



HAL
open science

Reconstruction parcimonieuse et semi-paramétrique en imagerie radio

Fadi Nammour

► **To cite this version:**

Fadi Nammour. Reconstruction parcimonieuse et semi-paramétrique en imagerie radio. Autre [cond-mat.other]. Université Paris Cité, 2021. Français. NNT : 2021UNIP7035 . tel-03564073

HAL Id: tel-03564073

<https://theses.hal.science/tel-03564073>

Submitted on 10 Feb 2022

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

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



Université
de Paris

Université de Paris

Ecole doctorale: Astronomie et d'Astrophysique d'Ile-de-France, 127
Laboratoire: Commissariat à l'Energie Atomique et aux Energies Alternatives

Thèse de doctorat en Astronomie et Astrophysique

Reconstruction Parcimonieuse et Semi-Paramétrique en Imagerie Radio

Fadi Nammour

Présentée et soutenue publiquement le 25 novembre 2021

Directeur: Dr Jean-Luc SARCK

Composition du jury:

Président:

Dr Laure BLANC-FERAUD, CNRS

Rapporteurs:

Pr Yves WIAUX, Heriot-Watt University, Royaume-Uni

Pr André FERRARI, Université Côte d'Azur

Examineurs:

Pr Simona MEI, Université de Paris

Dr Sandrine ANTHOINE, CNRS

Pr Chiara FERRARI, Université Côte d'Azur

Dr Jean-Luc STARCK, CEA

Invité:

Dr Julien N. GIRARD, Université de Paris

Dédicace

À Beyrouth, ma ville.

Entends-tu l'aurore
Comme un songe muet
Avancer la nuit?



Les funambules voiliers.

Résumé

Dans la décennie à venir, de nouveaux instruments au sol et dans l'espace seront mis en service pour sonder l'Univers. C'est le cas du télescope spatial Euclid en optique, et de l'interféromètre SKA en radio. La finalité est de faire avancer différents sujets d'étude comme ceux portant sur l'origine de l'Univers ou la répartition de la masse et de l'énergie dans celui-ci. À cet effet, il est crucial d'extraire les informations pertinentes des données telles que la forme des galaxies, leur flux et leur taille ; et ce avec une précision élevée. Mais ces données sont corrompues par les effets instrumentaux et les différentes sources de bruit (fond du ciel, mesure, etc.).

En traitement du signal, les données collectées sont modélisées à partir des données vérité terrain (non corrompues). En effet, la mesure par l'instrument (étalonné) est équivalente à une convolution du ciel par une Fonction d'Étalement du Point (FEP). De plus, s'ajoute à la mesure un bruit supposé additif et gaussien. Il s'agit donc d'un problème inverse mal posé et mal conditionné. Dans l'état de l'art, les méthodes de reconstruction d'image en optique offrent une reconstruction satisfaisante du profil et du flux des sources. Par contre la mesure de forme sur ces images reste problématique. L'utilisation d'autres méthodes que la reconstruction d'images est donc nécessaire. De même, dans le cas radio, la reconstruction d'image avec la méthode CLEAN et ses améliorations n'est pas optimale. Ceci a également poussé les astrophysiciens à développer d'autres méthodes comme la mesure de forme par ajustement dans l'espace de Fourier.

Ma thèse a pour but d'offrir des méthodes de reconstruction d'images qui permettent d'améliorer l'estimation des informations physiques des galaxies. J'ai commencé par développer une contrainte de forme qui, une fois ajoutée à un algorithme de reconstruction, permet de régulariser le problème afin d'obtenir une solution qui conserve les propriétés de la galaxie, notamment sa forme. Par la suite, j'ai développé un algorithme de reconstruction parcimonieuse semi-paramétrique en utilisant les ondelettes et la contrainte de forme à cette fin. Après, je me suis intéressé aux méthodes d'apprentissage profond qui sont récemment entrées dans l'état de l'art de différents domaines. J'ai alors implémenté deux méthodes d'apprentissage profond pour y ajouter la contrainte de forme et étudier son impact. La première méthode d'apprentissage profond est purement fondée sur l'apprentissage machine, tandis que la deuxième utilise de plus un algorithme proximal. Pour tester différents aspects de ces méthodes, j'ai développé au cours de ma thèse trois jeux de données différents. Le premier jeu de données est un jeu de données paramétrique en optique pour tester le concept de la contrainte de forme. Les deux jeux de données restants ont été construits spécifiquement pour les méthodes d'apprentissage profond avec plus de 50 000 objets chacun ; le premier est un jeu de données optique associant des images réelles du télescope Hubble Space Telescope (HST) à des FEP réalistes afin de simuler les réponses des télescopes HST et CFHT (de résolution angulaire inférieure). Le second jeu de données utilise des galaxies paramétriques simulées avec le catalogue réaliste T-RECS et des FEP réalistes du télescope MeerKAT. J'ai ainsi pu évaluer les limites et les avantages des méthodes que j'ai développées.

Mots clés: Astrophysique, Radio-interférométrie, Parcimonie, Régularisation, Apprentissage Profond, Traitement de l'image

Abstract

In the coming decade, new ground and space instruments will be brought into use to probe the universe. This is the case with the Euclid space telescope in optics, and the SKA interferometer in radio. The purpose is to advance various study subjects such as those relating to the origin of the universe or the distribution of mass and energy in it. To this end, it is crucial to extract relevant information from data such as the shape of galaxies, their flow and size; and this with high precision. But these data are corrupted by the instrumental effects and the different sources of noise (sky background, measurement, etc.).

In signal processing, the collected data is modeled from the ground truth data (non-corrupted). Indeed, the measurement by the (calibrated) instrument is equivalent to a convolution of the sky by a point spread function (PSF). Moreover, a noise that is supposed Gaussian is added to the measurement. This is therefore an ill-posed and ill-conditioned inverse problem. In the state of the art, optical image reconstruction methods offer a satisfactory reconstruction of the profile and flow of sources. On the other hand, the shape measurement on these images remains problematic. The use of methods other than image reconstruction is therefore necessary. Likewise, in the radio case, image reconstruction with the CLEAN method and its improvements is not optimal. This also prompted astrophysicists to develop other methods such as model fitting in Fourier space for shape measurement.

My thesis aims to provide image reconstruction methods that improve the estimation of the physical information of galaxies. I started by developing a shape constraint which, when added to a reconstruction algorithm, allows to regularize the problem in order to obtain a solution which preserves the properties of the galaxy, in particular its shape. Subsequently, I developed a sparse semi-parametric reconstruction algorithm using wavelets and the shape constraint for this purpose. Afterwards I focused on deep learning methods which have recently entered the state of the art in different fields. I implemented two deep learning methods to add to them the shape constraint and study its impact. The first deep learning method is purely based on machine learning while the second also uses a proximal algorithm. To test different aspects of these methods, I developed three different datasets during my thesis. The first dataset is a parametric optics dataset to test the concept of shape constraint. The remaining two datasets were built specifically for deep learning methods with over 50,000 objects each; the first is an optical dataset combining real images from the Hubble Space Telescope (HST) with realistic PSFs to simulate the responses of HST and CFHT telescopes (lower angular resolution). The second dataset uses parametric galaxies simulated with the realistic T-RECS catalog and realistic PSFs from the MeerKAT telescope. I was thus able to assess the limits and advantages of the methods I have developed.

Keywords: Astrophysics, Radio-Interferometry, Sparsity, Regularization, Deep Learning, Image Processing

Remerciements

Je tiens à remercier L. Blanc-Féraud, Y. Wiaux, A. Ferrari, S. Mei, S. Anthoine et C. Ferrari, en tant que jury de thèse, pour avoir élargi mes horizons sur mon travail de thèse à travers des critiques constructives et des discussions autour de mon manuscrit et ma soutenance de thèse.

Je souhaite également remercier Jean-Luc Starck pour avoir été mon encadrant et directeur de thèse. Durant mon stage et, ensuite, ma thèse, Jean-Luc m'a initié aux pratiques et codes des chercheurs tout en partageant son expertise et son savoir-faire.

De plus, j'aimerais faire preuve de reconnaissance envers Julien Girard pour avoir été mon co-encadrant. Durant trois années de travail avec Julien, j'ai découvert et exercé le quotidien du chercheur de manière collaborative, ludique et pédagogique. Julien a aussi été à l'écoute lors que je rencontrais des difficultés personnelles notamment liées à la situation dans mon pays natal ou à la pandémie.

D'autre part, je remercie François Lanusse du fond du cœur pour son support et son aide précieux qui m'ont été apportés pour mon dernier projet. Sa rigueur, sa flexibilité et son approche par "gamification", m'ont permis de rester motivé, surtout durant ma dernière année de thèse.

Je souhaite également remercier toute l'équipe du CosmoStat qui m'a chaleureusement accueilli avec Morgan Schmitz durant mon stage de fin d'études. Je me souviendrai pour toujours des conversations que nous avons partagées qu'elles soient professionnelles ou personnelles. Je me souviendrai également de l'amitié intense que j'ai partagé avec le groupe de stagiaires de 2018 et mes co-bureaux Hung Tran, Santiago Casas, Kostantinos Themelis, Austin Peel, Axel Guinot, Denise Lanzieri et Lisa Goh. L'interdisciplinarité permettait d'avoir des discussions variées et enrichissantes notamment sur l'optimisation et la séparation aveugle de sources avec Christophe Kervazo, Jérôme Bobin et Rémi Carloni ; sur l'apprentissage machine et le traitement d'images avec Benjamin Rémy, Samuel Farrens, Tobias Liaudat, Zaccharie Ramzi et Florent Sureau ; sur l'astrophysique avec Martin Kilbinger, Isabella Carucci, Melis Irfan, Sandrine Pires et ma jumelle de thèse, Virginia Ajani.

Finalement, je tiens à remercier toute ma famille qui m'a toujours encouragé et motivé à me dépasser, en m'assurant les conditions les plus propices pour ce faire.

Table des matières

Table des figures	xv
Liste des tableaux	xvii
Introduction	1
1 Imageries optique et radio	3
1.1 Imagerie optique	3
1.1.1 Les télescopes optiques	3
Le système optique	3
1.1.2 Limite de diffraction et Fonction d'Étalement du Point	4
Effets atmosphériques	5
Système d'acquisition	7
1.1.3 Les jeux de données visibles	7
COSMOS paramétrique	7
CFHT2HST	8
1.1.4 Conclusion	10
1.2 Imagerie Radio	10
1.2.1 Antenne et Réseau d'Antennes	11
Antenne	11
Réseau d'antennes	12
1.2.2 Interféromètre à deux antennes	13
Source ponctuelle	13
Source étendue	15
Systèmes de coordonnées (u, v, w) et (l, m, n)	16
1.2.3 Des visibilités à l'imagerie	17
Image observée et problème inverse	17
Méthodes classiques de déconvolution	18
1.2.4 Jeu de données	19
Catalogue	19
Génération des données	20
1.2.5 Conclusion	21
2 Problème inverse de déconvolution	23
2.1 Problème inverse	23
2.1.1 Définition d'un problème inverse	23
2.1.2 La déconvolution	24
2.2 Régularisation des problèmes inverses	27
2.2.1 Utilité de la régularisation	28
2.2.2 Exemples de régularisation	29
Régularisation de Tikhonov	29
Variation totale	30

	Projection sur un ensemble	32
2.3	Méthodes proximales	32
2.3.1	Opérateur proximal	32
2.3.2	Algorithmes proximaux	33
	Algorithme Forward-Backward Splitting	34
	Algorithme des directions alternées	34
2.4	Conclusion	36
3	Parcimonie et ondelettes	37
3.1	Parcimonie	37
3.1.1	Aperçu sur la parcimonie	37
3.1.2	Représentation parcimonieuse des signaux	39
3.2	Transformée en ondelettes	41
3.3	Ondelettes continues	42
3.4	Ondelettes orthogonales	44
3.4.1	Multi-résolution	44
3.4.2	Décomposition pyramidale	46
3.4.3	Ondelettes bi-orthogonales bi-dimensionnelle (2D)	48
3.5	Starlet	49
3.5.1	Transformée en ondelette à trous	50
3.5.2	Transformée en starlet	50
3.5.3	Deuxième Génération de Starlet	52
3.6	Conclusion	53
4	Reconstruction parcimonieuse et contrainte de forme	55
4.1	Contrainte de forme	55
4.1.1	Nécessité de la forme en astrophysique	55
4.1.2	Développement de la contrainte de forme	56
	Ellipticité	56
	Des ellipticités aux produits scalaires	57
	Fenêtrage	58
4.1.3	Reformulation de la contrainte	59
4.2	Algorithme SCORE	61
4.2.1	Déconvolution parcimonieuse	61
4.2.2	SCORE	64
4.3	Expériences numériques	66
4.3.1	Implémentation	66
4.3.2	Résultats	67
4.3.3	Conclusion	68
5	Reconstruction avec Apprentissage Profond	73
5.1	Apprentissage Machine	73
5.2	Apprentissage Profond	74
5.2.1	Architecture des réseaux de neurones	74
	Le neurone	75
	Réseau de neurones et couches	77
	Réseaux de neurones convolutifs	78
5.2.2	Entraînement	80
	Jeu de données	80

	Fonction de coût	82
	Optimisation	83
5.3	Apprentissage Profond et Déconvolution	84
5.3.1	Approche directe	84
	Déconvolution	85
	Débruitage	86
5.3.2	Approche modulaire	87
5.4	Expériences numériques	87
5.4.1	En optique	87
5.4.2	En radio	90
5.5	Conclusion	91
	Conclusion	101
	Bibliographie	105
	Annexes	
A	Expression de l'ellipticité des galaxies avec des produits scalaires	117
B	Jeu de données COSMOS paramétrique	119
C	Implémentation de l'ADMMnet	121
D	Communication et activité scientifiques	123
D.1	Publications	123
D.2	Conférences	123
D.3	Enseignements	123
D.4	Formations suivies	124
D.5	Adaptation de l'enseignement au confinement	124

Table des figures

1.1	Exemple de schéma d'un télescope de type Ritchey.	4
1.2	Exemple d'images avec un instrument à ouverture circulaire.	5
1.3	Exemple de convolution et d'ajout de bruit à l'image de Calisto.	6
1.4	Exemple de trois couples du jeu de données COSMOS paramétrique.	8
1.5	Exemple de trois couple du jeu de données CFHT2HST	10
1.6	Diagramme de rayonnement d'une antenne à ouverture rectangulaire.	12
1.7	Schéma d'un interféromètre à deux antennes.	14
1.8	Schéma d'un interféromètre observant une source étendue.	15
1.9	Schéma de l'algorithme CLEAN Cotton-Schwab.	19
1.10	Exemple de trois couples du jeu de données MeerKAT3600.	21
2.1	Exemple de diviseurs de zéro en convolution.	25
2.2	Exemple de déconvolution par division dans l'espace de Fourier.	27
2.3	Exemple de déconvolution avec un filtre de Tikhonov.	30
2.4	Exemple de débruitage avec une régularisation de variation totale.	31
2.5	Représentation des deux premières itérations de l'algorithme FBS.	35
3.1	Représentations spatiale et temporelle d'un sinus avec une fréquence de 1 Hz.	39
3.2	Exemples d'ondelette mère	43
3.3	Algorithme de transformée en ondelette rapide.	47
3.4	Fonction de remise à l'échelle et l'ondelette de la transformée en starlet 1D.	52
3.5	Tranformée en Starlet de l'image Ascent.	54
4.1	Écart dans la mesure de forme de deux images quasi-identiques.	57
4.2	Représentation des bandes de shearlets à trois échelles.	59
4.3	Résultats de débruitage de la galaxie #38 (RSB = 75)	69
4.4	MSE et δ_e en fonction du RSB, en débruitage.	70
4.5	Résultats de déconvolution de la galaxie #16 (RSB = 75).	71
4.6	MSE et δ_e en fonction du RSB, en déconvolution.	72
5.1	Schéma d'un neurone neurone artificiel.	75
5.2	Exemples de fonctions d'activation.	77
5.3	Exemple de réseau cyclique où la sortie est réinjectée en entrée.	77
5.4	Schéma de la première d'une couche de convolution.	79
5.5	Schéma de sur-échantillonnage et sous-échantillonnage.	80
5.6	Schéma du U-net.	81
5.7	Exemples de galaxies reconstruites avec le jeu de données CFHT2HST.	93
5.8	Erreurs moyennes de reconstruction en fonction de la magnitude pour le jeu de données CFHT2HST.	95
5.9	Exemples de galaxies reconstruites avec le jeu de données MeerKAT3600.	97
5.10	Erreurs moyennes de reconstruction en fonction de la magnitude pour le jeu de données MeerKAT3600.	99

