n)}$ and $E_2^{(n)}$, which are defined as follows :

- $E_1^{(n)}$ is the subspace of dimension K_n , spanned by the K_n singular vectors associated with the K_n largest singular values of matrix \mathbf{X}_n ; $E_1^{(n)}$ is called the signal subspace [1, 89, 139, 138].
- $E_2^{(n)}$ is the subspace of dimension $I_n - K_n$, spanned by the $I_n - K_n$ singular vectors associated with the $I_n - K_n$ smallest singular values of matrix \mathbf{X}_n ; $E_2^{(n)}$ is called the noise subspace [1, 89, 139, 138].

The dimensions K_1, K_2, \dots, K_N can be estimated by means of the well-known AIC (Akaike Information Criterion) or MDL (Minimum Description Length) criteria [132], which are entropy-based information criteria. Hence, one way to estimate signal tensor \mathcal{X} from noisy data tensor \mathcal{R} is to estimate $E_1^{(n)}$ in every n th-mode of \mathcal{R} . The following section presents three tensor filtering methods based on n th-mode signal subspaces. The first method is an extension of classical matrix filtering algorithms. It consists of a channel-by-channel SVD-based filtering.

The second filtering method is based on multimode PCA achieved by rank- (K_1, \dots, K_N) approximation. Two algorithms are presented for this case. The first algorithm is implicitly developed for an additive *white* and Gaussian noise assumption, whereas the second algorithm represents an improvement of the first one in the case of a *correlated* Gaussian noise. This improvement is achieved thanks to higher-order statistics.

The third method, the multiway Wiener filtering (Wmm- (K_1, \dots, K_N)), is an algorithm that extends the classical two-dimensional Wiener filtering to tensor data.

5.3.1 Channel-by-channel SVD-based filtering

The classical algebraical methods operate on two-dimensional data matrices and are based on the Singular Value Decomposition (SVD) [1, 12, 13], and on Eckart-Young theorem concerning the best lower rank approximation of a matrix [43] in the least-squares sense.

In the first method, a preprocessing is applied to the multidimensional and multiway data. It consists in splitting data tensor \mathcal{R} , representing the noisy multicomponent image into two-dimensional "slice matrices" of data, each representing a specific channel. According to the classical signal subspace methods [22], the left and right signal subspaces, corresponding to respectively the column and the row vectors of each slice matrix, are simultaneously determined by processing the SVD of the matrix associated with the data of the slice matrix. Let us consider the slice matrix $\mathcal{R}(:,:,i_3, \dots, i_j, \dots, i_N)$ of data tensor \mathcal{R} . Projectors \mathbf{P} on the left signal subspace, and \mathbf{Q} on the right signal subspace are built from respectively the left and the right singular vectors associated with the K largest singular values of $\mathcal{R}(:,:,i_3, \dots, i_j, \dots, i_N)$. The parameter K simultaneously defines the dimensions of the left and right signal subspaces. Applying the projectors \mathbf{P} and \mathbf{Q} on the slice $\mathcal{R}(:,:,i_3, \dots, i_j, \dots, i_N)$ amounts to compute its best lower rank- K matrix approximation [43] in the least-squares sense.

The filtering of each slice matrix of data tensor \mathcal{R} separately is called in the following "channel-by-channel" SVD-based filtering of \mathcal{R} . It consists of a first way to estimate the signal tensor \mathcal{X} and can be summarized by the following steps :

1. input : data tensor \mathcal{R} , left and right signal subspace dimension K .

```

for  $i_N = 1$  to  $I_N$  :
for  $i_{N-1} = 1$  to  $I_{N-1}$  :
:
for  $i_4 = 1$  to  $I_4$  :
for  $i_3 = 1$  to  $I_3$  :
```

(a) calculate matrix $\mathcal{R}(:,:,i_3, \dots, i_j, \dots, i_N)$ SVD :

$$\mathcal{R}(:,:,i_3, \dots, i_j, \dots, i_N) = \mathbf{U} \cdot \boldsymbol{\Sigma} \cdot \mathbf{V}^T,$$

where $\boldsymbol{\Sigma}$ is the core matrix regrouping the singular values of the matrix $\mathcal{R}(:,:,i_3, \dots, i_j, \dots, i_N)$, and $\mathbf{U} = [\mathbf{u}_1 \dots \mathbf{u}_{I_1}]$ and $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_{I_2}]$ are the matrices containing the left and right singular vectors defined respectively by \mathbf{u}_{i_1} and \mathbf{v}_{i_2} .

- (b) construct matrices $\mathbf{U}_K = [\mathbf{u}_1 \dots \mathbf{u}_K]$ and $\mathbf{V}_K = [\mathbf{v}_1 \dots \mathbf{v}_K]$ containing the K largest left and right eigenvectors of $\mathcal{R}(:,:,i_3, \dots, i_j, \dots, i_N)$;
- (c) compute the projector $\mathbf{P} = \mathbf{U}_K \mathbf{U}_K^T$ on the column signal subspace, and projector $\mathbf{Q} = \mathbf{V}_K \mathbf{V}_K^T$ on the row signal subspace.

(d) compute the two-dimensional slice matrices of the estimated expected signal $\hat{\mathcal{X}}$:

$$\hat{\mathcal{X}}(:,:,i_3,\dots,i_j,\dots,i_N) = \mathbf{P}\mathcal{R}(:,:,i_3,\dots,i_j,\dots,i_N)\mathbf{Q}$$

2. output : estimated expected signal : $\hat{\mathcal{X}}$.

Channel-by-channel SVD-based filtering is based on a common efficient method, but exhibits a major drawback : it does not take into account the relationships between the components of the processed tensor. Moreover, channel-by-channel SVD-based filtering is appropriate only on some conditions. For example, applying SVD-based filtering to an image is generally appropriate when the rows or columns of an image are redundant, that is, linearly dependent. In this case, the rank K of the image is equal to the number of linearly independent rows or columns. It is only in this case that it would be safe to throw out eigenvectors from $K + 1$ on. It is only in this special case that the noise subspace is orthogonal to the signal subspace. Otherwise, the noise simply increases the variance of the signal subspace and underestimating the signal subspace dimension would result in throwing out both signal and noise information. Thus, one would lose spatial resolution.

The next subsection presents a multiway filtering method that processes jointly, and not successively, each component of the data tensor.

5.3.2 Tensor filtering based on multimode PCA

5.3.2.1 White decorrelated Gaussian noise and second order statistics based method

Assuming that the dimension K_n of the signal subspace is known for all $n = 1$ to N , one way to estimate the expected signal tensor \mathcal{X} from the noisy data tensor $\mathcal{R} = \mathcal{X} + \mathcal{N}$, is to orthogonally project, for every n th-mode, the vectors of tensor \mathcal{R} on the n th-mode signal subspace $E_1^{(n)}$, for all $n = 1$ to N . This statement is equivalent to replace in (5.4) the filters $\mathbf{H}^{(n)}$ by the projectors $\mathbf{P}^{(n)}$ on the n th-mode signal subspace :

$$\hat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{P}^{(1)} \times_2 \cdots \times_N \mathbf{P}^{(N)}. \quad (5.5)$$

In this last formulation, projectors $\mathbf{P}^{(n)}$ are estimated thanks to a multimode PCA applied to data tensor \mathcal{R} . This multimode PCA-based filtering generalizes the classical matrix filtering methods [46, 49, 58, 60, 62, 82], and implicitly supposes that the additive noise is *white* and *Gaussian*.

In the vector or matrix formulation, the definition of the projector on the signal subspace is based on the eigenvectors associated with the largest eigenvalues of the covariance matrix of the set of observation vectors. Hence, the determination of the signal subspace amounts to determine the best approximation (in the least-squares sense) of the observation matrix or the covariance matrix.

As an extension to the vector and matrix cases, in the tensor formulation, the projectors on the n th-mode vector spaces are determined by computing the rank- (K_1, \dots, K_N) approximation of \mathcal{R} in the least-squares sense. From a mathematical point of view, the rank- (K_1, \dots, K_N)

approximation of \mathcal{R} is represented by tensor $\mathcal{R}^{K_1, \dots, K_N}$ which minimizes the quadratic tensor Frobenius norm $\|\mathcal{R} - \mathcal{B}\|^2$ subject to the condition that $\mathcal{B} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is a rank- (K_1, \dots, K_N) tensor. The description of TUCKALS3 algorithm, used in rank- (K_1, \dots, K_N) approximation is provided in subsection A.1.1.

A good approximation of the rank- (K_1, \dots, K_N) approximation can simply be achieved by computing the HOSVD- (K_1, \dots, K_N) of tensor \mathcal{R} [41, 92]. Indeed, the HOSVD- (K_1, \dots, K_N) of \mathcal{R} consists of the initialization step of TUCKALS3 algorithm, and hence can be considered as a suboptimal solution for the rank- (K_1, \dots, K_N) approximation of tensor \mathcal{R} [41]. This HOSVD-based technique has recently been used in [97] for denoising and source separation of multicomponent seismic waves.

5.3.2.2 Correlated Gaussian noise and higher-order statistics based method

In practice, the condition of noise whiteness is not always fulfilled. Hence, in the case of an additive *correlated* Gaussian noise, TUCKALS3 algorithm is theoretically incapable of providing a good estimation of the n th-mode signal subspaces since it is based on second order moments. A classical means to remove the Gaussian (noise) components is to use the higher-order statistics, and especially the higher-order cumulants. The tensor framework has been used to compute the fourth order cumulants as a means of solving the ICA problem [39]. Vasilescu and Terzopoulos introduced a multilinear ICA (N-mode ICA) for face recognition, which encodes the higher-order statistics associated with each mode of the tensor [128, 129]. The related methods are based on the well-known cumulant property stating that the higher-order cumulants of a Gaussian variable are null [80, 89].

As a consequence, in the case of an additive *correlated* Gaussian noise, a recent study [97] has proposed to improve the multimode PCA-based filtering, by incorporating into TUCKALS3 algorithm the fourth order cumulants instead of the second order moments.

From a practical point of view, second order matrices $\mathbf{C}^{(n),0}$ and $\mathbf{C}^{(n),k}$ at steps 2b and 3(a)iii of TUCKALS3 algorithm are replaced with the corresponding fourth order cumulants. In the following, we only present the details of the procedure for matrix $\mathbf{C}^{(n),k}$. Obtaining the details concerning $\mathbf{C}^{(n),0}$ is straightforward.

We assume that $\{\mathbf{r}_p^{(n)}, p = 1, \dots, M_n\}$, and $\{\mathbf{b}_p^{(n),k}, p = 1, \dots, M_n\}$ are the M_n realizations of two random vectors $\mathbf{r}^{(n)}$ and $\mathbf{b}^{(n),k}$. In practice, we take as the realizations of these two random vectors the n th-mode vectors of data tensors \mathcal{R} and $\mathcal{B}^{(n),k}$. Matrix $\mathbf{C}^{(n),k}$ reads :

$$\mathbf{C}^{(n),k} = \sum_{p=1}^{M_n} \mathbf{b}_p^{(n),k} \mathbf{r}_p^{(n)T}. \quad (5.6)$$

The fourth order cumulants associated with vectors $\mathbf{r}^{(n)}$ and $\mathbf{b}^{(n),k}$ are denoted by :

$$\mathcal{C}^{(n),k} = \text{Cum}(\mathbf{b}^{(n),k}, \mathbf{b}^{(n),kT}, \mathbf{r}^{(n)}, \mathbf{r}^{(n)T}), \quad (5.7)$$

where $\text{Cum}(\cdot)$ denotes the cumulant operator. $\mathcal{C}^{(n),k}$ is a fourth order super-symmetric tensor from $\mathbb{R}^{I_n \times I_n \times I_n \times I_n}$, whose generic term for indexes (i_1, i_2, j_1, j_2) , for centered variables, is given by [52, 80] :

$$\begin{aligned} (\mathcal{C}^{(n),k})_{i_1, i_2, j_1, j_2} &= E[b_{i_1}^{(n),k} b_{i_2}^{(n),k} r_{j_1}^{(n)} r_{j_2}^{(n)}] \\ &- E[b_{i_1}^{(n),k} r_{j_1}^{(n)}] E[b_{i_2}^{(n),k} r_{j_2}^{(n)}] - E[b_{i_1}^{(n),k} r_{j_2}^{(n)}] E[b_{i_2}^{(n),k} r_{j_1}^{(n)}] \end{aligned} \quad (5.8)$$

where $b_i^{(n),k}$ and $r_j^{(n)}$ are the i^{th} and j^{th} components of random vectors $\mathbf{b}^{(n),k}$ and $\mathbf{r}^{(n)}$, and $E[\cdot]$ is the expectation operator. The practical estimation of $(\mathcal{C}^{(n),k})_{i_1,i_2,j_1,j_2}$ is given by :

$$\begin{aligned} (\mathcal{C}^{(n),k})_{i_1,i_2,j_1,j_2} &= \frac{1}{M_n} \left(\sum_{p=1}^{M_n} \left(b_{i_1 p}^{(n),k} b_{i_2 p}^{(n),k} r_{j_1 p}^{(n)} r_{j_2 p}^{(n)} \right) \right) \\ &\quad - \frac{1}{M_n^2} \left(\sum_{p=1}^{M_n} \left(b_{i_1 p}^{(n),k} r_{j_1 p}^{(n)} \right) \right) \left(\sum_{p=1}^{M_n} \left(b_{i_2 p}^{(n),k} r_{j_2 p}^{(n)} \right) \right) \\ &\quad - \frac{1}{M_n^2} \left(\sum_{p=1}^{M_n} \left(b_{i_1 p}^{(n),k} r_{j_2 p}^{(n)} \right) \right) \left(\sum_{p=1}^{M_n} \left(b_{i_2 p}^{(n),k} r_{j_1 p}^{(n)} \right) \right). \end{aligned} \quad (5.9)$$

Here, $b_{ij}^{(n),k}$ and $r_{ij}^{(n)}$ are the elements at position (i, j) of tensors $\mathcal{B}^{(n),k}$ and \mathcal{R} n^{th} -mode flattening matrices $\mathbf{B}_n^{(n),k}$ and \mathbf{R}_n .

In the classical TUCKALS3 algorithm, the K_n n^{th} -mode signal subspace basis vectors, given by matrix $\mathbf{U}^{(n),k}$, are estimated by computing, at step 3a, the eigenvectors associated with the K_n largest eigenvalues of matrix $\mathbf{C}^{(n),k}$. This amounts to computing the best lower rank- K_n approximation of $\mathbf{C}^{(n),k}$. In [110] fourth order cumulants are used instead of the covariance matrix because of their ability to remove Gaussian noise. Indeed, the fourth order cumulants of Gaussian variables are null. Therefore, when dealing with an additive *correlated* Gaussian noise, we also use fourth order cumulants [97].

The main drawback of fourth order cumulants is the high computational load to build every fourth order cumulant tensor associated with the n^{th} -mode of the data tensor. This computational load depends on the size of the data tensor \mathcal{R} , that is, the values of I_n , for all $n = 1$ to N . One way to reduce the computational load has recently been proposed in [97] and consists in using the fourth order cumulant slice matrix. The cumulant slice matrix has initially been introduced in array processing for source localisation or directions-of-arrival (DOA) estimation [21, 139, 138]. In [52, 139, 138], it is proven that the signal subspace spanned by the eigenvectors associated with the largest eigenvalues of a cumulant slice matrix is the same as signal subspace obtained from the whole cumulant tensor defined in (5.7) [139, 138]. Therefore, we use only the eigenvectors of one cumulant slice matrix in our algorithm (see step 2(a)iii) because the other cumulant slice matrices provide redundant information. The use of the fourth order cumulant slice matrix provides a much faster algorithm [138]. In our application, the fourth order cumulant slice matrix $\mathbf{C}_q^{(n),k}$ can be defined, from (5.8), by fixing the q^{th} component of vector $\mathbf{b}^{(n),k}$ as follows :

$$\left(\mathbf{C}_q^{(n),k} \right)_{ij} = E \left[(b_q^{(n),k})^2 r_i^{(n)} r_j^{(n)} \right] - 2E \left[b_q^{(n),k} r_i^{(n)} \right] E \left[b_q^{(n),k} r_j^{(n)} \right]. \quad (5.10)$$

The practical estimation of $\left(\mathbf{C}_q^{(n),k} \right)_{ij}$ can be given by :

$$\left(\mathbf{C}_q^{(n),k} \right)_{ij} = \frac{1}{M_n} \left(\sum_{p=1}^{M_n} \left(b_{qp}^{(n),k} \right)^2 r_{ip}^{(n)} r_{jp}^{(n)} \right) - \frac{2}{M_n^2} \left(\sum_{p=1}^{M_n} b_{qp}^{(n),k} r_{ip}^{(n)} \right) \left(\sum_{p=1}^{M_n} b_{qp}^{(n),k} r_{jp}^{(n)} \right), \quad (5.11)$$

where $b_{ij}^{(n),k}$ and $r_{ij}^{(n)}$ are respectively the elements at position (i, j) in the n^{th} -mode flattening matrices $\mathbf{B}_n^{(n),k}$ and \mathbf{R}_n of tensors $\mathcal{B}^{(n),k}$ and \mathcal{R} .

As a consequence, in the case of an additive *correlated* Gaussian noise, the K_n n^{th} -mode signal subspace basis vectors can now be estimated by computing matrix $\mathbf{C}_q^{(n),k}$ lower rank- K_n approximation. Then, the fourth order cumulant slice matrix-based multimode PCA-based filtering is summarized in subsection A.1.2.

It was experimentally shown in [97], that when the parameter q involved in $\mathbf{C}_q^{(n),k}$ is chosen properly, multimode PCA filtering based on fourth order cumulants (denoted by rank- $\mathcal{C}(K_1, \dots, K_N)$) and on fourth order cumulant slice matrix (denoted by rank- $\mathbf{C}_1(K_1, \dots, K_N)$) give sensibly the same performances in regard to noise reduction in color images and multicomponent seismic waves.

5.3.3 Multiway Wiener filtering

Let \mathbf{R}_n , \mathbf{X}_n and \mathbf{N}_n be the n th-mode flattening matrices of tensors \mathcal{R} , \mathcal{X} and \mathcal{N} , respectively.

In the previous subsection, the estimation of signal tensor \mathcal{X} has been performed by projecting noisy data tensor \mathcal{R} on each n th-mode signal subspace. The n th-mode projectors have been estimated thanks to the use of multimode PCA achieved by rank- (K_1, \dots, K_N) approximation. In spite of the good results given by this method, it is possible to improve the tensor filtering quality by determining n th-mode filters $\mathbf{H}^{(n)}$, $n = 1$ to N , in (5.4), which optimize an estimation criterion. The most classical method is to minimize the mean squared error between the expected signal tensor \mathcal{X} and the estimated signal tensor $\hat{\mathcal{X}}$ given in (5.4) :

$$e(\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N)}) = E[\|\mathcal{X} - \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)}\|^2]. \quad (5.12)$$

Due to the criterion which is minimized, filters $\mathbf{H}^{(n)}$, $n = 1$ to N , can be called " n th-mode Wiener filters" [96].

According to the calculations presented in Appendix A.2, especially from (A.1) to (A.15), the minimization of (5.12) with respect to filter $\mathbf{H}^{(n)}$, for fixed $\mathbf{H}^{(m)}$, $m \neq n$, leads to the following expression of n th-mode Wiener filter :

$$\mathbf{H}^{(n)} = \gamma_{\mathbf{XR}}^{(n)} \mathbf{\Gamma}_{\mathbf{RR}}^{(n)}^{-1}, \quad (5.13)$$

where

$$\gamma_{\mathbf{XR}}^{(n)} = E \left[\mathbf{X}_n \mathbf{T}^{(n)} \mathbf{R}_n^T \right] \quad (5.14)$$

is the $\mathbf{T}^{(n)}$ -weighted covariance matrix between the random column vectors of signal \mathbf{X}_n and data \mathbf{R}_n , with :

$$\mathbf{T}^{(n)} = \mathbf{H}^{(1)} \otimes \cdots \otimes \mathbf{H}^{(n-1)} \otimes \mathbf{H}^{(n+1)} \otimes \cdots \otimes \mathbf{H}^{(N)}, \quad (5.15)$$

where \otimes stands for Kronecker product, and :

$$\mathbf{\Gamma}_{\mathbf{RR}}^{(n)} = E \left[\mathbf{R}_n \mathbf{Q}^{(n)} \mathbf{R}_n^T \right], \quad (5.16)$$

is the $\mathbf{Q}^{(n)}$ -weighted covariance matrix of the data \mathbf{R}_n , with :

$$\mathbf{Q}^{(n)} = \mathbf{T}^{(n)T} \mathbf{T}^{(n)}. \quad (5.17)$$

In order to obtain $\mathbf{H}^{(n)}$ through (5.13), we suppose that the filters $\{\mathbf{H}^{(m)}, m = 1$ to $N, m \neq n\}$ are known. Data tensor \mathcal{R} is available, but signal tensor \mathcal{X} is unknown. So, only the term $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$ can be derived, and not the term $\gamma_{\mathbf{XR}}^{(n)}$. Hence, some more assumptions on \mathcal{X} have to be made in order to overcome the indetermination over $\gamma_{\mathbf{XR}}^{(n)}$ [93, 96]. In the one-dimensional case, a classical assumption is to consider that a signal vector is a weighted combination of

the signal subspace basis vectors. In extension to the tensor case, [93, 96] have proposed to consider that the n th-mode flattening matrix \mathbf{X}_n can be expressed as a weighted combination of K_n vectors from the n th-mode signal subspace $E_1^{(n)}$:

$$\mathbf{X}_n = \mathbf{V}_s^{(n)} \mathbf{O}^{(n)}, \quad (5.18)$$

with $\mathbf{X}_n \in \mathbb{R}^{I_n \times M_n}$, and $\mathbf{V}_s^{(n)} \in \mathbb{R}^{I_n \times K_n}$ being the matrix containing the K_n orthonormal basis vectors of n th-mode signal subspace $E_1^{(n)}$. Matrix $\mathbf{O}^{(n)} \in \mathbb{R}^{K_n \times M_n}$ is a weight matrix and contains the whole information on expected signal tensor \mathcal{X} . This model implies that signal n th-mode flattening matrix \mathbf{X}_n is orthogonal to n th-mode noise flattening matrix \mathbf{N}_n , since signal subspace $E_1^{(n)}$ and noise subspace $E_2^{(n)}$ are supposed mutually orthogonal.

Supposing that noise \mathcal{N} in (5.3) is white, Gaussian and independent from signal \mathcal{X} , and introducing the signal model (5.18) in (5.13) leads to a computable expression of n th-mode Wiener filter $\mathbf{H}^{(n)}$ (see Appendix A.3) :

$$\mathbf{H}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \Lambda_{\mathbf{Rs}}^{(n)-1} \mathbf{V}_s^{(n)T}, \quad (5.19)$$

where $\gamma_{\mathbf{OO}}^{(n)} \Lambda_{\mathbf{Rs}}^{(n)-1}$ is a diagonal weight matrix given by :

$$\gamma_{\mathbf{OO}}^{(n)} \Lambda_{\mathbf{Rs}}^{(n)-1} = \text{diag} \left[\frac{\beta_1}{\lambda_1^\Gamma}, \dots, \frac{\beta_{K_n}}{\lambda_{K_n}^\Gamma} \right], \quad (5.20)$$

where $\lambda_1^\Gamma, \dots, \lambda_{K_n}^\Gamma$ are the K_n largest eigenvalues of $\mathbf{Q}^{(n)}$ -weighted covariance matrix $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$ (see (5.16)). Parameters $\beta_1, \dots, \beta_{K_n}$ depend on $\lambda_1^\gamma, \dots, \lambda_{K_n}^\gamma$ which are the K_n largest eigenvalues of $\mathbf{T}^{(n)}$ -weighted covariance matrix

$\gamma_{\mathbf{RR}}^{(n)} = \mathbb{E}[\mathbf{R}_n \mathbf{T}^{(n)} \mathbf{R}_n^T]$, according to the following relation :

$$\beta_{k_n} = \lambda_{k_n}^\gamma - \sigma_\Gamma^{(n)2}, \quad \forall k_n = 1, \dots, K_n \quad (5.21)$$

Superscript γ refers to the $\mathbf{T}^{(n)}$ -weighted covariance, and subscript Γ to the $\mathbf{Q}^{(n)}$ -weighted covariance. $\sigma_\Gamma^{(n)2}$ is the degenerated eigenvalue of noise $\mathbf{T}^{(n)}$ -weighted covariance matrix $\gamma_{\mathbf{NN}}^{(n)} = \mathbb{E}[\mathbf{N}_n \mathbf{T}^{(n)} \mathbf{N}_n^T]$. Thanks to the additive noise and the signal independence assumptions, the $I_n - K_n$ smallest eigenvalues of $\gamma_{\mathbf{RR}}^{(n)}$ are equal to $\sigma_\Gamma^{(n)2}$, and thus, can be estimated by the following relation :

$$\hat{\sigma}_\Gamma^{(n)2} = \frac{1}{I_n - K_n} \sum_{k_n=K_n+1}^{I_n} \lambda_{k_n}^\gamma. \quad (5.22)$$

In order to determine the n th-mode Wiener filters $\mathbf{H}^{(n)}$ that minimize the mean squared error (5.12), the Alternating Least Squares (ALS) algorithm has been proposed in [93, 96]. It is detailed in subsection A.1.3.

In subsection 5.3.2, we presented the adaptation of multimode PCA to the case of a non correlated Gaussian noise, by using higher-order statistics. In the same way, it is possible to use higher-order statistics for multiway Wiener filtering. For this, one should replace step 2aii-B by step 2aii of the ALS loop in subsection 5.3.2, and replace step 2aii-E by the computation of the cumulant slice $\mathbf{C}_q^{(n),k}$ associated with the fourth order cumulants of matrix $\mathbf{R}_n^{(n),k}$ and matrix $(\mathbf{R}_n^{(n),k})^T$. Elements of $\mathbf{C}_q^{(n),k}$ are given in (5.11).

5.4 Simulation results

In the following simulations, the channel-by-channel SVD-based filtering defined in subsection 5.3.1 and the rank- (K_1, \dots, K_N) approximation based multiway and multidimensional filtering are applied to the denoising of color images and multispectral images, and to the denoising of seismic signals. Color images, multispectral images, and seismic signals can be represented by a third order tensor from $\mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1 , I_2 , and I_3 take different values. In all these applications, the efficiency of denoising is tested in the presence of an additive Gaussian noise, either correlated or not.

A multidimensional and multiway white Gaussian noise \mathcal{N} which is added to signal tensor \mathcal{X} can be expressed as :

$$\mathcal{N} = \alpha \cdot \mathcal{G}, \quad (5.23)$$

where every element of $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is an independent realization of a normalized centered Gaussian law, and where α is a coefficient that permits to set the SNR in noisy data tensor \mathcal{R} .

When we process images impaired by correlated Gaussian noise, the noise which is added is a third order tensor defined by :

$$\mathcal{N}^c = \mathcal{N} \times_1 \mathbf{W}^{(1)} \times_2 \mathbf{W}^{(2)} \times_3 \mathbf{W}^{(3)}, \quad (5.24)$$

where every element of \mathcal{N} represents an independent realization of a white Gaussian noise, and $\mathbf{W}^{(n)}$ is a weight matrix in the n th-mode, $n = 1, 2, 3$.

In order to evaluate the performances of the overviewed tensor signal processing methods, a particular performance criterion is employed as proposed in [96, 97].

5.4.1 Performance criterion

Following the representation of (5.3), the multiway noisy data tensor is expressed as $\mathcal{R} = \mathcal{X} + \mathcal{N}$, where \mathcal{X} is the expected signal tensor and \mathcal{N} is the additive noise tensor. Let us define the Signal to Noise Ratio (SNR, in dB) in the noisy data tensor by :

$$SNR = 10 \log\left(\frac{\|\mathcal{X}\|^2}{\|\mathcal{N}\|^2}\right). \quad (5.25)$$

In order to *a posteriori* verify the quality of the estimated signal tensor, we use the Normalized Quadratic Error criterion (NQE) defined as follows :

$$NQE(\hat{\mathcal{X}}) = \frac{\|\hat{\mathcal{X}} - \mathcal{X}\|^2}{\|\mathcal{X}\|^2}. \quad (5.26)$$

The NQE criterion permits a quantitative comparison of the channel-by-channel SVD-based filtering and the rank- (K_1, K_2, K_3) approximation multiway and multidimensional filtering. Considering this criterion, we expect the rank- (K_1, K_2, K_3) approximation to give better results than the channel-by-channel SVD-based filtering method.

5.4.2 Denoising of color images

Denoising of color images has already been studied in several works [18, 99, 116]. Some solutions have been brought from the field of wavelet processing, exhibiting good results in terms of output SNR. These studies only concern bidimensional data, whereas the methods that we compare are adapted to the processing of third order tensors as a whole, and in particular to three-channel images. We focus on subspace-based methods. We first consider the channel-by-channel SVD-based filtering, the rank- (K_1, K_2, K_3) approximation and multi-way Wiener filtering (Wmm- (K_1, K_2, K_3)), applied to images impaired by an additive white Gaussian noise.

Then we present the results obtained with rank- (K_1, K_2, K_3) based on second order and higher-order statistics, applied to images impaired by an additive correlated Gaussian noise. We compare the performances of the methods applied in this subsection in terms of denoising efficiency and computational load.

5.4.2.1 Denoising of a color image impaired by additive Gaussian noise

Let us consider the "Sailboat" standard color image of Fig. 5.3(a) represented as a third order tensor $\mathcal{X} \in \mathbb{R}^{256 \times 256 \times 3}$. The ranks of the signal subspace for each mode are 30 for the 1st-mode, 30 for the 2nd-mode, and 2 for the 3rd-mode. This is fixed thanks to the following process : For Fig. 5.3(a), we took the standard non-noisy "sailboat" image and we artificially reduced the ranks of the non-noisy image, that is, we set the parameters (K_1, K_2, K_3) to $(30, 30, 2)$, thanks to the truncation of HOSVD. This permits to ensure that, for each mode, the rank of the signal subspace is lower than the corresponding dimension. This also permits to evaluate the performances of the filtering methods applied, independently from the accuracy of the estimation of the values of the ranks by MDL or AIC criterion.

Fig. 5.3(b) shows the noisy image resulting from the impairment of Fig. 5.3(a) and represented as $\mathcal{R} = \mathcal{X} + \mathcal{N}$. Third-order noise tensor \mathcal{N} is defined by relation (5.23) by choosing α such that, considering previous definition of (5.25), the SNR in the noisy image of Fig. 5.3(b) is 8.1dB. In these simulations, the value of the parameter K of channel-by-channel SVD-based filtering, the values of the dimensions of the row and column signal subspace are supposed to be known and fixed to 30. In the same way, parameters (K_1, K_2, K_3) of rank- (K_1, K_2, K_3) approximation are fixed to $(30, 30, 2)$.

The channel-by-channel SVD-based filtering of noisy image \mathcal{R} (see Fig. 5.3(b)) yields the image of Fig. 5.3(c), and rank- $(30, 30, 2)$ approximation of noisy data tensor \mathcal{R} yields the image of Fig. 5.3(d). The NQE, defined in (5.26), permits a quantitative comparison between channel-by-channel SVD-based filtering and rank- $(30, 30, 2)$ approximation. Figure 5.4, which presents the evolution of the NQE with respect to SNR varying from 3dB to 18dB, shows the NQE obtained with Wmm- $(30, 30, 2)$ is lower than the NQE obtained with the filtering with rank- $(30, 30, 2)$ approximation. For this simulation, the rank- (K_1, K_2, K_3) approximation gives better results than channel-by-channel SVD-based filtering according to the NQE criterion. From the resulting image, presented on Fig. 5.3(d), we notice that dimension reduction leads to a loss of spatial resolution. However the choice of a set of values K_1, K_2, K_3 which are small enough is the condition for an efficient noise reduction effect.

Therefore, a tradeoff should be considered between noise reduction and detail preservation. This tradeoff was discussed in [112]. We were interested in using the Minimum Description Length criterion [132], applied to the left singular values of the flattening matrices computed

over the successive n th-modes. As a rule of thumb, the MDL criterion overestimates the value of parameters K_1 , K_2 , and K_3 . This results in the preservation of the details in the processed image, at the expense of an efficient denoising.

Concerning the qualitative results obtained with this color image, we notice that the intra-class variance of the pixel values of each component (or color mode) of the resulting image is lower for the image obtained with Wmm-(30, 30, 2) than for those images obtained with other methods applied in this subsection. This permits, for example, to apply after denoising a high level classification method with a higher efficiency than when classification is applied after channel-by-channel SVD-based filtering or HOSVD-(30, 30, 2).

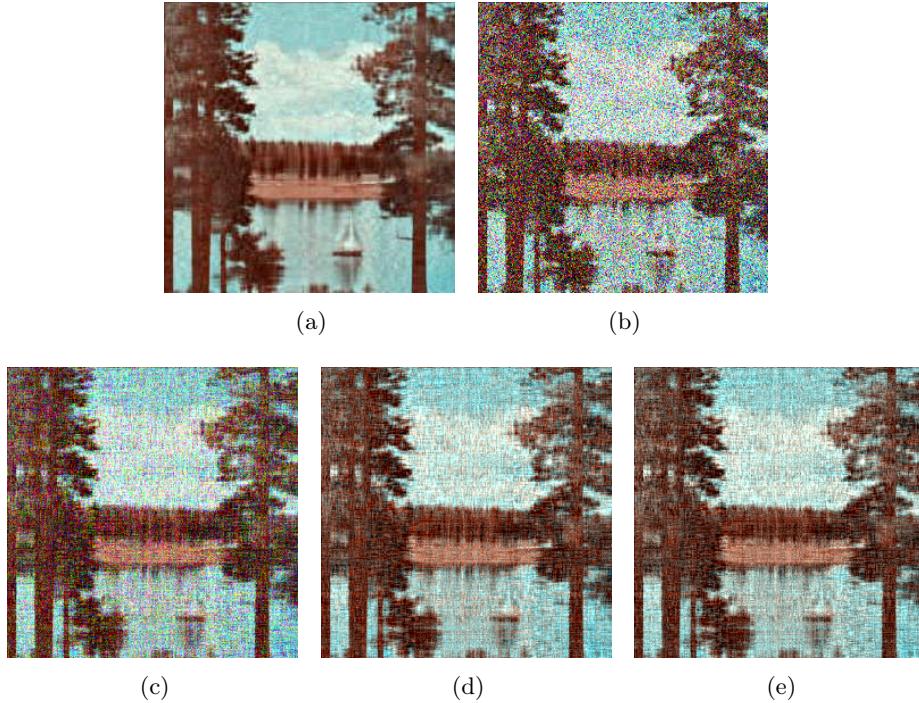


Figure 5.3 — (a) non-noisy image. (b) image to be processed, impaired by an additive white Gaussian noise, with SNR=8.1dB. (c) channel-by-channel SVD-based filtering of parameter $K=30$. (d) rank-(30, 30, 2) approximation. (e) Wmm-(30, 30, 2) filtering.

For the $256 \times 256 \times 3$ Sailboat image of Fig. 5.3, the computational times needed when *Matlab*[®] programs are used on a 3Ghz Pentium 4 processor running Windows are as follows. HOSVD-(30, 30, 2) lasts 1.61 sec., the channel-by-channel SVD-based filtering lasts 1.94 sec., the rank-(30, 30, 2) approximation run with 25 iterations lasts 54.1 sec. and Wmm-(30, 30, 2) run with 25 iterations lasts 40.0 sec. The results presented in Fig. 5.3 show that Wmm- (K_1, K_2, K_3) permits to obtain better results in terms of NQE, with a computational load which is lower than that of the rank- (K_1, K_2, K_3) approximation.

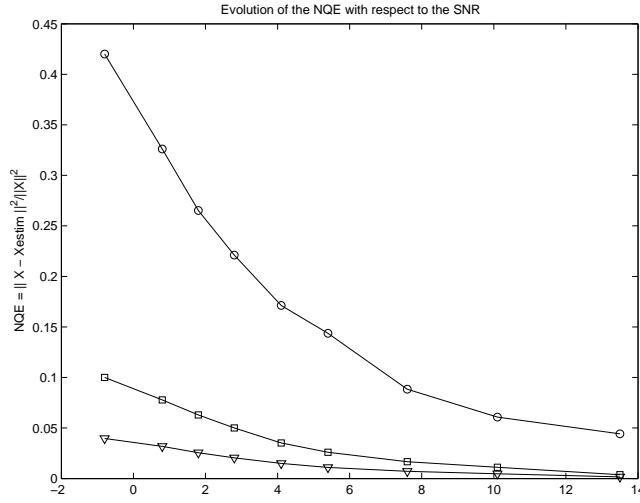


Figure 5.4 — NQE evolution with respect to SNR (dB) : channel-by-channel SVD-based filtering of parameter 30 (-o-), rank-(30, 30, 2) approximation (-□-), Wmm-(30, 30, 2) filtering (-▽-).

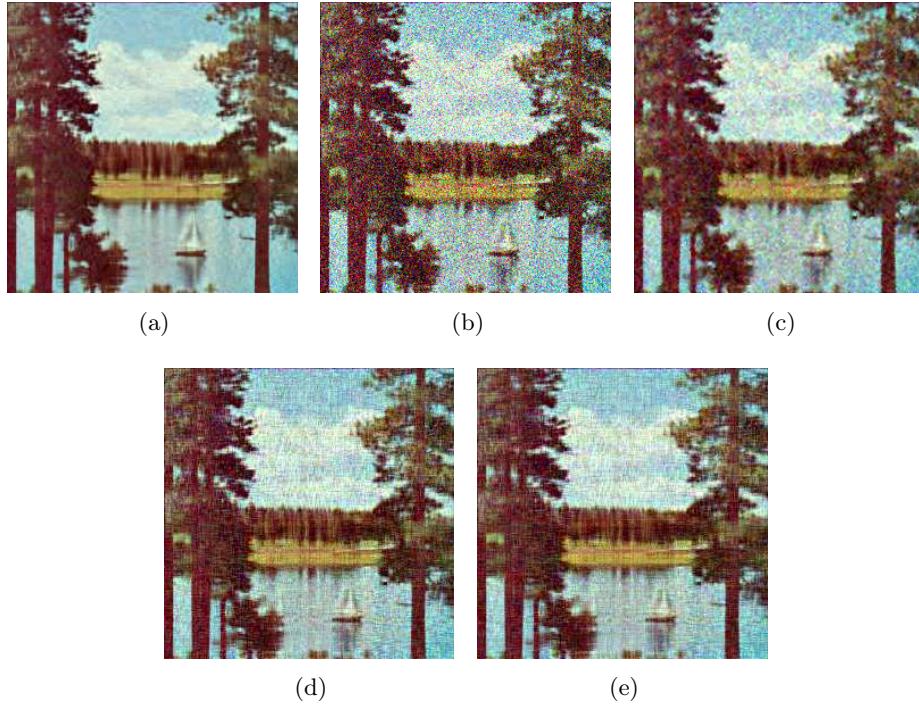


Figure 5.5 — (a) non-noisy image. (b) image to be processed, impaired by an additive white Gaussian noise, with SNR=8.1dB. (c) channel-by-channel Wiener-based filtering of parameter K=30. (d) rank-(30, 30, 3) approximation. (e) Wmm-(30, 30, 3) filtering.

In the next two examples we study the influence of the values of the n th-mode ranks. In the example of Fig. 5.5 we set, in the same way as in the previous example, the ranks

of the truncated image to $(30, 30, 3)$ (see Fig. 5.5(a)). Note that $K_3 = I_3 = 3$. Thus the assumption $K_3 < I_3$ is not fulfilled. We aim at studying the behavior of the proposed tensor filtering algorithms when the color mode rank is equal to the color mode dimension ($K_3 = I_3$). The truncated image is impaired by a non correlated Gaussian noise such that SNR=8.1dB (see Fig. 5.5(b)). The results obtained show that channel-by-channel Wiener-based filtering of parameter $K = 30$ (see Fig. 5.5(c)) is outperformed by rank-(30, 30, 3) approximation (see Fig. 5.5(d)) and Wmm-(30, 30, 3) (see Fig. 5.5(e)). Indeed, the proposed tensor filtering algorithms rely on an ALS loop which permits to take into account the relationships between the filters of each mode when multiway filters are used. In particular, concerning multiway Wiener filtering, it can be adapted to the case where it is applied with $K_3 = I_3$. For this, the weight matrix $\gamma_{\mathbf{O}\mathbf{O}}^{(3)} \boldsymbol{\Lambda}_{\mathbf{F}\mathbf{s}}^{(3)-1}$ of step 2(a)iiH of the multiway Wiener filtering algorithm presented in subsection 5.3.3 is set to identity. That is, $\mathbf{H}^{(3)}$ is replaced by $\mathbf{P}^{(3)}$. We adapted the algorithm in order to take into account the channel mode information for the computation of the two spatial filters thanks to the ALS loop.

This proves the interest of multiway filtering even in the case where the rank of the signal subspace along the third mode is equal to the number of channels.

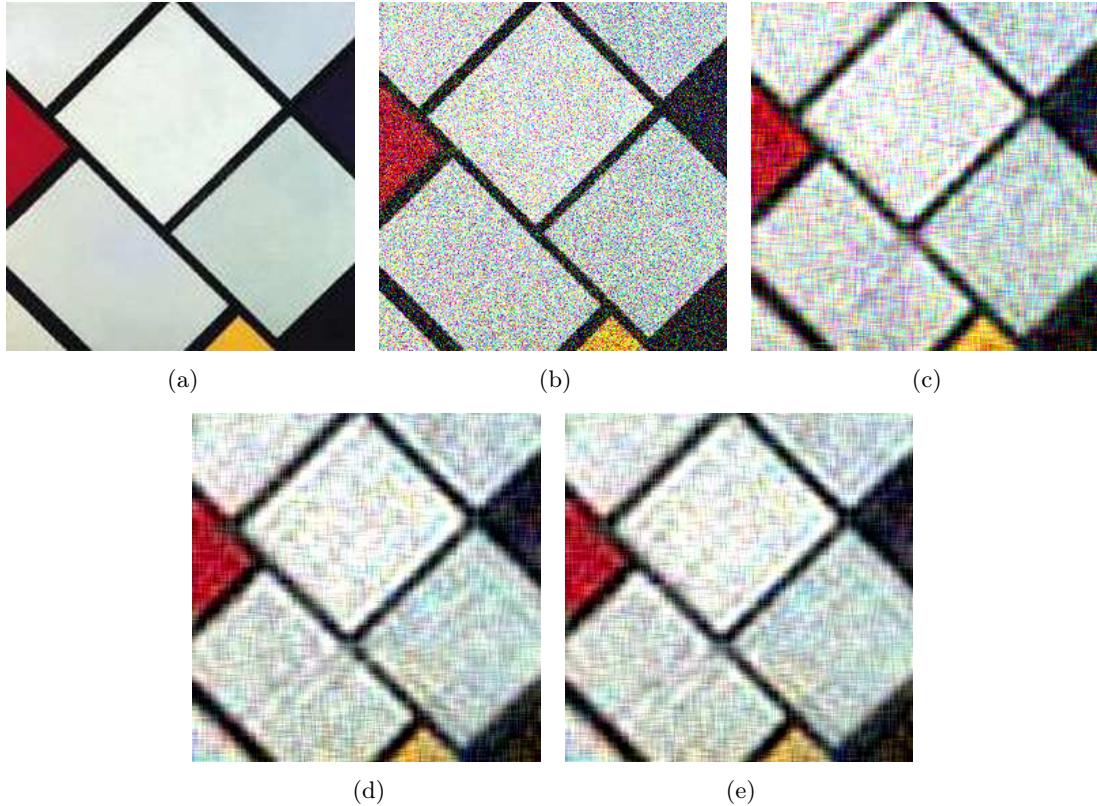


Figure 5.6 — (a) non-noisy image. (b) image to be processed, impaired by an additive white Gaussian noise, with SNR=8.0dB. (c) channel-by-channel SVD-based filtering of parameter $K=19$. (d) rank-(19, 19, 3) approximation. (e) Wmm-(19, 19, 3) filtering.

In the example of Fig. 5.6 we study the case where the ranks of the signal subspaces are underestimated for the spatial modes. Let us consider the "Mondriaan" standard color

image of Fig. 5.6 represented as a third order tensor $\mathcal{X} \in \mathbb{R}^{256 \times 256 \times 3}$. We set the ranks of the truncated image to $(150, 150, 3)$. The ranks along the spatial modes will be fixed intentionally to a value which is smaller than 150 when the reviewed methods are applied. Fig. 5.6(a) gives the non-noisy image, Fig. 5.6(b) shows the noisy image resulting from the impairment, with SNR=8.0dB, of the image of Fig. 5.6(a). Fig. 5.6(c) gives the result obtained with channel-by-channel SVD-based filtering of parameter $K = 19$. Fig. 5.6(d) gives the result obtained with rank-(19, 19, 3) approximation, Fig. 5.6(e) gives the result obtained with Wmm-(19, 19, 3) filtering. Note that choosing $(K_1, K_2, K_3) = (19, 19, 3)$ results in throwing out both signal and noise information along the spatial modes, as the ranks of the noisy image are $(150, 150, 3)$. Underestimating the ranks along the spatial modes induces some blurry effect in the result images : part of the spatial resolution is lost. The presented subspace-based algorithms perform well if there is a high level of redundancy in the column or row space or if the image exhibits many soft or blurry edges, and the n th-mode ranks are not underestimated.

5.4.2.2 HOSVD- (K_1, K_2, K_3) , rank- (K_1, K_2, K_3) approximation based on second order and higher-order statistics, applied to an image impaired by an additive correlated Gaussian noise

The purpose here is to compare methods based on second order statistics with methods based on higher-order statistics when an image is impaired by a correlated Gaussian noise. Fig. 5.7 shows the results obtained with the HOSVD- (K_1, K_2, K_3) , and the rank- (K_1, K_2, K_3) approximation based on second order and higher-order statistics, used for the denoising of an image impaired by an additive correlated Gaussian noise. We consider the non-noisy image of Fig. 5.7(a) whose ranks are fixed to $(30, 30, 2)$: we artificially reduced the ranks of the non-noisy image, that is, we set the parameters (K_1, K_2, K_3) to $(30, 30, 2)$, thanks to the truncation of HOSVD. This image is impaired by a correlated Gaussian noise (see (5.24)). Fig. 5.7(b) shows the noisy image. The result of HOSVD- (K_1, K_2, K_3) is given in Fig. 5.7(c), the result of rank- (K_1, K_2, K_3) approximation based on second order statistics is given in Fig. 5.7(d), the result of rank- $\mathcal{C}(K_1, K_2, K_3)$ approximation based on higher-order statistics is given in Fig. 5.7(e). The evolution of the NQE with respect to the SNR for HOSVD- $(30, 30, 2)$, rank- $\mathcal{C}(30, 30, 2)$ approximation based on fourth order cumulants, rank- $\mathbf{C}_1(30, 30, 2)$ approximation based on one slice of the fourth order cumulants, is represented on Fig. 5.8.

Main conclusions from Fig. 5.7 is that the methods based on fourth order cumulants give similar visual results and better results than HOSVD- $(30, 30, 2)$. Whatever the SNR, the methods based on fourth order cumulants give a lower NQE value than the methods based on second order statistics. The method based on fourth order cumulant slice matrix gives sensibly the same NQE values as the method based on fourth order cumulants.

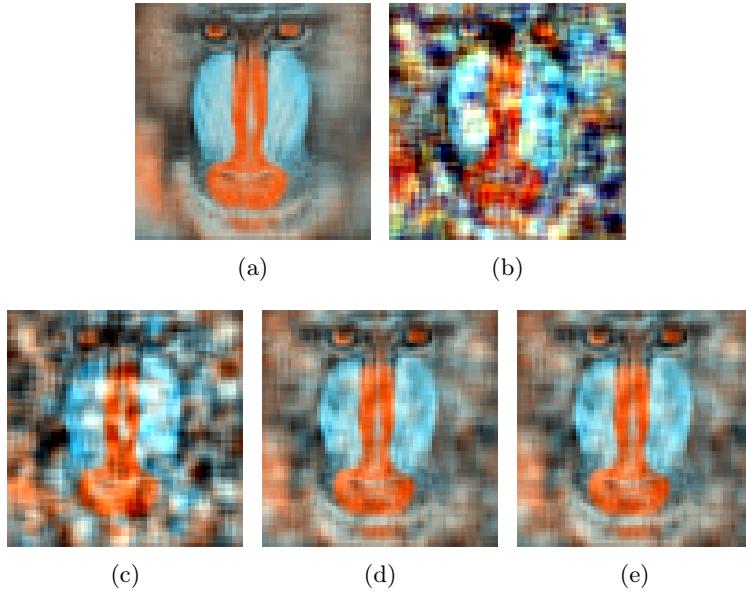


Figure 5.7 — (a) : initial non-noisy image. (b) : initial image with an additive correlated Gaussian noise, $SNR = 2.48dB$. (c) : HOSVD-(30,30,2). (d) : rank- $\mathcal{C}(30,30,2)$ approximation. (e) : rank- $\mathbf{C}_1(30,30,2)$ approximation.

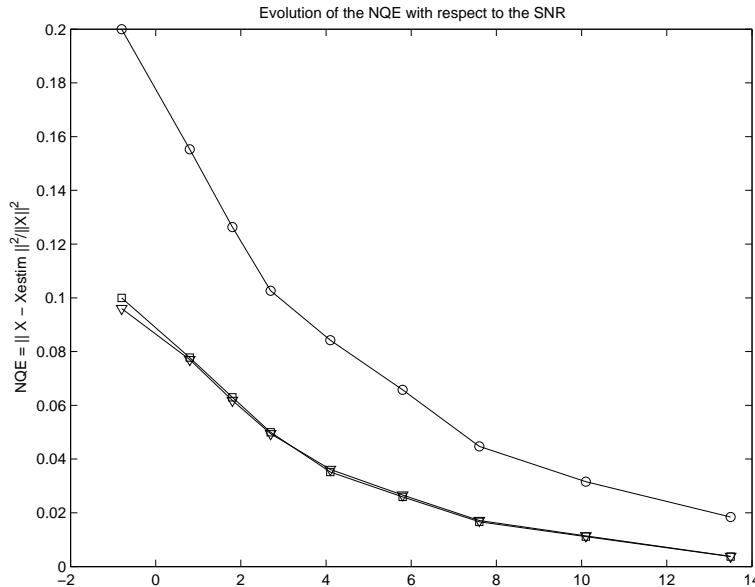


Figure 5.8 — Evolution of the NQE with respect to the SNR(dB) for each tensor filtering method : \circ : HOSVD-(30, 30, 2) ; ∇ : rank- \mathcal{C} (30, 30, 2) ; \square : rank- \mathcal{C}_1 (30, 30, 2).

For the $256 \times 256 \times 3$ Baboon image of Fig. 5.7, the computational times needed in the same conditions of processor and software as in previous subsection are the following : HOSVD-(30,30,2) lasts 1.61 sec., rank- \mathcal{C} (30,30,2) based filtering lasts 2h. 11 min. 40 sec., rank- \mathbf{C}_1 (30,30,2) lasts 3 min. 50 sec.

5.4.3 Denoising of multispectral images

The results obtained from the processing of a multispectral image composed of 72 rows, 160 columns and 100 spectral channels representing a truck are considered. This set of spectral images can be represented as a tensor $\mathcal{X} \in \mathbb{R}^{72 \times 160 \times 100}$. Images shown on Figs. 5.9(a) to 5.9(e) represent channels 30 to 34 of the multispectral image. To evaluate the performances of the reviewed methods, some signal-independent white Gaussian noise \mathcal{N} is added to \mathcal{X} and results in noisy tensor $\mathcal{R} = \mathcal{X} + \mathcal{N}$. Channels 30 to 34 of noisy multispectral image represented as \mathcal{R} are shown on Fig. 5.9(f) to 5.9(j), and correspond to a noise impairment level $SNR = -1\text{dB}$. Figures 5.9(k) to 5.9(o) represent channels 30 to 34 of the multispectral image obtained by applying channel-by-channel-based SVD-filtering to noisy image \mathcal{R} . Finally, Figs. 5.9(p) to 5.9(t) represent channels 30 to 34 of the multispectral image obtained after applying rank-(30,30,30) approximation to noisy image \mathcal{R} . This last simulation clearly shows that the rank-(30,30,30) approximation-based filtering gives better results than channel-by-channel SVD-based filtering in regard to denoising. Moreover, the evolution of the NQE with respect to the SNR varying from -1dB to 15dB , represented on Fig. 5.10 shows that the NQE obtained with Wmm- (K_1, K_2, K_3) is lower than the NQE obtained with a previously existing method.

For this simulation the estimation quality, respect to the NQE criterion, is better for rank- (K_1, K_2, K_3) approximation, compared to channel-by-channel SVD-based filtering. Superiority of rank- (K_1, K_2, K_3) approximation compared to channel-by-channel SVD-based filtering is confirmed.

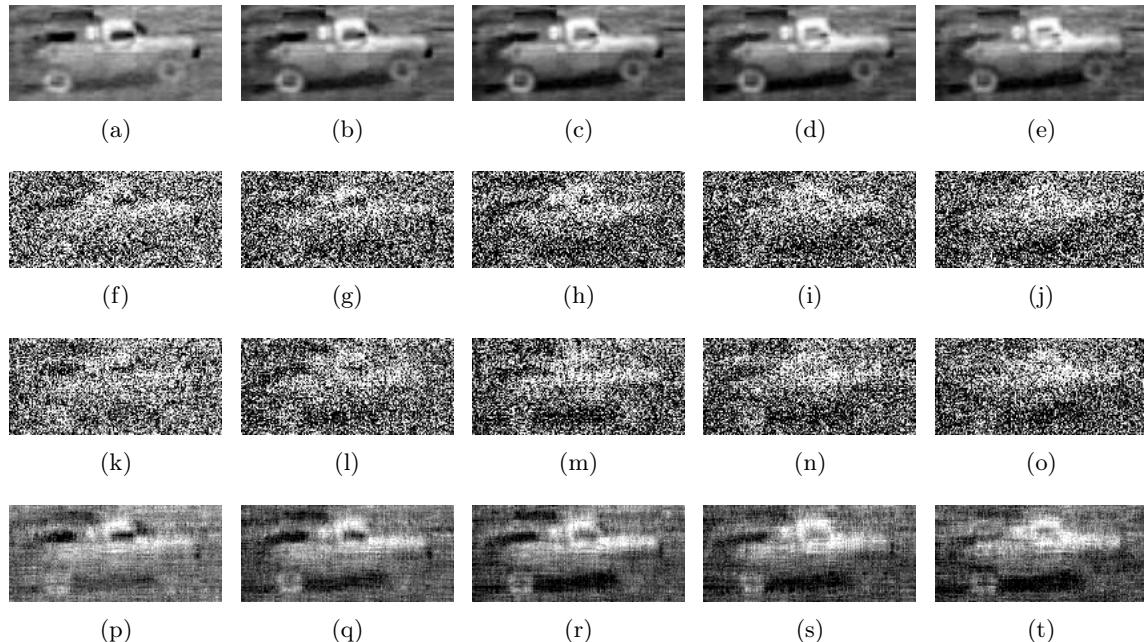


Figure 5.9 — Channels 30 to 34 of the processed multispectral images are presented : (a)-(e) : Non-noisy multispectral image. (f)-(j) : Impaired multispectral image. (k)-(o) : Results obtained with channel-by-channel SVD filtering. (p)-(t) : Results obtained with rank-(30,30,30) approximation.

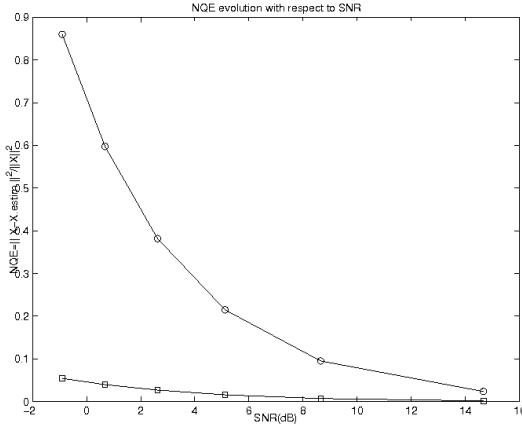


Figure 5.10 — NQE evolution with respect to SNR (from -1 to 15dB) : channel-by-channel SVD-based filtering of parameter 30 (-o-), and rank-(30, 30, 30) approximation (-□-).

According to the simulations performed on a color image and on a multispectral image, it is possible to conclude that the more channels the image is composed of, the better the denoising. This can be explained by a better estimation of projectors on 1st-mode and 2nd-mode signal subspaces in the case of the multispectral image. Indeed, the number of spectral channels in a multispectral image is much larger than in a color image. Equivalently, I_3 is much larger than 3, so M_1 and M_2 are much larger than for a color image, and the estimation of matrices $\mathbf{C}^{(1),k}$ and $\mathbf{C}^{(2),k}$ presented in (5.6) are computed with more realization vectors.

5.4.4 Statistical performances

The goal of the following simulation is to test the robustness to noise of channel-by-channel SVD-based filtering of parameter K and of rank- (K_1, K_2, K_3) approximation, with respect to the NQE criterion. We process "Sailboat" standard color image, impaired by an additive Gaussian noise, with SNR values varying from -0.7dB to 15dB . 100 trials are performed. For each trial one realization of additive Gaussian noise is simulated and added to the non-noisy image. The mean and standard deviation are computed over the NQE values obtained each time the channel-by-channel SVD-based filtering and the rank- (K_1, K_2, K_3) approximation are run. The evolution of the mean NQE :

$$m_{NQE} = \frac{1}{100} \sum_{i=1}^{100} NQE_i, \quad (5.27)$$

where index i refers to the i^{th} noise realization, is represented on Fig. 5.11(a), with respect to SNR. The evolution of the standard deviation of the NQE : $std_{NQE} = \sqrt{\frac{1}{100} \sum_{i=1}^{100} (NQE_i - m_{NQE})^2}$, is represented on Fig. 5.11(b), with respect to the SNR. Fig. 5.11 shows that the mean and standard deviation values of the NQE obtained with rank- (K_1, K_2, K_3) approximation and computed over 100 noise realizations are both lower than the mean and the standard deviation values obtained with channel-by-channel SVD-based filtering. Thus, for these simulations, the rank- (K_1, K_2, K_3) approximation gives better

results than channel-by-channel SVD-based filtering in regard to the robustness of tensor estimation and considering the NQE criterion.

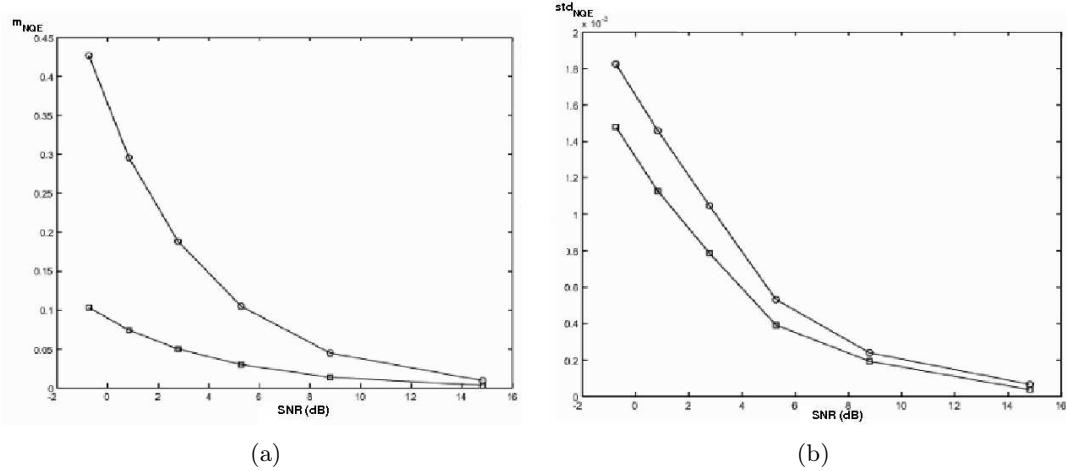


Figure 5.11 — (○) : results obtained with channel-by-channel SVD-based filtering of parameter 30 ; (□) : results obtained with rank-(30, 30, 2) approximation ; (a) : evolution of the mean NQE with respect to SNR (dB) ; (b) evolution of the standard deviation of NQE with respect to SNR (dB).

5.4.5 Filtering of a multicomponent seismic type signal

5.4.5.1 Filtering of a multicomponent seismic type signal impaired by an additive white Gaussian noise

In this simulation, a multicomponent seismic wave is received on a linear antenna composed of 10 sensors. The direction of propagation of the wave is assumed to be contained in a plane which is orthogonal to the antenna. The three components of the wave, represented as signal tensor \mathcal{X} , are called "Component 1", "Component 2" and "Component 3" and are represented in Fig. 5.12(a)-(c). In each seismic slice, the x-axis corresponds to the time sampling (200 or 100 time samples) and the y-axis corresponds to the spatial sensors (10 sensors). Each consecutive component presents a $\frac{\pi}{2}$ radian phase shift. The three components of noisy data tensor \mathcal{R} are represented in Fig. 5.12(d)-(f), where the additive noise is considered as white and Gaussian, and for which the SNR= -10 dB. The classical Wiener filtering of parameter K (Wcc- K) of each component, with a signal subspace dimension fixed to $K = 8$, permits to obtain the results presented in Fig. 5.12(g)-(i). The multimode PCA-based filtering achieved by applying HOSVD-(8,8,3) to noisy data tensor permits to obtain the results presented in Fig. 5.12(j)-(l). Finally, the results obtained with multiway Wiener filtering applied to the noisy data tensor are presented in Fig. 5.12(m)-(o). The evolution of the NQE with respect to the SNR (dB) is given in Fig. 5.13. As well as in the case of color image filtering, in this simulation, the best quality, in terms of noise reduction, is given by multiway Wiener filtering since, for all considered SNR values, the NQE values given by this method are lower than the values given by both HOSVD-(8, 8, 3) and Wcc-8.

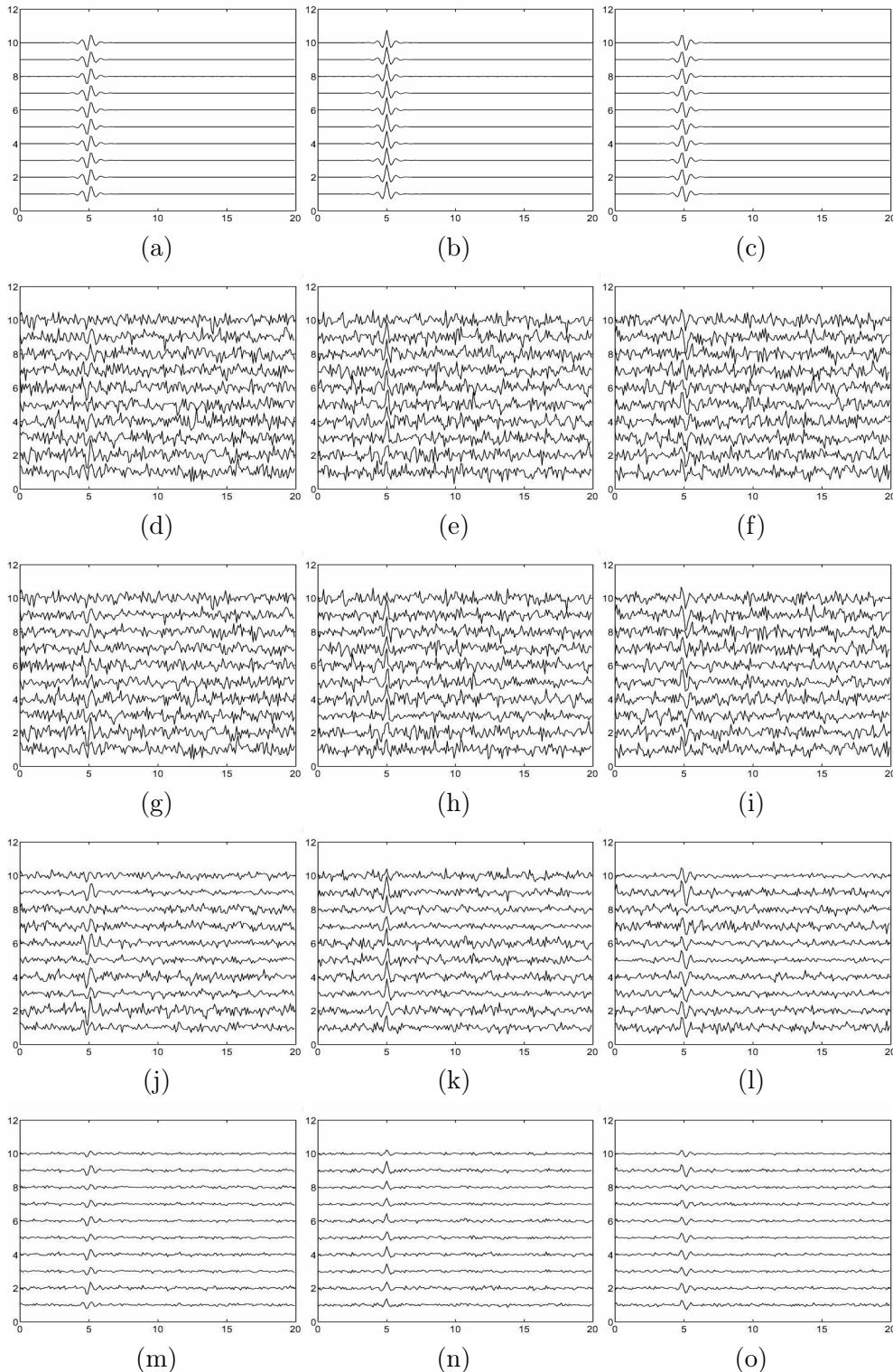
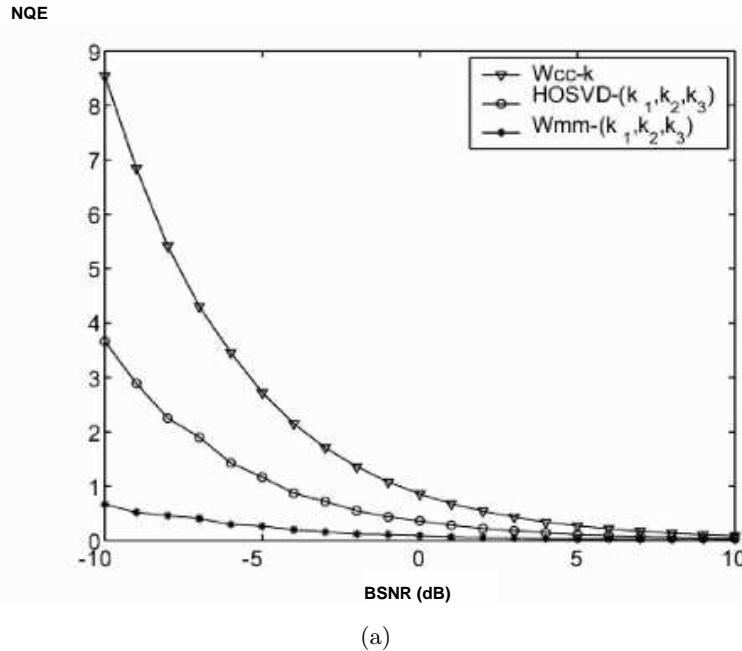


Figure 5.12 — non-noisy, impaired, and processed seismic wave : the three polarization components ; (a)-(c) : components 1, 2 and 3 of the non-noisy seismic wave ; (d)-(f) : components 1, 2 and 3 of the seismic wave, impaired by an additive white Gaussian noise (SNR= -10 dB) ; (g)-(i) : Wiener filtering applied component by component (Wcc-K), with rank $K = 8$; (j)-(l) : HOSVD-(K_1, K_2, K_3), with $(K_1, K_2, K_3)=(8, 8, 3)$; (m)-(o) : multiway Wiener filtering (Wmm-(K_1, K_2, K_3)), with $(K_1, K_2, K_3)=(8, 8, 3)$.