C.1 Exemple de reconstruction d'une galaxie avec l'ADMMnet simplifié en utilisant 500 itérations. 122

Liste des tableaux

1.1	Caractéristiques du jeu de données COSMOS paramétrique.	7
1.2	Caractéristiques du jeu de données CFHT2HST.	9
1.3	Caractéristiques du jeu de données MeerKAT3600.	21
4.1	Valeurs de γ_*	67
5.1	Résultats du jeu de données CFHT2HST.	89
5.2	Résultats du jeu de données MeerKAT3600.	90

Introduction

Dans les prochaines années, l'Univers sera observé avec une précision sans précédent avec le radio-interféromètre Square Kilometre Array (SKA) par exemple. Également d'autres télescopes seront mis en service comme Euclid et Vera C. Rubin pour le sonder. La finalité est de récolter des données afin d'apporter des éléments de réponse à des questions de domaines de recherche actifs en astrophysique, telles : Comment se forment et évoluent les galaxies ? Comment est répartie la masse dans l'Univers ? La première est associée au domaine de la formation et évolution des galaxies et la deuxième est associée à la cosmologie. Dans les deux cas, la forme de la galaxie est une information majeure ; en effet, dans le premier domaine cette information est utile pour classifier la galaxie. Dans le second domaine, cette information permet de déduire la carte de masse dans l'Univers. En effet, considérons un amas de galaxies dans l'espace, dans le régime du lentillage gravitationnel faible, un corps céleste massif, situé entre l'amas et l'observateur, induit un cisaillement gravitationnel des galaxies dans l'amas. En mesurant la forme de celles-ci, il est possible d'estimer le cisaillement et par conséquent la masse du corps massif. Il est nécessaire que l'estimation soit de haute précision puisque la contribution du cisaillement dans la mesure de forme d'une galaxie est de l'ordre de 1%. Les images brutes obtenues par les télescopes sont floutées par des effets instrumentaux et corrompues par un bruit additif provenant du fond du ciel observé, ce qui complique davantage l'estimation de la forme. En effet, il s'agit d'un problème inverse mal posé, plus spécifiquement mal conditionné. Avec les méthodes de reconstructions d'images standards, l'information sur la forme de la galaxie et les autres grandeurs physiques comme le flux ne sont pas préservées. Pour cette raison, différentes méthodes ont été développées dans les domaines optique et radio-interférométrique traitant le problème dans des espaces différents de l'espace image comme l'espace des moments quadripolaires ou l'espace des visibilitées, à titre d'exemples. Le but de cette thèse est de développer des méthodes de reconstruction d'images permettant de préserver les informations physiques des objets qui y sont contenus. Dans le chapitre 1, nous commençons par présenter le contexte, en détaillant l'imagerie optique et l'imagerie radio, respectivement. Nous posons également le problème inverse de déconvolution qui est commun aux domaines optique et radio. Nous présentons aussi les jeux de données que j'ai développés pour les expériences numériques dans notre étude. Dans le chapitre 2, nous introduisons les bases des problèmes inverses en nous attardant sur le problème inverse de déconvolution. Le chapitre 3 commence par aborder le concept de parcimonie qui est utilisé comme régularisation pour les problèmes inverses. Ensuite, la transformée en ondelette est introduite. Celle-ci est utilisée dans le chapitre 4 pour créer un cadre de déconvolution parcimonieuse. Ce cadre permet d'effectuer des expériences numériques servant de preuve de concept pour la contrainte de forme que j'ai développée. Cette dernière est utilisée pour construire l'algorithme de déconvolution parcimonieuse SCORE. Finalement dans le chapitre 5, nous présentons les bases de l'apprentissage machine en nous intéressant aux réseaux de neurones convolutifs. J'ai construit ensuite ShapeNet qui est une méthode d'apprentissage profond avec contrainte de forme et nous la comparons avec d'autres méthodes sur des jeux de données que j'ai développés.

Chapitre 1

Imageries optique et radio

1.1 Imagerie optique

L'histoire des civilisations est marquée par la présence du ciel, comme une source de questionnement et de mythes. Des lumières jusqu'à aujourd'hui, c'est avec des instruments toujours plus puissants que l'astronomie et l'astrophysique ont pu progresser. En effet, pour comprendre la nature de ces astres et de leur mouvement, Galilée développa la lunette astronomique vers la fin du XVI^e siècle, pour scruter le ciel en allant ainsi au-delà de l'observation à l'œil nu. De cet élan en 1610, il put alors observer pour la première fois les satellites de Jupiter. L'étude du mouvement relatif de la planète et des satellites permit à ce dernier de déduire que la Terre tourne autour du Soleil. Cette expérience fut l'acte de naissance de l'astronomie moderne. Depuis, les outils d'observations sont en constante amélioration, rendant de nos jours, l'observation à l'œil obsolète pour l'étude approfondie des objets astrophysiques. En effet, les caméras actuelles dans les télescopes permettent aujourd'hui par exemple d'avoir des observations à différents temps de pose (de l'imagerie ultrarapide à l'imagerie longue pose) et de capter des longueurs d'onde en dehors du domaine visible. Toujours avec une volonté d'améliorer la précision et notamment avec l'avènement du numérique, ces images sont prétraitées afin de prendre en compte les effets liés à la chaîne d'acquisition du signal (biais, bruits, déformations, etc.). Dans ce chapitre, nous allons dans un premier temps nous pencher sur la chaîne d'acquisition des images optiques pour, dans un deuxième temps, modéliser le système d'acquisition des images optiques et finalement présenter les jeux de données développés dans le cadre de notre étude.

1.1.1 Les télescopes optiques

Pour avoir des observations de plus en plus précises, les télescopes optiques sont généralement placés à l'abri de perturbations humaines et atmosphériques sur les sommets des montagnes ou des volcans comme les observatoires du Mauna Kea à Hawaï, allant même jusqu'à les placer dans l'espace comme le Hubble Space Telescope (HST). Néanmoins le principe de fonctionnement est le même. Dans cette partie, nous allons voir comment l'instrument fonctionne d'une part et d'une autre part, l'influence du milieu de propagation des ondes entre le télescope et l'objet observé.

Le système optique

Les télescopes sont constitués des éléments suivants :

- *Miroir primaire* : il s'agit d'un miroir concave qui va faire converger les rayons vers l'oculaire en passant par le miroir secondaire, voir figure 1.1. Contrairement à la lunette astronomique dans laquelle la lumière subit des réfractions à travers un

ensemble de lentilles, le miroir réfléchit les ondes et permet au télescope d'être totalement achromatique.

- *Miroir secondaire* : ce miroir permet de réfléchir les ondes venant du miroir primaire et de les rediriger vers le foyer du système optique, dont l'image est faite au travers d'un oculaire, si l'on observe à l'œil. Dans le cas d'un télescope de Newton, le miroir secondaire est plan.
- *Oculaire* : l'oculaire agit comme une loupe et permet d'agrandir l'image. Sa distance est réglable afin de faire coïncider son foyer objet avec le foyer image de l'objectif, rendant ainsi le télescope afocal. Notons que l'oculaire est généralement omis lorsque l'observation est faite avec une caméra.
- *Monture* : La monture est un support mobile de l'instrument qui permet de l'orienter de manière à pouvoir déplacer l'axe optique vers une cible. La monture est généralement motorisée, sur un télescope amateur haut de gamme ou professionnel.

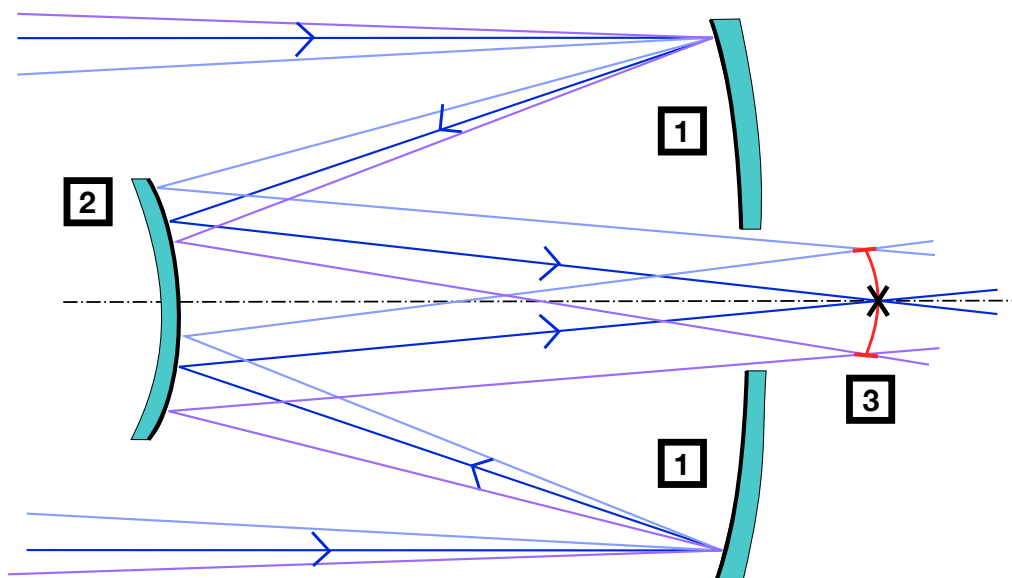


Figure 1.1 – Exemple de schéma d'un télescope de type Ritchey. Légende :
1.Miroir primaire 2.Miroir secondaire 3.Plan focal.

1.1.2 Limite de diffraction et Fonction d'Étalement du Point

En astrophysique, la séparation angulaire entre les objets observés est souvent très faible. Pour augmenter la résolution angulaire, il est possible d'augmenter la distance focale de l'instrument. Mais la capacité à distinguer deux points proches lors d'une observation est limitée par la *résolution* de l'instrument. La résolution est liée à la diffraction produite, au niveau de l'ouverture du télescope, par les fronts d'onde des rayons émis par l'objet observé. Dans le cas d'une ouverture circulaire, le motif de diffraction observé est un disque d'Airy comme illustré dans la figure 1.2 [Carroll et Ostlie 2007]. Chaque point source donne comme image la Fonction d'Étalement du Point (FEP), la figure 1.2a illustre le cas d'une ouverture circulaire. Considérons une source étendue, celle-ci peut-être exprimée comme un ensemble de sources ponctuelles. Les images de chacune de ces sources se somment l'une à l'autre pour former l'image de la source étendue, il s'agit d'une opération de *convolution*. Dans notre étude, nous considérons des images de support suffisamment réduit pour que la convolution soit localement linéaire et invariante par rapport à la translation. Dans la figure 1.3, nous

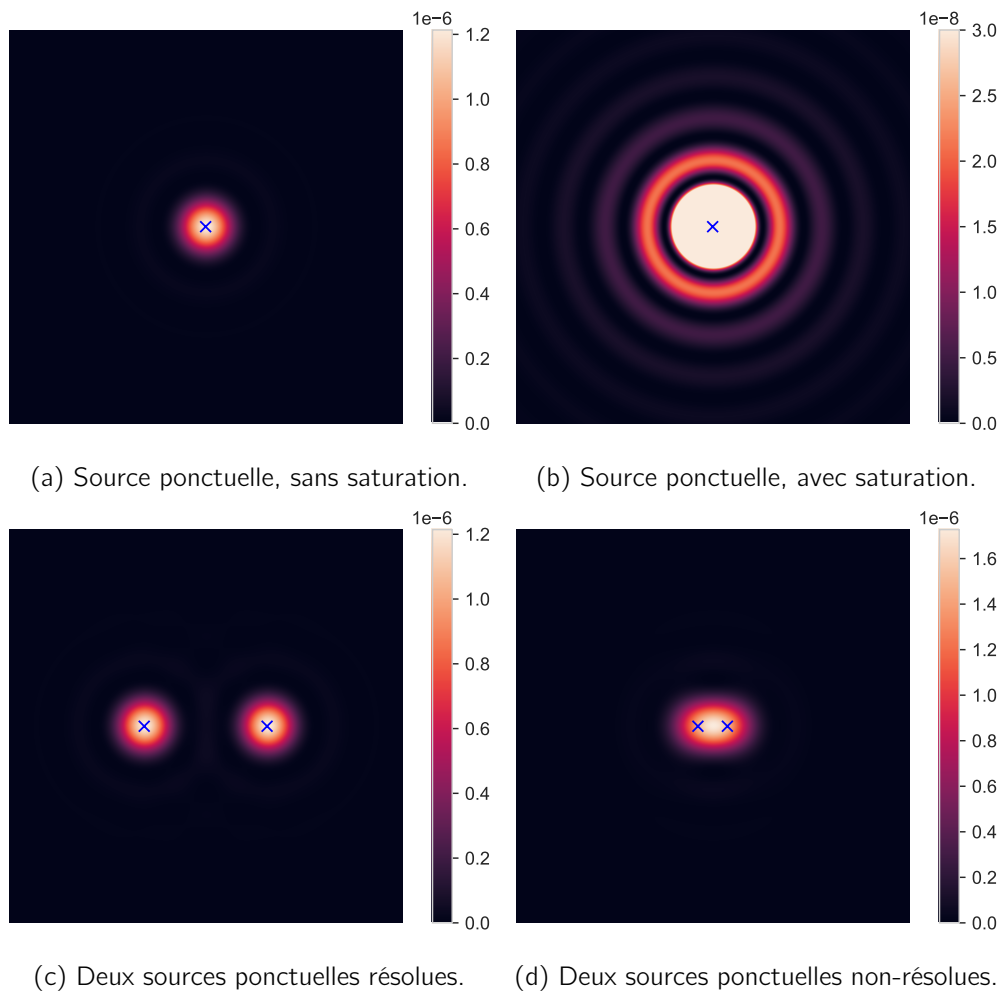


Figure 1.2 – Exemple d’images avec un instrument à ouverture circulaire. Dans chaque figure, l’image de sources ponctuelles est donnée. Les sources sont représentées par les croix bleues. Les figures 1.2a et 1.2b représentent la même image. Dans le premier cas, la plage de valeur de la barre de couleur s’étend du minimum au maximum des valeurs du disque d’Airy. Dans le deuxième cas, la plage de valeur ne s’étend pas jusqu’au maximum du disque ainsi le centre de l’image est saturé par contre la plage de valeur est adaptée pour la visualisation des anneaux externes du disque.

donnons un exemple de convolution de l’image de Calisto, un satellite de Jupiter, par un disque d’Airy qui est équivalent à l’observation du satellite avec un télescope à ouverture circulaire. Nous remarquons que l’image convoluée 1.3b est plus lisse que l’image originale 1.3a et possède moins de détails fins. Une présentation plus formelle de la convolution est donnée dans la partie 2.1.2. Dans la partie suivante, nous discutons les effets atmosphériques.