(a)

Figure 5.13 — Evolution of the NQE with respect to the SNR (dB) for each tensor filtering method. (∇) : Wiener filtering applied component by component (W_{cc-K}), with rank $K = 8$; (\circ) : HOSVD- (K_1, K_2, K_3) , with $(K_1, K_2, K_3) = (8, 8, 3)$; (\bullet) : Multiway Wiener filtering ($W_{mm}(K_1, K_2, K_3)$) with $(K_1, K_2, K_3) = (8, 8, 3)$.

5.4.5.2 Filtering of a multicomponent seismic type signal impaired by an additive correlated Gaussian noise

In this simulation, we consider a multicomponent seismic wave, impaired by a correlated Gaussian noise. The purpose here is to compare the performances of multiway filtering algorithms based on either second order moments or fourth order cumulants. Figures 5.14 and 5.15 show the efficiency, in terms of noise reduction, of rank- $\mathcal{C}(K_1, K_2, K_3)$ based filtering and rank- $\mathbf{C}_1(K_1, K_2, K_3)$ based filtering compared to rank- (K_1, K_2, K_3) approximation based on second order statistics, when seismic signals impaired by a correlated Gaussian noise are considered.

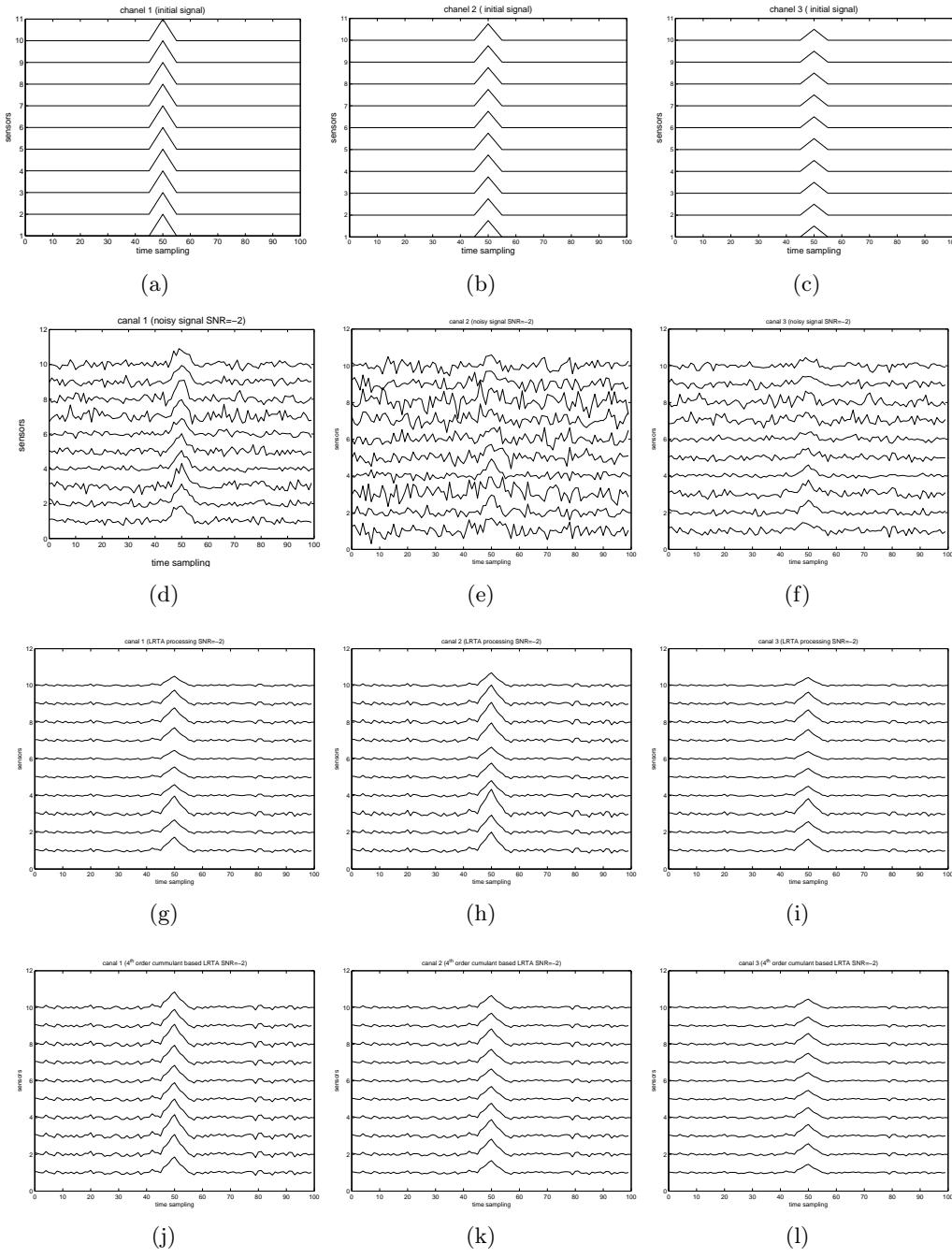


Figure 5.14 — Denoising of a multicomponent seismic wave impaired by an additive correlated Gaussian noise (SNR= -2 dB), using multiway filtering based on fourth order cumulants : comparison of rank- $\mathcal{C}(8,8,3)$, and rank- $\mathbf{C}_1(8,8,3)$: (a)-(c) : components 1 to 3 of the non-noisy seismic wave; (d)-(f) : noised signal; components 1 to 3 impaired by a correlated Gaussian noise (SNR= -2 dB); (g)-(i) : rank- $\mathcal{C}(8,8,3)$ based filtering; (j)-(l) : rank- $\mathbf{C}_1(8,8,3)$ based filtering.

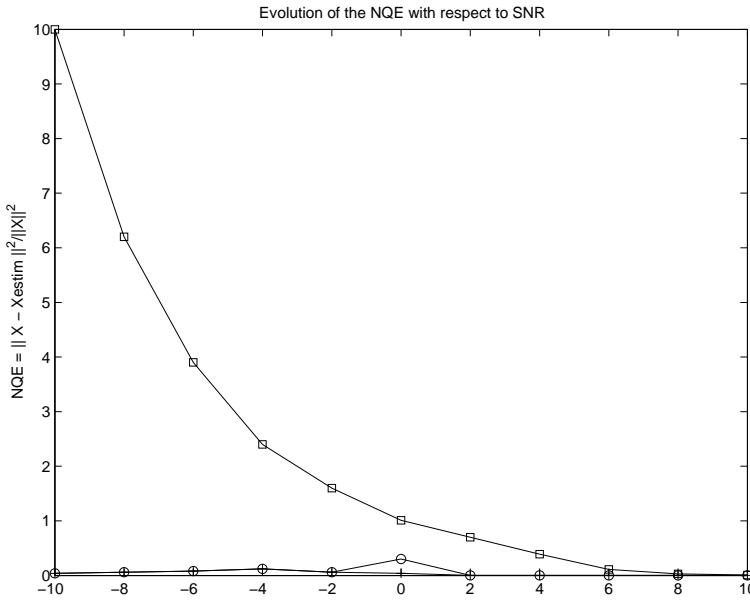


Figure 5.15 — Denoising of a multicomponent seismic wave impaired by an additive correlated Gaussian noise, by multiway filtering : Evolution of NQE with respect to SNR (dB) for rank-(8,8,3) approximation (\square), rank- $C_1(8,8,3)$ using fourth order cumulant slice matrix (\circ) and rank- $C(8,8,3)$ using fourth order cumulants (+).

The simulation results presented in section 5.4 exemplified the performance of several multiway filtering methods when applied to denoising of color images, a hyperspectral image, or seismic data. These methods assume that for each mode, the n th-mode rank is smaller than the n th-mode dimension. In the next section, we exemplify the case where this condition is not fulfilled and we show that a convenient data reorganization permits to fulfill the condition on n th-mode ranks. Starting from this remark, we propose to perform data reorganization of multichannel images. In the particular case when edges are present in images, in order to improve tensor denoising, we retrieve main directions in images and apply adequate rotations to the processed tensor. Rotated images exhibit lower n th-mode rank values.

5.5 Nonorthogonal tensor flattening for hyperspectral image filtering

The purpose of section 5.5 is to show the interest of line detection in images, for the possible improvement of tensor filtering methods. We show that when flattening is performed along main directions in a multichannel image, a better multiway Wiener filtering denoising performance is achieved. Details are preserved, which may yield in particular better further classification results on hyperspectral images.

Airborne hyperspectral imagers such as HYDICE [113] allow to study a scene that uses hundreds of contiguous spectral bands. A hyperspectral image (HSI) is a set of bidimensional images. Therefore, HSI data can be represented as multidimensional data : two spatial dimensions and one spectral dimension. In HSI, the acquired images are impaired by additive noise [106, 32, 14, 53]. This noise can affect the information extraction and scene interpretation

[104]. When using traditional noise removal methods, HSIs are split into vectors or matrices that are separately processed. Noise reduction is of great interest for target detection or classification [83, 32, 104]. Data are often decorrelated using PCA [14]. Then any filtering method could be applied. Several techniques exist to denoise each HSI band. Since each band is a 2-D image, one might consider using a Wiener filter. In [14] a denoising technique decorrelates the signal in space and across all channels. It uses a 2-D discrete wavelet transform and a discrete Fourier transform. Another noise removal method is the Maximum Noise Fraction (MNF) transform [53]. As a linear transformation, it uses essentially two cascaded PCA transformations. The first transformation decorrelates and rescales noise in the data. This results in transformed data in which noise has unit variance and no band to band correlation. The second transformation is a standard PCA of the noise-whitened data. The main shortcoming of these methods is the assumption of separability between spatial processing and spectral processing. Some recent methods propose to denoise HSI using wavelet thresholding technique [42, 33, 104]. *PCA – SWT* is one of these. Contrary to all existing methods, we consider HSI as three-dimensional data array or third order tensor. Thus multilinear algebra algorithms [76, 70, 126, 40] may be applied in this frame. It has been shown in chapter 5 that multidimensional Wiener filtering (*MWF*) is more efficient than classical filtering methods such as channel-by-channel SVD, Wiener filter, as well as other multidimensional filtering methods, namely truncation of higher-order SVD (HOSVD- (K_1, \dots, K_N)) and rank- (K_1, \dots, K_N) approximation. However, *MWF* generally introduces some blur and artifacts due to the usual flattening directions parallel to the dimensions of the multidimensional data. The blur and artifacts may degrade the performance of further target detection or classification algorithms. This study proposes an original approach to flatten HSI data along directions which are not parallel to the dimensions of the data. Section 5.5.2 presents the proposed algorithm : the Multidimensional Wiener Filtering on Rearranged data (*MWFR*). Section 5.5.3 presents real-world data results. Section 5.5.4 concludes the study.

5.5.1 Influence of n th-mode ranks on SNR

The singular value decomposition (SVD) of a matrix \mathbf{C} leads to : $\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \sum_{i=1}^K \lambda_i \mathbf{u}_i \mathbf{v}_i^T$, where \mathbf{U} and \mathbf{V} are respectively the matrices whose columns are matrix \mathbf{C} left and right singular vectors. K is the rank of matrix \mathbf{C} and λ_i is its i^{th} singular value. Usually, SVD uses a matrix for image representation. This row and column organization is involved in matrix products. Therefore, SVD is equivalent to an orthogonal projection along rows and columns. Thus, an oblique straight line (see Fig. 5.16-(a)) has full n th-mode ranks. Therefore, signal and noise subspaces cannot be distinguished. In opposite, a vertical or horizontal line has rank 1 in each n th-mode (see Fig. 5.16-(d)).

A gray scale image containing an oblique line (Fig. 5.16-(a)) can be represented as a full rank matrix. The truncation of its SVD leads to a loss of information. This is illustrated in Figs. 5.16-(b) and 5.16-(c) while either the first 80 or 110 components are kept. This is because the columns and rows of the HSI are not redundant, in opposite to the case of Fig. 5.16-(d). Indeed, it can be represented as a rank-1 matrix. So, its SVD truncation to the first component permits to reconstruct the image.

Consider two noisy images \mathcal{R}_a and \mathcal{R}_d containing the same features but otherwise disposed, such that $K_n^{(d)} < K_n^{(a)}$, $n = 1, 2$, as in Figs. 5.16-(a) and (d), where $K_n^{(l)}$ is the n th-mode rank of

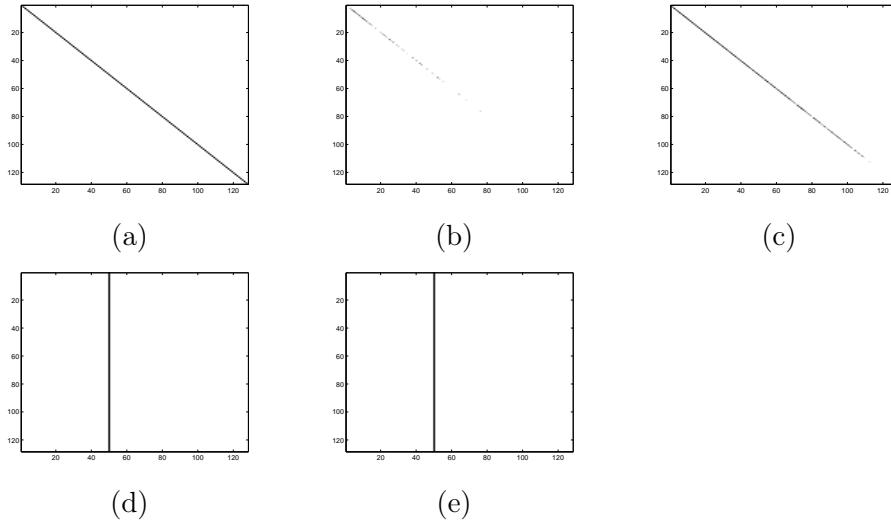


Figure 5.16 — $128 \times 128 \times 1$ gray image containing : (a) Oblique line, (b) SVD truncation to the 80^{th} eigenvalue, (c) SVD truncation to the 110^{th} eigenvalue; (d) Vertical line, (e) SVD truncation to the first eigenvalue.

$\mathcal{R}_l, l \in \{a, d\}$. Each n th-mode flattening matrix SNR can be defined as : $SNR = \frac{\sum_{i=1}^{K_n^{(l)}} \lambda_i^{(n)}}{\sum_{i=1}^{K_n^{(l)}} \sigma^2} = \frac{\sum_{i=1}^{K_n^{(l)}} \lambda_i^{(n)}}{K_n^{(l)} \cdot \sigma^2}$, where $\lambda_i^{(n)}$ is the i^{th} eigenvalue of the n th-mode signal flattening matrix \mathbf{X}_n and σ^2 is the additive white noise power. Assuming an equal signal energy in each n -dimension, $\sum_{i=1}^{K_n^{(a)}} \Lambda_i^{(n)} = \sum_{i=1}^{K_n^{(d)}} \Lambda_i^{(n)} = P$, where $\Lambda_i^{(n)} = \lambda_i^{(n)} + \sigma^2$ is the i^{th} eigenvalue of the n th-mode data flattening matrix \mathbf{R}_n , SNR becomes :

$$SNR(K_n^{(l)}) = \frac{P - K_n^{(l)} \cdot \sigma^2}{K_n^{(l)} \cdot \sigma^2} = \frac{P}{K_n^{(l)} \cdot \sigma^2} - 1 \quad (5.28)$$

Eq. (5.28) clearly emphasizes that $SNR(K_n^{(d)}) > SNR(K_n^{(a)})$. Thus, it is interesting to have the smallest n th-mode ranks values. Fig. 5.16 shows that n th-mode ranks decreases if data are rearranged : $K_1^{(d)} = K_2^{(d)} = 1$.

5.5.2 Main directions of tensors and MWFR algorithm

In this approach, the main arising problem is the estimation of HSIs main orientations. That is, the orientations in which the n th-mode ranks are the smallest. As an HSI is a collection of hundreds of images, the image mean along spectral mode can be used to provide the dominant directions in the hyperspectral cube. We consider that the main directions are straight lines.

The computational load of array processing-based algorithms are smaller than the computational load of the Hough Transform (see section 3.3). However, both lead to the same results. Thus, we adopt array processing-based algorithms to estimate main directions of HSI.

The proposed method called Multidimensional Wiener Filtering on Rearranged data (*MWFR*) can be summarized as follows :

1. Find the d main directions θ_k , $k = 1, \dots, d$ in the HSI tensor with the SLIDE algorithm.
2. For each main direction θ_k , $k = 1, \dots, d$:
 - (a) Flatten each sub-block along the main direction θ_k .
 - (b) Compute MWF .
3. Compute the average filtered HSI from the d main directions used for data rearrangement.

5.5.3 Real-world data : HYDICE HSI

This section gives some results concerning real-world data HSIs, obtained by HYDICE [113]. HYDICE is an airborne sensor which collects post-processed data in 210 wavelengths : $0.4 - 2.5 \mu m$. Spatial resolution is $1.5 m$ and spectral resolution is $10 nm$. In the following experiments, the signal-to-noise ratio is defined as $SNR = 10 \cdot \log \frac{\|\mathcal{X}\|^2}{\|\mathcal{N}\|^2}$. Before presenting some results, we introduce a quality criterion to quantify *a posteriori* the quality of the estimation :

$$QC(\hat{\mathcal{X}}) = 10 \cdot \log \left(\frac{\|\mathcal{X}\|^2}{\|\hat{\mathcal{X}} - \mathcal{X}\|^2} \right). \quad (5.29)$$

5.5.3.1 Noise level in HYDICE images

Fig. 5.17 concerns HSI02 hyperspectral image. The HSI images of HYDICE database usually exhibit low noise level, as exemplified in Fig. 5.17(a). Then, it is difficult to visualize the existing noise in the data. To show that there exists noise in the image, we emphasize it by transforming the data tensor into the spectral derivative domain [104], which is equivalent to a high-pass filter (see Fig. 5.17-(b)). We process the original HSI02 image with $MWFR$, and enhance the remaining noise by high-pass filtering (see Fig. 5.17-(c)). Fig. 5.17-(c) shows that the proposed $MWFR$ can remove some noise even from the HYDICE images which initially exhibit low noise level.

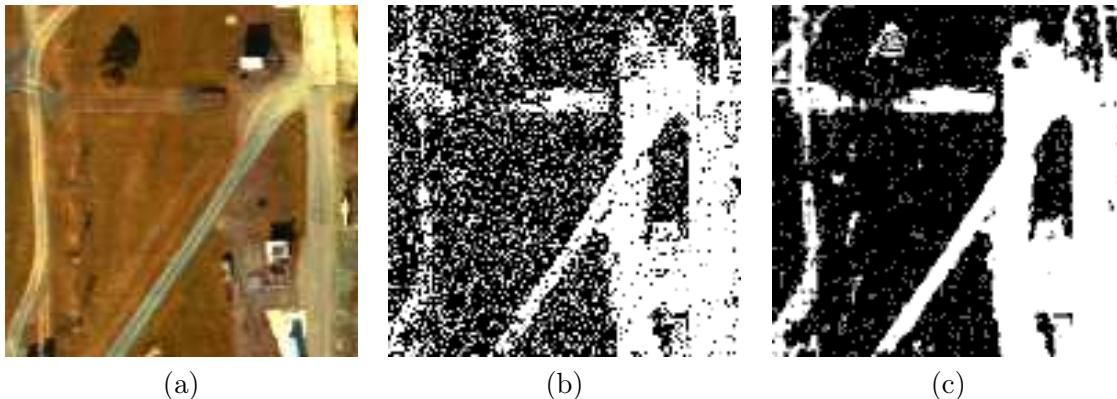


Figure 5.17 — 30^{th} spectral band : (a) acquired data, (b) derivative of acquired data, (c) $MWFR$ processed data.

5.5.3.2 HSI02 image

The image HSI02 is $128 \times 128 \times 148$ with an oblique road in the scene. A white noise has been added to the signal tensor (Fig. 5.17-(a)), with $SNR = 10.13$ dB. Fig. 5.18-(b) gives the result obtained by $PCA - SWT$ ($QC = 19.31$ dB), that is, a PCA to decorrelate channels followed by a soft thresholding of the wavelet transform. Fig. 5.18-(c) presents the result with MWF ($QC = 18.19$ dB). Although noise has been widely removed, targets are not clearly restored. For example, the oblique road is impaired by artifacts. It exhibits blurry edges.

SLIDE algorithm retrieves the main directions of the HSI : 0° , 34° and 90° . For the considered image, the proposed method is two-fold faster than the Hough Transform in estimating main directions. Fig. 5.18-(d) shows the improvement ($QC = 19.59$ dB) brought by main directions in $MWFR$ with the estimated n th-mode ranks $K_1 = K_2 = 31$ and $K_3 = 97$. The HSI restored by $MWFR$ (Fig. 5.18-(d)) contains more recognizable features, which will lead to a better classification or detection of targets.

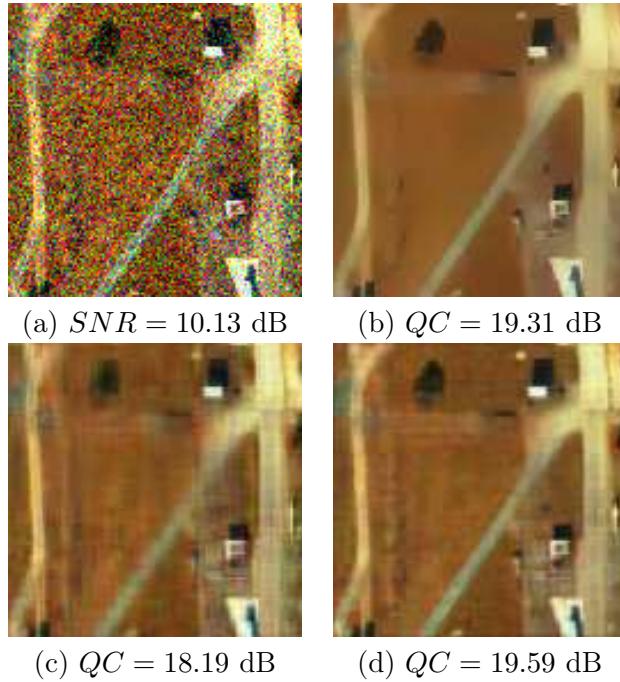


Figure 5.18 — Image HSI02 : (a) noisy tensor, (b) $PCA - SWT$, (c) MWF and (d) $MWFR$ with SLIDE estimated angles $\theta_k \in \{0^\circ, 34^\circ, 90^\circ\}$. ($K_1 = K_2 = 31, K_3 = 97$).

5.5.4 Balance about nonorthogonal tensor flattening

Although usual denoising methods consist in splitting data into vectors or matrices, the adopted approach considers hyperspectral images as whole tensors. Thus, multilinear algebra is used to provide adaptive tools. We proposed to adapt the MWF to hyperspectral images. SLIDE algorithm is adapted to retrieve main directions of HSI, which are required to rearrange data of flattening matrices involved in n th-mode Wiener filters. This rearrangement of data permits to reduce n th-mode ranks of the HSI, which increases the SNR . Comparative results on real-data HYDICE images have shown that the tensor representation yields a better

restoration compared to channel-by-channel denoising or to $PCA - SWT$, wavelet thresholding technique. $MWFR$ clearly outperforms MWF , especially for small targets which are not parallel to rows or columns. The $MWFR$ can be seen as a noise removal method for further applications such as classification or target detection in HSI.

5.6 Conclusion of chapter 5

In this chapter, an overview on new mathematical methods dedicated to multicomponent data is presented. Multicomponent data are represented as tensors, that is, multiway arrays, and the tensor filtering methods that are presented rely on multilinear algebra. First we present how to perform channel-by-channel SVD-based filtering. Then we review three methods that take into account the relationships between each component of a processed tensor. The first method consists of an extension of the classical SVD-based filtering method. In the case of an additive white Gaussian noise, the signal tensor is estimated thanks to a multimode PCA achieved by applying a lower rank- (K_1, \dots, K_N) approximation to the noisy data tensor, or a lower rank- (K_1, \dots, K_N) truncation of its HOSVD. This method is implicitly based on second order statistics and relies on the orthogonality between n th-mode noise and signal subspaces. The second presented method consists of an improvement of the multimode PCA-based tensor filtering in the case of an additive correlated Gaussian noise. In this case, the covariance matrix involved in TUCKALS3 algorithm is replaced with the fourth order cumulant matrix of the related vectors. We reviewed a low computational load procedure involving the fourth order cumulant slice matrix instead of fourth order cumulants. This improved multimode PCA provides good performances compared to the multimode PCA method based on second order statistics, as was shown in the case of noise reduction in color images and multicomponent seismic waves.

The third reviewed method is a multiway version of the classical Wiener filtering. In extension to the one-dimensional case, the n th-mode Wiener filters are estimated by minimizing the mean squared error between the expected signal tensor and the estimated signal tensor obtained by applying the n th-mode Wiener filters to the noisy data tensor thanks to the n th-mode product operator. An Alternating Least Squares algorithm has been presented to determine the optimal n th-mode Wiener filters. The performances of this multiway Wiener filtering and comparative results with multimode PCA have been presented in the case of additive white noise reduction in a color image and in a multicomponent seismic wave.

Finally, we proposed to use straight line estimation methods to improve HSI denoising in the case where main directions are expected. Nonorthogonal tensor flattening, performed along main directions in HSI, improves the denoising results.

Concluding remarks

THIS thesis was devoted to the study of subspace-based and optimization methods, applied to one-dimensional, bidimensional, and multidimensional signals. After reminding the principles of an array processing problem, we proved the interest of a fast optimization method which combines DIRECT and spline interpolation methods, in the resolution of a source characterization problem, when an antenna with a large number of sensors is used and in the presence of phase distortions. Starting from an existing signal generation formalism, we extended a straight line estimation method. We adapted existing array processing methods to the retrieval of free-form objects in an image. Finally, we presented the last advances in tensor processing methods and proposed an improvement in the case where straight edges are present in the processed tensor. For this we used the straight line estimation method presented in the first part of the manuscript.

Thus, in the first part, we focus on one-dimensional signals. In **chapter 1**, we first reminded the purposes of an array processing problem, the definition of an antenna as a set of sensors, of the medium and sources. We introduced the reader with the definition of the covariance matrix, and emphasized the interest of exploiting its eigen-elements. Within the measurement space spanned by the eigenvectors of the covariance matrix, we distinguished the useful signal subspace and the noise subspace. We focused on the interest of spatial smoothing in the case of correlated sources, and emphasized the orthogonality between signal and noise subspaces when useful signal and noise are uncorrelated. Starting from this assumption, the subspace-based MUSIC (MULTiple SIgnal Characterization) method retrieves directions-of-arrival by testing the orthogonality between a signal vector model and the noise subspace vectors. We explained the principles of a novel optimization method that combines either gradient or DIRECT algorithms with spline interpolation. In **chapter 2**, we consider the issue of direction-of-arrival estimation in the case where phase distortions are present. We proposed a novel optimization method, that combines DIRECT (DIviding RECTangles) algorithm with spline interpolation, in order to retrieve wavefront phase distortions. First, a rough estimate of several directions-of-arrival is obtained by MUSIC method. Then for each estimated value, used as an initialization parameter, phase shift values between a plane wavefront and the expected wavefront are estimated by DIRECT combined with spline interpolation. Phase shifts are canceled in the received signal realizations. Finally, MUSIC method is applied in a reduced interval around the initialization direction-of-arrival value. Owing to phase cancelation, several sources may be resolved inside this interval. The proposed algorithm was applied to the characterization of three sources in a noisy environment.

In the second part, we proposed several methods for image - bidimensional data- unders-

tanding applications. In particular, we described an existing formalism for signal generation out of an image. In this framework, a contour in an image is considered as a wavefront, whose propagation is simulated via a signal generation scheme. In **chapter 3**, we reminded the SLIDE (Subspace-based LIne DEtection) method that uses TLS-ESPRIT (Total Least Square-Estimation of Signal Parameters via Rotational Invariance Techniques) algorithm to retrieve straight line orientations. We proposed a faster method, namely the Propagator, which avoids the covariance matrix eigendecomposition, either by partitioning it or by working directly on the processed signal. Propagator exhibits a low computational load. For straight line estimation, we also proposed a novel high-resolution method for the estimation of straight line offsets. This method is faster than the extension of the Hough transform, especially when noisy images are processed, and copes with parallel lines. We extended straight line estimation methods to the estimation of any nearly linear contour. For this, we adapted gradient method and variable step gradient method to minimize a least-squares criterion between a model signal and the signal generated from the processed image. We proposed a generalization of this optimization method to the retrieval of several contours, in particular by minimizing a criterion based on propagator operator. We proposed an improved optimization method, that combines gradient method and spline interpolation. This method is fast and robust to noise impairment. By means of a comparison with GVF method, we emphasized the ability of the proposed optimization method to retrieve contours with high curvature in a noisy environment. Straight line and distorted contour estimation methods were applied to images coming from robotic vision, to aerial and medical images. In **chapter 4**, we extended the work devoted to rectilinear and nearly rectilinear contours to star-shaped contours. The most novel aspect of this chapter is the signal generation method upon a virtual circular antenna : we adapted a circular antenna to the processed image, to generate a linear phase signal and enable the use of high-resolution methods for possibly close concentric circle radius estimation. Signal generation upon linear antenna permits to locate the center of the expected features : the non-zero sections in the signals generated on the image left and top sides indicate the presence of features. Signal generation on linear antenna yields the axial parameters and center coordinates of ellipses. To retrieve rotated ellipses, we rotate the image until the non-zero section length in the left and bottom signals is the most different. When distorted contours are expected, optimization algorithms which are analogous to those applied for nearly straight contour retrieval are applied to signals generated on the proposed circular antenna. We consider several application cases such as biometrics, non-destructive control.

The third part is concerned with tensor -multidimensional data- signal processing, and demonstrates the usefulness of straight line retrieval methods for the improvement of denoising by multilinear subspace-based methods. Firstly, we presented the last advances in tensor signal processing. We reminded the principles of truncation of higher-order Singular Value Decomposition, HOSVD- (K_1, \dots, K_N) , which relies on the definition of signal subspace dimensions K_1, \dots, K_N . We reminded the principles of the Alternating Least-Squares optimization algorithm, which improves HOSVD- (K_1, \dots, K_N) and turns it into rank- (K_1, \dots, K_N) tensor approximation. We show how higher-order statistics, used in rank- (K_1, \dots, K_N) tensor approximation in place of second order statistics, improve tensor signal denoising in the case of correlated Gaussian noise. In particular, we propose a fast version of this method : we use fourth-order cumulant slice matrix in place of the whole fourth-order cumulant tensor. We give details about a multiway version of Wiener filtering,

which seeks for noise removal by minimizing the expectancy of the squared difference between expected and estimated tensor signals. Multiway Wiener filtering relies on data projection upon useful signal subspace. The estimation of projectors, for each n th-mode, is performed by an ALS loop. We performed several experiments on color images, considering particular cases where the signal subspace dimension is equal to the number of channels in the color mode, the case where the signal subspace dimension is underestimated, or where the image is impaired by correlated Gaussian noise. We also performed noise removal on multicomponent seismic wave. Secondly, we proposed an improvement of tensor signal denoising, using the work devoted to straight line estimation : When rectilinear features are present in the processed tensor, we show that nonorthogonal tensor flattening, and more precisely flattening along the orientations of edges, improves the denoising result.

Several prospects can be drawn from this work. SLIDE method has already been applied to the estimation of cinematic parameters of object trajectories. This led to valuable results in the case of constant speed motion. The advances proposed in the second part concerning distorted contour retrieval could be adapted to the characterization of non-constant speed trajectories.

In the field of tensor signal processing, the proposed nonorthogonal tensor flattening method reduces blurring effect which is due to the projection upon signal subspace in each mode. In particular, tensor signal processing methods preserve details. We could adapt this method for an application to target detection. Especially, HSI (Hyperspectral Images) data requires preprocessing such as noise reduction in order to increase the efficiency of classification algorithms. Indeed, a noise reduction processing decreases the spectral variability, which is useful for further classification and detection processing. Moreover the unreasonably large dimension of HSI not only increases computational complexity but also degrades classification accuracy. HSI images contain highly correlated bands, which results in spectral redundancy. Independent Component Analysis (ICA) is applied to reduce the number of spectral bands by selecting spectral components. It would be of great interest to combine the proposed tensor filtering algorithms with ICA, for a joint denoising and dimensionality reduction algorithm.

A

Appendix A : mathematical considerations about multiway filtering

A.1 Appendix : Alternating Least Squares algorithms

In this section, we detail the Alternating Least Squares (ALS) algorithms which are used in Rank- (K_1, \dots, K_N) tensor approximation, in the fourth order cumulant slice matrix-based multimode PCA-based filtering,

A.1.1 ALS algorithm in Rank- (K_1, \dots, K_N) approximation - TUCKALS3 algorithm

In this subsection, we present the ALS algorithm which is performed in Rank- (K_1, \dots, K_N) approximation.

1. **Input :** data tensor \mathcal{R} , and dimensions K_1, \dots, K_N of all n th-mode signal subspaces.
2. **Initialization** $k = 0$: For $n = 1$ to N , calculate the projectors $\mathbf{P}_0^{(n)}$ given by HOSVD- (K_1, \dots, K_N) :
 - (a) n th-mode flatten \mathcal{R} into matrix \mathbf{R}_n ;
 - (b) Compute the SVD of \mathbf{R}_n ;
 - (c) Compute matrix $\mathbf{U}_0^{(n)}$ formed by the K_n eigenvectors associated with the K_n largest singular values of \mathbf{R}_n . $\mathbf{U}_0^{(n)}$ is the initial matrix of the n th-mode signal subspace orthogonal basis vectors ;
 - (d) Form the initial orthogonal projector $\mathbf{P}_0^{(n)} = \mathbf{U}_0^{(n)} \mathbf{U}_0^{(n)T}$ on the n th-mode signal subspace ;
 - (e) Compute the HOSVD- (K_1, \dots, K_N) of tensor \mathcal{R} given by

$$\mathcal{B}_0 = \mathcal{R} \times_1 \mathbf{P}_0^{(1)} \times_2 \cdots \times_N \mathbf{P}_0^{(N)};$$
3. **ALS loop :**
 Repeat until convergence, that is, for example, while $\|\mathcal{B}_{k+1} - \mathcal{B}_k\|^2 > \epsilon$, $\epsilon > 0$ being a prior fixed threshold,

- (a) For $n = 1$ to N :
 - i. Form $\mathcal{B}^{(n),k}$:
 - $$\mathcal{B}^{(n),k} = \mathcal{R} \times_1 \mathbf{P}_{k+1}^{(1)} \times_2 \cdots \times_{n-1} \mathbf{P}_{k+1}^{(n-1)} \times_{n+1} \mathbf{P}_k^{(n+1)} \times_{n+2} \cdots \times_N \mathbf{P}_k^{(N)};$$
 - ii. n th-mode flatten tensor $\mathcal{B}^{(n),k}$ into matrix $\mathbf{B}_n^{(n),k}$;
 - iii. Compute matrix $\mathbf{C}^{(n),k} = \mathbf{B}_n^{(n),k} \mathbf{R}_n^T$;
 - iv. Compute matrix $\mathbf{U}_{k+1}^{(n)}$ composed of the K_n eigenvectors associated with the K_n largest eigenvalues of $\mathbf{C}^{(n),k}$. $\mathbf{U}_k^{(n)}$ is the matrix of the n th-mode signal subspace orthogonal basis vectors at the k^{th} iteration ;
 - v. Compute $\mathbf{P}_{k+1}^{(n)} = \mathbf{U}_{k+1}^{(n)} \mathbf{U}_{k+1}^{(n)T}$;
 - (b) Compute $\mathcal{B}_{k+1} = \mathcal{R} \times_1 \mathbf{P}_{k+1}^{(1)} \times_2 \cdots \times_N \mathbf{P}_{k+1}^{(N)}$;
 - (c) Increment k .
4. **Output** : the estimated signal tensor is obtained through $\hat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{P}_{k_{\text{stop}}}^{(1)} \times_2 \cdots \times_N \mathbf{P}_{k_{\text{stop}}}^{(N)}$. $\hat{\mathcal{X}}$ is the rank- (K_1, \dots, K_N) approximation of \mathcal{R} , where k_{stop} is the index of the last iteration after the convergence of TUCKALS3 algorithm.

In this algorithm, the second order statistics comes from the SVD of matrix \mathbf{R}_n at step 2b, which is equivalent, up to $\frac{1}{M_n}$ multiplicative factor, to the estimation of tensor \mathcal{R} n th-mode vectors [97]. The definition of M_n is given in (5.1). In the same way, at step 3(a)iii, matrix $\mathbf{C}^{(n),k}$ is, up to $\frac{1}{M_n}$ multiplicative factor, the estimation of the covariance matrix between tensor \mathcal{R} and tensor $\mathcal{B}^{(n),k}$ n th-mode vectors. According to step 3(a)i, $\mathcal{B}^{(n),k}$ represents data tensor \mathcal{R} filtered in every m th-mode but the n th-mode, by projection-filters $\mathbf{P}_l^{(m)}$, with $m \neq n$, $l = k$ if $m > n$ and $l = k + 1$ if $m < n$. TUCKALS3 algorithm has recently been used to process a multimode PCA in order to perform white noise removal in color images [94].

A.1.2 Fourth order cumulant slice matrix-based multimode PCA-based filtering

In this subsection, we present the ALS algorithm which is performed in Fourth order cumulant slice matrix-based multimode PCA-based filtering.

1. **Initialization** $k = 0$:

For all $n = 1$ to N , $\mathbf{P}_0^{(n)} = \mathbf{U}_0^{(n)} \mathbf{U}_0^{(n)T}$. $\mathbf{U}_0^{(n)}$ is the matrix of the K_n eigenvectors associated with the K_n largest eigenvalues of fourth order cumulant slice matrix $\mathbf{C}_q^{(n),0}$ of tensor \mathcal{R} n th-mode vectors.

2. **ALS loop** :

The steps (b) and (c) of the ALS loop are the same as in the algorithm "**rank- (K_1, \dots, K_N) approximation - TUCKALS3 algorithm**" described previously, and step (a) is replaced by :

- (a) For $n = 1$ to N :

i.
$$\mathcal{B}^{(n),k} = \mathcal{R} \times_1 \mathbf{P}_{k+1}^{(1)} \times_2 \cdots \times_{n-1} \mathbf{P}_{k+1}^{(n-1)} \times_{n+1} \mathbf{P}_k^{(n+1)} \times_{n+2} \cdots \times_N \mathbf{P}_k^{(N)};$$

- ii. Compute cumulant slice matrix $\mathbf{C}_q^{(n),k}$ associated with the fourth order cumulants of tensors \mathcal{R} and $\mathcal{B}^{(n),k}$ n th-mode vectors. Every element of $\mathbf{C}_q^{(n),k}$ is given in (5.11) ;
 - iii. Process matrix $\mathbf{C}_q^{(n),k}$ eigenvalue decomposition (EVD) and put the K_n eigenvectors associated with the K_n largest eigenvalues into $\mathbf{U}_{k+1}^{(n)}$;
 - iv. Compute projector $\mathbf{P}_{k+1}^{(n)} = \mathbf{U}_{k+1}^{(n)} \mathbf{U}_{k+1}^{(n)T}$;
3. **Output :** $\hat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{P}_{k_{stop}}^{(1)} \times_2 \cdots \times_N \mathbf{P}_{k_{stop}}^{(N)}$, with k_{stop} being the index of the last iteration after convergence of the algorithm.

Using the cumulant slice matrix in place of the whole cumulant tensor saves computational load and memory size.

A.1.3 ALS algorithm in multiway Wiener filtering

Alternating Least Squares (ALS) algorithm used in multiway Wiener filtering can be summarized in the following steps :

1. **Initialization** $k = 0 : \mathcal{R}^0 = \mathcal{R} \Leftrightarrow \mathbf{H}_0^{(n)} = \mathbf{I}_{In}$, Identity matrix, for all $n = 1$ to N .
2. **ALS loop :**
Repeat until convergence, that is, $\|\mathcal{R}^{k+1} - \mathcal{R}^k\|^2 < \epsilon$, with $\epsilon > 0$ a prior fixed threshold,
 - (a) for $n = 1$ to N :
 - i. Form $\mathcal{R}^{(n),k}$:

$$\mathcal{R}^{(n),k} = \mathcal{R} \times_1 \mathbf{H}_{k+1}^{(1)} \times_2 \cdots \times_{n-1} \mathbf{H}_{k+1}^{(n-1)} \times_{n+1} \mathbf{H}_k^{(n+1)} \times_{n+2} \cdots \times_N \mathbf{H}_k^{(N)}$$
;
 - ii. Determine $\mathbf{H}_{k+1}^{(n)} = \arg \min_{\mathbf{Z}^{(n)}} \|\mathcal{X} - \mathcal{R}^{(n),k} \times_n \mathbf{Z}^{(n)}\|^2$
subject to $\mathbf{Z}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ thanks to the following procedure :
 - A. n th-mode flatten $\mathcal{R}^{(n),k}$ into $\mathbf{R}_n^{(n),k} = \mathbf{R}_n (\mathbf{H}_{k+1}^{(1)} \otimes \cdots \otimes \mathbf{H}_{k+1}^{(n-1)} \otimes \mathbf{H}_k^{(n+1)} \otimes \cdots \otimes \mathbf{H}_k^{(N)})^T$, and \mathcal{R} into \mathbf{R}_n ;
 - B. Compute $\gamma_{\mathbf{RR}}^{(n)} = \mathbb{E}[\mathbf{R}_n \mathbf{R}_n^{(n),kT}]$,
 - C. Determine $\lambda_1^\gamma, \dots, \lambda_{K_n}^\gamma$, the K_n largest eigenvalues of $\gamma_{\mathbf{RR}}^{(n)}$;
 - D. For $k_n = 1$ to I_n , estimate $\sigma_\Gamma^{(n)2}$ thanks to (5.22) and for $k_n = 1$ to K_n , estimate β_{k_n} thanks to (5.21) ;
 - E. Compute $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)} = \mathbb{E}[\mathbf{R}_n^{(n),k} \mathbf{R}_n^{(n),kT}]$;
 - F. Determine $\lambda_1^\Gamma, \dots, \lambda_{K_n}^\Gamma$, the K_n largest eigenvalues of $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$;
 - G. Determine $\mathbf{V}_s^{(n)}$, the matrix of the K_n eigenvectors associated with the K_n largest eigenvalues of $\mathbf{\Gamma}_{\mathbf{RR}}^{(n)}$;
 - H. Compute the weight matrix $\gamma_{\mathbf{OO}}^{(n)} \Lambda_{\mathbf{Rs}}^{(n)-1}$ given in (5.20) ;
 - I. Compute $\mathbf{H}_{k+1}^{(n)}$, the n th-mode Wiener filter at the $(k+1)^{\text{th}}$ iteration, using (5.19) ;

- (b) Form $\mathcal{R}^{k+1} = \mathcal{R} \times_1 \mathbf{H}_{k+1}^{(1)} \times_2 \cdots \times_N \mathbf{H}_{k+1}^{(N)}$;
 - (c) Increment k ;
3. **output** : $\hat{\mathcal{X}} = \mathcal{R} \times_1 \mathbf{H}_{k_{stop}}^{(1)} \times_2 \cdots \times_N \mathbf{H}_{k_{stop}}^{(N)}$, with k_{stop} being the last iteration after convergence of the algorithm.

A.2 Appendix : n th-mode Wiener filter analytical expression

The following computations are related to section 5.3.3. They rely on the definitions and properties of tensors and multilinear algebra that can be found in [38, 40, 41].

The mean squared error involved in multiway Wiener filtering is given by relation :

$$\begin{aligned} e(\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N)}) &= E \left[\|\mathcal{X}\|^2 \right] - 2E \left[\langle \mathcal{X}, \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \rangle \right] \\ &\quad + E \left[\left\| \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \right\|^2 \right]. \end{aligned} \quad (\text{A.1})$$

The Frobenius norm of a tensor is also equal to the norm of any of its n th-mode flattening matrices. In order to determine the expression of filter $\mathbf{H}^{(n)}$ associated with fixed filters $\mathbf{H}^{(m)}$, for all $m \neq n$, the n th-mode flattening of (A.1) is processed.

Let us define matrix $\mathbf{F}_{\mathbf{XR}}^{(n)}$ as :

$$\mathbf{F}_{\mathbf{XR}}^{(n)} = \mathbf{X}_n \mathbf{T}^{(n)} \mathbf{R}_n^T. \quad (\text{A.2})$$

with :

$$\mathbf{T}^{(n)} = \mathbf{H}^{(1)} \otimes \cdots \otimes \mathbf{H}^{(n-1)} \otimes \mathbf{H}^{(n+1)} \otimes \cdots \otimes \mathbf{H}^{(N)}. \quad (\text{A.3})$$

Hence, for all $n = 1$ to N ,

$$\langle \mathcal{X}, \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \rangle = \text{tr} \left(\mathbf{F}_{\mathbf{XR}}^{(n)} \mathbf{H}^{(n)T} \right). \quad (\text{A.4})$$

Let us define matrix $\mathbf{G}_{\mathbf{RR}}^{(n)}$ as :

$$\mathbf{G}_{\mathbf{RR}}^{(n)} = \mathbf{R}_n \mathbf{Q}^{(n)} \mathbf{R}_n^T, \quad (\text{A.5})$$

with :

$$\mathbf{Q}^{(n)} = \mathbf{T}^{(n)T} \mathbf{T}^{(n)}$$

$$\mathbf{Q}^{(n)} = \mathbf{H}^{(1)T} \mathbf{H}^{(1)} \otimes \cdots \otimes \mathbf{H}^{(n-1)T} \mathbf{H}^{(n-1)} \otimes \mathbf{H}^{(n+1)T} \mathbf{H}^{(n+1)} \otimes \cdots \otimes \mathbf{H}^{(N)T} \mathbf{H}^{(N)}. \quad (\text{A.6})$$

Hence, for all $n = 1$ to N :

$$\left\| \mathcal{R} \times_1 \mathbf{H}^{(1)} \times_2 \cdots \times_N \mathbf{H}^{(N)} \right\|^2 = \text{tr} \left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} \mathbf{H}^{(n)T} \right). \quad (\text{A.7})$$

Minimization of mean squared error $e(\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N)})$

The expression of the n th-mode flattened mean squared error $e(\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N)})$ is the following :

$$e(\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N)}) = E \left[\|\mathbf{X}_n\|^2 \right] - 2E \left[\text{tr} \left(\mathbf{F}_{\mathbf{XR}}^{(n)} \mathbf{H}^{(n)T} \right) \right] + E \left[\text{tr} \left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} \mathbf{H}^{(n)T} \right) \right]. \quad (\text{A.8})$$

Assuming that m -mode filters $\mathbf{H}^{(m)}$ are fixed for all $m \neq n$, mean squared error $e(\mathbf{H}^{(1)}, \dots, \mathbf{H}^{(N)})$ is minimal when its gradient with respect to n th-mode filter $\mathbf{H}^{(n)}$ is null :

$$\mathbf{grad}(e) = \left[\frac{\partial e}{\partial \mathbf{H}^{(1)}}, \dots, \frac{\partial e}{\partial \mathbf{H}^{(N)}} \right]^T \quad (\text{A.9})$$

that is, when $\frac{\partial e}{\partial \mathbf{H}^{(n)}}$ are conjointly null for all $n = 1$ to N . Let us study $\frac{\partial e}{\partial \mathbf{H}^{(n)}}$ for a given n th-mode. The n th-mode filters $\mathbf{H}^{(m)}$ are supposed to be fixed for all $m \in \{1, \dots, N\} - \{n\}$. Then $\frac{\partial e}{\partial \mathbf{H}^{(n)}} = 0$ implies that :

$$E \left[\frac{\partial}{\partial \mathbf{H}^{(n)}} \text{tr} \left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} \mathbf{H}^{(n)T} \right) \right] = 2E \left[\frac{\partial}{\partial \mathbf{H}^{(n)}} \text{tr} \left(\mathbf{F}_{\mathbf{XR}}^{(n)} \mathbf{H}^{(n)T} \right) \right], \quad (\text{A.10})$$

We compute then the derivatives on both sides in (A.10), taking into account the fact that $\mathbf{G}_{\mathbf{RR}}^{(n)}$ and $\mathbf{F}_{\mathbf{XR}}^{(n)}$ are independent from $\mathbf{H}^{(n)}$:

$$\frac{\partial}{\partial \mathbf{H}^{(n)}} \text{tr} \left(\mathbf{F}_{\mathbf{XR}}^{(n)} \mathbf{H}^{(n)T} \right) = \mathbf{F}_{\mathbf{XR}}^{(n)}. \quad (\text{A.11})$$

$$\frac{\partial}{\partial \mathbf{H}^{(n)}} \text{tr} \left(\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)} \mathbf{H}^{(n)T} \right) = 2\mathbf{H}^{(n)} \mathbf{G}_{\mathbf{RR}}^{(n)}. \quad (\text{A.12})$$

Expression of $\mathbf{H}^{(n)}$, n th-mode Wiener filter

Replacing (A.11) and (A.12) into expression (A.10) leads to the expression of $\mathbf{H}^{(n)}$ n th-mode Wiener filter associated with fixed $\mathbf{H}^{(m)}$ m -mode filters, $m \neq n$:

$$\mathbf{H}^{(n)} = \gamma_{\mathbf{XR}}^{(n)} \boldsymbol{\Gamma}_{\mathbf{RR}}^{(n)-1}, \quad (\text{A.13})$$

where :

$$\gamma_{\mathbf{XR}}^{(n)} = E \left[\mathbf{F}_{\mathbf{XR}}^{(n)} \right] \quad (\text{A.14})$$

is the $\mathbf{T}^{(n)}$ -weighted covariance matrix between the signal \mathbf{X}_n and the data \mathbf{R}_n , and :

$$\boldsymbol{\Gamma}_{\mathbf{RR}}^{(n)} = E \left[\mathbf{G}_{\mathbf{RR}}^{(n)} \right] \quad (\text{A.15})$$

is the $\mathbf{Q}^{(n)}$ -weighted correlation matrix of the data.

A.3 Appendix : Assumptions and related expression of the n th-mode Wiener filter

The following computations are related to section 5.3.3. Let us consider matrices $\mathbf{T}^{(n)}$ and $\mathbf{Q}^{(n)}$ defined in (A.3) and (A.6). Their generic (i, j) -terms are denoted respectively by $T_{ij}^{(n)}$ and by $Q_{ij}^{(n)}$.

Weight matrix term independence

The terms of weight matrix $\mathbf{O}^{(n)} \in \mathbb{R}^{K_n \times M_n}$ are supposed mutually independent :

$$\mathbb{E}[o_{kl} o_{mn}] = \alpha_{kl} \delta_{km} \delta_{ln}, \quad (\text{A.16})$$

whatever k and $m \in \{1, \dots, K_n\}$, l and $n \in \{1, \dots, M_n\}$ and where α_{kl} is not null.

White and Gaussian noise condition

White and Gaussian noise condition applied to the n th-mode flattening \mathbf{N}_n can be expressed by :

$$\mathbb{E}[n_{kl} n_{pq}] = \sigma_n^2 \delta_{kp} \delta_{lq}, \quad (\text{A.17})$$

where $(k, p) \in \{1, \dots, K_n\}^2$, $(l, q) \in \{1, \dots, M_n\}^2$ and σ_n^2 is the n th-mode noise power.

Noise and signal independence

The condition on noise and signal independence can be expressed by :

$$\mathbb{E}[x_{kl} n_{pq}] = 0, \quad (\text{A.18})$$

for all $(k, p) \in \{1, \dots, K_n\}^2$ and $(l, q) \in \{1, \dots, M_n\}^2$. Hence, $\mathbf{T}^{(n)}$ and $\mathbf{Q}^{(n)}$ -weighted (\mathbf{X}, \mathbf{N}) -covariance matrices are null :

$$\begin{aligned} \gamma_{\mathbf{X}\mathbf{N}}^{(n)} &= \gamma_{\mathbf{N}\mathbf{X}}^{(n)} = \mathbf{0}, \\ \Gamma_{\mathbf{X}\mathbf{N}}^{(n)} &= \Gamma_{\mathbf{N}\mathbf{X}}^{(n)} = \mathbf{0}. \end{aligned} \quad (\text{A.19})$$

Indeed, their (i, j) -term is :

$$\begin{aligned} \left(\gamma_{\mathbf{X}\mathbf{N}}^{(n)}\right)_{ij} &= \sum_{k=1}^{M_n} \sum_{l=1}^{M_n} T_{kl}^{(n)} \mathbb{E}[x_{ik} n_{jl}], \\ \left(\Gamma_{\mathbf{X}\mathbf{N}}^{(n)}\right)_{ij} &= \sum_{k=1}^{M_n} \sum_{l=1}^{M_n} Q_{kl}^{(n)} \mathbb{E}[x_{ik} n_{jl}]. \end{aligned} \quad (\text{A.20})$$

Expressions of weighted covariance matrices

Covariance matrix $\gamma_{\mathbf{R}\mathbf{R}}^{(n)}$

As $\mathbf{R}_n = \mathbf{X}_n + \mathbf{N}_n$, the expression of $\gamma_{\mathbf{R}\mathbf{R}}^{(n)}$ reads :

$$\gamma_{\mathbf{R}\mathbf{R}}^{(n)} = \gamma_{\mathbf{X}\mathbf{X}}^{(n)} + \gamma_{\mathbf{X}\mathbf{N}}^{(n)} + \gamma_{\mathbf{N}\mathbf{X}}^{(n)} + \gamma_{\mathbf{N}\mathbf{N}}^{(n)}. \quad (\text{A.21})$$

So according to (A.19), $\gamma_{\mathbf{RR}}^{(n)}$ weighted covariance matrix can be expressed by :

$$\gamma_{\mathbf{RR}}^{(n)} = \gamma_{\mathbf{XX}}^{(n)} + \gamma_{\mathbf{NN}}^{(n)}. \quad (\text{A.22})$$

Moreover :

$$\gamma_{\mathbf{XR}}^{(n)} = \gamma_{\mathbf{XX}}^{(n)} + \gamma_{\mathbf{XN}}^{(n)} = \gamma_{\mathbf{XX}}^{(n)}. \quad (\text{A.23})$$

Covariance matrix $\Gamma_{\mathbf{RR}}^{(n)}$

Relations (A.21), (A.22), and (A.23) hold as well for $\Gamma_{\mathbf{RR}}^{(n)}$:

$$\Gamma_{\mathbf{RR}}^{(n)} = \Gamma_{\mathbf{XX}}^{(n)} + \Gamma_{\mathbf{XN}}^{(n)} + \Gamma_{\mathbf{NX}}^{(n)} + \Gamma_{\mathbf{NN}}^{(n)},$$

and :

$$\Gamma_{\mathbf{RR}}^{(n)} = \Gamma_{\mathbf{XX}}^{(n)} + \Gamma_{\mathbf{NN}}^{(n)}. \quad (\text{A.24})$$

Moreover :

$$\Gamma_{\mathbf{XR}}^{(n)} = \Gamma_{\mathbf{XX}}^{(n)} + \Gamma_{\mathbf{XN}}^{(n)} = \Gamma_{\mathbf{XX}}^{(n)}. \quad (\text{A.25})$$

Expressions of $\Gamma_{\mathbf{NN}}^{(n)}$ and $\gamma_{\mathbf{NN}}^{(n)}$

According to (A.17), the (i, j) -term of $\Gamma_{\mathbf{NN}}^{(n)}$ is the following :

$$\left(\Gamma_{\mathbf{NN}}^{(n)}\right)_{ij} = \sum_{k=1}^{M_n} \sum_{l=1}^{M_n} Q_{kl}^{(n)} \mathbb{E}[n_{ik} n_{jl}] = \sigma_{\Gamma}^{(n)2} \delta_{ij}, \quad (\text{A.26})$$

with

$$\sigma_{\Gamma}^{(n)2} = \text{tr}(\mathbf{Q}^{(n)}) \sigma_n^2. \quad (\text{A.27})$$

Hence :

$$\Gamma_{\mathbf{NN}}^{(n)} = \sigma_{\Gamma}^{(n)2} \mathbf{I}_{I_n}. \quad (\text{A.28})$$

The (i, j) -term of $\gamma_{\mathbf{NN}}^{(n)}$ can also be expressed by :

$$\left(\gamma_{\mathbf{NN}}^{(n)}\right)_{ij} = \sum_{k=1}^{M_n} \sum_{l=1}^{M_n} T_{kl}^{(n)} \mathbb{E}[n_{ik} n_{jl}] = \sigma_{\gamma}^{(n)2} \delta_{ij},$$

with :

$$\sigma_{\gamma}^{(n)2} = \text{tr}(\mathbf{T}^{(n)}) \sigma_n^2.$$

Hence :

$$\gamma_{\mathbf{NN}}^{(n)} = \sigma_{\gamma}^{(n)2} \mathbf{I}_{I_n}. \quad (\text{A.29})$$

Expressions of $\Gamma_{\mathbf{XX}}^{(n)}$ and $\gamma_{\mathbf{XX}}^{(n)}$

Considering the signal model (5.18) :

$$\gamma_{\mathbf{XX}}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)T}, \quad (\text{A.30})$$

where

$$\gamma_{\mathbf{OO}}^{(n)} = \mathbb{E} \left[\mathbf{O}^{(n)T} \mathbf{T}^{(n)} \mathbf{O}^{(n)} \right]. \quad (\text{A.31})$$

According to (A.16), the generic term of $\gamma_{\mathbf{OO}}^{(n)}$ is :

$$\left(\gamma_{\mathbf{OO}}^{(n)} \right)_{ij} = \sum_{k=1}^{M_n} \sum_{l=1}^{M_n} T_{kl}^{(n)} \mathbb{E} [n_{ik} n_{jl}] = \beta_i \delta_{ij}, \quad (\text{A.32})$$

where, for all $i = 1$ to K_n :

$$\beta_i = \sum_{k=1}^{M_n} T_{kk}^{(n)} \alpha_{ik} \quad (\text{A.33})$$

and where α_{ik} is defined in (A.16). So, $\gamma_{\mathbf{OO}}^{(n)}$ is a diagonal matrix :

$$\gamma_{\mathbf{OO}}^{(n)} = \begin{bmatrix} \beta_1 & 0 \\ \ddots & \ddots \\ 0 & \beta_{K_n} \end{bmatrix}. \quad (\text{A.34})$$

The matrix $\Gamma_{\mathbf{XX}}^{(n)}$ is also expressed as :

$$\Gamma_{\mathbf{XX}}^{(n)} = \mathbf{V}_s^{(n)} \Gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)T}, \quad (\text{A.35})$$

where $\Gamma_{\mathbf{OO}}^{(n)}$ is the diagonal matrix :

$$\Gamma_{\mathbf{OO}}^{(n)} = \begin{bmatrix} \epsilon_1 & 0 \\ \ddots & \ddots \\ 0 & \epsilon_{K_n} \end{bmatrix}, \quad (\text{A.36})$$

and :

$$\epsilon_i = \sum_{k=1}^{M_n} Q_{kk}^{(n)} \alpha_{ik}, \quad (\text{A.37})$$

where α_{ik} is defined in (A.16).

Final expression of $\mathbf{H}^{(n)}$, n th-mode Wiener filter

According to (A.23) and (A.30) :

$$\gamma_{\mathbf{XR}}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)T}. \quad (\text{A.38})$$

According to (A.24), (A.28) and (A.35) :

$$\Gamma_{\mathbf{RR}}^{(n)} = \mathbf{V}_s^{(n)} \Gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)T} + \sigma_{\Gamma}^{(n)2} \mathbf{I}_{I_n},$$

which can be expressed as :

$$\boldsymbol{\Gamma}_{\mathbf{RR}}^{(n)} = \left[\mathbf{V}_s^{(n)} \mathbf{V}_b^{(n)} \right] \begin{bmatrix} \boldsymbol{\Gamma}_{\mathbf{OO}}^{(n)} + \sigma_{\Gamma}^{(n)2} \mathbf{I}_{K_n} & 0 \\ 0 & \sigma_{\Gamma}^{(n)2} \mathbf{I}_{I_n - K_n} \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^{(n)T} \\ \mathbf{V}_b^{(n)T} \end{bmatrix}, \quad (\text{A.39})$$

with $\mathbf{V}_b^{(n)} \in \text{St}(I_n, I_n - K_n)$ the column-wise orthogonal matrix containing the noise subspace basis vectors. The assumption of noise and signal independence implies that the noise and signal subspaces are orthogonal :

$$\mathbf{V}_s^{(n)T} \mathbf{V}_b^{(n)} = \mathbf{0}. \quad (\text{A.40})$$

Let us call :

$$\boldsymbol{\Lambda}_s^{(n)} = \boldsymbol{\Gamma}_{\mathbf{OO}}^{(n)} + \sigma_{\Gamma}^{(n)2} \mathbf{I}_{K_n}, \quad (\text{A.41})$$

and :

$$\boldsymbol{\Lambda}_b^{(n)} = \sigma_{\Gamma}^{(n)2} \mathbf{I}_{I_n - K_n}. \quad (\text{A.42})$$

Inserting the last expressions of $\gamma_{\mathbf{XR}}^{(n)}$ and $\boldsymbol{\Gamma}_{\mathbf{RR}}^{(n)}$ (see (A.38) and (A.39)) into Wiener nth-mode filter expression (A.13) leads to :

$$\mathbf{H}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)T} \left[\mathbf{V}_s^{(n)} \mathbf{V}_b^{(n)} \right] \begin{bmatrix} \boldsymbol{\Lambda}_s^{(n)-1} & 0 \\ 0 & \boldsymbol{\Lambda}_b^{(n)-1} \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^{(n)T} \\ \mathbf{V}_b^{(n)T} \end{bmatrix}, \quad (\text{A.43})$$

which can be expressed as :

$$\mathbf{H}^{(n)} = \left[(\mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)T} \mathbf{V}_s^{(n)}) (\mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \mathbf{V}_s^{(n)T} \mathbf{V}_b^{(n)}) \right] \begin{bmatrix} \boldsymbol{\Lambda}_s^{(n)-1} \mathbf{V}_s^{(n)T} & 0 \\ 0 & \boldsymbol{\Lambda}_b^{(n)-1} \mathbf{V}_b^{(n)T} \end{bmatrix} \quad (\text{A.44})$$

Considering noise and signal orthogonality condition (A.40) and the fact that $\mathbf{V}_n^{(n)} \mathbf{V}_n^{(n)T} = \mathbf{I}_{K_n}$, the final Wiener nth-mode filter expression becomes :

$$\mathbf{H}^{(n)} = \mathbf{V}_s^{(n)} \gamma_{\mathbf{OO}}^{(n)} \boldsymbol{\Lambda}_{\mathbf{rs}}^{(n)-1} \mathbf{V}_s^{(n)T}. \quad (\text{A.45})$$

Appendix B : Synthèse du manuscrit

Les progrès concernant la théorie du traitement d'antenne ont été très importants au cours des trente dernières années. Les moyens mis en oeuvre pour résoudre le cas de sources proches ou corrélées, qui font face au bruit corrélé Gaussien, ont abouti à des méthodes précises, rapides, et robustes au bruit. Parmi ces méthodes se distinguent les méthodes par sous-espaces et d'optimisation. Elles ont été appliquées sous la forme d'algorithmes rapides grâce aux progrès des calculateurs. Ce travail de thèse est guidé par la volonté d'adapter à des problèmes de traitement d'image les méthodes du traitement d'antenne. Pour cela des méthodes de génération de signal à partir d'une image ont été développées. Elles permettent l'adaptation des méthodes algébriques du traitement d'antenne à la caractérisation d'objets dans des images ou au débruitage des images couleur et hyperspectrales. Le cas des signaux sismiques multidimensionnels est également considéré.