Effets atmosphériques

La résolution d’un télescope au sol ne peut s’améliorer indéfiniment avec la taille du miroir principal. C’est dû aux turbulences atmosphériques qui sont causées par des différences de température et densité sur des distances de plusieurs centimètres voire plusieurs mètres. Nous obtenons ainsi des zones où la lumière est réfractée dans des directions quasi

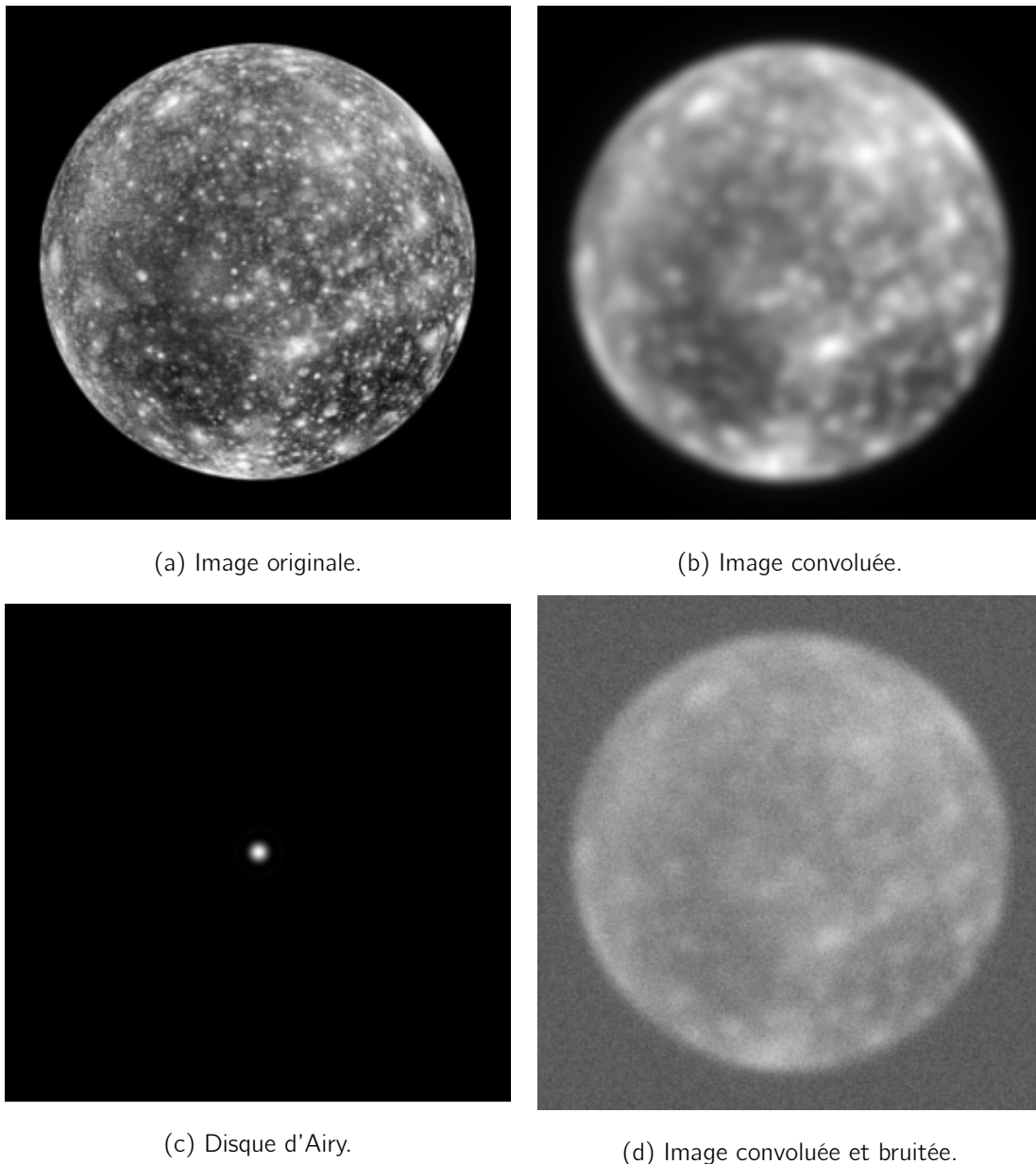


Figure 1.3 – Exemple de convolution et d'ajout de bruit à l'image du satellite de Jupiter, Calisto. La FEP utilisée est un disque d'Airy. Source : NASA/JPL/DLR, Photojournal : PIA03456.

aléatoires. L'image obtenue contient un flou supplémentaire qui est mis en relief dans le cas des sources ponctuelles. Au passage, pour cette raison, les étoiles scintillent le soir. La qualité de l'image d'une source pour une localisation donnée au sol est appelée *seeing*. En présence d'atmosphère turbulente, le trajet non droit des rayons étale davantage la réponse du télescope, ce qui dégrade la résolution angulaire du télescope. Le modèle de Kolmogorov [Tatarskii 1961] est communément utilisé pour décrire la statistique de la turbulence. Pour limiter les effets atmosphériques, les télescopes sont construits à haute altitude et les télescopes spatiaux s'affranchissent complètement de ces effets [Carroll et Ostlie 2007].

Système d'acquisition

Pour capturer l'image formée, une caméra est placée dans le plan focal du télescope. Les capteurs de la caméra sont généralement basés sur un *dispositif à transfert de charge* (Charge Coupled Device (CCD), en anglais). Ainsi chaque pixel de l'image obtenue correspond à une partie du ciel observé sous un certain angle solide, il s'agit de l'*échelle de pixel* sur le ciel. En observant, un bruit de fond va venir s'ajouter aux images. Il s'agit d'un bruit de comptage de photons qui est modélisé par un bruit de Poisson. Nous pouvons aussi l'approcher par un bruit blanc gaussien additif. Dans la figure 1.3d, l'image correspond à une observation (simulée) convoluée puis bruitée du satellite Calisto, faite avec un télescope à ouverture circulaire. L'image *vérité terrain* du satellite est donnée dans la figure 1.3a. Formellement, nous posons le modèle en équation, en notant y l'image observée et x l'image vérité terrain, comme suit :

$$y = h * x + b \quad , \quad (1.1)$$

avec $*$ l'opérateur de convolution et h le noyau de convolution correspondant aux effets de diffraction liés au télescope et aux éventuels effets atmosphériques et b correspondant au bruit additif (d'autres effets et d'autres sources de bruit peuvent entrer en jeu, telles les aberrations du système optique et le bruit de lecture (readout noise, en anglais), mais sont négligés dans notre étude). La FEP déforme l'image idéale du ciel et le bruit la détériore aussi, il est alors important de pouvoir corriger ces effets pour que les mesures effectuées par les astrophysiciens à partir de ces images soient les plus précises possibles. Une approche commune pour ce faire est de reconstruire x à partir de y : c'est le *problème inverse de déconvolution*. Cette classe de problèmes est discutée en détail dans le chapitre 2.

1.1.3 Les jeux de données visibles

Pour tester et évaluer les méthodes développées au cours de notre étude, j'ai développé deux jeux de données optiques basés sur le catalogue COSMOS.

COSMOS paramétrique

J'ai développé le jeu de données COSMOS paramétrique dans le but de créer un environnement de travail contrôlé afin de pouvoir correctement cerner la performance des méthodes développées d'un point de vue théorique, si nécessaire. Les caractéristiques de ce jeu de données sont dans le tableau 1.1

Caractéristiques communes			
Échelle du pixel		0,05''	
Taille du support		96 × 96	
Rapports Signal à Bruit (RSB)		40 ; 75 ; 150 ; 380	
Nombre d'objets par RSB		300	
Galaxie		FEP	
Profil	Sersic [Trujillo et al. 2001]	Profil	Moffat [Trujillo et al. 2001]
Type	Optique	Type	HST

Table 1.1 – Caractéristiques du jeu de données COSMOS paramétrique.

Les galaxies simulées dans ce jeu de données, ont des caractéristiques semblables à des observations spatiales avec le télescope HST (résolution angulaire, rapport signal à bruit (RSB), etc.). Les paramètres utilisés pour générer les galaxies sont tirées du catalogue COSMOS [Mandelbaum et al. 2018]. Ce catalogue contient : i) des images de galaxies obtenues avec le télescope HST et ii) les FEP correspondantes (pour plus de détails, voir l'appendice B). Les observations dans ce jeu de données sont construites à partir de l'équation 1.1. Le bruit est ajouté en choisissant son écart-type, noté σ , en fonction du RSB désiré, défini par :

$$\text{RSB}(y) = \frac{\|x * h\|_2}{\sigma}, \quad (1.2)$$

avec $\|\cdot\|_2$ la norme euclidienne, x l'image vérité terrain, h la FEP du télescope. Le signal utile correspond à $x * h$. Dans la figure 1.4, nous donnons trois exemples d'images tirées du jeu de données. La taille du support choisi pour les images permet de faire des expériences numériques plus rapidement et aussi simuler le fait qu'on travaillera sur des petites imagettes. Le code utilisé pour générer le jeu de données est écrit en Python 3.6 et utilise les bibliothèques GalSim [Rowe et al. 2015] et est publiquement disponible sur GitHub¹.

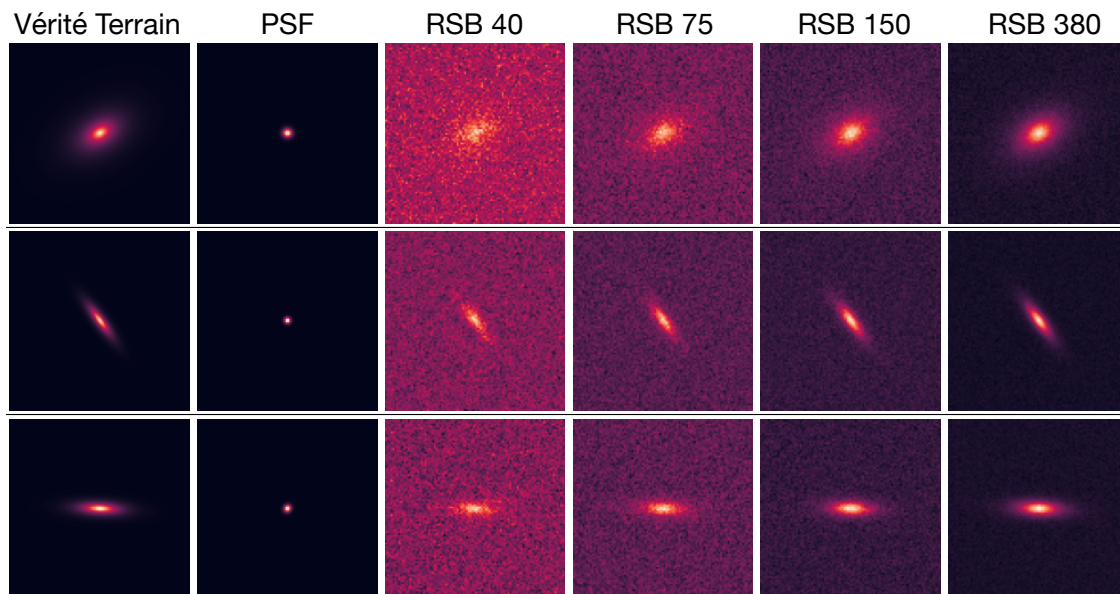


Figure 1.4 – Exemple de trois couples générés avec le jeu de données COSMOS paramétrique. Chaque couple contient la galaxie vérité terrain, la FEP, et les observations avec quatre niveaux de RSB : 40, 75, 150 et 380.

CFHT2HST

J'ai développé le jeu de données CFHT2HST pour obtenir à la fois des données réalistes et adaptées pour l'apprentissage profond. L'idée est d'utiliser des images vérité terrain, également appelées images cibles, qui correspondent à de vraies galaxies. Cette approche est limitée par le fait que toutes les images de galaxies qui existent sont floutées et contaminées par du bruit. Par contre, d'un instrument à l'autre, la qualité de l'observation peut considérablement varier. Alors nous avons choisi d'avoir des images vérité terrain obtenues avec le

1. lien vers le dépôt GitHub https://github.com/CosmoStat/score/blob/master/reproducible_research/generate_data.py.

télescope spatial HST qui possède une résolution élevée donc une FEP avec un support réduit, et ses observations sont relativement peu bruitées. Pour générer les images observées, également appelées images d'entrée, nous avons simulé des observations semblables à celles du télescope Canada-France-Hawaï (Canada France Hawaii Telescope (CFHT), en anglais) qui est au sol. Ces images sont floutées et bruitées davantage comme le montre la figure 1.5. Le jeu de données est formé de :

- *galaxie cible* : la galaxie cible sert de vérité terrain. Dans un contexte d'apprentissage machine, nous avons choisi des galaxies obtenues à partir de vraies images de galaxies. Il s'agit des images du catalogue COSMOS qui correspondent à des observations par HST convoluées par la FEP cible qui détermine ainsi la résolution de l'image (voir point suivant). Les images réelles permettent d'éviter les biais de modélisation, cependant celles-ci sont contaminées par un léger bruit additif correspondant au bruit de fond réel dans l'espace. La présence de ce bruit dans la vérité terrain est problématique pour la comparaison des performances de différentes méthodes de reconstruction. En effet, ce bruit biaise les mesures de qualité. À cette fin, nous avons également créé un jeu de donnée avec des galaxies cibles paramétriques ajustées sur les galaxies réelles.
- *FEP cible* : La FEP cible permet de définir la résolution de la galaxie cible. En effet, dans un souci de réalisme, comme il n'existe pas d'image de galaxie réelle à une résolution infinie, le rôle de la FEP est de restreindre la résolution de la galaxie cible à une résolution qui pourrait être obtenue avec un vrai télescope. De plus, comme les variations de la FEP cible sont minimales vis-à-vis de celles de la FEP d'entrée, nous avons créé une FEP cible unique qui permet d'obtenir toutes les galaxies cibles.
- *galaxie d'entrée* : cette galaxie correspond à une observation au sol qui est considérablement plus floue et plus bruitée que l'image de la galaxie cible. Ces images sont obtenues en convoluant les images du catalogue COSMOS avec la FEP d'entrée puis en ajoutant un bruit avec le même écart-type pour toutes les images. La valeur de l'écart type est calculée en tenant compte des conditions d'observation et des propriétés du télescope CFHT ².
- *FEP d'entrée* : c'est la FEP utilisée pour obtenir l'image d'entrée. Cette FEP est générée en utilisant le modèle de Kolmogorov[Racine 1996].

Les caractéristiques de ce jeu de données sont dans le tableau 1.2.

Échelle du pixel	0,187''
Taille du support	64 × 64
Nombre d'objets	51 000
Écart-type du bruit	23.59

Table 1.2 – Caractéristiques du jeu de données CFHT2HST.

Le code utilisé pour générer le jeu de données est écrit en Python 3.7.5 et utilise les bibliothèques `GalSim` [Rowe et al. 2015] et `Galaxy2Galaxy` [Lanusse et al. 2021], `TensorFlow` 1.15.2 [Abadi et al. 2015] et est publiquement disponible sur GitHub ³

2. Lien vers un Jupyter Notebook détaillant le calcul et montrant un exemple pour une galaxie : <https://github.com/CosmoStat/ShapeDeconv/blob/master/data/CFHT/HST2CFHT.ipynb>.

3. lien vers le dépôt GitHub https://github.com/fadinamour/galaxy2galaxy/tree/cfht2hst_prblm.

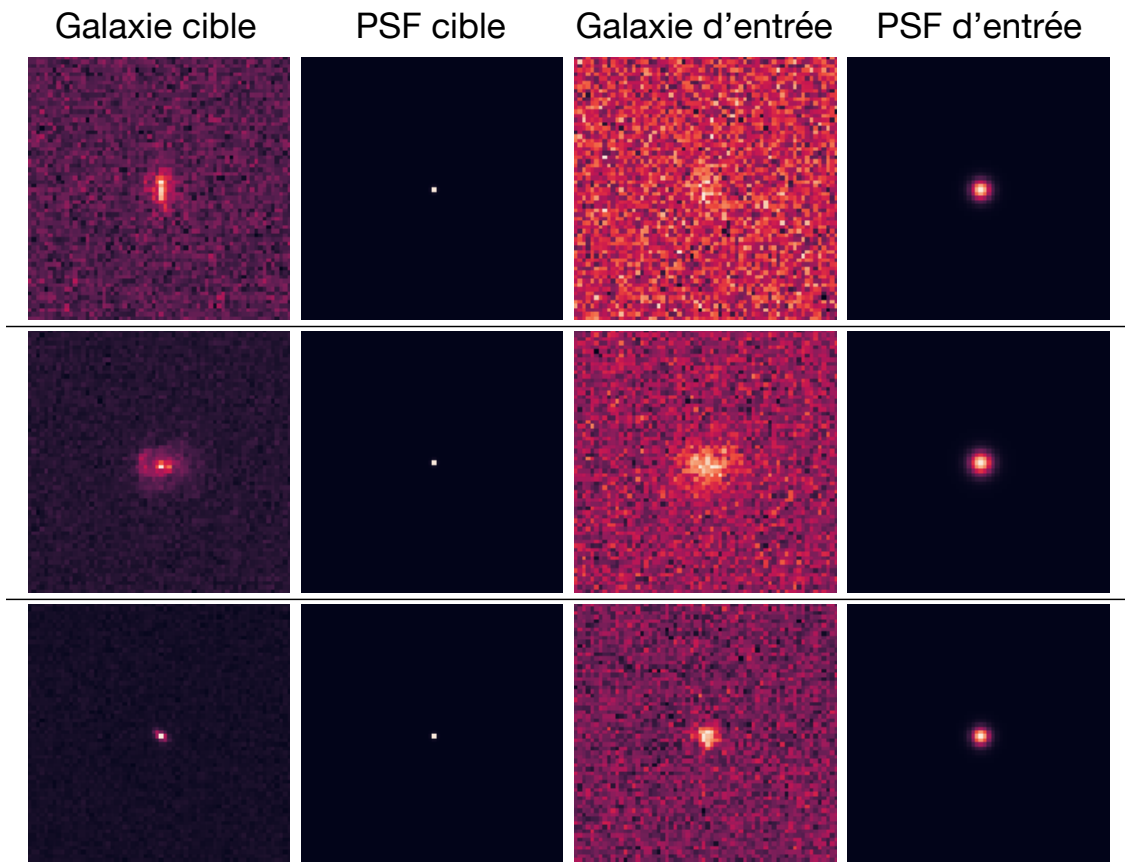


Figure 1.5 – Exemple de trois couples (galaxie cible, FEP cible, galaxie d'entrée, FEP d'entrée) générés avec le jeu de données CFHT2HST.