L'objectif de cette synthèse est de donner les grandes lignes du manuscrit, qui se divise en trois parties. La première partie permet de comprendre un problème en traitement d'antenne, et propose une application à la localisation de sources dans le cas où l'antenne est constituée d'un grand nombre de capteurs. La deuxième partie montre comment les méthodes de traitement d'antenne et d'optimisation sont adaptées à l'estimation de contours dans des images. La troisième partie propose une approche tensorielle du problème de débruitage de données.

B.1 Introduction générale

Le traitement et l'analyse de données ont été développés et encouragés par des progrès considérables dans les technologies de calcul numérique, de vidéo, et dans les systèmes d'acquisition comme les capteurs. Un ensemble de capteurs forme une antenne. Lorsque l'on considère un problème de traitement d'antenne, de façon générale, on suppose que les signaux reçus sur chaque capteur ont un contenu en fréquences centré autour d'une fréquence donnée. Ainsi, pour chaque capteur, on choisit la composante de la transformée de Fourier du signal correspondant à cette fréquence d'intérêt. On obtient ainsi un signal unidimensionnel : une composante par capteur. Une antenne est utilisée pour la localisation de sources. Une antenne peut prendre des formes variables. Les antennes circulaires sont couramment utilisées pour des signaux de télécommunications, les antennes linéaires sont utilisées par exemple pour la détection d'objets. En particulier, des antennes à grand nombre de capteurs ont été utilisées, pour améliorer les résultats de localisation. La détection et la localisation d'objets est d'un grand

intérêt pour des applications civiles et militaires, pour la détection de mines, pipe-lines, ou câbles. Résoudre un problème de traitement d'antenne consiste à déterminer l'origine d'ondes émises par des sources pouvant être proches. Des moyens importants ont été investis dans le domaine du traitement d'antenne pour résoudre des cas difficiles tels que le cas de sources de directions d'arrivée proches, le cas de sources corrélées, le cas où un bruit corrélé dégrade les données.

La démarche adoptée lors de cette thèse provient du constat suivant : les méthodes du traitement d'antenne ont atteint un niveau de perfectionnement élevé, et pourraient être profitables aux domaines du traitement d'image et du traitement de données multidimensionnelles. Il nous est donc apparu d'un grand intérêt, dans le cadre de cette thèse, d'établir des analogies entre les méthodes du traitement d'antenne, qui sont par nature des méthodes de traitement de signaux unidimensionnels, et des problèmes de traitement d'image. Or les images sont soit des données bidimensionnelles, lorsque l'on considère des images binaires ou à niveaux de gris, soit multidimensionnelles lorsque l'on considère des images couleur ou des images hyperspectrales. Nous avons considéré le problème d'estimation de contour dans des images binaires ou à niveaux de gris. Une première difficulté est de transcrire le contenu d'une image en un signal. Pour cela une méthode de génération de signal particulière consiste à adapter une antenne à l'image à traiter. Cette antenne est soit linéaire soit circulaire, en fonction du type d'objet recherché dans l'image. Pour appliquer les méthodes du traitement d'antenne à un problème de traitement de données multidimensionnelles, en particulier celui du débruitage de données, il est possible de choisir un autre formalisme. Nous avons choisi, pour pouvoir appliquer les méthodes algébriques du traitement d'antenne à un tel problème, de considérer chaque vecteur colonne d'une entité multidimensionnelle comme une réalisation de signal. On donne dans la suite de cette sous-section des détails concernant les enjeux des traitements appliqués aux signaux, images, ou données multidimensionnelles.

Les données sont bidimensionnelles quand on réalise de l'acquisition d'images. Le traitement d'images consiste à produire à partir de données image brutes des images qui sont en un certain sens de meilleure qualité. Améliorer la qualité d'une image qui a été dégradée par un flou ou du bruit est nécessaire dans le cas où les capteurs bougent ou introduisent un bruit parasite. L'analyse d'images a pour objet d'extraire de l'information. La localisation d'objets est une tâche caractéristique d'analyse d'images. Une telle tâche est réalisée facilement par le cerveau humain qui possède une capacité à extraire de l'information pertinente à partir d'une image et à interpréter une image en un nombre réduit de paramètres. Les données fournies par l'analyse d'images sont soit des valeurs de paramètres soit une description des objets présents dans l'image.

De nombreux phénomènes dépendent de plusieurs variables physiques telles que celles associées à l'espace, le temps, la longueur d'onde, la couleur ou la polarisation. Dans ce contexte, l'acquisition conduit à des données multidimensionnelles. Pour enregistrer, étudier, et extraire de l'information utile concernant ces phénomènes, plusieurs capteurs multidimensionnels ou multicomposantes sont apparus lors des cinquante dernières années. Les données qui proviennent de ces capteurs sont appelées multidimensionnelles. Les données multidimensionnelles sont modélisées par des tableaux, dont chaque entrée est associée à un paramètre physique, tel que le temps, l'espace, ou la longueur d'onde. Un tel tableau est appelé tenseur [92, 38], et chaque dimension du tenseur est appelée n -mode ; le nombre de

n -modes est appelé l'ordre du tenseur.

Les données sont échantillonnées de sorte qu'un signal est un ensemble d'échantillons, une image est un ensemble de pixels, et un tenseur est un tableau multidimensionnel. Donc, le traitement du signal, le traitement d'image ou le traitement du signal tensoriel est mis en oeuvre par des méthodes algébriques. Ces traitements reposent, en général, sur des statistiques d'ordre deux [133, 114, 90, 5, 3] ou d'ordre supérieur [80, 89, 108]. Le traitement le plus courant est basé sur la matrice de covariance (statistique d'ordre deux), en particulier sa décomposition en valeurs singulières (ou Singular Value Decomposition, SVD) [43, 51]. Des méthodes fiables pour extraire des paramètres d'un tableau de données sont fondées sur le concept de sous-espace signal [115, 114, 90, 3]. La sélection d'un sous-espace de dimension réduite permet d'extraire de l'information utile, en particulier des paramètres à retrouver [5, 3]. La connaissance, ou l'estimation par des critères statistiques [132], de la dimension du sous-espace "signal utile" permet de réaliser une partition de l'espace des mesures en un sous-espace signal qui contient l'information désirée, et un sous-espace bruit, qui contient toute l'information non désirée.

Nous associerons, aux méthodes par sous-espaces, des méthodes d'optimisation, qui ont pour objet d'estimer un ensemble de paramètres inconnus, en minimisant un critère, en général une distance au carré entre un modèle de données et des mesures. On distingue deux catégories de méthodes d'optimisation : les méthodes globales et les méthodes locales. Les méthodes globales retrouvent le minimum global d'une fonction, elles ne convergent pas vers un minimum qui n'est que local, aux dépens d'une charge de calcul élevée. Par exemple, la méthode DIviding RECTangles (DIRECT) [64], ou les algorithmes génétiques [50] sont des méthodes d'optimisation globales. Les méthodes locales sont en général plus rapides que les méthodes globales. Une méthode locale classique pour la résolution de problèmes d'optimisation non linéaires et sans contraintes est la méthode du gradient dite aussi de Newton [48], dont il existe une version rapide -à pas variable-.

Il est d'un grand intérêt de combiner des méthodes par sous-espaces et d'optimisation pour obtenir une estimation optimale au sens d'un critère donné *a priori* d'un ensemble de paramètres, ou d'éléments caractéristiques à partir d'un signal, d'une image, ou de données tensorielles. En traitement d'antenne, la méthode par sous-espaces dite du "Propagateur" a été associée à une méthode d'optimisation pour améliorer l'estimation de directions d'arrivée en présence de distorsions de phase [90]. Dans le domaine du traitement d'images, la méthode DIRECT [64] a été appliquée. Dans le domaine du traitement du signal tensoriel, un algorithme de moindres carrés alternés, (ou Alternating Least Squares ALS) qui aboutit à une décomposition tensorielle particulière a été proposée dans [76, 77]. Un algorithme ALS a aussi été proposé dans [96] pour résoudre un problème de filtrage de données tensorielles, pour une application en débruitage.

L'objectif de ce manuscrit est de proposer et d'associer plusieurs méthodes, soit par sous-espaces, soit d'optimisation, pour le traitement et (ou) l'analyse de données, dans les domaines du traitement d'antenne pour les signaux unidimensionnels, du traitement d'images pour les signaux bidimensionnels, et du traitement du signal tensoriel pour les signaux multi-dimensionnels. Les grands principes, et les améliorations proposées, en traitement d'antenne, d'image, et du signal tensoriel sont les suivants :

- Dans un problème en traitement d'antenne, plusieurs réalisations de signal, collectées à divers instants, sont disponibles. La dimension du sous-espace signal est égale au nombre

de sources. En pratique, si le nombre de sources n'est pas connu *a priori*, ce nombre peut être estimé à partir des données par un critère statistique tel que le critère MDL (Minimum Description Length) [132]. Dans ce cadre, nous introduisons de nouvelles méthodes d'optimisation dans un problème de traitement d'antenne concernant des antennes distordues à grand nombre de capteurs.

- L'objectif de l'analyse d'image est d'extraire de l'information d'intérêt à partir d'une image ou d'un ensemble d'images. Le rehaussement d'image a pour but de produire des images qui présentent une meilleure qualité, selon un critère donné. La déconvolution et le débruitage sont des applications du rehaussement d'images. Une approche originale de l'estimation de contours consiste à placer l'image à traiter dans un contexte de propagation d'onde. Plus précisément, un contour est considéré comme un front d'onde figé, et le fond d'image est considéré comme un milieu de propagation [5]. Dans ce cadre de travail, quand une image est traitée, l'on n'a pas accès à plusieurs signaux, acquis au cours du temps. On peut donc se demander comment calculer des statistiques d'ordre deux, et notamment une matrice de covariance. Dans [3, 5], Aghajan a proposé un formalisme spécifique qui permet de transcrire le contenu d'une image en un ensemble de réalisations de signaux. Ce formalisme permet d'appliquer des méthodes par sous-espaces [114]. Dans ce cadre nous adaptons des algorithmes de traitement d'antenne et d'optimisation à un problème d'analyse d'images, en particulier pour retrouver des contours linéaires, approximativement linéaires, circulaires et approximativement circulaires.
- Dans un problème de traitement du signal tensoriel, il est possible d'étendre une approche par sous-espaces, avec une condition *a priori* sur la dimension du sous-espace signal [92]. La projection des données sur le sous-espace signal , selon chaque mode, conduit à un débruitage des données [96]. En supposant que pour tout n -mode le rang (dimension du sous-espace signal) est inférieur à la dimension, on distingue un sous-espace signal et un sous-espace bruit. L'estimation des opérateurs de projection peut être raffinée par un algorithme d'optimisation dit Alternating Least Squares (ALS). Nous présentons les dernières avancées en traitement du signal multidimensionnel, qui reposent sur des méthodes par sous-espaces et d'optimisation. Nous montrons l'intérêt de l'estimation de directions privilégiées pour l'amélioration du traitement du signal tensoriel.

B.2 Corps du manuscrit

Nous présentons dans cette section les principaux enjeux et résultats correspondant à chaque chapitre de la thèse.

B.2.1 Résumé du chapitre 1 : Principes des méthodes de traitement d'antenne et d'optimisation

Nous posons un problème de traitement d'antenne et décrivons des méthodes haute résolution du traitement d'antenne : MUSIC (MULTiple SIgnal Characterization), TLS-ESPRIT (Total-Least-Squares - Estimation of Signal Parameters via Rotational Invariance Techniques), et propagateur. Nous proposons une nouvelle méthode d'optimisation.

B.2.1.1 Bases de traitement d'antenne

Le traitement d'antenne consiste à analyser des signaux collectés sur un ensemble de capteurs, pour obtenir des informations sur des sources -direction d'arrivée ou direction-of-arrival (DOA), amplitude. La méthode de formation de voies [75] est la première méthode développée pour la localisation de sources. Cette méthode est dite "à faible résolution", elle ne peut pas distinguer des sources proches. Les méthodes haute résolution ont été développées pour distinguer des sources proches. Ces méthodes sont fondées sur l'orthogonalité entre sous-espace signal et sous-espace bruit.

Définitions

Les objectifs du traitement d'antenne sont d'estimer les caractéristiques (direction d'arrivée, amplitude) de sources d'émission. Les signaux reçus dépendent de l'antenne (N capteurs identiques alignés et indicés par $i = 1, \dots, N$), le milieu, les sources.

En fixant à zéro la phase du signal pour le premier capteur (indiqué par 1) et en notant s l'amplitude du signal émis, le signal reçu par le premier capteur est $z(1) = s$.

Pour une source placée à un azimuth θ , le délai de réception td entre deux capteurs pour une onde plane provenant de cette source est : $td = \Delta * \sin(\theta)/c$ où Δ est la distance inter-capteur, et c la célérité de l'onde. Ce temps de délai conduit à une valeur de décalage de phase $\varphi = 2 * \pi * f * td$ où f est la fréquence d'émission de la source.

Un capteur d'indice i reçoit le signal :

$$z(i) = s \cdot \exp(j(i-1)\varphi), \quad i = 1, \dots, N. \quad (\text{B.1})$$

Quand il existe d sources d'émission, dans un environnement bruité, le signal reçu s'exprime de la façon suivante :

$$\mathbf{z} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \exp(j\varphi_1) & \exp(j\varphi_2) & \dots & \exp(j\varphi_d) \\ \dots & \dots & \dots & \dots \\ \exp(j(N-1)*\varphi_1) & \exp(j(N-1)*\varphi_2) & \dots & \exp(j(N-1)*\varphi_d) \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \dots \\ s_d \end{bmatrix} + \begin{bmatrix} n(1) \\ n(2) \\ \dots \\ n(N) \end{bmatrix} \quad (\text{B.2})$$

Où $\mathbf{z} = [z(1), \dots, z(N)]^T$, T signifiant transposée, est un vecteur contenant toutes les composantes de signal, $[s_1, \dots, s_d]^T$ est un vecteur contenant les valeurs d'amplitude des sources, et $[n(1), \dots, n(N)]^T$ est un vecteur contenant les composantes de bruit. En définissant $a_i(\varphi_k) = \exp(j(i-1)*\varphi_k)$, chaque composante $z(i)$ de \mathbf{z} est exprimée par :

$$z(i) = \sum_{k=1}^d a_i(\varphi_k) s_k + n(i), \quad i = 1, \dots, N. \quad (\text{B.3})$$

Nous obtenons un modèle pour le signal reçu par une antenne de N capteurs, quand d sources sont présentes.

On pose $\mathbf{a}(\varphi_k) = [a_1(\varphi_k), a_2(\varphi_k), \dots, a_N(\varphi_k)]^T$. Les signaux peuvent être exprimés sous la forme vectorielle suivante :

$$\mathbf{z} = \mathbf{A}(\Phi) \cdot \mathbf{s} + \mathbf{n}, \quad (\text{B.4})$$

$\mathbf{A}(\Phi) = [\mathbf{a}(\varphi_1), \dots, \mathbf{a}(\varphi_d)]$ est une matrice de taille $N \times d$ de type Vandermonde, appelée matrice de transfert ou matrice directionnelle.

B.2.1.2 Statistiques d'ordre deux : matrice interspectrale et distinction entre sous-espace signal et sous-espace bruit

A partir des signaux reçus sur une antenne, il est d'un grand intérêt d'exploiter des statistiques d'ordre deux. En particulier, la décomposition en éléments propres de la matrice interspectrale conduit à une distinction entre sous-espace signal et sous-espace bruit. Nous considérons un ensemble de K réalisations $\{\mathbf{z}_l, l = 1, \dots, K\}$. Une matrice interspectrale est exprimée comme suit :

$$\mathbf{R}_{zz} = \mathbb{E} [\mathbf{z}_l \mathbf{z}_l^H] \quad (\text{B.5})$$

où H signifie transposé conjugué. Une estimée en est donnée par :

$$\mathbf{R}_{zz} = \frac{1}{K} \sum_{l=1}^K \mathbf{z}_l \mathbf{z}_l^H, \quad (\text{B.6})$$

Sous forme matricielle :

$$\mathbf{R}_{zz} = \frac{1}{K} \mathbf{Z} \mathbf{Z}^H \quad (\text{B.7})$$

Le signal est supposé être indépendant du bruit ; les réalisations de bruit \mathbf{n}_l sont décorrélées, donc la matrice de covariance du bruit est diagonale. La matrice de covariance du bruit est $\sigma^2 \mathbf{I}$. La partie signal utile est supposée indépendante de la partie bruit, donc la matrice de covariance des signaux reçus est la somme de la matrice de covariance de la partie signal utile et de la matrice de covariance de la partie bruit. En utilisant Eq. (B.4) dans Eq. (B.7), on obtient :

$$\mathbf{R}_{zz} = \mathbf{A}(\Phi) \mathbf{R}_{ss} \mathbf{A}^H(\Phi) + \sigma^2 \mathbf{I} \quad (\text{B.8})$$

avec

$$\mathbf{R}_{ss} = \mathbb{E} [\mathbf{s}_l \mathbf{s}_l^H], \quad (\text{B.9})$$

et

$$\mathbb{E} [\mathbf{n}_l \mathbf{n}_l^H] = \sigma^2 \mathbf{I} \quad (\text{B.10})$$

La décomposition en éléments propres de la matrice de covariance est utilisée en général pour caractériser des sources par des méthodes par sous-espaces (sous-espace bruit, sous-espace signal) [115, 107, 114]. La décomposition en éléments propres de la matrice de covariance est exprimée comme suit :

$$\mathbf{R}_{zz} = \sum_{i=1}^N \lambda_i \mathbf{e}_i \mathbf{e}_i^H \quad (\text{B.11})$$

Où $\{\lambda_i, i = 1, \dots, N\}$ sont les valeurs propres de la matrice de covariance, et $\{\mathbf{e}_i, i = 1, \dots, N\}$ sont les vecteurs propres de la matrice de covariance.

Distinction entre sous-espace signal et sous-espace bruit

Les vecteurs propres de la matrice de covariance engendrent l'espace des mesures. Dans l'espace des mesures, l'on distingue le sous-espace signal et le sous-espace bruit. Quand la

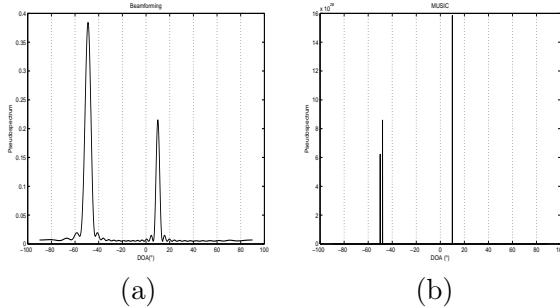


Figure B.1 — Pseudospectres : a)Formation de voies ; b)MUSIC

dimension du sous-espace signal est d , *i. e.*, quand il y a d sources dans le problème considéré, les d vecteurs propres associés aux d plus grandes valeurs propres engendrent le sous-espace signal, et les autres vecteurs propres recouvrent le sous-espace bruit.

B.2.1.3 Méthodes haute résolution

Nous présentons trois méthodes haute résolution. Leur propriété principale est qu'elles distinguent des valeurs de direction d'arrivée proches.

B.2.1.4 Méthode MUSIC

La méthode MUSIC (MULTiple SIgnal Characterization) sélectionne les vecteurs propres qui ne sont pas associés avec les valeurs propres dominantes, *i.e.* les vecteurs propres du sous-espace bruit. MUSIC est fondée sur l'orthogonalité entre sous-espace signal et sous-espace bruit.

La figure 1.2(a) présente le pseudospectre obtenu avec la méthode classique de la formation de voies [75], la figure 1.2(b) présente le pseudospectre obtenu avec la méthode haute résolution MUSIC. MUSIC résout deux sources de directions d'arrivée (-50° et -48°), alors que la méthode de formation de voies n'y parvient pas. On remarque que MUSIC exige la recherche de maxima du pseudospectre.

B.2.1.5 Méthode TLS-ESPRIT

TLS-ESPRIT est appliquée dans le cas particulier où l'antenne est composée de deux sous-antennes identiques [114]. Il existe une relation linéaire entre les mesures correspondant à la première antenne et les mesures correspondant à la deuxième antenne. L'expression des directions d'arrivée provient des valeurs propres d'une matrice qui exprime cette relation linéaire.

B.2.1.6 Méthode du Propagateur

La propriété principale de la méthode Propagateur est qu'elle n'est pas fondée sur la décomposition en éléments propres d'une matrice de covariance. Cette méthode est donc plus rapide que les autres méthodes haute résolution. Propagateur exploite une relation linéaire

entre les lignes de la matrice de transfert et repose sur la partition de cette matrice de transfert. Pour mettre en oeuvre la méthode, on construit à partir des données un opérateur, qui est une matrice orthogonale à tout vecteur directionnel.

B.2.1.7 Algorithme d'optimisation associé à une méthode d'interpolation par splines

DIRECT (Dividing RECTangles) est une méthode itérative d'optimisation globale et robuste au bruit, mais elle présente un inconvénient majeur : sa charge de calcul, qui augmente drastiquement quand le nombre d'inconnues augmente. Un algorithme d'interpolation fournit $N > O$ valeurs, à partir de O valeurs (ou noeuds). Nous proposons de réduire le nombre d'inconnues retrouvées par DIRECT ou l'algorithme du gradient, grâce à une interpolation par splines polynomiales cubiques [134], à chaque itération de DIRECT ou du gradient. Faire usage de l'interpolation accélère drastiquement l'algorithme DIRECT : moins il y a de noeuds d'interpolation, moins le résultat obtenu est fiable, mais plus l'algorithme est rapide.

B.2.1.8 Conclusion du chapitre 1

Nous avons introduit des concepts de base concernant le traitement d'antenne, et les méthodes haute résolution qui permettent d'estimer des valeurs de directions d'arrivée qui peuvent être proches. Ces méthodes sont fondées sur la distinction entre sous-espace signal et sous-espace bruit. Nous avons présenté une nouvelle méthode d'optimisation à la fois robuste et rapide. Dans la suite de ce travail de thèse nous faisons appel aux méthodes par sous-espaces et d'optimisation présentées dans le chapitre 1.

B.2.2 Résumé du chapitre 2 : Estimation de distorsions de phase par DIRECT associé à l'interpolation par splines

Une cause importante de mauvaises performances en localisation de sources est la déviation de l'antenne par rapport à sa structure rectiligne supposée. C'est le cas par exemple lorsque des antennes flexibles sont utilisées en acoustique sous-marine. On étudie ici la localisation de sources en présence d'erreurs de phase. Il est nécessaire de supprimer les fluctuations de phase d'un signal si l'on veut résoudre un problème de localisation, et en particulier distinguer des sources proches. Des travaux existants proposent des solutions adaptées au cas où les antennes comportent peu de capteurs. Nous proposons un nouvel algorithme qui est adapté aux antennes composées d'un nombre important de capteurs, et qui présente une charge de calcul faible. Notre méthode est fondée sur l'orthogonalité entre sous-espace bruit et sous-espace signal, et sur la méthode DIRECT accélérée par l'interpolation par splines. Nous illustrons les performances des méthodes proposées en les appliquant à la caractérisation de trois sources. Ce travail a été publié dans :

J. Marot and S. Bourennane, "Phase Distortion Estimation by DIRECT and spline interpolation algorithms", IEEE Signal Processing Letters, Vol. 14, no. 7, pp : 461-464, July 2007.

B.2.2.1 Position du problème

La figure 2.1 représente des fronts d'onde distordus incidents sur une antenne distordue (capteurs 1, 2, ..., N). Une limitation des méthodes haute résolution fondées sur l'estimation

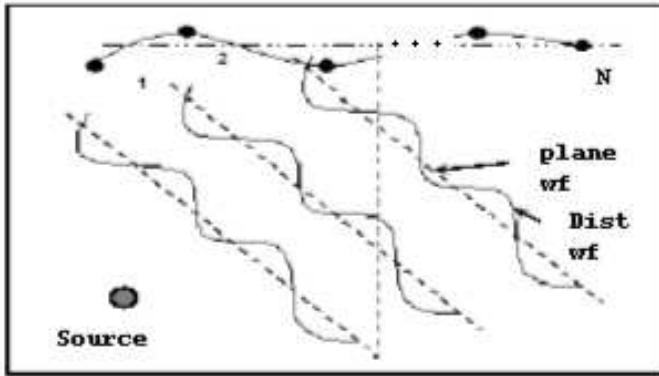


Figure B.2 — Fronts d'onde distordus reçus sur une antenne composée d'un grand nombre de capteurs.

d'une matrice de covariance est qu'elles ne donnent pas un résultat satisfaisant quand le front d'onde incident diffère du modèle de front d'onde à phase linéaire. Nous considérons le cas où l'antenne est composée de 100 capteurs.

B.2.2.2 Résultats : Caractérisation de trois sources

La figure 2.2 illustre le problème à traiter : plusieurs fronts d'onde distordus sont incidents sur une antenne distordue.

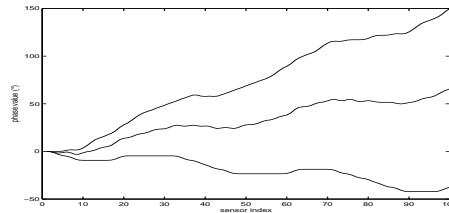


Figure B.3 — Evolution de la phase de trois fronts d'onde en fonction de l'indice des capteurs.

La figure 2.3 montre le pseudospectre de MUSIC obtenu lorsque trois sources sont présentes, et selon que les distorsions de phase du signal ne sont pas annulées, ou le sont de façon plus ou moins précise, en les estimant par un nombre plus ou moins grand d'itérations de la méthode d'optimisation.

B.2.2.3 Conclusion du chapitre 2

Un nouvel algorithme pour la caractérisation de fronts d'onde distordus et l'amélioration de la résolution en présence de distorsions de phase a été présenté. La combinaison de la

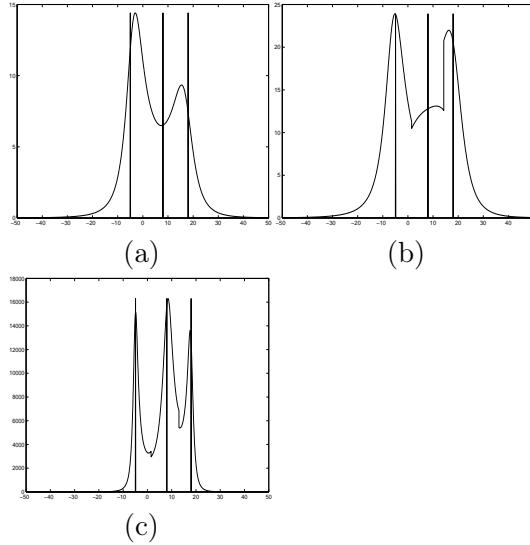


Figure B.4 — Pseudospectres de MUSIC, représentés dans l'intervalle $[-50^\circ, 50^\circ]$, superposés aux valeurs attendues (lignes verticales) : (a) sans annulation des distorsions de phase ; (b) et (c) avec l'estimation -10 (resp. 25) itérations de DIRECT- et annulation des erreurs de phase.

méthode DIRECT et de l'interpolation par splines conduit à un algorithme rapide même pour une antenne à grand nombre de capteurs.

B.2.3 Résumé du chapitre 3 : Estimation de contours rectilignes et approximativement rectilignes

Nous traitons de l'estimation de contours rectilignes et approximativement rectilignes dans des images par des méthodes haute résolution et d'optimisation. Il a été montré qu'un formalisme approprié transpose l'estimation de droites en un problème de traitement d'antenne. La méthode Subspace-based Line Detection (SLIDE) conduit à des modèles où l'orientation et l'offset des droites sont les paramètres à retrouver. Nous proposons plusieurs méthodes haute résolution qui sont adaptées à l'estimation d'orientations de droites. En particulier, nous proposons d'adapter la méthode haute résolution Propagateur. Nous utilisons les méthodes d'optimisation présentées dans le chapitre 1. En particulier la méthode DIRECT est robuste à la présence de bruit, et l'interpolation par splines aboutit à un contour continu, qui respecte des contraintes de continuité et de courbure.

Ce travail a été publié dans :

- S. Bourennane and J. Marot, "Estimation of straight line offsets by a high-resolution method", *IEE Vision Image and Signal Processing*, vol. 153, pp. 224-229, 2006., dans
- S. Bourennane and J. Marot, "Contour estimation by array processing methods", *Applied signal processing*, article ID 95634, 15 pages, 2006., dans
- J. Marot and S. Bourennane, "Propagator method for an application to contour estimation", *Pattern Recognition Letters*, March 2007, dans
- S. Bourennane and J. Marot, "Line parameters estimation by array processing methods", *IEEE ICASSP'05*, Vol. 4, pp. 965-968, Mar. 2005, et dans

S. Bourennane and J. Marot, "Optimization and interpolation for distorted contour estimation", IEEE ICASSP'06, Vol. 2, pp. 717-20, April 2006.

B.2.3.1 Modèle de données, génération de signal à partir de l'image

Soit I l'image à traiter (voir la figure 3.1 (a)). Il est possible d'établir une analogie entre localisation de sources [107, 133] en traitement d'antenne et la détection de droite en traitement d'image.

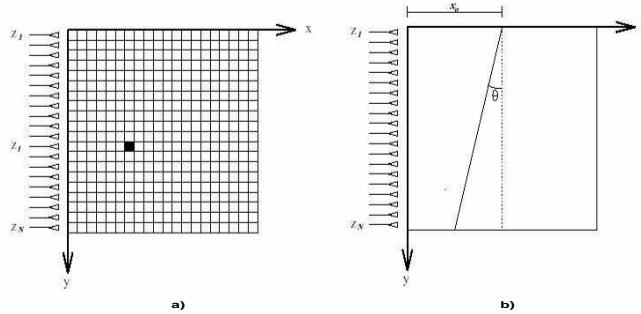


Figure B.5 — Modèle d'image : (a) Image-matrice muni d'un système de coordonnées et d'une antenne rectiligne de N capteurs équidistants. (b) Un droite caractérisée par son angle θ et son offset x_0 .

Pour cela, des signaux sont générés à partir de l'image [3] : on crée artificiellement, à partir des N lignes de l'image, N entrées pour une antenne composée de N capteurs équidistants positionnés le long de l'image. Un formalisme adopté dans [5] permet de générer un signal, par le calcul suivant :

$$z(i) = \sum_{k=1}^C I(i, k) \exp(-j\mu k), \quad i = 1, \dots, N \quad (\text{B.12})$$

Où $\{I(i, k); i \in \{1, \dots, N\}; k \in \{1, \dots, C\}\}$ dénotent les valeurs des pixels de l'image, et où μ est un paramètre de propagation. Quand une image contient d droites et des pixels de bruit, Eq. (3.1) suit un modèle de traitement d'antenne (voir Eq. (B.4)). Donc nous appliquons des méthodes haute résolution aux signaux générés, pour obtenir les paramètres orientation et offset des droites à retrouver. En particulier, les méthodes haute résolution distinguent des droites d'orientation pouvant être très proches. Nous proposons d'associer une méthode haute résolution appelée MFBLP (Modified Forward Backward Linear Prediction) à un schéma de génération à vitesse variable, pour estimer les offsets de droites parallèles pouvant être très proches. La méthode proposée est plus rapide que la méthode de l'extension de la transformée de Hough, en particulier quand des images bruitées sont considérées.

B.2.3.2 Estimation de contours distordus

Nous proposons d'adapter des algorithmes d'optimisation pour faire face au cas de contours qui ne sont plus simplement rectilignes, mais approximativement rectilignes et distordus. L'algorithme d'optimisation proposé est appliqué à un signal obtenu par schéma de génération à

vitesse constante, il est initialisé par une droite qui passe par une portion approximativement rectiligne du contour à retrouver, et dont les paramètres sont fournis par l'algorithme d'estimation de droites proposé. La figure 3.9 illustre la méthode proposée par un exemple de contour distordu et une droite d'initialisation.

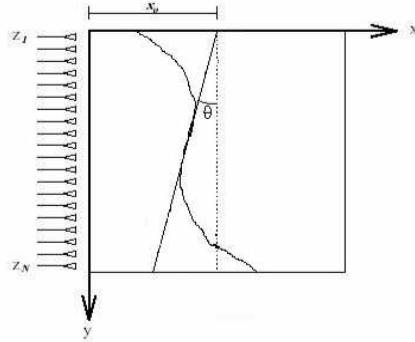


Figure B.6 — Un modèle d'image contenant un contour distordu.

B.2.3.3 Quelques résultats : estimation de contours rectilignes et approximativement rectilignes

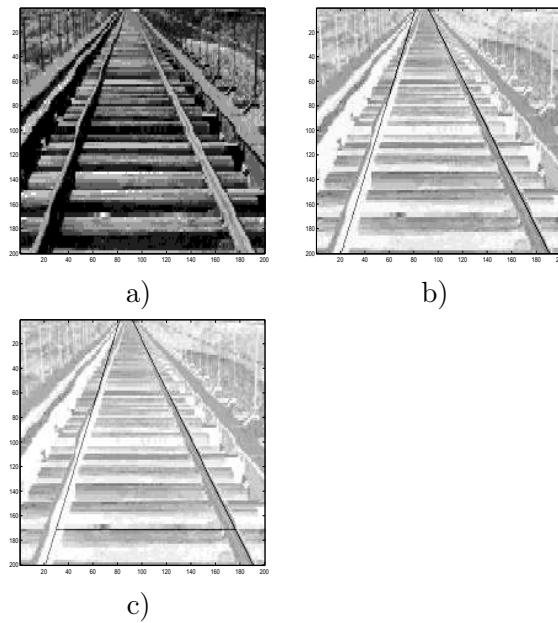


Figure B.7 — (a) Image transmise à la commande automatique d'un véhicule se déplaçant sur une voie ferrée. (b) Détection des rails. (c) Localisation de la première traverse.

La figure 3.2(a) est une photographie prise par une caméra et transmise à la commande automatique d'un véhicule se déplaçant sur les rails. L'estimation des deux valeurs d'orientation, incluant la génération de signal et la méthode TLS-ESPRIT, requiert 0.063 sec. L'estimation des valeurs d'offset requiert 1.1 sec. Pour comparaison, la méthode de l'extension de la transformée de Hough, utilisée avec la connaissance *a priori* des angles requiert 47 sec. pour trouver les deux valeurs d'offset.

La figure 3.24 présente les résultats obtenus sur des images aériennes. Le contour d'initialisation et le résultat final sont superposés à l'image à traiter. La colonne (a) contient l'image à traiter, la colonne (b) contient l'initialisation par la méthode proposée, la colonne (c) contient le résultat final obtenu par la méthode proposée, la colonne (d) contient l'initialisation choisie pour l'algorithme GVF, et qui permet à GVF d'obtenir un bon résultat final, la colonne (e) contient le résultat obtenu par GVF.

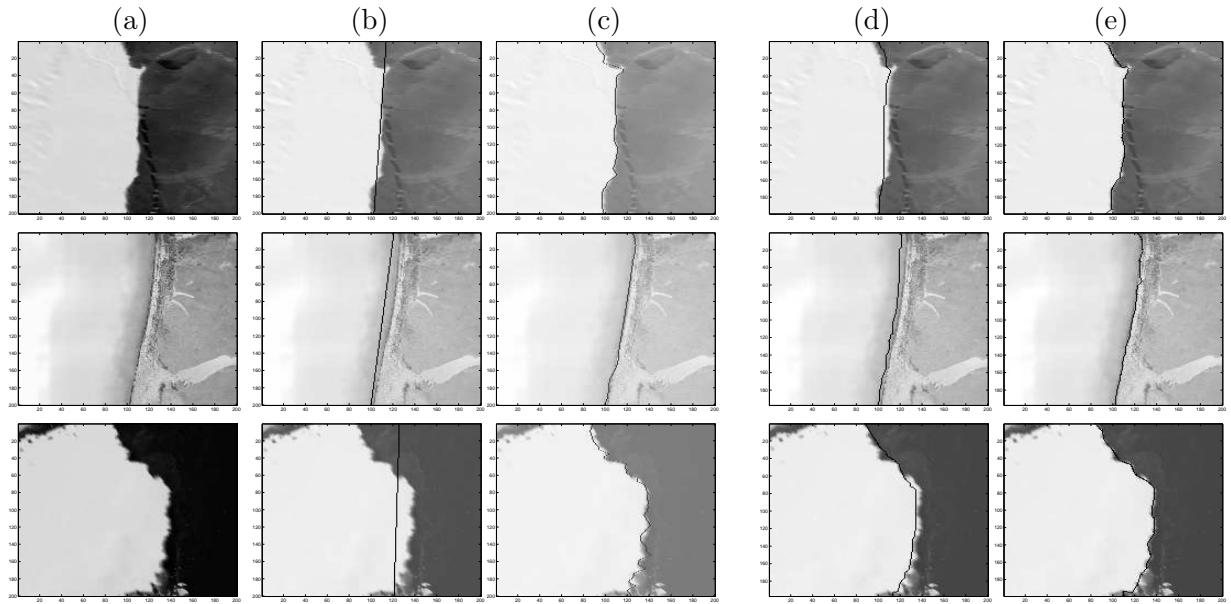


Figure B.8 — Analyse d'images aériennes : détection de côte. (a) image à traiter, (b) initialisation, (c) résultat obtenu avec la méthode proposée, (d) initialisation pour GVF, (e) résultat obtenu avec GVF.

B.2.3.4 Conclusion du chapitre 3

Nous avons considéré un problème d'analyse d'image : l'estimation de droites et de contours approximativement rectilignes et distordus. Pour cela, nous avons adapté des méthodes par sous-espaces du traitement d'antenne, et des méthodes d'optimisation. Pour transcrire le contenu de l'image en un signal unidimensionnel, nous avons adopté un schéma de génération à vitesse constante et un schéma de génération à vitesse variable. Ces schémas de génération conduisent à des modèles de signaux où les orientations et offsets des droites sont des valeurs de fréquence à estimer. Ces paramètres peuvent être de valeur très proches, et nous considérons des environnements bruités. Or, les méthodes haute résolution du traitement d'antenne sont censées donner de bons résultats dans ces conditions. Donc, nous avons adapté les méthodes TLS-ESPRIT et Propagateur pour estimer l'orientation de droites. En particulier, la version du propagateur, qui présente l'avantage de ne pas passer par le calcul d'une matrice de covariance, présente une charge de calcul faible. Nous avons également adapté la méthode haute résolution MFBLP à l'estimation de valeurs d'offsets (pouvant être proches) de droites parallèles.

Nous avons proposé une méthode d'optimisation pour estimer des contours distordus, et nous avons adapté l'algorithme d'optimisation du gradient, pour estimer des distorsions de phase à partir du signal généré à partir de l'image. Les distorsions de phase sont proportionnelles aux décalages entre la courbe distordue à retrouver et la droite d'initialisation. Cette procédure est généralisée à l'estimation de plusieurs contours par l'association des méthodes Propagateur et du gradient. Nous proposons également un nouvel algorithme d'optimisation qui est rapide, et conduit à un contour continu. Nous avons comparé les résultats obtenus avec cette méthode avec les résultats obtenus avec la méthode GVF (Gradient Vector Flow). En particulier, la méthode proposée estime des contours de courbure élevée, et donne de bons résultats avec des photographies de scènes. Ces méthodes reposent sur la génération de signal sur une antenne linéaire. Nous proposons dans la suite une méthode de génération de signal virtuel sur une antenne circulaire.

B.2.4 Résumé du chapitre 4 : Estimation de contours circulaires et approximativement circulaires

Les formes circulaires sont recherchées couramment en traitement d'images. Nous proposons une nouvelle méthode pour estimer plusieurs valeurs de rayons, pouvant être proches, et étendons l'estimation de cercles à l'estimation de contours approximativement circulaires et distordus. En particulier, nous développons et validons un nouveau modèle pour la génération de signal virtuel par une antenne circulaire.

Cette antenne circulaire conduit à des signaux à phase linéaire qui suivent un modèle auquel sont adaptées les méthodes haute résolution. Donc, les mêmes méthodes haute résolution et d'optimisation que celles qui ont été appliquées dans le cas de l'estimation de contours rectilignes et approximativement rectilignes peuvent être adaptées à l'estimation de contours circulaires et approximativement circulaires.

Nous proposons une méthode d'estimation du centre par le biais de génération de signal sur une antenne linéaire. Un schéma de génération à vitesse variable sur l'antenne circulaire proposée fournit un signal à phase linéaire, dont les fréquences sont proportionnelles aux valeurs des rayons des cercles présents dans l'image. La méthode du gradient, ou la méthode plus robuste combinant DIRECT et interpolation par splines étendent la méthode proposée

à la segmentation de contours étoilés quelconques. L'estimation de contours multiples pouvant ne pas être concentriques est aussi considérée. Pour évaluer les performances des méthodes proposées, nous les comparons à des méthodes moindres carrés, à la transformée de Hough, et à GVF. Nous appliquons la méthode proposée à des images artificielles, et à des photographies de scènes.

Ce travail a été publié dans :

J. Marot and S. Bourennane, "Subspace-Based and DIRECT Algorithms for Distorted Circular Contour Estimation", IEEE trans. on Image Processing, vol. 16, no. 9, pp. 2369-2378, september 2007, dans

J. Marot and S. Bourennane, "Array processing and fast Optimization Algorithms for Distorted Circular Contour Retrieval", Advances in signal processing, article ID 57354, 13 pages, february 2007, et dans

J. Marot and S. Bourennane, "Array processing approach for object segmentation in images", IEEE ICASSP'07, Vol. 1, pp. 621-24, April 2007.

B.2.4.1 Position du problème et génération de signal

Position du problème

Nous considérons le problème de l'estimation de rayon, et de l'estimation de distorsions entre un contour étoilé et un cercle qui est une approximation de ce contour. Nous proposons une antenne circulaire qui permet une génération de signal particulière. Nous mettons en valeur les propriétés de la phase des signaux générés. Ces signaux suivent des modèles auxquels sont adaptées les méthodes haute résolution du traitement d'antenne, et les méthodes d'optimisation. La figure 4.1(a) présente une image binaire I comme une matrice de dimensions $N \times N$. Chaque élément représente un pixel d'image. Un objet dans l'image est composé de pixels frontière de valeur 1, et le fond est composé de pixels de valeur 0. L'objet coïncide avec un cercle de rayon r et de centre de coordonnées (l_c, m_c) . La figure 4.1(b) montre une sous-image extraite de l'image de départ, de telle sorte que son extrémité supérieure gauche est le centre du cercle. Cette sous-image est associée à un repère de coordonnées polaires (ρ, θ) . Chaque pixel du contour à retrouver a les coordonnées suivantes : $r + \Delta\rho, \theta$. $\Delta\rho$ est le décalage entre un pixel du contour et le pixel du cercle qui a la même coordonnée θ . Nous recherchons des contours étoilés, *i. e.* des contours décrits par la relation $\rho = f(\theta)$, où f est une fonction de $[0, 2\pi]$ dans \mathbb{R}_+ .

Génération de signal

Nous faisons une analogie entre l'estimation d'un contour circulaire dans une image et la caractérisation de front d'onde en traitement d'antenne. Notre idée est d'obtenir un signal à phase linéaire à partir d'une image contenant un quart de cercle. Pour cela, nous utilisons une antenne circulaire. Selon que l'on met en oeuvre un schéma de génération à vitesse constante ou variable, la phase du signal généré est constante ou varie linéairement en fonction de l'indice des capteurs. Un quart de cercle de rayon r et une antenne circulaire sont représentés sur la figure 4.2. L'antenne a la forme d'un quart de cercle, centré sur le côté supérieur gauche, et qui

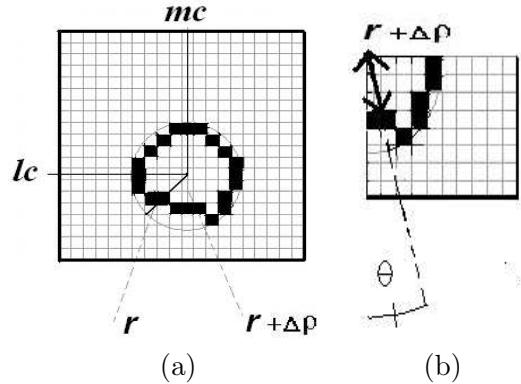


Figure B.9 — (a) Contour approximativement circulaire ; (b) Quart inférieur droit du contour, et coordonnées de pixels dans un repère de coordonnées polaires (ρ, θ) centré sur le centre du cercle. r est le rayon du cercle. $\Delta\rho$ est la valeur de décalage entre un pixel du contour et le pixel du cercle qui a la même valeur de coodonnée θ .

passe par le côté inférieur droit de la sous-image extraite. Une telle antenne est adaptée aux sous-images contenant chaque quart du contour à retrouver (voir la figure 4.2). En pratique, la sous-image extraite est tournée de une ou plusieurs fois 90° de façon à ce que le côté supérieur gauche corresponde au centre estimé. L'antenne a un rayon R_a tel que $R_a = \sqrt{2}N_s$, où N_s est le nombre de lignes ou colonnes dans la sous-image. Quand nous considérons la sous-image qui inclut la partie inférieure droite du contour à retrouver, la relation suivante est valable : $N_s = \max(N - l_c, N - m_c)$ où l_c et m_c sont les coordonnées verticale et horizontale du centre du contour à retrouver dans un repère cartésien centré sur le côté supérieur gauche de l'image à traiter de départ (voir la figure 4.1). Les coordonnées l_c et m_c sont estimées par une méthode passant par la génération d'un signal sur une antenne linéaire.

Le principe de génération de signal sur une antenne circulaire est le suivant : les directions adoptées pour la génération partent du côté supérieur gauche de la sous-image et atteignent chacune un capteur. L'antenne est composée de S capteurs, donc il y a S composantes de signal.

Soit D_i , la direction faisant un angle θ_i avec l'axe vertical et qui passe par l'angle supérieur gauche de l'image. La i^{me} composante ($i = 1, \dots, S$) du signal \mathbf{z} généré à partir de l'image est donnée par :

$$z(i) = \sum_{\substack{l,m=1 \\ (l,m) \in D_i}}^{l,m=N_s} I(l,m) \exp(-j\mu\sqrt{l^2 + m^2}), \quad (\text{B.13})$$

L'entier l (resp. m) indice les lignes (resp. les colonnes) de l'image. j vaut pour $\sqrt{-1}$. μ est le paramètre (ou constante) de propagation [7]. Chaque capteur indicé par i est associé à une direction de génération de signal D_i dont l'orientation est $\theta_i = \frac{(i-1)\cdot\pi/2}{S}$. Dans l' Eq. (4.1), le terme $(l,m) \in D_i$ signifie que seuls les pixels de l'image qui appartiennent à D_i sont considérés pour la génération de la i^{me} composante de signal. Les signaux générés sur l'antenne circulaire proposée conviennent aux méthodes haute résolution, qui estiment des valeurs de rayon pouvant être proches.

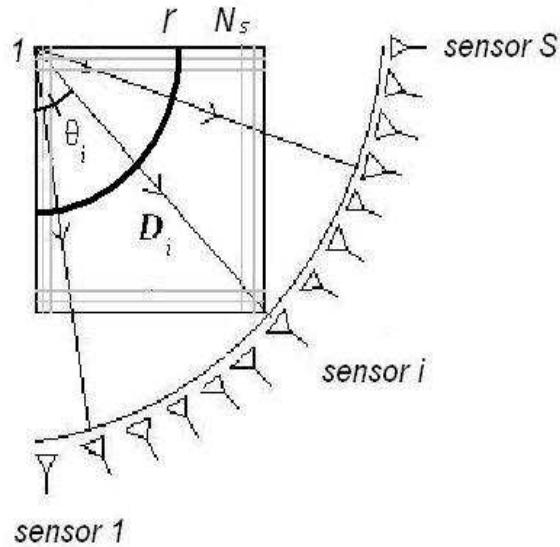


Figure B.10 — Sous-image, associée à une antenne circulaire comportant S capteurs.

B.2.4.2 Estimation de cercles multiples avec des centres et des rayons différents

De façon générale, une image contient plusieurs cercles qui peuvent être concentriques ou pas, et avoir des rayons différents ou pas (voir Fig. 4.3). Pour appliquer la méthode proposée passant par la génération d'un signal sur une antenne circulaire, les coordonnées du centre de chaque cercle sont nécessaires. Pour estimer ces coordonnées, nous générerons un signal à paramètre constant sur le côté gauche et le côté supérieur de l'image. La l^{me} composante de signal, générée à partir de la l^{me} ligne de l'image, est donnée par : $z_{lin}(l) = \sum_{m=1}^N I(l, m) \exp(-j\mu m)$, où μ est le paramètre de propagation [5]. Les sections non nulles des signaux, comme il est montré sur les côtés gauche et droit de l'image (voir la figure 4.3), indiquent la présence d'objets. La largeur de chaque section non nulle sur le signal généré sur la gauche (resp. sur le dessus) de l'image donne la hauteur (resp. la largeur) de l'objet correspondant. Le milieu de chaque section non nulle sur le signal du côté gauche (resp. du côté supérieur) conduit à une valeur approximative de la coordonnée l_c (resp. m_c) du centre de chaque objet. La hauteur, la largeur et le centre de chaque objet permet de sélectionner des régions d'images et de traiter ces régions successivement. Chaque région contient un contour ou un ensemble de contours concentriques. La méthode proposée, fondée sur une antenne circulaire, détecte le nombre de cercles concentriques pour chaque centre estimé, dans chaque région d'image. Les valeurs des rayons des cercles qui recouvrent au mieux les objets sont alors estimées. Ensuite, une méthode d'optimisation peut être utilisée pour affiner l'estimation de chaque contour.

B.2.4.3 Quelques résultats

Nous avons étudié la robustesse de la méthode proposée et de "gradient vector flow" à la dégradation par un bruit additif et à l'importance de la courbure du contour à retrouver. Les résultats statistiques présentés ci-dessous sont obtenus avec 15 images contenant chacune un

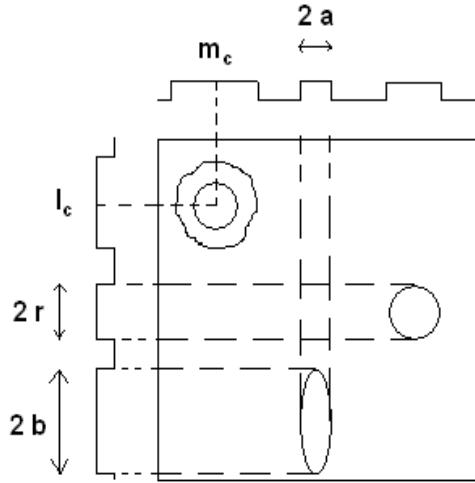


Figure B.11 — Modèle pour une image contenant plusieurs contours approximativement circulaires ou elliptiques. r est le rayon du cercle, a et b sont les paramètres axiaux de l'ellipse.

cercle distordu différent, caractérisé par une amplitude de distorsions variable. L'amplitude de distortion maximale et l'écart type des distorsions est différente pour chaque image. Les contours les moins et les plus distordus sont présentés sur les figures B.12(a) et (b) respectivement. La figure B.12 présente aussi les résultats obtenus avec la méthode proposée. Un bruit aléatoire est ajouté à un pourcentage variable des pixels de l'image : 0, 1, 2, 4, 6, et 8%. Les images non bruitées d'origine présentent des valeurs de pixel 1 (contour à retrouver) ou 0 (fond d'image). Les images sont dégradées par un bruit Gaussien de moyenne m_{No} et d'écart type σ_{No} égaux respectivement à 20% et 1% de la valeur d'un pixel du contour à retrouver. Pour chaque quart d'image, les temps de calcul sont respectivement 0.28 sec. pour la génération de signal, 0.15 sec. pour faire tourner l'algorithme du gradient à pas variable -50 itérations sont nécessaires- ; et 12 sec. pour faire tourner l'algorithme GVF. L'algorithme proposé est donc cinq fois plus rapide que GVF. Les résultats sont donnés en terme de valeur moyenne d'erreur sur la coordonnée radiale de chaque pixel. Cette valeur moyenne d'erreur est notée ME .

Le tableau B.1 présente les valeurs moyennes d'erreur ME pour les paramètres de distorsion et de pourcentage de bruit. Le tableau B.1 présente les valeurs d'erreur moyenne ME pour toutes les images et valeurs de pourcentage de bruit. La première colonne de la table B.1 présente le couple de valeurs (amplitude de distortion maximale ; écart type des distorsions), pour toutes les images. Le tableau B.1 montre que les valeurs d'erreur moyenne sont comprises entre 1.44 et 3.43 pixels pour la méthode proposée et entre 1.40 et 4.33 pixels pour GVF. Les valeurs moyennes ME_d des valeurs d'erreur ME obtenues pour chaque amplitude de distorsion montrent que GVF est limité par les valeurs de courbure élevées. Les valeurs d'erreur

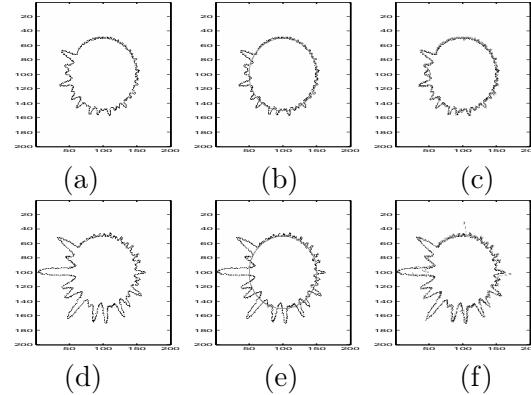


Figure B.12 — Exemples d’images à traiter contenant le contour le moins (a) et le plus distordu (d), l’initialisation (b,e) et l’estimation finale utilisant la méthode proposée (c,f).

Valeurs moyennes d’erreur : $ME_\rho = 1.4$ pixel et 2.7 pixels.

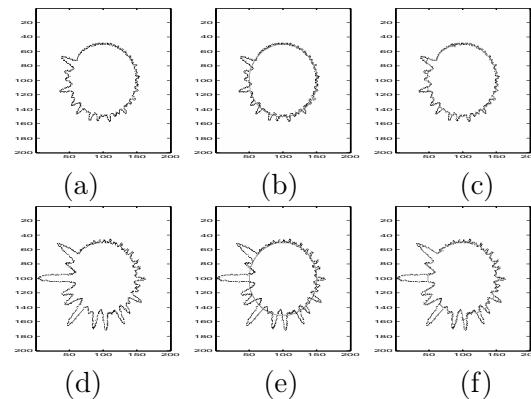


Figure B.13 — Exemples d’images à traiter contenant le contour le moins (a) et le plus distordu (d), l’initialisation (b,e) et l’estimation finale utilisant la méthode proposée (c,f).

Valeurs moyennes d’erreur : $ME_\rho = 1.4$ pixel et 4.1 pixels.

augmentent rapidement quand la valeur d’amplitude de distortion maximale augmente. Les valeurs moyennes ME_n des valeurs d’erreur obtenues pour chaque valeur de pourcentage de bruit mettent en lumière le fait que la méthode proposée obtient de meilleurs résultats que GVF. Les erreurs obtenues avec GVF sont la conséquence de son incapacité à progresser dans les sections à forte concavité, alors que la méthode proposée estime les valeurs de distortion quelles qu’elles soient. Les erreurs obtenues avec la méthode proposée viennent de fluctuations de phase inattendues : le bruit dégrade la phase des signaux générés.

n. dist.	Valeurs de ME (pixels)													
	0%		1%		2%		4%		6%		8%		Mean ME_d	
(9.28 ; 1.66)	(A) 1.44	(B) 1.40	(A) 1.83	(B) 1.42	(A) 2.00	(B) 1.43	(A) 2.05	(B) 1.44	(A) 2.08	(B) 1.45	(A) 2.10	(B) 1.47	(A) 1.91	(B) 1.43
(11.84 ; 1.97)	(A) 1.51	(B) 1.62	(A) 1.96	(B) 1.64	(A) 2.06	(B) 1.66	(A) 2.07	(B) 1.68	(A) 2.12	(B) 1.69	(A) 2.23	(B) 1.70	(A) 1.99	(B) 1.66
(14.68 ; 2.42)	(A) 1.68	(B) 1.67	(A) 1.98	(B) 1.69	(A) 2.09	(B) 1.72	(A) 2.10	(B) 1.72	(A) 2.16	(B) 1.73	(A) 2.27	(B) 1.74	(A) 2.04	(B) 1.71
(17.76 ; 2.93)	(A) 1.70	(B) 1.78	(A) 2.00	(B) 1.85	(A) 2.11	(B) 1.86	(A) 2.12	(B) 1.87	(A) 2.18	(B) 1.88	(A) 2.28	(B) 1.89	(A) 2.06	(B) 1.85
(18.36 ; 3.46)	(A) 1.72	(B) 2.07	(A) 2.02	(B) 2.08	(A) 2.13	(B) 2.08	(A) 2.14	(B) 2.08	(A) 2.19	(B) 2.11	(A) 2.30	(B) 2.12	(A) 2.08	(B) 2.09
(21.05 ; 3.07)	(A) 1.81	(B) 2.10	(A) 2.17	(B) 2.10	(A) 2.24	(B) 2.10	(A) 2.25	(B) 2.11	(A) 2.26	(B) 2.12	(A) 2.32	(B) 2.15	(A) 2.17	(B) 2.11
(21.55 ; 3.62)	(A) 1.82	(B) 2.28	(A) 2.18	(B) 2.29	(A) 2.26	(B) 2.31	(A) 2.27	(B) 2.32	(A) 2.32	(B) 2.33	(A) 2.38	(B) 2.34	(A) 2.20	(B) 2.31
(24.88 ; 4.25)	(A) 1.84	(B) 2.59	(A) 2.19	(B) 2.61	(A) 2.30	(B) 2.62	(A) 2.32	(B) 2.63	(A) 2.55	(B) 2.64	(A) 2.56	(B) 2.65	(A) 2.29	(B) 2.62
(28.34 ; 5.01)	(A) 2.02	(B) 2.69	(A) 2.29	(B) 2.71	(A) 2.46	(B) 2.72	(A) 2.67	(B) 2.74	(A) 2.77	(B) 2.75	(A) 2.79	(B) 2.76	(A) 2.50	(B) 2.72
(31.85 ; 5.21)	(A) 2.18	(B) 3.04	(A) 2.37	(B) 3.11	(A) 2.52	(B) 3.12	(A) 2.85	(B) 3.14	(A) 2.89	(B) 3.18	(A) 2.90	(B) 3.19	(A) 2.61	(B) 3.13
(35.39 ; 5.86)	(A) 2.19	(B) 3.12	(A) 2.47	(B) 3.16	(A) 2.60	(B) 3.19	(A) 2.89	(B) 3.21	(A) 2.90	(B) 3.34	(A) 2.96	(B) 3.41	(A) 2.66	(B) 3.23
(38.90 ; 6.61)	(A) 2.30	(B) 3.43	(A) 2.51	(B) 3.51	(A) 2.62	(B) 3.55	(A) 2.95	(B) 3.59	(A) 3.00	(B) 3.61	(A) 3.01	(B) 3.62	(A) 2.73	(B) 3.55
(42.33 ; 7.42)	(A) 2.42	(B) 3.69	(A) 2.63	(B) 3.71	(A) 2.67	(B) 3.82	(A) 3.00	(B) 3.83	(A) 3.13	(B) 3.84	(A) 3.19	(B) 3.85	(A) 2.84	(B) 3.79
(45.63 ; 8.28)	(A) 2.43	(B) 4.05	(A) 2.89	(B) 4.06	(A) 2.96	(B) 4.07	(A) 3.14	(B) 4.08	(A) 3.20	(B) 4.09	(A) 3.34	(B) 4.09	(A) 2.99	(B) 4.07
(48.75 ; 9.22)	(A) 2.77	(B) 4.21	(A) 3.11	(B) 4.22	(A) 3.16	(B) 4.23	(A) 3.30	(B) 4.27	(A) 3.38	(B) 4.28	(A) 3.43	(B) 4.33	(A) 3.19	(B) 4.25
Mean ME_n	1.98	2.64	2.30	2.67	2.41	2.69	2.54	2.71	2.60	2.73	2.67	2.75	2.41	2.69

Tableau B.1 — Valeurs d'erreur ME (en pixel) obtenues avec la méthode proposée (a) et avec GVF (b), en fonction des paramètres de distortion (amplitude maximum ; écart type) -dist.-, et du pourcentage de bruit -n.-.

B.2.4.4 Conclusion du chapitre 4

Nous avons considéré l'estimation de contours circulaires distordus dans des images par des méthodes de traitement d'antenne et d'optimisation. Nous avons démontré l'efficacité d'une antenne circulaire pour la génération de signaux à phase linéaire à partir d'images contenant des contours circulaires. Ceci permet d'utiliser des méthodes haute résolution et des méthodes d'optimisation pour l'estimation de contours circulaires distordus dans des images. Un schéma de propagation à vitesse variable et le critère MDL estiment le nombre de cercles et la valeur des rayons. En mettant en oeuvre un schéma de propagation à vitesse constante et une méthode d'optimisation, nous retrouvons des contours étoilés. L'algorithme du gradient, ou DIRECT combiné à l'interpolation par splines, estiment les décalages entre un cercle d'initialisation et le contour à retrouver. Nous généralisons ces algorithmes à l'estimation de cercles multiples avec des centres et des rayons différents, et à l'estimation d'ellipses inclinées. Nous avons appliqué les méthodes proposées à des images artificielles et à des photographies, et comparé les résultats obtenus avec ceux obtenus par estimation des moindres carrés, transformée de Hough, et "gradient vector flow". L'estimation d'un cercle dans une image bruitée a permis d'illustrer la robustesse au bruit de notre méthode. En considérant des cercles concentriques, nous avons montré la capacité de nos méthodes à distinguer deux cercles concentriques. Par une expérience concernant un contour interrompu dans une image bruitée, nous avons montré que DIRECT combiné à l'interpolation par splines conduit à un résultat continu dans des conditions difficiles. La méthode de génération de signal proposée et notre méthode d'optimi-

sation utilisant le gradient conduit à de meilleurs résultats en terme de biais sur l'estimation de la position des pixels que la méthode GVF, quand on considère des contours à courbure importante. Ces expériences montrent que la méthode proposée est rapide et prometteuse pour l'estimation de contours. Une perspective qui se dégage de ce travail consiste en l'estimation de contours circulaires ou elliptiques qui s'intersectent.

B.2.5 Résumé du chapitre 5 : Approximation de tenseur de rang inférieur et filtrage multimodal

Nous présentons des méthodes récentes de filtrage fondées sur l'approximation de rang inférieur pour débruiter des signaux tensoriels. Dans cette approche, les données multicomposantes sont représentées par des tenseurs, *i. e.* des tableaux multidimensionnels, et les méthodes de filtrage présentées reposent sur l'algèbre multilinéaire. Nous présentons les données tensorielles, et un résumé de leur propriétés. Nous introduisons la formulation tensorielle du problème classique de débruitage, et des notations propres au filtrage tensoriel. Premièrement, nous expliquons comment la SVD tranche par tranche traite successivement chaque composante du tenseur de données. Deuxièmement, nous considérons deux méthodes qui prennent en compte les relations entre chaque matrice tranche du tenseur considéré. Ces deux méthodes sont fondées sur le concept de sous-espace signal pour chaque n -mode. La première méthode pour l'estimation de tenseur est fondée sur une PCA (Principal Component Analysis) multimodale obtenue par une approximation de rang- (K_1, \dots, K_N) . La seconde méthode est une nouvelle version tensorielle du filtrage de Wiener. Nous présentons des résultats comparatifs où les différentes méthodes de filtrage multimodal sont appliquées à la réduction de bruit dans des images couleur, dans les images hyperspectrales, et dans les signaux sismiques multicomposante. Nous présentons une méthode de déploiement de tenseur non orthogonale, qui a pour but d'améliorer les résultats du débruitage de données quand des directions principales (lignes de fuite, frontières, limites) sont présentes dans le tenseur à traiter. Nous considérons une application à l'imagerie hyperspectrale.

Ce travail a été publié dans :

D. Muti, S. Bourennane and J. Marot, "Lower-Rank Tensor Approximation and Multiway Filtering", SIAM journal on applied mathematics, accepté Septembre 2007.

B.2.5.1 Les données tensorielles

Les données tensorielles étendent le concept classique de vecteur de données. La mesure d'un signal multidimensionnel \mathcal{X} par des capteurs multidimensionnels, auquel s'ajoute un bruit \mathcal{N} , résulte en un tenseur de données \mathcal{R} tel que :

$$\mathcal{R} = \mathcal{X} + \mathcal{N}. \quad (\text{B.14})$$

\mathcal{R} , \mathcal{X} et \mathcal{N} sont des tenseurs d'ordre N de $\mathbb{R}^{I_1 \times \dots \times I_N}$. Les tenseurs \mathcal{N} et \mathcal{X} représentent les parties bruit et signal des données respectivement. Nous supposons que le bruit \mathcal{N} est indépendant du signal \mathcal{X} , et que le rang n -modal K_n est inférieur à la dimension n -modale I_n ($K_n < I_n$, pour tout $n = 1$ à N). Donc il est possible d'étendre l'approche classique par sous-espaces aux données tensorielles en supposant que, quel que soit le n -mode, l'espace vectoriel $E^{(n)}$ est la somme directe de deux sous-espaces orthogonaux, *i. e.* $E_1^{(n)}$ et $E_2^{(n)}$, qui sont définis comme suit :

- $E_1^{(n)}$ est le sous-espace de dimension K_n , engendré par les K_n vecteurs singuliers associés aux K_n plus grandes valeurs singulières de la matrice \mathbf{X}_n ; $E_1^{(n)}$ est appelé le sous-espace signal [1, 89, 139, 138].
- $E_2^{(n)}$ est le sous-espace de dimension $I_n - K_n$, engendré par les $I_n - K_n$ vecteurs singuliers associés aux $I_n - K_n$ plus petites valeurs singulières de la matrice \mathbf{X}_n ; $E_2^{(n)}$ est appelé le sous-espace bruit [1, 89, 139, 138].