1.1.4 Conclusion

Dans cette partie, nous avons commencé par introduire le problème inverse de déconvolution en optique. Nous avons également présenté les deux jeux de données que j'ai développés. Le premier, COSMOS paramétrique, est utilisé pour effectuer la preuve de concept de la contrainte de forme dans le chapitre 4. Ce choix est justifié par le fait que les propriétés des données sont contrôlées. Le deuxième jeu de données, CFHT2HST est utilisé dans le chapitre 5 pour effectuer de l'apprentissage profond avec un nombre d'exemples suffisamment grand pour cette tâche et il s'agit également d'un jeu de données réalistes, ce qui permet d'avoir une évaluation plus pertinente des méthodes développées. Dans la suite nous présentons le cas de l'imagerie radio-interférométrique.

1.2 Imagerie Radio

Âgée de moins d'un siècle, la radio-astronomie est une discipline en pleine floraison. Actuellement, plusieurs radio télescopes de grande envergure sont entrain d'être déployés, par exemple le projet du télescope SKA qui à terme, va s'étendre sur les continents africain, SKA1-MID, et océanien, SKA-LOW, pour délivrer des images radio-astronomiques avec un débit de 1 terabit par seconde ($\text{Tb}\cdot\text{s}^{-1}$) [Scaife 2020]. Pour préparer la venue de cet observatoire, nous pouvons également citer les télescopes précurseurs LOw Frequency ARray (LOFAR) (aux basses fréquences dans la gamme 30-250 MHz) [Haarlem et al. 2013] et

MeerKAT (dans la gamme 1-10 GHz) [Booth et Jonas 2012] permettant déjà une grande couverture spectrale de la fenêtre radio observable depuis le sol. Ces télescopes sont constitués de réseaux d'antennes fonctionnant notamment en mode interféromètre. Dans ce chapitre, nous commencerons par aborder succinctement le fonctionnement d'une antenne avant de voir comment elles s'assemblent en réseau pour construire un interféromètre (ou un réseau phasé). Enfin, nous allons voir comment les données radio-interférométriques peuvent être inversées afin de former des images radio, en tenant compte des contraintes astrométriques.

1.2.1 Antenne et Réseau d'Antennes

Pour observer le ciel en radio, il est possible soit d'utiliser une antenne unique soit un réseau d'antennes. Dans cette section nous allons voir dans un premier temps, les propriétés de l'élément de base d'un interféromètre : l'antenne. Puis dans un second temps, nous allons nous intéresser à comment il est possible de combiner les signaux d'antennes, supposées identiques, afin de former un interféromètre radio.

Antenne

Une antenne est un dispositif qui effectue la conversion entre les ondes électromagnétiques et le signal électrique (plus précisément entre une onde se propageant dans un espace guidé et une onde en espace libre). Les ondes électromagnétiques captées par une antenne dans le régime utilisé en radioastronomie sont amplement distantes de l'antenne pour placer le cadre de travail dans la *zone de Fraunhofer*, également appelée *champ lointain* (far field, en anglais). Par ailleurs, nous pouvons déduire les propriétés en émission ou réception de ces antennes avec la simplification de la théorie de la diffraction au sens de Fraunhofer. Dans ce cas, l'onde est supposée sphérique (mais peut être considérée localement comme une onde plane) et l'énergie totale rayonnée dans une sphère ayant pour centre la source est supposée constante. Ainsi dans un milieu transparent, homogène et isotrope, les champs électrique et magnétique en un point ayant une distance r de la source, vont connaître une diminution de leur amplitude en $1/r$ par rapport à la source et de leur énergie en $1/r^2$.

En radioastronomie, les antennes sont utilisées en tant que récepteurs d'onde radio. Par contre, les propriétés d'une antenne énoncées ci-dessous, sont les mêmes que ce soit en émission ou en réception tel que garantit le théorème de réciprocité de Carson :

- *la directivité* : c'est le caractère directionnel d'une antenne à recevoir ou émettre une majorité de la fraction de l'énergie dans une direction donnée. Pour calculer la directivité, nous définissons la puissance totale, p_0 , émise par la source la puissance, $p(\theta, \phi)$, reçue par une antenne située dans une direction (θ, ϕ) , et la puissance, p_{iso} , reçue par une antenne isotrope située dans la même direction [Boyer 2011] :

$$p_{iso} = \frac{p_0}{4\pi} \quad (1.3)$$

La directivité s'écrit alors [Boyer 2011] :

$$d(\theta, \phi) = \frac{p(\theta, \phi)}{p_{iso}} \quad (1.4)$$

- *le gain* : en pratique, une antenne n'absorbe pas toute la puissance émise par une source dans sa direction mais une fraction de celle-ci. Ainsi nous définissons le gain pour une direction (θ, ϕ) par $g(\theta, \phi) = \eta d(\theta, \phi)$ avec $\eta \in [0, 1]$ le rendement total de l'antenne.

- *l'aire effective* : cette aire correspond à l'aire équivalente d'une surface perpendiculaire au vecteur d'onde de l'onde émise par la source, recevant la même puissance que l'antenne. Cette grandeur est également liée à la *sensibilité* de l'antenne qui indique le seuil du signal le plus faible que l'antenne peut détecter.
- *le diagramme de rayonnement* : c'est un diagramme qui représente le gain de l'antenne en fonction de sa direction que ce soit une représentation bi-dimensionnelle (2D) ou tri-dimensionnelle (3D). Un diagramme présente en général un lobe primaire et des lobes secondaires. Pour observer une cible avec une antenne, le lobe primaire de l'antenne est pointé vers cette cible, les signaux reçus via les lobes secondaires sont traités comme du bruit. Nous pouvons alors définir la résolution angulaire d'une antenne comme la *largeur de faisceau à mi-puissance* (Full Width at Half Maximum (FWHM), en anglais et également appelée *ouverture à -3 dB* du lobe primaire (voir figure 1.6).

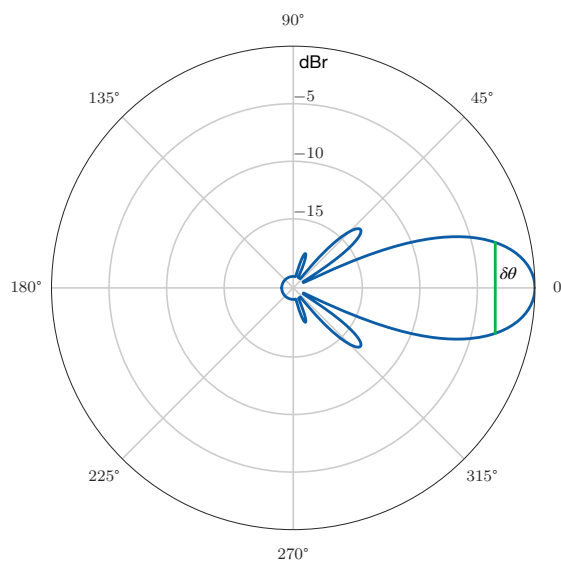


Figure 1.6 – Diagramme de rayonnement d'une antenne à ouverture rectangulaire. La puissance est exprimée en décibel relatif, noté dBr, qui correspond à une normalisation par le maximum. $\delta\theta$ est la largeur de faisceau à mi-puissance.

Il est important de noter que les propriétés d'une antenne sont directement liées à sa géométrie, par exemple la résolution est liée à son envergure totale. Effectivement, la résolution angulaire d'une antenne est généralement approximée par $\delta\theta = \lambda/D$ (en radians), avec D la dimension la plus importante de l'antenne.

Réseau d'antennes

Les besoins croissants en sensibilité et en résolution angulaire ont poussé les astronomes à fabriquer des antennes radio de plus en plus grandes. La résolution angulaire et la sensibilité étant respectivement proportionnelles à $1/D$ et $1/D^2$, il était justifié de construire des répliques géantes d'antennes paraboliques, comme le radiotélescope d'Arecibo qui avait un diamètre de 300 m. Cependant, la difficulté croissante de construction, d'opération et de maintenance de ces télescopes ont rendu impossible des structures très grande surface collectrice. D'ailleurs, c'est le 1er décembre 2020 que le télescope Arecibo s'est effondré suite à une rupture de câbles et à un défaut de maintenance [Anish Roshi et al. 2021]. C'est au

contraire en morcelant l'ouverture en petits morceaux que l'idée d'utiliser les antennes dans un réseau s'est formée. Chaque petit morceau est constitué d'un petit radiotélescope dont les dimensions et les contraintes techniques sont facilement contrôlées. Utiliser un réseau d'antennes permet de former un diagramme de rayonnement virtuel, associé à une antenne fictive aux propriétés de sensibilité et de résolution accrues qui ne seraient pas forcément atteignables avec un télescope physique unique. Pour illustrer l'apport des réseaux d'antennes, le 10 avril 2019, le télescope Event Horizon Telescope (EHT) avec une configuration s'étendant sur tous les continents du globe, a permis d'obtenir la première image d'un trou noir de la galaxie Messier 87 (M87) avec une résolution angulaire théorique de 25 microarcsecondes (μas) [Event Horizon Telescope Collaboration et al. 2019].

En général, il existe deux catégories de réseaux d'antennes :

- *les réseaux phasés* : les ondes captées par chaque antenne subissent un décalage de phase pour obtenir des signaux cohérents qui seront ensuite sommés pour obtenir le signal d'intérêt provenant d'un faisceau synthétique. Il est alors possible de pointer électroniquement la direction d'observation en contrôlant le décalage de phase. Ces décalages de phase peuvent être obtenus par des délais vrais entre les signaux des antennes (avec des câbles ou par des décalages d'échantillons) ou des coefficients complexes sur les signaux numérisés.
- *les interféromètres* : les ondes captées par chaque antenne sont d'abord mélangés deux à deux dans un corrélateur (par leur somme ou leur produit de corrélation) puis le résultat est ensuite moyenné temporellement en sortie du corrélateur. Le signal à la sortie est stocké pour ensuite subir un post-traitement incluant nettoyage, rééchantillonnage et étalonnage avant d'être utilisé pour l'imagerie. Chaque échantillon du ciel peut être considéré approximativement comme un échantillon de la transformée de Fourier du ciel. Il s'agit d'une *synthèse d'ouverture*.

Dans le cadre de notre travail, nous allons nous restreindre à l'usage du mode interférométrie qui est détaillé dans les sections qui suivent.

1.2.2 Interféromètre à deux antennes

Les premiers radio-interféromètres ont été construits dans les années 1950 et ont mené à la découverte notamment des quasars et l'étude des premières sources radio découvertes dans les années 1930 par, entre autres, Jansky et Reber. Le tout premier radio-interféromètre consistait de deux antennes, à placer en regard du nombre prévu d'antennes pour SKA qui sera de l'ordre de 10^5 . Cependant le principe de fonctionnement est le même et peut à chaque fois se ramener au cas d'un interféromètre à deux antennes. Pour cela, nous l'expliquerons dans ce qui suit en se basant sur le cas le plus simple : l'interféromètre à 2 antennes avec un seul corrélateur.

Source ponctuelle

Dans la figure 1.7, les antennes sont supposées identiques et séparées d'une distance b , appelée *ligne de base*. Considérons une source ponctuelle émettant une onde plane monochromatique de fréquence ν , atteignant les antennes de sorte à former un angle θ avec la ligne de base au niveau de l'antenne 1. Le champ électrique de cette onde induit une tension $e_1(t) = a \cos(2\pi\nu(t - \tau))$ à l'antenne 1 et une tension $e_2(t) = a \cos(2\pi\nu t)$ à l'antenne 2, a étant l'amplitude et τ le délai géométrique qui vaut $b \cdot \cos(\theta)/c$. Ainsi les signaux sont

multipliés et moyennés dans le corrélateur comme suit :

$$e_1(t)e_2(t) = a^2 \cos(2\pi\nu t) \cos(2\pi\nu(t - \tau)) \quad (1.5)$$

$$= \frac{a^2}{2} [\cos(4\pi\nu t - 2\pi\tau) + \cos(2\pi\nu\tau)] \quad (1.6)$$

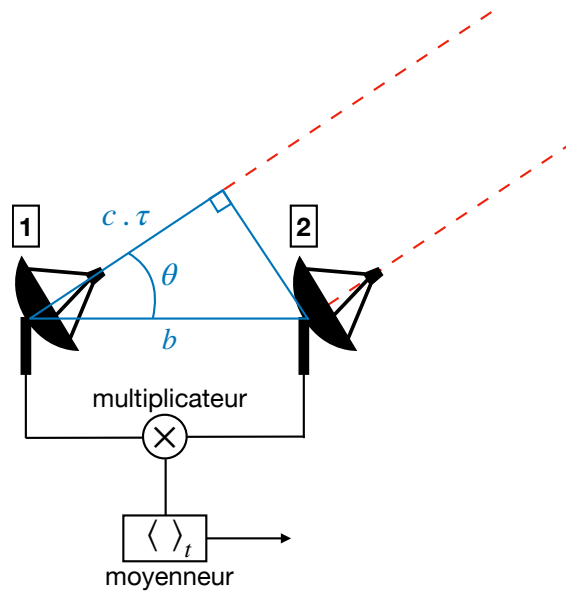


Figure 1.7 – Schéma d'un interféromètre à deux antennes.

Ce produit est moyenné sur une durée suffisamment longue pour annuler les oscillations rapides causées par $4\pi\nu t$, l'argument dépendant explicitement du temps, ce qui donne :

$$R(\tau) = \langle e_1(t)e_2(t) \rangle_t \quad (1.7)$$

$$= \frac{a^2}{2} \cos(2\pi\nu\tau) \quad (1.8)$$

Notons que R dépend de τ et présente des modulations lentes décrivant le motif d'interférence du signal émis par la source et reçu par cette ligne de base. Cette grandeur n'est pas fixe dans le temps car le mouvement de la Terre mène au déplacement apparent de la source dans le ciel, ce qui fait varier l'angle θ et par conséquent le délai τ . En pratique, ce délai est compensé au sein de l'interféromètre de sorte à rester constant définissant ainsi le *délai de référence*, τ_0 . Avec la correction continue de ce délai τ , le produit de corrélation $R(\tau)$ est ainsi toujours maximum, en ce qui permet une observation en mode "poursuite" d'une source (placée en *centre de phase*).

Maintenant, en notant λ la longueur d'onde correspond à la fréquence ν , la phase ϕ d'une frange s'écrit alors :

$$\phi = 2\pi\nu\tau \quad (1.9)$$

$$= 2\pi\nu b \frac{\cos(\theta)}{c} \quad (1.10)$$

$$= 2\pi \frac{b \cos(\theta)}{\lambda} \quad (1.11)$$

Nous pouvons alors définir résolution angulaire d'un interféromètre comme étant la distance angulaire projetée sur le ciel entre deux franges brillantes consécutives et qui vaut environ λ/b . Nous retrouvons ainsi une analogie avec la résolution d'une antenne où la résolution ne dépend plus explicitement de l'ouverture D du télescope mais de la distance $b \cos(\theta)$ de la *ligne de base projetée*, vue depuis la direction de référence. L'interfrange est chromatique et dépend donc explicitement de b , de l'angle d'incidence θ ainsi que de la longueur d'onde d'observation λ . La collection de toutes les mesures apportées par chaque interféromètre d'un réseau sera importante pour l'étape d'imagerie par synthèse d'ouverture. Dans le cas général, les sources observées en radio-astronomie ne sont pas forcément toutes monochromatiques ou ponctuelles, pour cela il faut élargir notre étude à un cas plus réaliste dans la partie qui suit.

Source étendue

En se basant sur la figure 1.8, considérons désormais une source réelle étendue déterminée par l'angle Ω sur la sphère céleste. Le vecteur \mathbf{b} , de l'espace correspond maintenant à la ligne de base, \mathbf{s}_0 est un vecteur unité pointant vers la direction θ_0 qui est la direction de référence de la mesure (ou centre de phase), \mathbf{s} est un vecteur pointant de l'interféromètre vers une portion élémentaire d'angle solide $d\Omega$ de la source. Comme une source réelle n'est pas généralement monochromatique, notons $i_\nu(\mathbf{s})$ l'intensité de la portion $d\Omega$ pour une bande de fréquences centrée en ν et de largeur $\Delta\nu$. Ainsi en adaptant l'équation 1.7 pour des antennes d'aire effective $a(\mathbf{s})$ en direction de \mathbf{s} , la sortie du corrélateur s'écrit [Thompson, Moran et Swenson 2017] :

$$dR = a(\mathbf{s})i_\nu(\mathbf{s})\Delta\nu \cos(2\pi\nu\tau) d\Omega \quad , \quad (1.12)$$

Les coordonnées $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ et $(\mathbf{l}, \mathbf{m}, \mathbf{n})$ sont détaillées dans la partie suivante. Maintenant τ peut s'exprimer sous la forme d'un produit scalaire de $\mathbf{b} \cdot \mathbf{s}$ divisé par c (homogène donc à un temps).

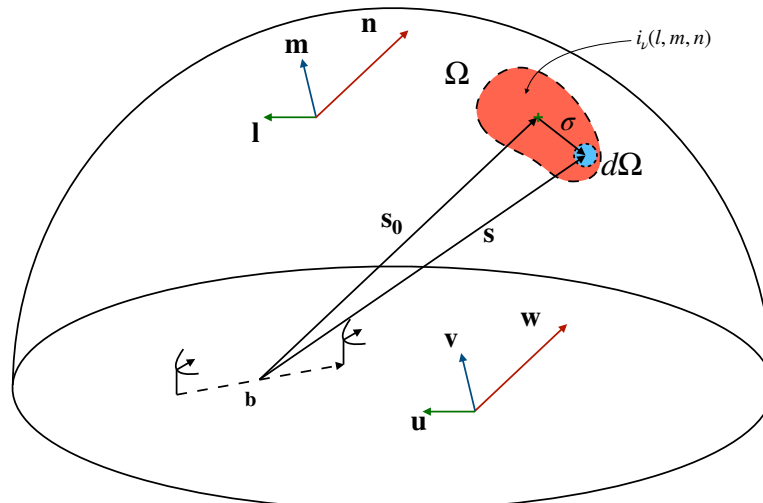


Figure 1.8 – Schéma d'un interféromètre observant une source étendue.

Comme la majorité des sources en astrophysiques sont spatialement incohérentes, il est légitime de faire l'hypothèse que la sortie du corrélateur, correspondant à l'ensemble de la

source d'extension Ω , est la somme de celles des portions élémentaires, ce qui donne :

$$I = \Delta\nu \int_{\Omega} a(\mathbf{s}) i_{\nu}(\mathbf{s}) \cos\left(2\pi\nu \frac{\mathbf{b} \cdot \mathbf{s}}{c}\right) d\Omega \quad . \quad (1.13)$$

En choisissant une *position de référence* \mathbf{s}_0 basée sur la compensation du délai τ_0 , nous introduisons $\boldsymbol{\sigma} = \mathbf{s} - \mathbf{s}_0$ qui est un vecteur existant sur la calotte sphérique définie par Ω (cf. figure 1.8). L'équation 1.13 se réécrit à l'aide des formules d'addition trigonométriques :

$$I = \Delta\nu \cos\left(2\pi\nu \frac{\mathbf{b}\mathbf{s}_0}{c}\right) j_1 - \Delta\nu \sin\left(2\pi\nu \frac{\mathbf{b}\mathbf{s}_0}{c}\right) j_2 \quad , \quad (1.14)$$

avec

$$j_1 = \int_{\Omega} a(\boldsymbol{\sigma}) i_{\nu}(\boldsymbol{\sigma}) \cos\left(2\pi\nu \frac{\mathbf{b}\boldsymbol{\sigma}}{c}\right) d\Omega \quad (1.15)$$

$$j_2 = \int_{\Omega} a(\boldsymbol{\sigma}) i_{\nu}(\boldsymbol{\sigma}) \sin\left(2\pi\nu \frac{\mathbf{b}\boldsymbol{\sigma}}{c}\right) d\Omega \quad (1.16)$$

Alors nous énonçons le *théorème de van Cittert-Zernike* qui relie la fonction de visibilité complexe V à la distribution d'intensité sur la région Ω (ce théorème est généralisable à tout le ciel visible) [Michelson 1891 ; Taylor et al. 1999] :

$$V(\mathbf{b}) = j_1 - ij_2 \quad (1.17)$$

$$= \int_{\Omega} a(\boldsymbol{\sigma}) i_{\nu}(\boldsymbol{\sigma}) \exp\left(-2\pi i \nu \frac{\mathbf{b}\boldsymbol{\sigma}}{c}\right) d\Omega \quad (1.18)$$

On appelle également $V(\mathbf{b})$ visibilité qui est homogène à une densité de flux mesurée en Jansky (Jy) tel que $1 \text{ Jy} = 10^{-26} \text{ W.m}^{-2}.\text{Hz}^{-1}$.

Systèmes de coordonnées $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ et $(\mathbf{l}, \mathbf{m}, \mathbf{n})$

Pour mieux appréhender la fonction de visibilité à partir de l'équation 1.18, nous utilisons le repère $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ et son dual $(\mathbf{l}, \mathbf{m}, \mathbf{n})$ tels que dans la figure 1.8. Le repère $(\mathbf{l}, \mathbf{m}, \mathbf{n})$ est un trièdre droit qui a pour origine le centre de phase de pointage \mathbf{s}_0 ainsi les coordonnées \mathbf{l} , \mathbf{m} et \mathbf{n} sont les composantes du vecteur \mathbf{s} quelconque. L'axe \mathbf{l} pointe vers l'Est, \mathbf{m} vers le Nord et \mathbf{n} va dans la direction sortante le long de la ligne de visée. Le repère $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ est un repère 3D dont les axes sont colinéaires aux axes du repère $(\mathbf{l}, \mathbf{m}, \mathbf{n})$ et permet de repérer les coordonnées des vecteurs lignes de bases projetées par rapport à la direction de pointage \mathbf{s}_0 . Pour plus de détails à propos des définitions de ces repères, voir [Taylor et al. 1999]. Ainsi $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ est un repère pour la position des visibilités et $(\mathbf{l}, \mathbf{m}, \mathbf{n})$ est celui pour la position de la source observée. Dans ce système de coordonnées, nous avons [Taylor et al. 1999] :

$$\frac{\mathbf{b}\boldsymbol{\sigma}}{\lambda} = ul + vm + w(n-1) \quad , \quad (1.19)$$

$$d\Omega = \frac{dl dm}{n} \quad , \quad (1.20)$$

$$n = \sqrt{1 - l^2 - m^2} \quad (1.21)$$

Ainsi l'équation 1.18 se réécrit :

$$V(u, v, w) = \int_{\Omega} a(\boldsymbol{\sigma}) i_{\nu}(\boldsymbol{\sigma}) \exp\left[-2\pi i \left(ul + vm + w(\sqrt{1 - l^2 - m^2} - 1)\right)\right] \frac{dl dm}{\sqrt{1 - l^2 - m^2}} \quad (1.22)$$

Pour notre étude, les images couvriront un champ réduit du ciel donc les valeurs de l et m sont suffisamment petites pour avoir :

$$V(u, v) \approx \int_{\Omega} a(\boldsymbol{\sigma}) i_{\nu}(\boldsymbol{\sigma}) \exp[-2\pi i (ul + vm)] dl dm \quad (1.23)$$

Dans ce cas, la distribution de luminosité de la source est confinée dans une région du ciel localement assimilable à une surface plane. En conséquence, cette simplification lie la fonction de visibilité complexe à sa transformée de Fourier 2D de la distribution d'intensité. Ainsi dans un interféromètre, chaque ligne de base permet l'accès à une composante de visibilité qui correspond à un échantillon de la transformée de Fourier de la luminosité du ciel. Dans la section suivante, nous allons voir comment, à partir de ces échantillons que sont les visibilités, nous pouvons reconstruire une approximation de l'image du ciel observé.

1.2.3 Des visibilités à l'imagerie

À l'instar des télescopes optiques, les télescopes radio permettent également d'obtenir des images du ciel. Dans le cas de l'interférométrie radio par synthèse d'ouverture, ces images sont obtenues à partir des visibilités mesurées.

Image observée et problème inverse

Les lignes de bases longues sont sensibles aux hautes fréquences spatiales du ciel tandis que les lignes de bases courtes couvrent les basses fréquences spatiales. Ainsi un interféromètre à N antennes est en fait une série de filtres spatiaux dont la bande passante dépend de la longueur de la ligne de base, la direction dans le ciel et la fréquence. La collection de tous ces échantillons de la fonction de visibilité porte des informations spatiales (il s'agit de *fréquences spatiales*), dans le plan de Fourier, caractéristique de la source.

Les données brutes mesurées par l'interféromètre sont contaminées par un bruit additif. Tout d'abord, ces données subissent une étape d'étalonnage mais comme cela est hors le cadre de notre étude nous allons supposer que nos données sont parfaitement étalonnées. Dans un premier temps, les échantillons de visibilités sont projetés sur une grille uniforme (étape dite de *gridding*) puis pondérés. Il existe plusieurs sortes de pondération qui permettent d'optimiser soit la résolution angulaire soit le RSB, effectivement, en donnant un poids élevé aux visibilités représentant les hautes fréquences nous augmentons artificiellement la résolution angulaire mais également le bruit. À l'inverse, une pondération *naturelle* donnera un poids statistique supérieur aux informations mesurées par les lignes de base courtes, apportant un meilleur RSB au détriment de la résolution angulaire. La pondération de Briggs [Briggs 1995] permet d'avoir un équilibre entre le RSB et la résolution angulaire. Dans notre cas, nous considérons une pondération *uniforme* où toutes les visibilités reçoivent le même poids, cette pondération permet d'avoir le plus haut RSB. Notons alors M , l'opérateur qui tient compte de l'échantillonnage des visibilités, une fois la projection sur la grille et la pondération effectuées. Échantillonner des points de la fonction de visibilité complexe est équivalent à masquer les points non échantillonnés, pour cela M est souvent appelé *masque*. Notons ϵ , le bruit additif sur les visibilités, nous exprimons alors la fonction

de visibilité observée V_y en fonction de la vraie visibilité V par :

$$V_y = M(V + \epsilon) \quad . \quad (1.24)$$

D'après le théorème de la convolution, l'équation 1.24 devient :

$$\mathcal{F}^{-1}(V_y) = \mathcal{F}^{-1}(M) * \mathcal{F}^{-1}(V) + \mathcal{F}^{-1}(M) * \mathcal{F}^{-1}(\epsilon) \quad , \quad (1.25)$$

avec \mathcal{F} la transformée de Fourier. En notant $y = \mathcal{F}^{-1}(V_y)$, $h = \mathcal{F}^{-1}(M)$, $x = \mathcal{F}^{-1}(V)$ et $\eta = \mathcal{F}^{-1}(M) * \mathcal{F}^{-1}(\epsilon)$, le problème se récrit :

$$y = h * x + \eta \quad . \quad (1.26)$$

On appelle y l'image observée dite *dirty image*, en anglais, h la FEP dite *dirty beam* en anglais. Comme dans le cas optique, pour accéder à l'image vérité terrain x , il faut déconvoluer l'observation bruitée y . Il s'agit alors également d'un problème inverse de déconvolution.

Méthodes classiques de déconvolution

Dans l'état de l'art en imagerie par radio-interférométrie, deux familles de méthodes sont communément utilisées pour reconstruire les images : des méthodes variationnelles comme la Méthode d'Entropie Maximale (MEM) (Maximum Entropy Method, en anglais) et des méthodes itératives comme la méthode CLEAN.

D'une part, comme son nom l'indique, la solution obtenue après la méthode MEM est celle qui maximise une entropie donnée pour l'image [Frieden 1972 ; Narayan et Nityananda 1986 ; Wernecke 1977]. Cette solution correspond à une image ayant des pixels positifs avec une plage de valeur réduite.

D'autre part, le principe de CLEAN est de supposer que la source est constituée d'un ensemble fini de points source. Ainsi l'algorithme va restituer ces points et leur intensité dans une approche de *Matching Pursuit* [Bergeaud et Mallat 1995]. L'ensemble de ces points est convolué avec un noyau lisse construit à partir de la FEP (correspondant au *dirty beam* dans l'algorithme), il s'agit du *CLEAN beam* (également appelé *restoring beam*). Généralement le CLEAN beam correspond à une fonction gaussienne 2D ajustée sur la FEP. La solution est alors de la forme :

$$\tilde{x} = a * g + r \quad , \quad (1.27)$$

avec g le CLEAN beam, a la carte des points reconstruits correspondant aux sources détectées par CLEAN, également appelés les *composantes de CLEAN* (CLEAN components, en anglais) et r le résidu. La version originale proposée dans [Högbom 1974] est illustré dans l'algorithme 1.1. Parmi les différentes versions qui l'ont suivie, il existe deux variantes majeures. La première est la version de [Clark 1980] consistant à optimiser l'algorithme en utilisant la Fast Fourier Transform (FFT) et structurant l'algorithme en suivant une stratégie de cycles mineurs, se déroulant dans l'espace image, et de cycles majeurs se déroulant dans l'espace des visibilitées. La deuxième version est celle de [Schwab 1984] qui est aussi une amélioration de la version de Clark se basant sur les cycles majeurs et mineurs pour ajouter également un va-et-vient entre l'espace des visibilitées projetées (avec coordonnées discrètes) et celui des visibilitées non projetées comme le montre la figure 1.9.

L'algorithme CLEAN est plus rapide que MEM et est également plus répandu en imagerie radio. Néanmoins dans les deux cas, la reconstruction des sources ponctuelles est satisfaisante, par contre elle est plus problématique dans le cas des sources étendues. Dans cet esprit, d'autres méthodes ont été explorées dans les dernières années, certaines sont

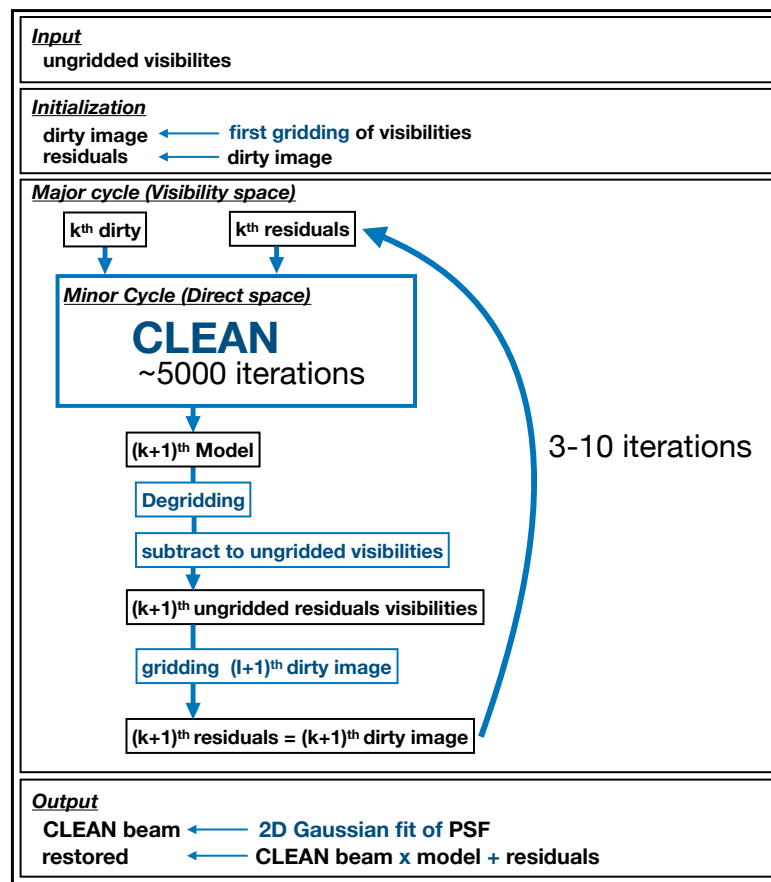


Figure 1.9 – Schéma de l’algorithme CLEAN Cotton-Schwab. Les variables sont notées en noir et les opérations en bleu.

basées sur la parcimonie, par exemple [Carin 2009 ; Carrillo, McEwen et Wiaux 2012, 2014 ; Garsden et al. 2015 ; Wiaux et al. 2009].

1.2.4 Jeu de données

Pour pouvoir effectuer des expériences numériques dans la suite, il est important de pouvoir les tester sur un jeu de données pertinent. En particulier, pour l’apprentissage profond (Deep Learning, en anglais), il est nécessaire d’avoir un nombre massif d’images, d’au moins plusieurs dizaines de milliers, pour obtenir des résultats pertinents. À notre connaissance, un jeu de données radio public de cette ampleur et convenable à notre cadre de travail n’existe pas. Pour cela, nous avons développé notre propre jeu de données composé de plus de 50 000 images avec des FEP réalistes semblables à celles du télescope MeerKAT et des galaxies paramétriques avec des propriétés réalistes tirés du catalogue Tiered Radio Extragalactic Continuum Simulation (T-RECS) [Bonaldi et al. 2019]. Les détails sont donnés dans ce qui suit.