Les dimensions de sous-espace K_1, K_2, \dots, K_N peuvent être estimées par les critères statistiques AIC (Akaike Information Criterion) ou MDL (Minimum Description Length) [132], qui sont des critères basés sur l'entropie. Donc, une méthode pour estimer le tenseur signal \mathcal{X} à partir du tenseur de données \mathcal{R} est d'estimer $E_1^{(n)}$ dans chaque n -mode de \mathcal{R} .

B.2.5.2 Quelques résultats de débruitage de données tensorielles

L'objectif de la figure 5.3 est de comparer les résultats obtenus avec la SVD tranche par tranche, l'approximation de rang-(30, 30, 2), le filtrage de Wiener multimodal de rang-(30, 30, 2) (Wmm-(30, 30, 2)).

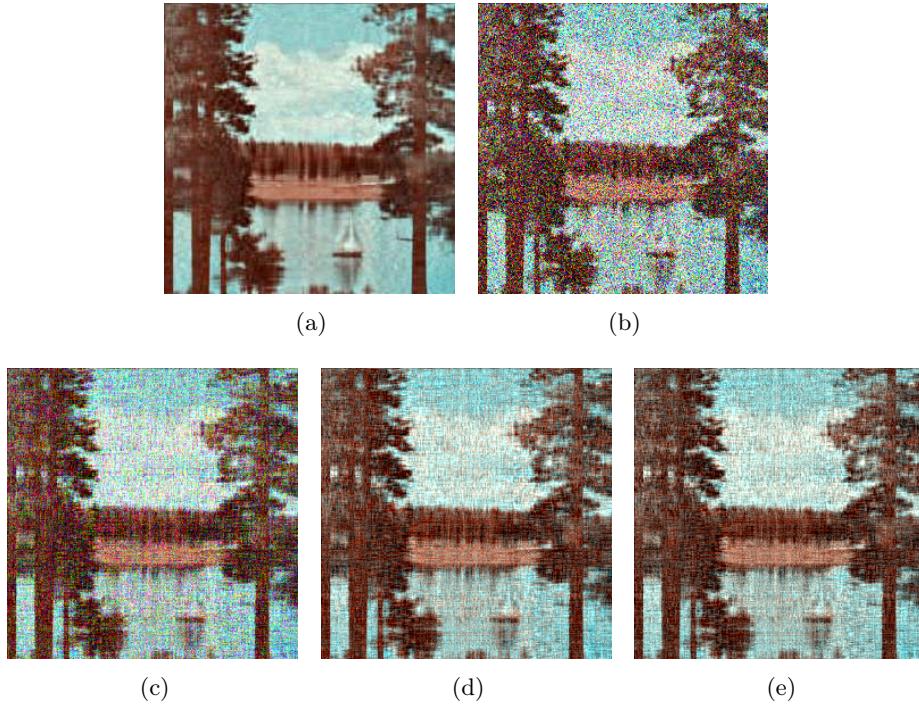


Figure B.14 — (a) Image non bruitée, (b) Image à traiter, dégradée par un bruit additif, Gaussien et blanc, avec un SNR=8.1dB. (c) Filtrage par SVD tranche par tranche de paramètre $K=30$. (d) Approximation de rang-(30, 30, 2) (e) Filtrage de Wiener multimodal de rang-(30, 30, 2).

D'après les résultats obtenus, la réduction de dimension conduit à une baisse de la résolution spatiale. Le choix des rangs n -modaux est établi par un compromis entre la préservation des détails et l'efficacité du débruitage. On note que le filtrage de Wiener multimodal conduit à une meilleure préservation des détails que les autres méthodes.

B.2.5.3 Conclusion du chapitre 5

Une vue d'ensemble de nouvelles méthodes mathématiques dédiées aux données multicomposantes est présentée. Les données multicomposantes sont représentées comme des tenseurs, *i. e.* des tableaux multidimensionnels, et les méthodes de filtrage tensoriel qui sont présentées reposent sur des méthodes d'algèbre linéaire, plus précisément des méthodes par sous-espaces, et des méthodes d'optimisation. Premièrement, nous montrons comment appliquer un filtrage par SVD tranche par tranche. Ensuite, nous présentons trois méthodes qui prennent en compte les relations entre chaque matrice tranche d'un tenseur à traiter.

La première méthode est une PCA multimodale appliquée par une approximation de rang- (K_1, \dots, K_N) au tenseur de données bruité, ou par une troncature de rang- (K_1, \dots, K_N) inférieur de sa HOSVD (higher order singular value decomposition ou décomposition en valeurs singulières d'ordre supérieur). Cette méthode est fondée sur des statistiques d'ordre deux et repose sur l'orthogonalité entre le sous-espace bruit et le sous-espace signal dans chaque n -mode. La seconde méthode consiste en une amélioration de la première, dans le sens où elle permet de traiter une image bruitée par un bruit Gaussien corrélé. Dans ce cas le calcul de statistiques d'ordre deux est remplacé par le calcul de cumulants d'ordre quatre. Nous avons décrit une procédure passant par le calcul de cumulants d'ordre quatre qui présente une charge de calcul faible, par l'utilisation d'une matrice tranche de cumulants. Cette méthode a donné de bons résultats sur des images et des signaux sismiques bruités par un bruit corrélé.

La troisième méthode considérée est une version multimodale du filtrage de Wiener. Le nom de cette méthode provient du critère minimisé, une erreur quadratique moyenne, entre le tenseur signal à retrouver et le tenseur estimé. Un algorithme ALS (Alternating Least Squares ou moindres carrés alternés) a été utilisé pour déterminer de façon optimale les filtres de Wiener n -modaux. Les performances de ce filtrage de Wiener multimodal et des résultats comparatifs avec la PCA multimodale ont été présentés dans le cas de la réduction de bruit blanc additif dans des images couleur et dans des signaux sismiques multicomposante.

Finalement, nous utilisons la méthode d'estimation de contours rectilignes pour améliorer le débruitage de données hyperspectrales dans le cas où des directions principales sont présentes dans l'image à traiter. Un déploiement non orthogonal des données, réalisé le long des directions principales des images hyperspectrales, améliore les résultats de débruitage.

Nous montrons que quand le déploiement est réalisé le long de directions de frontières dans l'image, on obtient de meilleures performances de débruitage par le filtrage de Wiener multidimensionnel. Les détails sont préservés, ce qui pourrait conduire à de futures applications en détection et classification d'objets dans les images hyperspectrales.

B.3 Conclusion générale

Cette thèse est dédiée à l'étude de méthodes par sous-espaces et d'optimisation, appliquées à des signaux unidimensionnels, bidimensionnels, et multidimensionnel. Nous rappelons les principes d'un problème en traitement d'antenne, nous prouvons l'intérêt d'une méthode d'optimisation rapide qui combine DIRECT et une méthode d'interpolation par splines pour la résolution d'un problème de caractérisation de sources, quand une antenne avec un grand nombre de capteurs est utilisée et en présence de distorsions de phase. En partant d'un formalisme dans lequel un signal est généré à partir d'une image, nous avons étendu une méthode d'estimation de contours rectilignes. Nous avons adapté des méthodes existantes de traite-

ment d'antenne au problème de l'estimation d'objets de forme quelconque dans des images. Finalement, nous avons présenté les dernières avancées des méthodes de traitement du signal tensoriel et proposé une amélioration dans le cas où des frontières rectilignes sont présentes dans le tenseur à traiter. Pour cela nous avons utilisé la méthode d'estimation de contour rectiligne présentée dans la première partie du manuscrit.

Ainsi, dans la première partie, nous nous sommes concentrés sur des signaux unidimensionnels. Dans le **chapitre 1**, nous avons rappelé les éléments et objectifs d'un problème de traitement d'antenne, la définition d'une antenne en tant qu'ensemble de capteurs, du milieu et des sources. Nous avons introduit les définitions de la matrice de covariance. Dans l'espace des mesures engendré par les vecteurs propres de la matrice de covariance, nous avons distingué le sous-espace signal (utile) et le sous-espace bruit. Nous avons mis l'accent sur l'orthogonalité entre sous-espace signal et sous-espace bruit lorsque le signal et le bruit sont décorrélés. En partant de cette hypothèse, la méthode par sous-espace MUSIC (MULTiple SIgnal Characte-riZation) estime des directions d'arrivée en testant l'orthogonalité entre un modèle de vecteur signal et les vecteurs du sous-espace bruit. Nous avons expliqué les principes d'une nouvelle méthode d'optimisation qui combine soit la méthode du gradient soit la méthode DIRECT combinée à l'interpolation par splines. Dans le **chapitre 2**, nous considérons l'estimation de directions d'arrivée dans le cas où des distorsions de phase sont présentes. Nous utilisons la combinaison de l'algorithme DIRECT (DIviding RECTangles) avec l'interpolation par splines pour estimer des distorsions de phase de fronts d'onde. Premièrement, une estimée grossière de plusieurs directions d'arrivée est obtenue par la méthode MUSIC. Ensuite, pour chaque valeur estimée, utilisée comme un paramètre d'initialisation, les valeurs de distorsions de phase entre un front d'onde plan et le front d'onde qui donne lieu aux mesures sont estimées par DIRECT combiné à l'interpolation par splines. Les décalages de phase sont supprimés dans les réalisations de signal. Finalement, la méthode MUSIC est appliquée dans un intervalle réduit autour des directions d'arrivée ayant servi d'initialisation. Du fait de l'annulation des distorsions de phase, plusieurs sources peuvent être résolues dans cet intervalle. L'algorithme proposé a été appliqué à la caractérisation de trois sources dans un environnement bruité.

Dans la seconde partie, nous avons proposé plusieurs méthodes pour des applications en analyse d'images - données bidimensionnelles. En particulier, nous avons décrit un formalisme existant pour la génération de signal à partir d'une image. Dans ce cadre de travail, un contour dans une image est considéré comme un front d'onde, dont la propagation est simulée via un schéma de propagation (ou schéma de génération de signal). Dans le **chapitre 3**, nous avons rappelé la méthode SLIDE (Subspace-based LIne DEtection) qui utilise l'algorithme TLS-ESPRIT (Total Least Square-Estimation of Signal Parameters via Rotational Invariance Techniques) pour retrouver l'orientation de lignes droites. Nous avons proposé une méthode plus rapide, appelée le Propagateur, qui évite la décomposition en éléments propres de la matrice de covariance, soit en réalisant une partition de cette matrice, soit en travaillant directement sur le signal. Le propagateur présente une charge de calcul faible. Pour l'estimation de contours rectilignes, nous avons aussi proposé une nouvelle méthode pour l'estimation d'offsets de lignes droites. Cette méthode est plus rapide que l'extension de la transformée de Hough, en particulier quand des images bruitées sont traitées, et distingue des droites parallèles. Nous avons étendu les méthodes d'estimation de droites à l'estimation de contours approximativement rectilignes. Pour cela, nous avons adapté la méthode du gradient à pas

variable pour minimiser un critère des moindres carrés entre un signal modèle et un signal générée à partir de l'image. Nous avons proposé une généralisation de cette méthode d'optimisation à l'estimation de plusieurs contours, en minimisant un critère fondé sur l'opérateur du propagateur. Nous avons proposé une méthode d'optimisation améliorée, qui combine la méthode du gradient et l'interpolation par splines. Cette méthode est rapide et robuste au bruit. Par une comparaison avec la méthode GVF, nous avons insisté sur les capacités de la méthode proposée à retrouver des contours de courbure importante dans un environnement bruité. Les méthodes d'estimation de contours rectilignes et approximativement rectilignes ont été appliquées à des images de vision robotique, à des images aériennes et d'imagerie médicale. Dans le **chapitre 4**, nous avons étendu le travail dédié aux contours rectilignes et approximativement rectilignes aux contours étoilés. L'aspect le plus nouveau du **chapitre 4** consiste en la génération d'un signal sur une antenne circulaire : nous adaptons une antenne circulaire à l'image à traiter, pour générer un signal à phase linéaire, et pour pouvoir utiliser des méthodes haute résolution pour l'estimation de valeurs de rayons de cercles concentriques pouvant être proches. La génération de signal sur une antenne linéaire placée sur les côtés de l'image permet de localiser le centre des objets : les sections non nulles des signaux générés sur les côtés gauche et supérieur de l'image indiquent la présence d'objets. La génération de signal sur les côtés de l'image donne également accès aux paramètres axiaux d'ellipses. Pour retrouver des ellipses inclinées, nous réalisons une rotation de l'image jusqu'à ce que la longueur des sections non nulles dans les signaux générés sur chaque côté soit de différence maximale. Quand des contours distordus sont attendus, des algorithmes d'optimisation analogues à ceux appliquées pour l'estimation de contours approximativement rectilignes sont appliqués aux signaux générés sur l'antenne circulaire. Nous considérons plusieurs cas d'application comme en biométrie, et contrôle non destructif.

La troisième partie est consacrée au traitement du signal tensoriel -de données multidimensionnelles-, et démontre l'utilité des méthodes d'estimation de directions privilégiées pour l'amélioration du débruitage par des méthodes par sous-espace. Premièrement, nous avons présenté les dernières avancées en traitement du signal tensoriel. Nous avons rappelé les principes de la troncature de la décomposition en valeurs singulières d'ordre supérieur, la HOSVD- (K_1, \dots, K_N) , qui repose sur la définition des dimensions de sous-espace signal K_1, \dots, K_N . Nous avons rappelé les principes de l'algorithme d'optimisation ALS (Alternating Least Squares), qui améliore la troncature de la HOSVD- (K_1, \dots, K_N) et en fait une approximation de rang- (K_1, \dots, K_N) inférieur d'un tenseur. Nous montrons comment les statistiques d'ordre supérieur, utilisées dans l'approximation de tenseur de rang- (K_1, \dots, K_N) à la place des statistiques d'ordre deux, améliore le débruitage de signal tensoriel dans le cas d'un bruit Gaussien corrélé. En particulier, nous proposons une version rapide de cette méthode : nous utilisons une matrice tranche de cumulants d'ordre quatre, à la place du tenseur de cumulants complet. Nous donnons des détails sur une version multilinéaire du filtrage de Wiener. Le filtrage de Winer multidimensionnel repose sur l'utilisation du sous-espace signal utile. L'estimation des projecteurs sur le sous-espace signal utile, pour chaque n -mode, est réalisée par une boucle ALS (Alternating Least Squares ou moindres carrés alternés). Nous avons réalisé plusieurs expériences sur des images couleur, en considérant en particulier des cas où la dimension du sous-espace signal est égale au nombre de canaux dans le mode couleur, dans le cas où la dimension du sous-espace signal est sous-estimée, ou dans le cas où l'image est dégradée par un bruit Gaussien corrélé. Nous avons aussi réalisé du débruitage sur des ondes sismiques multicomposante. Deuxièmement, nous avons proposé une amélioration

du débruitage de signal tensoriel, en utilisant le travail dédié à l'estimation de contours rectilignes : Quand des frontières rectilignes sont présentes dans le tenseur à traiter, nous montrons qu'un déploiement non orthogonal des données, et plus précisément un déploiement selon ces directions principales, améliore le résultat du débruitage.

Plusieurs perspectives se dégagent de ce travail. La méthode SLIDE a déjà été appliquée à l'estimation de paramètres cinématiques de trajectoires d'objets. Cela a conduit à de bons résultats dans le cas de mouvements à vitesse constante. Les avancées proposées dans la seconde partie concernant l'estimation de contour distordus pourraient être adaptées à la caractérisation de trajectoires à vitesse non constante.

Dans le domaine du traitement du signal tensoriel, la méthode proposée de déploiement non orthogonal de tenseur améliore le rapport signal sur bruit du tenseur signal obtenu. En particulier, les méthodes de traitement du signal tensoriel préservent les détails. Nous pourrions adapter cette méthode pour une application en détection de cible. En particulier, les données hyperspectrales requièrent un prétraitement tel que la réduction de bruit pour augmenter l'efficacité des algorithmes de classification. En effet, une procédure de réduction de bruit diminue la variabilité spectrale, ce qui est utile pour de futures applications en classification et détection. De plus, la taille importante des données hyperspectrales augmente la charge de calcul et dégrade les résultats de classification. Les images hyperspectrales contiennent des bandes fortement corrélées, ce qui résulte en une redondance spectrale. L'analyse en composantes indépendantes (ou Independent Component Analysis, ICA) est appliquée à la réduction du nombre de bandes spectrales en sélectionnant des composantes spectrales. Il pourrait être d'un grand intérêt de combiner les algorithmes de filtrage avec ICA, pour un débruitage et une réduction de dimension conjoints.

Liste des figures

1.1	Elements of an array processing problem. Δ is the inter-sensor distance, θ is the direction-of-arrival.	10
1.2	Pseudospectra : a)Beamforming; b)MUSIC	16
1.3	Principles of DIRECT method.	21
2.1	Distorted wavefronts received on a distorted antenna composed of a large number of sensors.	24
2.2	Evolution of the phase of the three impinging wavefronts as a function of the sensor index.	27
2.3	Pseudospectra of MUSIC, represented in the interval $[-50^\circ, 50^\circ]$, superimposed to the expected values (vertical lines) : (a) without cancellation of phase distortions ; (b) and (c) with the estimation with 10 (resp. 25) iterations of DIRECT and cancellation of the phase distortions.	27
3.1	The image model (see [4]) : (a) The image-matrix provided with the coordinate system and the rectilinear array of N equidistant sensors. (b) A straight line characterized by its angle θ and its offset x_0	35
3.2	(a) Image transmitted to the automatic command of a vehicle that is moving on a railway for the servicing of the railways. (b) Detection of the rails for the progress of the vehicle. (c) Localization of the first crosspiece that the vehicle has to replace. The process is iterated crosspiece after crosspiece : photography, detection of the rails and detection of the next crosspiece.	41
3.3	(a) Noisy image containing a convex polygon the summits of which we aim at detecting. (b) Superposition of the segments that fit the sides of the polygon, estimated by our method, and the original image.	41
3.4	The main direction of a set of points	42
3.5	(a) Two parallel curves : orientation estimation by Propagator and (b) Offset estimation by MFBILP, (c) Offset estimation by Hough transform.	42
3.6	(a) Two nearly parallel curves. (b) Orientation estimation by Propagator, offset estimation by MFBILP.	43
3.7	Proposed method : region segmentation by global threshold.	46

3.8	Detection of the pectoral muscle : (a) Processed image (b) Superposition of the initialization image and the edge enhanced processed image (c) Superposition of the estimation by automatic threshold and the edge enhanced processed image (d) Result obtained with levelset method.	47
3.9	A model for an image containing a distorted curve.	49
3.10	An image the contour of which exemplifies the distorted curves our method can cope with.	55
3.11	(a)Initialization of the method (b)Determination of the vector $\hat{\mathbf{x}}$ (c)Junction of the different parts of the curve by determining the coefficients p_l	55
3.12	An image and the steps of the method : Original image, initialization, estimation obtained before sticking sections together, final estimation. The method manages to return the shifts.	56
3.13	The main results obtained by distorted contour estimation : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Difference between the initial image and the estimation	56
3.14	Distorted contour estimation on a noisy image : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Superposition of the initial image and the estimation.	57
3.15	The case of an image containing two distorted curves : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Difference between the initial image and the estimation	57
3.16	Results obtained by the method for the estimation of several curves. One curve is a straight line : a)Image to be treated b)Initialization c)Estimation obtained by the proposed method d)Difference between the initial image and the estimation	58
3.17	A practical situation is simulated. A vehicle and the two sides of the road are retrieved : a)Initial image b)First estimation of the borders of the road c)Estimation of the sides of the vehicle d)Refined estimation of the borders of the road	58
3.18	Aerial image processing : (a) Initial image; (b) Result of the edge detector. Proposed method : (c) Superposition of the initial image and the initialization straight line; (d) Superposition of the initial image and the estimation. GVF method : (e) Initialization ; (f) Superposition of the initial image and the estimation.	61
3.19	Aerial image processing : (a) Initial image; (b) Result of the edge detector. Proposed method : (c) Initialization ; (d) Superposition of the initial image and the estimation.	61
3.20	Hand-made image, common initialization : (a) Example of hand-made image ; (b) Result of the initialization ; (c) Superposition initial image and result obtained by the proposed method ; (d) Superposition initial image and result obtained by GVF.	63
3.21	Combination of gradient algorithm and spline interpolation : increasing number of nodes permits an increased resolution.	65

3.22	(a) Two nearly parallel curves. (b) Orientation estimation by Propagator, offset estimation by MFBLP. Distortion retrieval by the proposed optimization method : (c) estimation, (d) superposition estimation and processed.	67
3.23	(a) Least distorted contour : initialization, results obtained (b) Most distorted contour : initialization, results obtained by the proposed method and GVF respectively.	68
3.24	Aerial image analysis : coast detection. (a) processed image, (b) initialization, (c) result with the proposed method, (d) initialization for GVF, (e) result with GVF.	70
3.25	Aerial image analysis : road detection. (a) processed image, (b) initialization, (c) result with the proposed method, (d) initialization for GVF, (e) result with GVF.	71
3.26	Industrial application : default detection in tissue. (a) processed image, (b) initialization, (c) result with the proposed method, (d) initialization for GVF, (e) result with GVF.	71
4.1	(a) Circular-like contour ; (b) Bottom right quarter of the contour and pixel coordinates in the polar system (ρ, θ) having its origin on the center of the circle. r is the radius of the circle. $\Delta\rho$ is the value of the shift between a pixel of the contour and the pixel of the circle having same coordinate θ	76
4.2	Sub-image, associated with a circular array composed of S sensors.	77
4.3	Model for an image containing several nearly circular or elliptic features. r is the circle radius, a and b are the axial parameters of the ellipse.	79
4.4	(a) Processed (b) Superposition processed and result obtained after applying a least-squares method for circle fitting : $ME_\rho = 4.5$ pixels, (c) Superposition processed and result obtained after applying the Hough transform : $ME_\rho = 0.7$ pixel, (d) Superposition processed and result obtained after applying the proposed method : $ME_\rho = 0.3$ pixel.	82
4.5	(a) Processed, (b) Result (superimposed), with the proposed method for radius estimation or equivalently with Hough transform or a least-squares fitting method : $ME_\rho = 0.1$ (resp. 0.3 for Hough transform and 0.4 for the least-squares fitting method).	83
4.6	Biased center and radius estimation, proposed optimization algorithm using DIRECT and spline interpolation. (a) Processed (b) Initialisation by the least-squares method, (c) Initialisation by the Hough transform, (d) Initialisation by the proposed method, (e) Superposition processed and final result, $ME_\rho = 0.9$ pixel.	84
4.7	Examples of processed images containing the less (a) and the most (d) distorted circles, initialization (b,e) and estimation using the proposed method (c,f). $ME_\rho = 1.4$ pixel and 2.7 pixels.	85
4.8	Examples of processed images containing the less (a) and the most (d) distorted circles, initialization (b,e) and estimation using GVF algorithm (c,f). $ME_\rho = 1.4$ pixel and 4.1 pixels.	86

4.9	Multiple circles : (a) Processed (b) Signal generated upon the left side of the image (c) Initialization (d) Result with gradient method.	87
4.10	Two circles with different centers and same radius. (a) and (b) Processed images, modulus of the generated signals; (c) and (d) Superposition processed and result with gradient method.	88
4.11	Ellipse (and respectively rotated ellipse) fitting : (a) and (c) Processed ; (b) and (d) Superposition processed and result obtained after applying the proposed method for ellipse fitting : $ME_\rho = 0.7$ (respectively $ME_\rho = 0.6$) pixel.	89
4.12	(a) Processed image, (b) result of Canny operator (c) Initialisation, (d) Result obtained with gradient method, (e) Result obtained with GVF, (f) Result obtained with DIRECT combined with spline interpolation.	90
4.13	(a) Processed image, (b) Initialisation, (c) Result obtained with the proposed optimization method.	91
4.14	(a) Processed image, (b) Initialisation, (c) Result obtained with the proposed optimization method.	91
5.1	'Lena' standard color image and its tensor representation.	98
5.2	2nd-mode flattening of tensor $\mathcal{A} : \mathbf{A}_2$	99
5.3	(a) non-noisy image. (b) image to be processed, impaired by an additive white Gaussian noise, with SNR=8.1dB. (c) channel-by-channel SVD-based filtering of parameter K=30. (d) rank-(30,30,2) approximation. (e) Wmm-(30,30,2) filtering.	108
5.4	NQE evolution with respect to SNR (dB) : channel-by-channel SVD-based filtering of parameter 30 (-o-), rank-(30,30,2) approximation (-□-), Wmm-(30,30,2) filtering (-▽-).	109
5.5	(a) non-noisy image. (b) image to be processed, impaired by an additive white Gaussian noise, with SNR=8.1dB. (c) channel-by-channel Wiener-based filtering of parameter K=30. (d) rank-(30,30,3) approximation. (e) Wmm-(30,30,3) filtering.	109
5.6	(a) non-noisy image. (b) image to be processed, impaired by an additive white Gaussian noise, with SNR=8.0dB. (c) channel-by-channel SVD-based filtering of parameter K=19. (d) rank-(19,19,3) approximation. (e) Wmm-(19,19,3) filtering.	110
5.7	(a) : initial non-noisy image. (b) : initial image with an additive correlated Gaussian noise, $SNR = 2.48$ dB. (c) : HOSVD-(30,30,2). (d) : rank- \mathcal{C} (30,30,2) approximation. (e) : rank- \mathbf{C}_1 (30,30,2) approximation.	112
5.8	Evolution of the NQE with respect to the SNR(dB) for each tensor filtering method : o : HOSVD-(30,30,2) ; ▽ : rank- \mathcal{C} (30,30,2) ; □ : rank- \mathbf{C}_1 (30,30,2).	112
5.9	Channels 30 to 34 of the processed multispectral images are presented : (a)-(e) : Non-noisy multispectral image. (f)-(j) : Impaired multispectral image. (k)-(o) : Results obtained with channel-by-channel SVD filtering. (p)-(t) : Results obtained with rank-(30,30,30) approximation.	113

5.10 NQE evolution with respect to SNR (from -1 to 15dB) : channel-by-channel SVD-based filtering of parameter 30 (-o-), and rank-(30,30,30) approximation (-□-).	114
5.11 (o) : results obtained with channel-by-channel SVD-based filtering of parameter 30 ; (□) : results obtained with rank-(30,30,2) approximation ; (a) : evolution of the mean NQE with respect to SNR (dB) ; (b) evolution of the standard deviation of NQE with respect to SNR (dB).	115
5.12 non-noisy, impaired, and processed seismic wave : the three polarization components ; (a)-(c) : components 1, 2 and 3 of the non-noisy seismic wave ; (d)-(f) : components 1, 2 and 3 of the seismic wave, impaired by an additive white Gaussian noise (SNR= -10 dB) ; (g)-(i) : Wiener filtering applied component by component (Wcc- K), with rank $K = 8$; (j)-(l) : HOSVD-(K_1, K_2, K_3), with $(K_1, K_2, K_3)=(8, 8, 3)$; (m)-(o) : multiway Wiener filtering (Wmm-(K_1, K_2, K_3)), with $(K_1, K_2, K_3)=(8, 8, 3)$	116
5.13 Evolution of the NQE with respect to the SNR (dB) for each tensor filtering method. (▽) : Wiener filtering applied component by component (Wcc- K), with rank $K = 8$; (o) : HOSVD-(K_1, K_2, K_3), with $(K_1, K_2, K_3)=(8, 8, 3)$; (●) : Multiway Wiener filtering (Wmm-(K_1, K_2, K_3)) with $(K_1, K_2, K_3)=(8, 8, 3)$	117
5.14 Denoising of a multicomponent seismic wave impaired by an additive correlated Gaussian noise (SNR= -2 dB), using multiway filtering based on fourth order cumulants : comparison of rank- $\mathcal{C}(8, 8, 3)$, and rank- $\mathbf{C}_1(8, 8, 3)$: (a)-(c) : components 1 to 3 of the non-noisy seismic wave ; (d)-(f) : noised signal ; components 1 to 3 impaired by a correlated Gaussian noise (SNR= -2 dB) ; (g)-(i) : rank- $\mathcal{C}(8, 8, 3)$ based filtering ; (j)-(l) : rank- $\mathbf{C}_1(8, 8, 3)$ based filtering.	118
5.15 Denoising of a multicomponent seismic wave impaired by an additive correlated Gaussian noise, by multiway filtering : Evolution of NQE with respect to SNR (dB) for rank-(8, 8, 3) approximation (□), rank- $\mathbf{C}_1(8, 8, 3)$ using fourth order cumulant slice matrix (○) and rank- $\mathcal{C}(8, 8, 3)$ using fourth order cumulants (+).	119
5.16 $128 \times 128 \times 1$ gray image containing : (a) Oblique line, (b) SVD truncation to the 80 th eigenvalue, (c) SVD truncation to the 110 th eigenvalue ; (d) Vertical line, (e) SVD truncation to the first eigenvalue.	121
5.17 30 th spectral band : (a) acquired data, (b) derivative of acquired data, (c) MWFR processed data.	122
5.18 Image HSI02 : (a) noisy tensor, (b) PCA-SWT , (c) MWF and (d) MWFR with SLIDE estimated angles $\theta_k \in \{0^\circ, 34^\circ, 90^\circ\}$. ($K_1 = K_2 = 31, K_3 = 97$).	123
B.1 Pseudospectres : a)Formation de voies ; b)MUSIC	145
B.2 Fronts d'onde distordus reçus sur une antenne composée d'un grand nombre de capteurs.	147
B.3 Evolution de la phase de trois fronts d'onde en fonction de l'indice des capteurs.	147

B.4 Pseudospectres de MUSIC, représentés dans l'intervalle $[-50^\circ, 50^\circ]$, superposés aux valeurs attendues (lignes verticales) : (a) sans annulation des distorsions de phase ; (b) et (c) avec l'estimation -10 (resp. 25) itérations de DIRECT- et annulation des erreurs de phase.	148
B.5 Modèle d'image : (a) Image-matrice muni d'un système de coordonnées et d'une antenne rectiligne de N capteurs équidistants. (b) Un droite caractérisée par son angle θ et son offset x_0	149
B.6 Un modèle d'image contenant un contour distordu.	150
B.7 (a) Image transmise à la commande automatique d'un véhicule se déplaçant sur une voie ferrée. (b) Détection des rails. (c) Localisation de la première traverse.	150
B.8 Analyse d'images aériennes : détection de côte. (a) image à traiter, (b) initialisation, (c) résultat obtenu avec la méthode proposée, (d) initialisation pour GVF, (e) résultat obtenu avec GVF.	151
B.9 (a) Contour approximativement circulaire ; (b) Quart inférieur droit du contour, et coordonnées de pixels dans un repère de coordonnées polaires (ρ, θ) centré sur le centre du cercle. r est le rayon du cercle. $\Delta\rho$ est la valeur de décalage entre un pixel du contour et le pixel du cercle qui a la même valeur de coodonnée θ	154
B.10 Sous-image, associée à une antenne circulaire comportant S capteurs.	155
B.11 Modèle pour une image contenant plusieurs contours approximativement circulaires ou elliptiques. r est le rayon du cecle, a et b sont les paramètres axiaux de l'ellipse.	156
B.12 Exemples d'images à traiter contenant le contour le moins (a) et le plus distordu (d), l'initialisation (b,e) et l'estimation finale utilisant la méthode proposée (c,f). Valeurs moyennes d'erreur : $ME_\rho = 1.4$ pixel et 2.7 pixels.	157
B.13 Exemples d'images à traiter contenant le contour le moins (a) et le plus distordu (d), l'initialisation (b,e) et l'estimation finale utilisant la méthode proposée (c,f). Valeurs moyennes d'erreur : $ME_\rho = 1.4$ pixel et 4.1 pixels.	157
B.14 (a) Image non bruitée, (b) Image à traiter, dégradée par un bruit additif, Gaus-sien et blanc, avec un SNR=8.1dB. (c) Filtrage par SVD tranche par tranche de paramètre K=30. (d) Approximation de rang-(30, 30, 2) (e) Filtrage de Wiener multimodal de rang-(30, 30, 2).	160

Liste des tableaux

2.1	<i>ME</i> and <i>RMSE</i> values over angle estimation for several <i>SNR</i> values.	28
3.1	Experimental and theoretical data concerning the complexity of the high-resolution based method and the extension of the Hough transform method. .	44
3.2	<i>ME</i> and <i>Std</i> values (in pixel) obtained with the proposed method and GVF versus amplitude and period values.	63
3.3	<i>ME</i> values (in pixel) obtained with the proposed method (A) and with GVF (B), versus distortion (maximum amplitude ; standard deviation) -dist.- and noise percentage values -n..	68
3.4	<i>Std</i> values (in pixel) obtained with the proposed method (A) and with GVF (B), versus distortion (maximum amplitude ; standard deviation) -dist.- and noise percentage values -n..	69
4.1	<i>ME</i> values (in pixel) obtained with the proposed method (a) and with GVF (b), versus distortion (maximum amplitude ; standard deviation) -dist.-, and noise percentage -n--.	86
4.2	Real-world images : ME values (in pixel).	91
B.1	Valeurs d'erreur <i>ME</i> (en pixel) obtenues avec la méthode proposée (a) et avec GVF (b), en fonction des paramètres de distortion (amplitude maximum ; écart type) -dist.-, et du pourcentage de bruit -n--.	158

Bibliographie

- [1] K. ABED-MERAIM, H. MAÎTRE et P. DUHAMEL, *Blind multichannel image restoration using subspace-based method*, dans *IEEE International Conf. on Acoustics Systems and Signal Processing*, Hong Kong, China, April 6-10, 2003.
- [2] K. ABED-MERAIM et Y. HUA, *Multi-line fitting and straight edge detection using polynomial phase signals*, Conference Records of the 31st Asilomar Conference on Signal, Systems and Computers, vol. 2, 1997.
- [3] H. K. AGHAJAN, *Subspace Techniques for Image Understanding and Computer Vision.*, Phd thesis, Stanford University, Stanford, California, USA, Nov. 1995.
- [4] H. K. AGHAJAN et T. KAILATH, *A subspace fitting approach to super resolution multi-line fitting and straight edge detection*, Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. 3, 1992.
- [5] H. K. AGHAJAN et T. KAILATH, *Sensor array processing techniques for super resolution multi-line-fitting and straight edge detection*, IEEE Transactions on Image Processing, vol. 2, no. 4, pp. 454–465, Oct. 1993.
- [6] H. K. AGHAJAN et T. KAILATH, *Slide : subspace-based line detection*, Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. 5, 1993.
- [7] H. K. AGHAJAN et T. KAILATH, *Slide : Subspace-based line detection*, IEEE trans. on Pattern Analysis and Machine Intelligence, vol. 16, no. 11, pp. 1057–1073, Nov. 1994.
- [8] H. K. AGHAJAN, B. KHALAJ et T. KAILATH, *Estimation of multiple 2-d uniform motions by slide : subspace-based line detection*, IEEE Transactions on Image Processing, vol. 8, no. 4, pp. 517–526, Apr. 1999.
- [9] Y. AHN et H. KIM, *Approximation of circular arcs by b  zier curves*, Journal of Computational and Applied Mathematics, 1997.
- [10] Y. AHN, Y. KIM et Y. SHIN, *Approximation of circular arcs and of a set curves by bezier curves of high degree*, Journal of Computational and Applied Mathematics, vol. 167, no. 2, pp. 405–416, June 2004.
- [11] O. ALTER et G. GOLUB, *Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations*, Proc. Nat. Acad. Sci. USA, vol. 102, no. 49, pp. 17559–17564, October 2005.
- [12] H. ANDREWS et C. PATTERSON, *Singular value decomposition and digital image processing*, IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 24, no. 1, pp. 26–53, Feb. 1976.