Catalogue

La base T-RECS a pour but d’offrir des catalogues d’objets radio réalistes afin de simuler des observations à des profondeurs variées permettant de prédire les résultats qui

Algorithme 1.1 CLEAN (Högbom 1974)

tâche : déconvoluer x à partir de y et h .

paramètres : $g \in]0, 1]$ gain.

initialisation : $r^{(0)} \leftarrow y$, $a^{(0)} \leftarrow 0$, $i \leftarrow 0$, $\sigma \leftarrow$ estimation de l'écart-type du bruit dans y .

tant que $\max(r^{(i)}) > 3\sigma$ **faire**

$(p, q) \leftarrow \operatorname{argmax}(r^{(i)})$

Soustraction d'une fraction du maximum en utilisant la FEP centrée et mise à l'échelle.

Ajout de $f = g.r^{(i)}[p, q]$ aux coordonnées (p, q) de l'image modèle $a^{(i)}$.

$i \leftarrow i + 1$

fin tant que

renvoyer : $x^{(i)}$

seront obtenus avec les télescopes radio existant et à venir tel que SKA. Les catalogues contiennent deux types majeurs de galaxies : les *galaxies à formation d'étoiles* (Star Forming Galaxy (SFG), en anglais) et les *Noyaux Actifs de Galaxies* (NAG) (Active Galactic Nucleii (AGN), en anglais). La plage de fréquences de la base T-RECS s'étend de 50 MHz à 20 GHz et le champ de vision est de 25 deg^2 .

Génération des données

Le jeu de données généré contient plusieurs dizaines de milliers de couples de galaxies et de FEP dont les caractéristiques sont dans le tableau 1.3. Dans nos simulations, nous créons la vérité terrain à partir des informations géométriques du catalogue T-RECS. Pour ce qui est de la FEP, nous utilisons des réalisations simulées de la FEP observationnelle de l'instrument MeerKAT pour des durées d'observations typiques de 2h. En effet, la FEP est complètement déterminée par : i) la longueur d'onde d'observation, ii) le temps d'intégration de l'observation, iii) la distribution du réseau « vue » depuis la source pendant l'observation.

Pour simuler une variété des cas d'observations en poursuite de source réalistes, nous avons généré des réalisations de FEP intégrées sur 2h pour des directions de source aléatoires. Cependant, du fait que le réseau est au sol donc contraint par l'horizon, toutes les directions du ciel ne permettent pas d'observer des sources pendant 2h. Il existe moins de directions du ciel permettant d'observer une source pendant 8h consécutives, qu'il n'y a de possibilités d'observer. D'autre part, les FEP intégrées sur des temps courts (par exemple 1h) ont un aspect moins lisses et présentent des lobes secondaires importants par rapport aux FEP intégrées sur des temps longs (par exemple 8h). En effet, un temps d'observation plus grand permet d'accumuler davantage d'échantillons variés dans le plan de Fourier, ce qui diminue les artefacts dus à l'incomplétude du masque d'échantillonnage. Alors la durée de 2h est un compromis permettant d'explorer différentes formes de FEP afin que le jeu de données contienne multiples directions possibles.

Le code utilisé pour générer le jeu de données est écrit en Python 3.6 et utilise les bibliothèques `GalSim` [Rowe et al. 2015], `TensorFlow` 1.15.2 et est publiquement disponible sur GitHub⁴. Le bruit est rajouté à la volée et est réglé suivant les applications. Le niveau de bruit est mesuré avec le pic RSB (peak SNR, en anglais), noté PSNR et défini comme le rapport du maximum du signal utile sur l'écart-type du bruit (définition non standard mais utile aux radioastronomes). Ainsi la figure 1.10 donne un exemple de trois objets produits avec notre jeu de données.

4. lien vers le dépôt GitHub https://github.com/fadinammour/galaxy2galaxy/tree/radio_data.

Caractéristiques communes			
Échelle du pixel		0,58''	
Fréquence		3600 MHz	
Taille du support		128 × 128	
Nombre d'objets		51 000	
Galaxie		FEP	
Type	SFG	Type	MeerKAT
Profil	exponentiel	Nombre d'antennes	64
Taille minimale	6,5 pixels	Temps d'observation	2 heures
Taille maximale	80 pixels	Pas de temps	300 secondes

Table 1.3 – Caractéristiques du jeu de données MeerKAT3600.

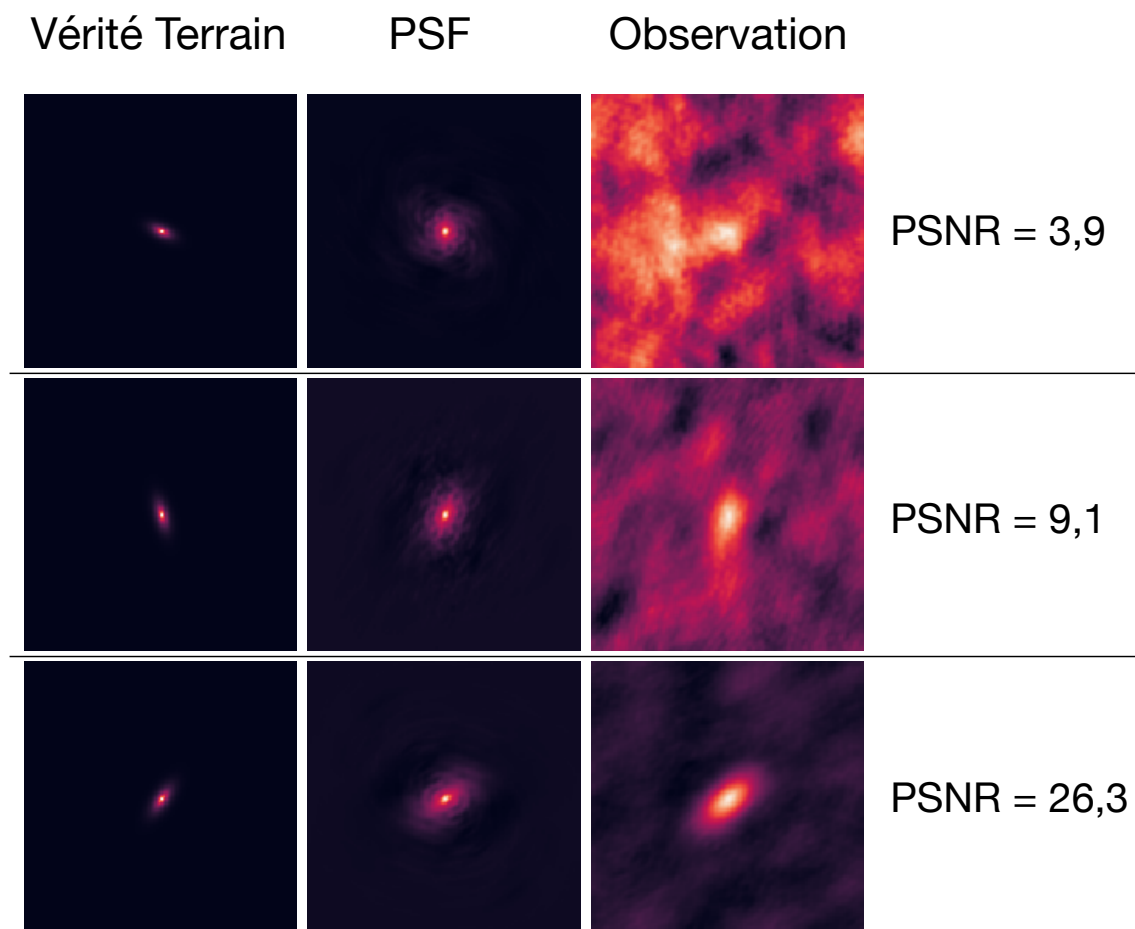


Figure 1.10 – Exemple de trois couples (vérité terrain, FEP, observations) générés avec le jeu de données MeerKAT3600.

1.2.5 Conclusion

Dans cette partie, nous avons présenté l'imagerie par synthèse d'ouverture en interférométrie radio. Nous avons introduit le problème inverse de déconvolution associé et montré qu'il représente des enjeux communs au problème en optique (la différence majeure étant la

présence de corrélation dans le bruit dans le cas radio). Enfin, nous avons développé un jeu de données radio réaliste et adapté à l'apprentissage profond par la diversité et le nombre des objets le constituant. Dans le chapitre 5, nous montrons les résultats de déconvolution sur ce jeu de données, en utilisant des méthodes CLEAN, parcimonieuses et d'apprentissage profond. Mais avant, nous allons présenter en détail le problème inverse de déconvolution dans le chapitre suivant.

Chapitre 2

Problème inverse de déconvolution

Le problème de restauration d'image appartient à la vaste classe des problèmes inverses. Ces derniers sont rencontrés dans différents domaines comme la sismologie et le biomédical à titre d'exemples. La résolution des problèmes inverses est un domaine de recherche actif, ainsi au cours des années, plusieurs approches ont été développées comme l'utilisation de régularisation ou l'apprentissage machine. Ainsi dans ce chapitre, nous allons commencer par aborder les problèmes inverses pour ensuite parler de la régularisation de ceux-ci et des algorithmes proximaux pour les résoudre.

2.1 Problème inverse

Que ce soit en océanographie, chimie, traitement d'images, astrophysique, biomédical ou sismologie, les problèmes inverses sont présents dans ces domaines et sont souvent *mal posés* (nous reviendrons sur cette notion plus tard). Dans cette section, nous allons commencer par définir les termes relatifs aux problèmes inverses puis nous allons nous intéresser au cas de la déconvolution.

2.1.1 Définition d'un problème inverse

Généralement, un problème inverse correspond à la situation où nous souhaitons étudier un signal physique x_v implicite dont nous pouvons mesurer ses effets sur une grandeur y en modélisant l'ensemble des opérations menant de x_v à y par l'opérateur h tel que :

$$y = h(x_v) \quad . \quad (2.1)$$

Le problème est alors dit inverse car il faut inverser l'opérateur h pour reconstruire le signal d'intérêt x_v à partir de l'observation y . Mais il n'existe pas de garantie sur l'inversibilité de h . Par exemple, en imagerie radio, nous avons un problème inverse lié à la diffraction où nous cherchons à déterminer la luminosité d'une galaxie à partir d'échantillons parcellaires de son motif de diffraction.

Nous distinguons deux cas de problèmes inverses :

- *problème inverse non-linéaire* : cette classe de problème inverse possède un opérateur h qui n'est pas linéaire. Souvent dans ces cas, le problème correspond à une équation différentielle, par exemple pour l'étude de la distribution de la température dans un matériau inhomogène [Kern 2002].
- *problème inverse linéaire* : cette classe de problèmes inverses correspondant au cas où l'opérateur h est linéaire. À titre d'exemple nous pouvons évoquer la reconstruction des images des tissus mous, comme le cerveau, avec l'Imagerie par Résonance Magnétique (IRM) [Ramzi et al. 2021a].

Dans notre cas, nous nous intéressons au problème inverse de déconvolution qui est linéaire, pour cela, nous allons considérer uniquement cette catégorie de problème inverse par la suite.

Dans la résolution des problèmes inverses, suivant les propriétés de l'opérateur h , il peut y avoir des complications. Pour pouvoir mieux cerner ces dernières, des critères ont été définis, il s'agit des trois conditions de Hadamard pour qu'un problème soit bien posé [Hadamard 1907] :

1. l'existence d'une solution x pour toute observation y .
2. l'unicité de la solution x .
3. la dépendance continue de la solution x vis-à-vis des observations y .

Ainsi tout problème inverse linéaire qui ne vérifie pas ces conditions est dit *mal posé*. La plupart des problèmes inverses rencontrés dans le monde réel sont mal posés. Effectivement, notons premièrement que le modèle physique menant à l'opérateur h contient potentiellement un biais de modèle, de plus les observations sont souvent contaminées par du bruit et l'injectivité de l'opérateur h n'est pas garantie [Kern 2002]. En particulier, les problèmes inverses de déconvolution que nous détaillons dans la suite, sont généralement des problèmes inverses mal posés.

2.1.2 La déconvolution

La déconvolution est un problème inverse lié à l'imagerie. En effet, dans la chaîne d'acquisition d'une image, deux parties sont généralement modélisées par des opérateurs de convolution pour tenir compte des aberrations qu'elles engendrent : le milieu de propagation du signal comme l'atmosphère dans le cas d'un télescope par exemple; et l'instrument d'acquisition où dans le cas des systèmes optiques la résolution est limitée par la diffraction. Alors pour corriger ces effets, il est nécessaire de déconvoluer les images acquises qui peuvent également être contaminées par du bruit.

Pour commencer, considérons le cas sans bruit. Les produits de convolution successifs sont équivalents à la convolution par le produit des noyaux de convolution. Donc sans perte de généralité, nous allons considérer par la suite uniquement un seul produit de convolution entre l'image vérité-terrain et un noyau de convolution donné. Nous notons l'image acquise avec f dans $\mathbf{L}^2(\mathbb{R}^2)$, A l'opérateur de convolution de $\mathbf{L}^2(\mathbb{R}^2)$ dans $\mathbf{L}^2(\mathbb{R}^2)$ et le noyau de convolution H dans $\mathbf{L}^2(\mathbb{R}^2)$. Dans notre étude, nous utilisons des images à support restreint nous permettant de faire l'hypothèse que le produit de convolution est linéaire et invariante par translation. Alors l'opération de convolution est définie par :

$$\forall x \in \mathbb{R}^2, Af(x) = (H * f)(x) = \int_{\mathbb{R}^2} H(x - z)f(z)dz \quad . \quad (2.2)$$

Le théorème de la convolution donne alors :

$$\forall \nu \in \mathbb{R}^2, \mathcal{F}(H * f)(\nu) = \mathcal{F}(H)(\nu) \cdot \mathcal{F}(f)(\nu) \quad , \quad (2.3)$$

où ν est la fréquence spatiale et \mathcal{F} l'opérateur de la transformée de Fourier. Le produit de convolution dans l'espace direct devient donc dans l'espace de Fourier une simple multiplication. L'espace de Fourier facilite alors l'interprétation et la manipulation des convolutions (en plus d'être l'espace de mesure naturel des données interférométriques). Nous constatons qu'il est possible d'inverser le produit de convolution en divisant par $\mathcal{F}(H)$ à condition que :

$$\forall \nu \in \mathbb{R}^2, \mathcal{F}(H) \neq 0 \quad , \quad (2.4)$$

Si H représente la FEP (Point Spread Function (PSF), en anglais) d'un instrument alors cette condition n'est pas satisfaite. En effet, les instruments utilisés dans l'imagerie ont une bande de fréquence limitée et ne recouvrent pas toutes les fréquences, notamment les hautes fréquences [Bertero et Boccacci 2020]. Dans la figure 2.1, nous donnons un exemple de signal 1D avec un noyau de convolution tel que leurs supports fréquentiels sont disjoints. De ce fait, leur produit de convolution est nul. Il s'agit alors d'un *signal fantôme* également dit *signal invisible*. Dans le cas de la reconstruction d'un signal, si nous rajoutons un signal fantôme à la solution, la somme est elle aussi une solution. Donc dans ce cas, la solution n'est pas unique. Et le problème inverse de déconvolution est mal posé.

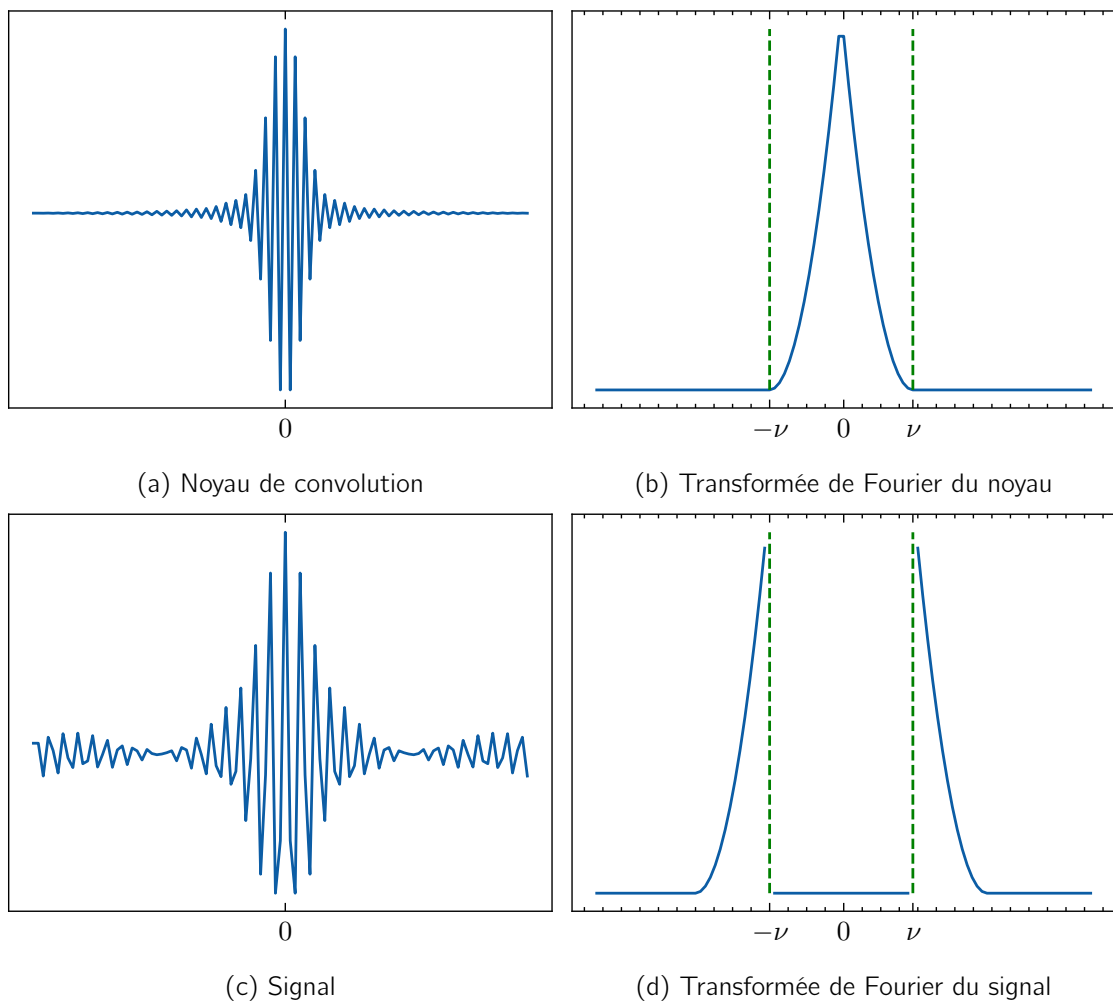


Figure 2.1 – Exemple de noyau et de signal non nuls dont le produit de convolution est nul. Dans la colonne de droite, l'axe des abscisses représente le domaine direct. Dans la colonne de gauche l'axe des abscisses représente le domaine fréquentiel. Dans les quatre figures, l'axe des ordonnées représente les amplitudes.

Maintenant considérons le cas de discret. Considérons x dans \mathbb{R}^n le vecteur codant l'image, k dans \mathbb{R}^n le vecteur codant le noyau de convolution et H la matrice circulante de convolution de taille $n \times n$. Notons que H est l'homologue A dans le cas discret. Il existe différentes manières de considérer les bords de k , cela se fait généralement suivant la nature

du problème et du signal. Nous écrivons alors le problème comme suit :

$$\forall i \in \{1, \dots, n\}, \quad Hx[i] = (h * x)[i] = \sum_{k=0}^n h[i-k]x[k] \quad , \quad (2.5)$$

et les coefficients de H s'écrivent :

$$\forall (i, j) \in \{1, \dots, n\}^2, \quad H[k, i] = h[i-k] \quad . \quad (2.6)$$

Le théorème de la convolution reste valide dans le cas discret avec pour toute fréquence spatiale $\nu = (\nu_1, \nu_2)$:

$$\mathcal{F}(h * x)[\nu] = \mathcal{F}(h)[\nu] \cdot \mathcal{F}(x)[\nu] \quad . \quad (2.7)$$

Avec les mêmes considérations que dans le cas continu, le problème inverse est toujours mal posé. Il est possible de reconstruire x à partir du produit de convolution. En effet, pour toute fréquence ν dans le support de $\mathcal{F}(h)$, nous avons :

$$\mathcal{F}(\hat{x})[\nu] = \frac{\mathcal{F}(h * x)[\nu]}{\mathcal{F}(h)[\nu]} \quad , \quad (2.8)$$

et pour les fréquences ν en dehors du support de $\mathcal{F}(h)$, nous forçons $\mathcal{F}(\hat{x})[\nu]$ à 0. Ainsi le signal reconstruit \hat{x} ne contient pas de signal fantôme et est la solution avec l'énergie minimale.

Maintenant considérons un cas plus réaliste où l'image acquise y est également corrompue par un bruit additif b alors :

$$y = Hx_v + b = h * x_v + b \quad , \quad (2.9)$$

Nous pouvons alors reconstruire la solution \tilde{x} de manière analogue à \hat{x} , ce qui donne pour toute fréquence ν dans le support de $\mathcal{F}(h)$:

$$\mathcal{F}(\tilde{x})[\nu] = \mathcal{F}(\hat{x})[\nu] + \frac{\mathcal{F}(b)[\nu]}{\mathcal{F}(h)[\nu]} \quad . \quad (2.10)$$

Le second terme dans le membre à droite est une erreur de propagation. Cette erreur est majorée par le *conditionnement*, noté a dans \mathbb{R}_+ et défini comme :

$$a = \frac{\lambda_{\max}(H)}{\lambda_{\min}(H)} = \frac{\max(\mathcal{F}(h))}{\min(\mathcal{F}(h))} \quad , \quad (2.11)$$

avec $\lambda_{\max}(H)$ et $\lambda_{\min}(H)$ les valeurs propres maximale et minimale de H respectivement. Ces valeurs correspondent respectivement aux valeurs maximale et minimale de la transformée de Fourier de h car cette dernière correspond au vecteur des valeurs propres de H . En général, dans les problèmes de déconvolution discrets cette valeur est élevée. Effectivement, plus la grille utilisée pour échantillonner le signal est fine et plus la valeur est importante. À titre d'exemple, dans le cas 1D, une FEP gaussienne échantillonnée sur 32 points dans l'intervalle $[-3; 3]$ a un conditionnement de 4, 35.10³ et la même fonction échantillonnée sur 128 points dans l'intervalle $[-6; 6]$ a un conditionnement de 3, 42.10⁷ [Bertero et Boccacci 2020]. Pour le cas 2D, la figure 2.2 montre qu'une perturbation légère dans l'image cause de fortes oscillations dans la reconstruction. Donc le problème de déconvolution discret ne respecte pas la troisième condition de Hadamard, il s'agit d'un problème mal conditionné. Plus en détails, les oscillations sont liées à la taille de la FEP dans l'espace de Fourier : si les valeurs

de la FEP décroissent rapidement à partir de la fréquence centrale alors la division par la FEP devient instable ; notamment, en présence de bruit dans les hautes fréquences.

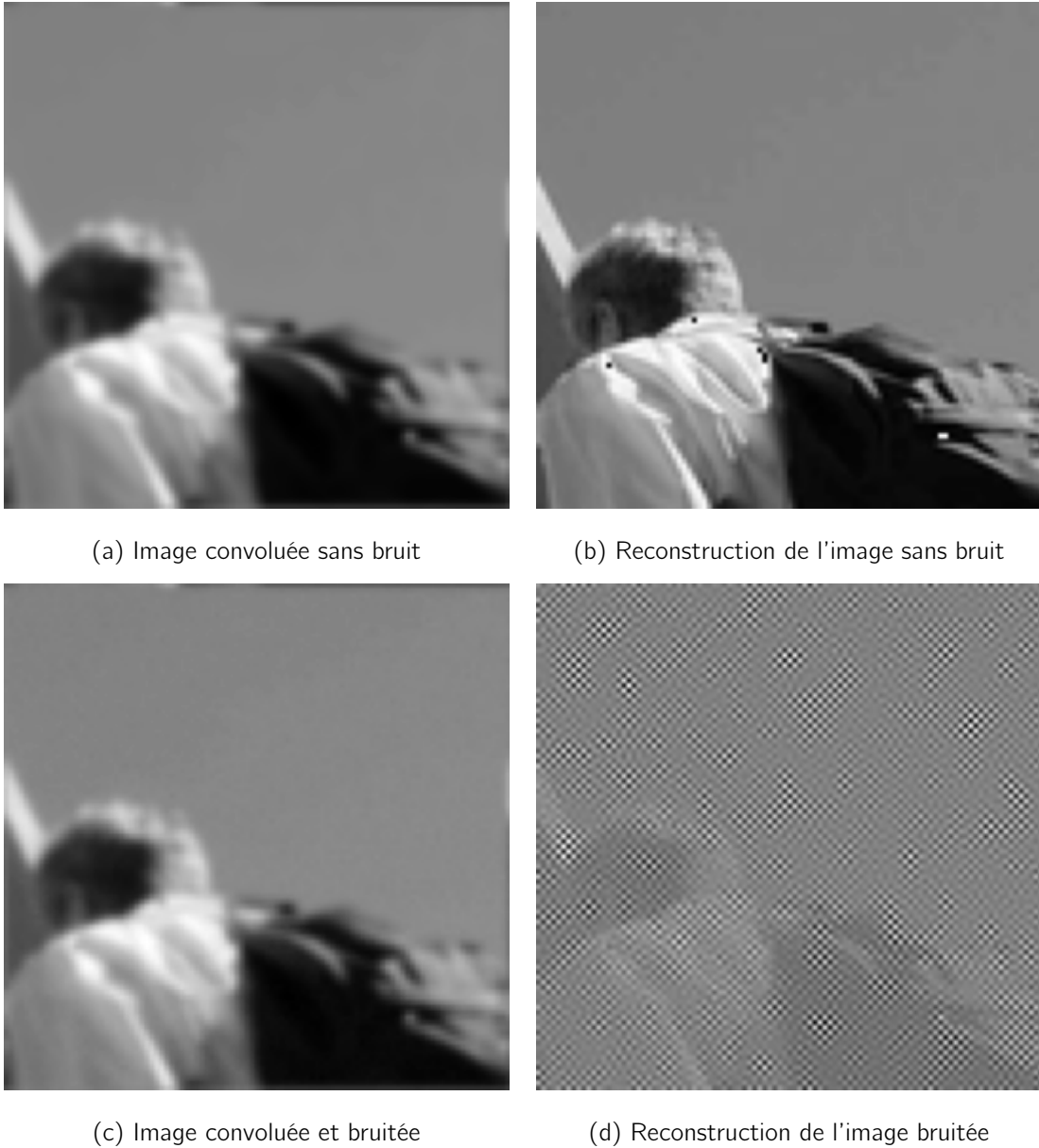


Figure 2.2 – Exemple de déconvolution effectuée en divisant par la transformée de la FEP dans l'espace de Fourier. La taille des images est 101×101 . La FEP utilisée pour générer les images convoluées a un profil gaussien, une taille de 101×101 et est échantillonnée sur l'intervalle $[-55; 55]$. Le conditionnement de la FEP vaut $a = 2, 51 \cdot 10^3$. Source de l'image : SciPy.

2.2 Régularisation des problèmes inverses

Souvent dans les problèmes inverses, le signal à estimer possède des propriétés connues ou supposées a priori. Par exemple, l'énergie du signal peut être majorée ou la représentation du signal peut être parcimonieuse dans un espace donné. Ainsi en tenant compte de ces

informations dans les méthodes de reconstruction, nous pouvons restreindre l'espace des solutions admises pour le problème et obtenir une solution satisfaisante. En d'autres termes, s'agit d'une *régularisation* du problème.

2.2.1 Utilité de la régularisation

Pour mieux appréhender le concept de régularisation, considérons le résidu, noté r , d'une solution x_0 de l'équation 2.9 :

$$r(x_0) = y - Hx_0 \quad (2.12)$$

L'énergie du résidu est donc majorée comme suit :

$$\|r(x_0)\|_2^2 = \|y - Hx_0\|^2 \leq \|y - Hx\|^2 = \|b\|^2, \forall x \in \mathbb{R}^n. \quad (2.13)$$

L'énergie du résidu doit alors être inférieure à celle du bruit donc nous pouvons exclure toute solution telle que l'énergie de son résidu est supérieure à celle du bruit. En cherchant à obtenir l'écart le plus faible par rapport aux données au sens des *moindres carrés*, nous notons x_1 la solution ayant un résidu d'énergie minimale :

$$x_1 = \underset{x}{\operatorname{argmin}} \|y - Hx\|_2^2 \quad (2.14)$$

En notant $I(x) = \|r(x)\|_2^2$, x_1 est par définition la valeur qui annule son gradient :

$$\nabla I(x_1) = 2H^T(Hx_1 - y) = 0, \quad (2.15)$$

$$\iff x_1 = (H^T H)^{-1} H^T y. \quad (2.16)$$

Il est nécessaire que $(H^T H)^{-1}$ soit inversible mais comme argumenté pour la solution \tilde{x} , même si l'inversion est possible, pour un problème mal conditionné, l'opérateur $(H^T H)^{-1}$ est instable en présence de bruit et donne des solutions insatisfaisantes.

Avec une information a priori sur le signal x , nous pouvons ajouter un terme de régularisation. Ce terme s'exprime comme l'image d'une fonction f que la solution x_f doit minimiser :

$$x_f = \underset{x}{\operatorname{argmin}} \|y - Hx\|_2^2 + \lambda f(x), \quad (2.17)$$

avec λ dans \mathbb{R}_+ , une pondération. De ce fait, l'espace des solutions est restreint et les solutions obtenues sont plus stables. La fonction à minimiser pour reconstruire x est appelée fonction de coût. En la notant par I_f , celle-ci s'exprime pour tout x dans \mathbb{R}^n comme :

$$I_f(x) = \|y - Hx\|_2^2 + \lambda f(x). \quad (2.18)$$

La pondération λ permet d'établir un compromis entre le premier terme, appelé *terme d'attache aux données* et le terme de régularisation. Généralement, la valeur de λ est choisie en fonction du conditionnement du problème et du niveau de bruit. En effet, si le niveau de bruit est élevé (ou si l'opérateur H est instable avec le bruit) alors une valeur élevée de λ augmente l'influence du terme de régularisation et stabilise la solution. Par contre, une valeur excessivement élevée de λ détériore la qualité de la reconstruction qui dans ce cas possède un résidu avec une énergie élevée.

Par exemple, il est commun dans les problèmes de reconstruction de régulariser la solution en contraignant l'énergie du signal. Cette régularisation est appelée régression d'arête

(ridge regression, en anglais). Dans ce cas, f est le carré de la norme euclidienne du signal :

$$l_f(x) = \|y - Hx\|_2^2 + \lambda \|x\|_2^2 \quad . \quad (2.19)$$

Soit x_2 la solution qui minimise cette fonction. Alors x_2 s'exprime comme [Bertero et Boccacci 2020] :

$$x_2 = (H^T H + \lambda I_n)^{-1} H^T y \quad . \quad (2.20)$$

Il s'agit d'un filtrage de Tikhonov. Dans la figure 2.3, nous montrons un cas d'utilisation de ce filtre pour 3 valeurs différentes de λ . Pour $\lambda = 1.10^{-3}$, la régularisation a une faible influence et la solution présente des oscillations. Pour $\lambda = 1.10^{-1}$, la solution contient un faible niveau de bruit mais nous pouvons voir que l'image reconstruite possède des contours mieux définis que l'image d'entrée. Finalement, pour $\lambda = 10$, la solution est plus floue que l'entrée ; ceci est dû à la valeur élevée de λ qui lisse la solution. Il existe des méthodes pour estimer la valeur de λ . Dans [Bertero et Boccacci 2020], plusieurs méthodes sont détaillées comme la régularisation avec l'énergie de la solution connue, la régularisation avec l'énergie résiduelle connue, la méthode de Miller, la validation croisée, l'étude de la courbe en L et la méthode interactive.

En régularisant le problème de déconvolution, nous obtenons des reconstructions plus stables comme le montre l'exemple de la régression d'arête. Dans la suite nous allons voir quelques autres régularisations.

2.2.2 Exemples de régularisation

La régularisation d'un problème inverse est basée sur les spécificités du problème et du signal à reconstruire. Il existe des régularisations qui sont souvent utilisées. Par exemple, la régularisation parcimonieuse est développée dans la section 3.1. Nous donnons également d'autres exemples de régularisation.