- [13] H. ANDREWS et C. PATTERSON, *Singular Value Decomposition (SVD) image coding*, IEEE Trans. on Communications, vol. 24, no. 4, pp. 425–432, April 1976.
- [14] I. ATKINSON, F. KAMALABADI et D. JONES, *Wavelet-based hyperspectral image estimation*, Proc-IGARS, vol. 2, pp. 743–745, 2003.
- [15] J. AUJOL, G. AUBERT et L. BLANC-FÉRAUD, *Wavelet-based level set evolution for classification of textured images*, IEEE Transactions on Image Processing, vol. 12, no. 12, pp. 1634–1641, dec. 2003.
- [16] N. AYACHE et O. FAUGERAS, *Hyper : a new approach for the recognition and positionning of two-dimensionnal objects*, IEEE trans. on Pattern Analysis and Machine Intelligence, vol. 8, no. 1, pp. 44–54, Jan. 1986.
- [17] B. BADER et T. KOLDA, *Algorithm 862 : Matlab tensor classes for fast algorithm prototyping*, ACM Transactions on Mathematical Software,, vol. 32, no. 4, December 2006.
- [18] E. BALA et A. ERTÜZÜN, *A multivariate thresholding technique for image denoising using multiwavelets*, ASP, (pp. 1205–1211), 08 2005.
- [19] D. BALLARD, *Generalizing the hough transform to detect arbitrary shapes*, Pattern Recognition, vol. 13, no. 2, pp. 111–122, Apr. 1981.
- [20] J. BORKOWSKI, B. MATUSZEWSKI, J. MROCZKA et L.-K. SHARK, *Geometric matching of circular features by least-squares fitting*, Pattern Recognition Letters, vol. 23, no. 7, pp. 885–894, May 2002.
- [21] S. BOURENNANE et A. BENDJAMA, *Locating wide band acoustic sources using higher-order statistics*, Applied Acoustics, vol. 63, no. 3, pp. 235–251, May 2001.
- [22] S. BOURENNANE, A. BENJAMA et M. FRIKEL, *Seismic wave separation based on higher-order statistics*, dans *IEEE International Conf. on Digital Signal Processing and its Applications*, Moscow, Russia, 1998.
- [23] S. BOURENNANE et M. FRIKEL, *Localization of the wideband sources with estimation of the antenna shape*, IEEE-Workshop on Statistical & Array Processing, 1996.
- [24] S. BOURENNANE et J. MAROT, *Line parameters estimation by array processing methods*, Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. 4, 2005.
- [25] S. BOURENNANE et J. MAROT, *Contour estimation by array processing methods*, EUR-ASIP Journal on Advances on Signal Processing, vol. article ID 95634 15 pages, 2006.
- [26] S. BOURENNANE et J. MAROT, *Estimation of straight line offsets by a high-resolution method*, IEE proceedings Journal on Vision, Image and Signal processing, vol. 153, no. 2, pp. 224–229, Apr. 2006.
- [27] S. BOURENNANE et J. MAROT, *Optimization and interpolation for distorted contour estimation*, Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. 2, 2006.
- [28] S. BOURENNANE et J. MAROT, *Propagator method for an application to contour estimation*, Pattern Recognition Letters, vol. 28, no. 12, pp. 1556–1562, Sept. 2007.
- [29] P. BRIGGER, J. HOEG et M. UNSER, *B-spline snakes : A flexible tool for parametric contour detection*, IEEE Transactions on Image Processing, vol. 9, no. 9, pp. 1484–1496, 2000.

- [30] R. BRO, *Multi-way analysis in the food industry*, PhD thesis, Royal Veterinary and Agricultural University, Denmark, 1998.
- [31] J. CAROLL et J. CHANG, *Analysis of individual differences in multidimensional scaling via n-way generalization of eckart-young decomposition*, Psychometrika, vol. 35, no. 3, pp. 283–319, 09 1970.
- [32] C. CHANG et Q. DU, *Estimation of number of spectrally distinct signal sources in hyperspectral imagery*, IEEE-TGARS, vol. 42, no. 3, pp. 608–619, 2004.
- [33] Y.-C. CHANG, B.-K. SHYU, C.-Y. CHO et J.-S. WANG, *Adaptive post-processing for region-based fractal image compression*, dans *Data Compression Conference*, (p. 549), 2000.
- [34] J. CHENG et S. W. FOO, *Dynamic directional gradient vector flow for snakes*, IEEE Transactions on Image Processing, vol. 15, no. 6, pp. 1563–1571, Jun. 2006.
- [35] G. COATH et P. MUSUMECI, *Adaptive arc fitting for ball detection in robocup*, Proc. APRS Workshop on Digital Image Computing, 2003.
- [36] S. CONNELL et A. JAIN, *Template-based online character recognition*, Pattern Recognition, vol. 34, no. 1, pp. 1–14, Jan. 2001.
- [37] J. CRAWFORD, *A noniterative method for fitting circular arcs to measured points*, Nuclear Instruments and Methods in Physics Research, 1983.
- [38] L. DE LATHAUWER, *Signal Processing Based on Multilinear Algebra*, Phd thesis, K.U. Leuven, E.E. Dept. (ESAT), Belgium, Sept. 1997.
- [39] L. DE LATHAUWER, B. DE MOOR et J. VANDEWALLE, *Dimensionality reduction in higher-order-only ica*, Proc. IEEE Signal Processing Workshop on higher-order Statistics, (pp. 316–320), 07 1997.
- [40] L. DE LATHAUWER, B. DE MOOR et J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM Journal on Matrix Analysis and Applications, vol. 21, no. 4, pp. 1253–1278, April 2000.
- [41] L. DE LATHAUWER, B. DE MOOR et J. VANDEWALLE, *On the best rank-(1) and rank- (r_1, \dots, r_N) approximation of higher-order tensors*, SIAM Journal on Matrix Analysis and Applications, vol. 21, no. 4, pp. 1324–1342, April 2000.
- [42] D. DONOHO, *De-noising by soft-thresholding*, IEEE-IT, vol. 41, pp. 613–627, 1995.
- [43] Y. ECKART et G. YOUNG, *The approximation of a matrix by another of lower rank*, Psychometrika, vol. 1, pp. 211–218, 1936.
- [44] R. J. FERRARI, R. M. RANGAYYAN, J. E. L. DESAUTELS, R. A. BORGES et A. F. FRÈRE, *Automatic identification of the pectoral muscle in mammograms*, IEEE trans. on Medical Imaging, vol. 23, no. 2, pp. 232–245, Feb. 2004.
- [45] M. A. T. FIGUEIREDO et A. JAIN, *Unsupervised learning of finite mixture models*, IEEE trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 3, pp. 381–396, Mar. 2002.
- [46] F. FREIRE et T. ULRYCH, *Application of SVD to vertical seismic profiling*, Geophysics, vol. 53, pp. 778–785, 1988.
- [47] W. GANDER, G. GOLUB et R. STREBEL, *Least-squares fitting of circles and ellipses*, BIT, 1994.

- [48] P. GILL et W. MURRAY, *Quasi-newton methods for unconstrained optimization*, Journal of Instruments and Mathematical Applications, vol. 9, 1972.
- [49] F. GLANGEAUD et J. MARI, *Wave separation*, Technip IFP édition, 1994.
- [50] D. GOLDBERG, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wisley, New-York (USA), 1989.
- [51] G. GOLUB et C. VAN LOAN, *Matrix computations*, 3rd ed., Baltimore, The John Hopkins University Press édition, 1996.
- [52] E. GONEN et J. MENDEL, *Application of cumulants to array processing-part III : blind beamforming for coherent signal*, IEEE Trans. on Signal Processing, vol. 45, no. 9, pp. 2252–64, Sept. 1997.
- [53] A. GREEN, M. BERMAN et M. CRAIG, *A transformation for ordering multispectral data in terms of image quality with implications for noise removal*, IEEE-TGARS, vol. 26, no. 1, pp. 65–74, 1988.
- [54] N. GUIL, J. VILLABA et E. L. ZAPATA, *A fast hough transform for segment detection*, IEEE Transactions on Image Processing, vol. 5, no. 5, pp. 787–792, May 1996.
- [55] B. HALDER, H. AGHAJAN et T. KAILATH, *Propagation diversity enhancement to the subspace-based line detection algorithm*, Proceedings of SPIE, vol. 2424, 1995.
- [56] M. HANKE et P. HANSEN, *Regularization methods for large-scale problems*, Survey of mathematics for Industry, vol. 3, 1993.
- [57] R. HARSHMAN et M. LUNDY, *Research methods for multimode data analysis*, (pp. 122–215), Praeger, New York, 1984.
- [58] M. HEMON et D. MACE, *The use of Karhunen-Loeve transform in seismic data prospecting*, Geophysical Prospecting, vol. 26, pp. 600–626, 1978.
- [59] J. HÄSTAD, *Tensor rank is np-complete*, Journal of Algorithms, vol. 11, pp. 644–654, september 1990.
- [60] K. HSU, *Wave separation and feature extraction of acoustic well-logging waveforms of triaxial recordings by singular value decomposition*, Geophysics, vol. 55, pp. 176–184, 1990.
- [61] J. ILLINGWORTH et J. KITTNER, *A survey of the hough transform*, Proc. of the IEEE Int. Conf. on Computer Vision Graphics and Image Processing, vol. 44, no. 1, pp. 87–116, Oct. 1988.
- [62] G. JACKSON, I. MASON et S. GREENHALGH, *Principal component transforms of triaxial recordings by singular value decomposition*, Geophysics, vol. 56, no. 4, pp. 176–184, 1991.
- [63] M. JANSSON, A. SWINDEHURST et B. OTTERSTEN, *Weighted subspace fitting for general array error models*, IEEE trans. on Signal Processing, vol. 46, no. 9, pp. 2484–2498, sep. 1998.
- [64] D. JONES, C. PERTUNEN et B. STUCKMAN, *Lipschitzian optimization without the lipschitz constant*, Journal of Optimization Theory and Applications, vol. 79, no. 1, pp. 157–181, Oct. 1993.
- [65] V. KARIMAKI, *Effective circle fitting for particle trajectories*, Nuclear Instruments and Methods in Physics Research, 1991.

- [66] I. KAROUI, R. FABLET, J.-M. BOUCHER et J.-M. AUGUSTIN, *Region-based segmentation using texture statistics and level-set methods*, Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. 2, 2006.
- [67] N. KARSSEMEIJER, *Automated classification of parenchymal patterns in mammograms*, Physics in Medicine and Biology, vol. 43, no. 2, pp. 365–378, Feb. 1998.
- [68] I. KASA, *A circle fitting procedure and its error analysis*, IEEE trans. on Instruments and Measures, vol. 25, no. 1, pp. 8–14, Mar. 1976.
- [69] M. KASS, A. WITKIN et D. TERZOPoulos, *Snakes : Active contour model*, Int. J. of Computer Vision, vol. 1, no. 4, pp. 321–331, May 1988.
- [70] H. KIERS, *Towards a standardized notation and terminology in multiway analysis*, Journal of Chemometrics, vol. 14, pp. 105–122, 2000.
- [71] N. KIRIATY et A. BRUCKSTEIN, *On navigating between friends and foes*, IEEE trans. on Pattern Analysis and Machine Intelligence, vol. 13, no. 6, pp. 602–606, June 1991.
- [72] N. KIRYATI et A. BRUCKSTEIN, *What's in a set of points ? [straight line fitting]*, IEEE trans. on Pattern Analysis and Machine Intelligence, vol. 14, no. 4, pp. 496–500, Apr. 1992.
- [73] E. KOFIDIS et P. REGALIA, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM Journal on Matrix Analysis and Applications, vol. 23, no. 3, pp. 863–884, March 2002.
- [74] T. KOLDA, *Orthogonal tensor decomposition*, SIAM Journal on Matrix Analysis and Applications, vol. 23, no. 1, pp. 243–255, March 2001.
- [75] H. KRIM et M. VIBERG, *Two decades of array signal processing research*, 1996.
- [76] P. KROONENBERG, *Three-mode principal component analysis*, DSWO press, Leiden, 1983.
- [77] P. KROONENBERG et J. DE LEEUW, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, vol. 45, no. 1, pp. 69–97, Mar. 1980.
- [78] J. KRUSKAL, *Rank, decomposition, and uniqueness for 3-way and N-way arrays*, Multiway Data Analysis, elsevier, Amsterdam, The Netherlands : North Holland édition, 1988.
- [79] S. M. KWOK, R. CHANDRASEKHAR, Y. ATTICKIOUZEL, M. T. RICKARD et A. F. FRÈRE, *Automatic pectoral muscle segmentation on mediolateral oblique view mammograms*, IEEE trans. on Medical Imaging, vol. 23, no. 9, pp. 1129–1140, Feb. 2004.
- [80] J. LACOUME, P. AMBLARD et P. COMON, *Statistiques d'ordre supérieur pour le traitement du signal*, Paris, Masson édition, 1997.
- [81] U. LANDAU, *Estimation of a circular arc center and its radius*, Proc. of the IEEE Int. Conf. on Computer Vision Graphics and Image Processing, 1986.
- [82] X. LIU, *Ground roll suppression using the Karhunen-Loeve transform*, Geophysics, vol. 64, no. 2, pp. 564–566, 1991.
- [83] D. MANOLAKIS, C. SIRACUSA et G. SHAW, *Hyperspectral subpixel target detection using the linear mixing model : Analysis of hyperspectral image data*, IEEE-TGARS, vol. 39, no. 7, pp. 1392–1409, 2001.

- [84] J. MAROT et S. BOURENNANE, *Array processing and fast optimization algorithms for distorted circular contour retrieval*, EURASIP Journal on Advances on Signal Processing, vol. article ID 57354, p. 13 pages, 2007.
- [85] J. MAROT et S. BOURENNANE, *Phase distortion estimation by direct and spline interpolation algorithms*, IEEE Signal Processing Letters, vol. 14, no. 7, pp. 461–464, 2007.
- [86] J. MAROT et S. BOURENNANE, *Subspace-based and direct algorithms for distorted circular contour estimation*, IEEE Transactions on Image Processing, vol. 16, no. 9, pp. 2369–2378, Sept. 2007.
- [87] J. MAROT, S. BOURENNANE et M. ADEL, *Array processing approach for object segmentation in images*, Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. 1, 2007.
- [88] P. MARTIN, P. REFREGIER, F. GALLAND et F. GUERAULT, *Nonparametric statistical snake based on the minimum stochastic complexity*, IEEE Transactions on Image Processing, vol. 15, no. 9, pp. 2762–2770, Sept. 2006.
- [89] J. MENDEL, *Tutorial on higher-order statistics (spectra) in signal processing and system theory : theoretical results and some applications*, dans *Proc. of the IEEE*, vol. 79, (pp. 278–305), March 1991.
- [90] J. MUNIER et G. Y. DELISLE, *Spatial analysis using new properties of the cross-spectral matrix*, IEEE trans. on Signal Processing, vol. 39, no. 3, pp. 746–749, Mar. 1991.
- [91] J. MUNIER et S. MARCOS, *Source localization using a distorted antenna*, Proc. of the IEEE Int. Conf. on Acoustic, Speech and Signal Processing, vol. 4, 1989.
- [92] D. MUTI, *Traitement du signal tensoriel. Application aux images en couleurs et aux signaux sismiques.*, Phd thesis, Institut Fresnel, Université Paul Cézanne, Aix-Marseille III, Marseille, France, december 2 2004.
- [93] D. MUTI et S. BOURENNANE, *Multidimensional estimation based on a tensor decomposition*, dans *IEEE Workshop on Statistical Signal Processing*, St Louis, USA, Sept. 28 - Oct. 1, 03.
- [94] D. MUTI et S. BOURENNANE, *Multidimensional signal processing using lower rank tensor approximation*, dans *IEEE Int. Conf. on Acoustics, Systems and Signal Processing*, Hong Kong, China, April 6-10, 2003.
- [95] D. MUTI et S. BOURENNANE, *Traitement du signal par décomposition tensorielle*, dans *GRETsi Symp.*, Paris, France, September 8-11 2003.
- [96] D. MUTI et S. BOURENNANE, *Multidimensional filtering based on a tensor approach*, Signal Proceesing Journal, Elsevier, vol. 85, no. 12, pp. 2338–2353, May 2005.
- [97] D. MUTI et S. BOURENNANE, *Multiway filtering based on fourth order cumulants*, Applied Signal Processing, EURASIP, vol. 7, pp. 1147–1159, May 2005.
- [98] D. MUTI, S. BOURENNANE et J. MAROT, *Lower-rank tensor approximation and multiway filtering*, SIAM Journal on Matrix Analysis and Applications (SIMAX), 2007.
- [99] R. NEELAMANI, H. CHOI et R. BARANIUK, *Forward : Fourier-wavelet regularized deconvolution for ill-conditioned systems*, IEEE Transactions on SP, vol. 52, no. 2, pp. 418–433, Feb. 2004.
- [100] A. P.-C. NG, *Direction-of-arrival estimates in the presence of wavelength, gain and phase errors*, IEEE trans. on Signal Processing, vol. 4, no. 1, pp. 225–232, jan. 1995.

- [101] P. O'LEARY, M. HARKER et P. ZSOMBOR-MURRAY, *Direct and least-squares fitting of coupled geometric objects for metric vision*, IEE proceedings Journal on Vision, Image and Signal processing, vol. 152, no. 6, pp. 687–694, dec. 2005.
- [102] S. OSHER et R. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New-York (USA), 2002.
- [103] S. OSHER et J. SETHIAN, *Fronts propagating with curvature-dependent speed : algorithms based on hamilton-jacobi formulations*, Journal of Computational Physics, vol. 79, 1988.
- [104] H. OTHMAN et S.-E. QIAN, *Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage*, IEEE-TGARS, vol. 44, no. 2, pp. 397–408, 2006.
- [105] N. PARAGIOS et R. DERICHE, *Geodesic active regions and level set methods for supervised texture segmentation*, Int. J. of Computer Vision, vol. 46, no. 3, pp. 223–247, Feb. 2002.
- [106] L. PARRA, C. SPENCE, P. SAJDA, A. ZIEHE et K. ULLER, *Unmixing hyperspectral data*, ANIPS, vol. 12, pp. 942–948, 2000.
- [107] A. PAULRAJ et T. KAILATH, *Eigenstructure methods for direction of arrival estimation in the presence of unknown noise fields*, IEEE trans. on Acoustics, Speech and Signal Processing, vol. 34, no. 1, pp. 13–20, Febr. 1986.
- [108] J. PESQUET et E. MOREAU, *Cumulant-based independence measures for linear mixture*, IEEE Transactions on Information Theory, vol. 47, no. 5, pp. 1947–1956, July 2001.
- [109] S. PILLAI et B. KWON, *Forward/backward spatial smoothing techniques for coherent signal identification*, IEEE trans. on Acoustics, Speech and Signal Processing, vol. 37, no. 1, pp. 8–15, Jan. 1989.
- [110] B. PORAT et B. FRIEDLANDER, *Direction finding algorithms based on higher-order statistics*, IEEE Trans. on Signal Processing, vol. 39, no. 9, pp. 2016–2024, Sept. 1991.
- [111] F. PRECIOSO, M. BARLAUD, T. BLU et M. UNSER, *Robust real-time segmentation of images and videos using a smooth-spline snake-based algorithm*, IEEE Transactions on Image Processing, vol. 14, no. 7, pp. 910–924, July 2005.
- [112] C. PREZA, M. MILLER, L. THOMAS JR. et J. McNALLY, *Regularized method for reconstruction of three-dimensional microscopic objects from optical sections*, J. Opt. Soc. Am. A, vol. 9, pp. 219–228, 1992.
- [113] L. J. RICKARD, R. W. BASEDOW, E. F. ZALEWSKI, P. R. SILVERGLATE et M. LANDERS, *HYDICE : an airborne system for hyperspectral imaging*, dans *Proc. SPIE, Imaging Spect. of the Terrestrial Environment, Gregg Vane ; Ed.* (édité par G. VANE), vol. 1937, (pp. 173–179), 1993.
- [114] R. ROY et T. KAILATH, *Esprit : Estimation of signal parameters via rotational invariance techniques*, IEEE trans. on Acoustics, Speech and Signal Processing, vol. 37, no. 7, pp. 984–995, July 1989.
- [115] R. SCHMIDT, *Multiple emitter location and signal parameters estimation*, IEEE trans. on Acoustics, Speech and Signal Processing, vol. 4, no. 3, pp. 276–280, Mar. 1983.
- [116] L. SHE et B. ZHENG, *Multiwavelets based denoising of sar images*, dans *Proceedings 5th international Conference on Signal Processing*, (pp. 321–324), Beijing, China, 08 2000.

- [117] J. SHEINVALD et N. KIRYATI, *On the magic of slide*, Machine Vision and Applications, vol. 9, no. 5, 1997.
- [118] N. SIDIROPOULOS et R. BRO, *On the uniqueness of multilinear decomposition of N-way arrays*, Journal of Chemometrics, vol. 14, pp. 229–239, 2000.
- [119] N. SIDIROPOULOS, R. BRO et G. GIANNAKIS, *Parallel factor analysis in sensor array processing*, IEEE Trans. on Signal Processing, vol. 48, no. 8, pp. 2377–2388, August 2000.
- [120] N. SIDIROPOULOS, G. GIANNAKIS et R. BRO, *Blind parafac receivers for DS-CDMA systems*, IEEE Trans. on Signal Processing, vol. 48, no. 3, pp. 810–823, 2000.
- [121] A. SMILDE, R. BRO et P. GELADI, *Multi-way analysis : Applications in the Chemical Sciences*, Wiley, 2004.
- [122] C. L. TISSE, L. MARTIN, L. TORRES et M. ROBERT, *Person identification technique using human iris recognition*, International Conference on Vision Interface, 2002.
- [123] L. TUCKER, *Some mathematical notes on three-mode factor analysis*, Psychometrika, vol. 31, no. 3, pp. 279–311, Sept. 1966.
- [124] D. TUFTS et R. KUMARESAN, *Estimation of frequencies of multiple sinusoids : making linear prediction perform like maximum likelihood*, Proceedings of IEEE, vol. 70, no. 9, pp. 975–989, Sept. 1982.
- [125] M. UNSER, A. ALDROUBI et M. EDEN, *B-spline signal processing : Part i-theory ; part ii-efficient design and applications*, IEEE Transactions on Image Processing, vol. 41, no. 2, pp. 821–848, Feb. 1993.
- [126] M. VASILESCU et D. TERZOPoulos, *Multilinear analysis of image ensembles : Tensorfaces*, ECCV, (pp. 447–460), 2002.
- [127] M. VASILESCU et D. TERZOPoulos, *Multilinear image analysis for facial recognition*, dans *IEEE Int. Conf. on Pattern Recognition (ICPR2002)*, vol. 2, Quebec city, Canada, 11-15, August 2002.
- [128] M. VASILESCU et D. TERZOPoulos, *Multilinear independent components analysis*, Learning 2004, Snowbird, UT, 04 2004.
- [129] M. VASILESCU et D. TERZOPoulos, *Multilinear independent components analysis*, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 547–553, 06 2005.
- [130] M. P. WACHOWIAK et T. M. PETERS, *Parallel Optimization Approaches for Medical Image Registration*, pp. 781–788, Springer Berlin, Heidelberg, Berlin, 2004.
- [131] H. WANG et N. AHUJA, *Facial expression decomposition*, dans *9th IEEE International Conf. on Computer Vision (ICCV2003)*, vol. 2, Nice, France, 13-16, October 2003.
- [132] M. WAX et T. KAILATH, *Detection of signals by information theoretic criteria*, IEEE trans. on Acoustics, Speech and Signal Processing, vol. 33, no. 2, pp. 387–392, Apr. 1985.
- [133] R. WILLIAMS, S. PRASAD et A. MAHALANABIS, *An improved spatial smoothing technique for bearing estimation in multipath environment*, IEEE trans. on Acoustics, Speech and Signal Processing, vol. 36, no. 4, pp. 425–432, Apr. 1988.
- [134] G. WOLBERG et I. ALFY, *Monotonic cubic spline interpolation*, Proc. of Computer Graphics International, 1999.

- [135] X. XIANGHUA et M. MIRMEHDI, *Rags : region-aided geometric snake*, IEEE Transactions on Image Processing, vol. 13, no. 5, pp. 640–652, May 2004.
- [136] G. XU, E. SEGAWA et S. TSUJI, *Robust active contours with intensive parameters*, Pattern Recognition, vol. 27, no. 7, pp. 879–884, 1994.
- [137] S. XU et J. PRINCE, *Gradient vector flow, a new external force for snakes*, Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1997.
- [138] N. YUEN et B. FRIEDLANDER, *DOA estimation in multipath : an approach using fourth order cumulant*, IEEE Trans. on Signal Processing, vol. 45, no. 5, pp. 1253–63, 1997.
- [139] N. YUEN et B. FRIEDLANDER, *Asymptotic performance analysis of blind signal copy using fourth order cumulant*, International Jour. Adaptative Contr. Signal Proc., vol. 48, pp. 239–65, Mar. 96.
- [140] T. ZHANG et G. GOLUB, *Rank-one approximation to high order tensor*, SIAM Journal on Matrix Analysis and Applications, vol. 23, no. 2, pp. 534–550, November 2001.
- [141] S. C. ZHU et A. YUILLE, *Region competition : Unifying snakes, region growing, and bayes/mdl for multiband image segmentation*, IEEE trans. on Pattern Analysis and Machine Intelligence, vol. 18, no. 9, pp. 884–900, Sept. 1996.

Résumé :

Cette thèse est consacrée aux méthodes par sous-espaces et d'optimisation, qui sont développées et adaptées dans trois contextes : en traitement d'antenne, en analyse d'images, en traitement du signal tensoriel. Dans la première partie, des définitions concernant un problème de traitement d'antenne sont présentées, l'intérêt des statistiques d'ordre deux est mis en valeur : l'on distingue le sous-espace signal, et le sous-espace bruit. L'orthogonalité entre ces deux sous-espaces fonde les méthodes haute résolution du traitement d'antenne. Nous proposons une nouvelle méthode d'optimisation appliquée à la localisation de sources en présence de distorsions de phase, pour un grand nombre de capteurs. Dans la seconde partie de ce manuscrit, nous proposons des méthodes par sous-espaces rapides pour l'estimation de l'orientation et de l'offset de contours rectilignes. Nous présentons plusieurs méthodes d'optimisation pour l'estimation de contours distordus, approximativement rectilignes ou étoilés. Nous adaptons une antenne circulaire virtuelle à l'image à traiter, conduisant à des signaux à phase linéaire à partir de cercles concentriques. Des méthodes haute résolution du traitement d'antenne distinguent alors des valeurs de rayons pouvant être très proches. Nous considérons des cercles distordus, des ellipses. Dans la troisième partie du manuscrit, nous rappelons les définitions concernant les tenseurs, et donnons un état de l'art sur les méthodes de traitement du signal multidimensionnel, qui reposent sur la projection des données sur le sous-espace signal, et sur des méthodes d'optimisation : la troncature de rang- (K_1, \dots, K_N) de la HOSVD, l'approximation de rang- (K_1, \dots, K_N) inférieur d'un tenseur. Nous présentons une version multidimensionnelle du filtrage de Wiener. Nous proposons une procédure nonorthogonale de déploiement de tenseur, selon les directions principales du tenseur pour l'amélioration du filtrage tensoriel.

Abstract :

This thesis is devoted to subspace-based and optimization methods, developed and adapted in three contexts : in array processing, image understanding, tensor signal processing. In the first part, we present definitions about array processing, the interest of second-order statistics is emphasized : we distinguish signal and noise subspaces. Orthogonality between these subspaces is the basis of high-resolution methods of array processing. We propose a novel optimization method applied to source localization in presence of phase distortions and for a large number of sensors. In the second part, we propose fast subspace-based methods for the estimation of the orientation and offset of rectilinear contours. We present several optimization methods for the estimation of distorted contours, approximately linear, or star-shaped. We adapt a virtual circular antenna to the processed image, which yields linear phase signals from concentric circles. High-resolution methods of array processing distinguish between possibly close radius values. We consider distorted circles and ellipses. In the third part of the manuscript, we recall definitions about tensors, and provide a state-of-the-art concerning multiway signal processing, which rely on the projection of data upon signal subspace, and on optimization methods : rank- (K_1, \dots, K_N) truncation of HOSVD, lower rank- (K_1, \dots, K_N) tensor approximation. We present a multiway version of Wiener filtering. We propose a nonorthogonal tensor flattening procedure along tensor main directions for tensor filtering improvement.

Mots clés : Antenne, statistiques d'ordre deux, méthodes par sous-espaces, méthodes haute résolution, front d'onde distordu, estimation, traitement d'antenne, optimisation, interpolation par splines, image, contour, antenne linéaire, antenne circulaire, algèbre multilinéaire, décomposition tensorielle, tableau multidimensionnel, approximation de rang inférieur, filtrage.

Key words : Antenna, second order statistics, subspace-based methods, high-resolution methods, distorted wavefront, estimation, array processing, optimization, spline interpolation, image, contour, linear antenna, circular antenna, multilinear algebra, tensor decomposition, multiway array, lower rank tensor approximation, filtering.

Adresse :

Institut Fresnel, UMR CNRS 6133 et Université Paul Cézanne (Aix-Marseille III),
D.U. de Saint Jérôme, 13397 Marseille - France.