Régularisation de Tikhonov

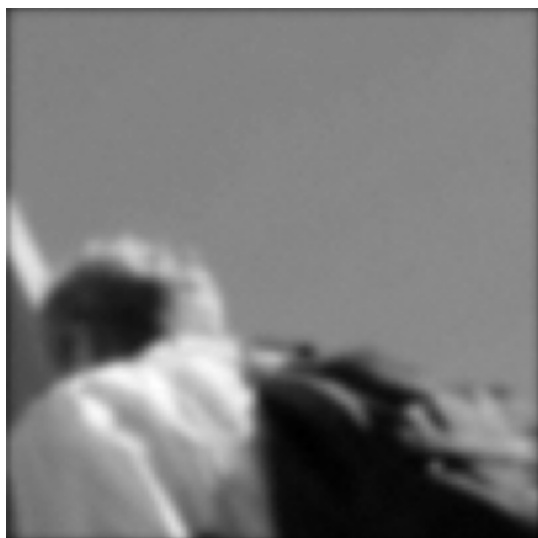
Dans la section précédente, nous avons étudié un cas particulier de la régularisation de Tikhonov. En général, la régularisation de Tikhonov s'écrit à partir de l'équation 2.18 comme suit :

$$l_t(x) = \|y - Hx\|_2^2 + \lambda \|\Gamma x\|_2^2 \quad , \quad (2.21)$$

avec Γ une matrice de taille $n \times n$. Alors le terme de régularisation de Tikhonov $\|\Gamma x\|_2^2$ correspond à l'énergie d'une propriété du signal. Dans l'exemple donné dans l'équation 2.19, Γ est égale à la matrice identité. Dans le cas où Γ correspond à un opérateur de dérivation alors le terme de régularisation favorise une solution avec des contours ayant une énergie minimale. La solution x_t qui minimise la fonction de coût l_t , a une forme close. Comme elle annule le gradient de l_t :

$$\nabla l_t(x_t) = 2 \left[H^T (Hx_t - y) + \lambda \Gamma^T \Gamma x_t \right] = 0 \quad , \quad (2.22)$$

$$\iff x_t = (H^T H + \lambda \Gamma^T \Gamma)^{-1} H^T y \quad (2.23)$$



(a) Image convoluée et bruitée

(b) Reconstruction pour $\lambda = 1.10^{-3}$ (c) Reconstruction pour $\lambda = 1.10^{-1}$ (d) Reconstruction pour $\lambda = 10$

Figure 2.3 – Exemple de déconvolution effectuée avec un filtre de Tikhonov. La taille des images est 101×101 . La FEP utilisée pour générer les images convoluées a un profil gaussien, une taille de 101×101 et est échantillonnée sur l'intervalle $[-50; 50]$. Dans les images reconstruites, λ correspond à la pondération de la régularisation. Source de l'image : SciPy.

Variation totale

En 1D, la variation totale d'une fonction f permet de mesurer sa longueur d'arc sur le segment $[a; b]$. En 2D, si nous considérons une image comme une surface alors la variation totale mesure la déformation de cette surface. Le bruit dans une image augmente la déformation de la surface. À cet effet, l'utilisation de la variation totale comme régularisation permet de pénaliser les solutions contenant du bruit. Une des variantes répandues de la

fonction de coût s'écrit alors comme [Chambolle et al. 2009] :

$$f(x) = \|y - x\|_2^2 + \lambda \|\nabla x\|_{2,1} \quad , \quad (2.24)$$

avec $\|\nabla x\|_{2,1} = \sum_k \sqrt{(\nabla_i x[k])^2 + (\nabla_j x[k])^2}$ où $\nabla_i x$ est le gradient de x suivant les lignes et $\nabla_j x$ suivant les colonnes. Nous donnons un exemple à la figure 2.4. Pour une valeur élevée de λ , la reconstruction est plus lisse et tend vers une fonction constante par morceaux. Notons que le terme de régularisation n'est pas différentiable partout. C'est également le cas de l'exemple suivant.



(a) Image convoluée et bruitée



(b) Image bruitée



(c) Reconstruction pour $\lambda = 10$



(d) Reconstruction pour $\lambda = 40$

Figure 2.4 – Exemple de débruitage avec une régularisation de variation totale. La taille des images est 101×101 . Un bruit blanc gaussien est ajouté tel que l'image 2.4ba un RSB de 12,72. Dans les images reconstruites, λ correspond à la pondération de la régularisation. Source de l'image : SciPy.

Projection sur un ensemble

Une autre façon commune de régulariser est la projection sur un ensemble. En effet, l'information a priori que le signal x à estimer appartient un ensemble donné \mathcal{C} est formellement notée à l'aide de l'indicateur de Moreau $\iota_{\mathcal{C}}$ avec :

$$\forall x \in \mathbb{R}^n, \quad \iota_{\mathcal{C}}(x) = \begin{cases} 0 & \text{si } x \in \mathcal{C} \\ +\infty & \text{sinon.} \end{cases}, \quad (2.25)$$

Le terme de régularisation est ajouté à la fonction de coût sans pondération dans ce cas et f s'écrit :

$$f(x) = \|y - Hx\|_2^2 + \iota_{\mathcal{C}}(x), \quad (2.26)$$

Par exemple, si les images représentent des intensités donc les valeurs sont positives. Alors nous utilisons la fonction indicatrice de l'ensemble des images à coefficients positives, notée ι_+ , telle que :

$$\forall x \in \mathbb{R}^n, \quad \iota_+(x) = \begin{cases} 0 & \text{si } \forall k \in \{1, \dots, n\}, x[k] \geq 0 \\ +\infty & \text{sinon.} \end{cases}, \quad (2.27)$$

Dans une fonction de coût, il est possible d'utiliser plusieurs termes de régularisation à la fois. Dans le cas général, les fonctions de coût ne sont pas différentiables. Pour les minimiser, il faut alors utiliser des outils avancés comme les algorithmes proximaux qui sont introduits dans la section suivante.

2.3 Méthodes proximales

Les algorithmes proximaux sont des algorithmes itératifs permettant de minimiser des fonctions qui ne sont pas différentiables. Ils peuvent être vus comme l'extension des algorithmes de descente de gradient aux fonctions non-différentiables en utilisant les opérateurs proximaux. Dans cette partie nous allons commencer par aborder les opérateurs proximaux pour ensuite parler des algorithmes proximaux et donner quelques exemples.

2.3.1 Opérateur proximal

Les opérateurs proximaux sont un outil efficace pour manipuler des fonctions non-différentiables. Ils ont été introduits par Moreau [Moreau 1965] comme la généralisation des opérateurs de projection.

Avant de définir l'opérateur proximal d'une fonction, commençons par donner la définition d'une fonction proximable. Soit \mathcal{H} un espace de Hilbert et f une fonction de \mathcal{H} dans $\mathbb{R} \cup \{-\infty, +\infty\}$. La fonction f est dite proximable si et seulement si f est :

- *convexe* : $\forall (x, y) \in \mathcal{H}$, si $f(x) < +\infty$ et $f(y) < +\infty$ alors $\forall \lambda \in [0, 1]$,

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y). \quad (2.28)$$

- *propre* : $\forall x \in \mathcal{H}, f(x) > -\infty$ et $\exists y \in \mathcal{H}, f(y) < +\infty$.
- *semi-continue inférieurement* : tous les sous-ensembles $(\{x \in \mathcal{H}, f(x) \geq a\})_{a \in \mathbb{R}}$ sont fermés.

En notant, $\|\cdot\|$ la norme canonique de \mathcal{H} , l'opérateur proximal d'une fonction f proximable est unique et est défini comme :

$$\forall x \in \mathcal{H}, \quad \text{prox}_f(x) = \underset{y \in \mathcal{H}}{\text{argmin}} \frac{1}{2} \|x - y\|^2 + f(y) \quad . \quad (2.29)$$

En considérant les deux termes minimisés par l'opérateur proximal en un point x dans \mathcal{H} , nous pouvons l'interpréter comme l'antécédent du minimum de f au voisinage de x .

Pour montrer que les opérateurs proximaux sont une généralisation des opérateurs de projection, considérons un sous-ensemble \mathcal{C} de \mathcal{H} et f la fonction indicatrice de Moreau du sous-ensemble \mathcal{C} telle que définie dans l'équation 2.25, alors l'opérateur proximal de f s'écrit :

$$\forall x \in \mathcal{H}, \quad \text{prox}_f(x) = \underset{y \in \mathcal{H}}{\text{argmin}} \frac{1}{2} \|x - y\|^2 + \iota_{\mathcal{C}}(y) = \underset{y \in \mathcal{C}}{\text{argmin}} \frac{1}{2} \|x - y\|^2 \quad . \quad (2.30)$$

Par définition, il s'agit de la projection orthogonale sur le sous-ensemble \mathcal{C} .

Les propriétés intéressantes de l'opérateur proximal sont données dans [Combettes et Pesquet 2011]. Nous en citons les suivantes :

1. la translation

$$\forall (x, y) \in \mathcal{H}, \quad \text{prox}_{f(\cdot - y)}(x) = y + \text{prox}_f(x - y) \quad , \quad (2.31)$$

2. la remise à l'échelle

$$\forall x \in \mathcal{H}, \forall \rho \in]-\infty, +\infty[, \quad \text{prox}_{f(\rho \cdot)} = \frac{1}{\rho} \text{prox}_{\rho^2 f}(\rho x) \quad , \quad (2.32)$$

3. la séparabilité : soit $\{f_i\}_{1 \leq i \leq n}$ une famille de fonctions de \mathcal{H} dans $\mathbb{R} \cup +\infty$, propres convexes et semi-continues inférieurement et f la fonction de \mathcal{H}^n dans $\mathbb{R} \cup +\infty$ définie par :

$$\forall a = (a_i)_{1 \leq i \leq n} \in \mathcal{H}^n, \quad f(a_1, \dots, a_n) = \sum_{i=1}^n f_i(a_i) \quad (2.33)$$

alors

$$\text{prox}_f(a) = (\text{prox}_{f_i}(a_i))_{1 \leq i \leq n} \quad . \quad (2.34)$$

Les opérateurs proximaux permettent le développement d'algorithmes proximaux qui sont utilisés pour la résolution de problèmes inverses. Dans la suite nous aborderons les algorithmes proximaux.

2.3.2 Algorithmes proximaux

Les algorithmes proximaux permettent de minimiser les fonctions de coût qui ne sont pas différentiables. Plusieurs termes de régularisation sont non-différentiables, par exemple la fonction indicatrice de Moreau et la norme ℓ_1 . Dans la thèse, nous intéressons aux deux algorithmes suivants :

Algorithme Forward-Backward Splitting

L'algorithme Forward-Backward Splitting (FBS) [Combettes et Wajs 2005] est un algorithme proximal simple. Il est utilisé pour résoudre un problème de la forme suivante :

$$x = \operatorname{argmin}_{z \in \mathcal{H}} f(z) + g(z) \quad . \quad (2.35)$$

avec f une fonction propre, convexe et différentiable et g une fonction proximable. Souvent f correspond au terme d'attache aux données et g au terme de régularisation comme dans l'équation 2.26.

L'algorithme est itératif et alterne entre deux étapes :

- *pas forward* : également dit pas explicite, il s'agit de la descente de gradient liée à f ,
- *pas backward* : également dit pas implicite, il s'agit de l'application de l'opérateur proximal de g .

Comme illustré dans la figure 2.5, le pas forward rapproche l'itéré du minimum global de f et le pas backward l'éloigne de celle-ci. L'algorithme est donné par : La convergence de cet

Algorithme 2.1 Forward Backward Splitting (FBS)

tâche : estimer x qui minimise $f + g$.

paramètres : $\beta > 0$, A critère d'arrêt

initialisation : $x^{(0)}$, $i \leftarrow 0$

tant que non(A) **faire**

$$x^{(i+1)} \leftarrow \operatorname{prox}_{\beta g} \left(x^{(i)} - \beta \nabla f(x^{(i)}) \right)$$

$$i \leftarrow i + 1$$

fin tant que

renvoyer : $x^{(i)}$

algorithme est garantie pour $\beta < 2/\alpha$ avec α la constante de Lipschitz du gradient de f , noté ∇f . Il est possible d'accélérer la vitesse de convergence de l'algorithme en modifiant l'expression de l'itéré [Beck et Teboulle 2009 ; Nesterov 2013]. D'ailleurs, des améliorations de l'algorithme FBS ont été apportées comme le FBS généralisé [Raguét, Fadili et Peyré 2013] qui permet de minimiser une fonction de coût de la forme $f + \sum_{i=1}^n g_i$ avec $\{g_i\}_{1 \leq i \leq n}$ des fonctions proximables. Un autre exemple est celui de l'algorithme des directions alternées détaillé dans la suite. Citons également les algorithmes proximaux de Douglas-Rachford [Douglas et Rachford 1956] et l'algorithme de Chambolle-Pock [Chambolle et Pock 2010].

Algorithme des directions alternées

L'algorithme des directions alternées (Alternating Direction Method of Multipliers (ADMM), en anglais) [Boyd, Parikh et Chu 2011] a pour but de minimiser une fonction de coût de la forme $f + g$ avec f et g des fonctions proximables de \mathbb{R}^n dans \mathbb{R} . Le principe de cette méthode est de séparer la variable à estimer en deux variables et les contraindre à être égales. Ce qui donne la formulation suivante :

$$\begin{cases} \min_{x,v} f(x) + \lambda g(v) \\ \text{t.q. } x = v \end{cases} \quad (2.36)$$

avec λ dans \mathbb{R}_+ une pondération.

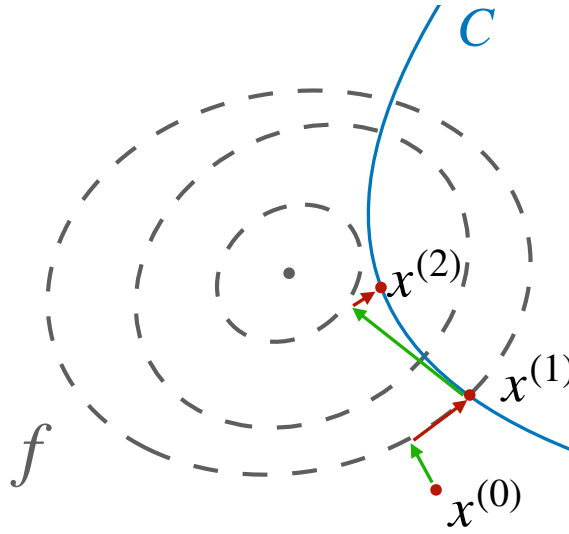


Figure 2.5 – Représentation des deux premières itérations de l'algorithme FBS. La fonction à minimiser est $f + \iota_C$. Les lignes grises représentent les lignes de niveau de f et le point gris son minimum global. La ligne bleue correspond à la frontière de l'ensemble \mathcal{C} . Les flèches vertes représentent les pas explicites et les flèches rouges représentent les pas implicites.

Le lagrangien augmenté correspondant s'écrit [Chan, Wang et Elgendy 2016] :

$$\mathcal{L}(x, v, u) = f(x) + \lambda g(v) + u^\top (x - v) + \frac{\rho}{2} \|x - v\|_2^2 \quad (2.37)$$

La résolution du problème se fait itérativement, en mettant à jour une des trois variables x , v et u et en fixant les deux autres :

$$\begin{cases} x^{(k+1)} = \underset{x}{\operatorname{argmin}} f(x) + \frac{\rho}{2} \|x - \tilde{x}^{(k)}\|_2^2, \\ v^{(k+1)} = \underset{v}{\operatorname{argmin}} \lambda g(v) + \frac{\rho}{2} \|v - \tilde{v}^{(k)}\|_2^2, \\ \bar{u}^{(k+1)} = \bar{u} + (x^{(k+1)} - v^{(k+1)}) \end{cases}, \quad (2.38)$$

avec $\bar{u} = u^{(k)}/\rho$; $\tilde{x}^{(k)} = v^{(k)} - \bar{u}^{(k)}$ et $\tilde{v}^{(k)} = x^{(k+1)} + \bar{u}^{(k)}$. Une variante intéressante de l'ADMM est le *Plug-and-Play ADMM* [Venkatakrishnan, Bouman et Wohlberg 2013] qui montre que l'étape de mise à jour de v est équivalente à l'utilisation d'un débruiteur gaussien, noté \mathcal{D}_σ où σ est l'écart-type du bruit :

$$v^{(k+1)} = \mathcal{D}_\sigma(\tilde{v}^{(k)}), \quad (2.39)$$

avec $\sigma = \sqrt{\lambda/\rho}$. L'algorithme s'écrit comme suit :

Il est possible d'accélérer l'algorithme en augmentant la valeur de ρ à chaque itération, il s'agit d'un schéma de continuation [Ng 2002].

Dans le cas où f est un terme d'attache aux données alors il est possible d'utiliser un débruiteur quelconque pour régulariser le problème sans explicitement définir g . Cette propriété est utilisée dans une méthode de reconstruction dans la partie 5.3.2 traitant l'apprentissage profond.

Algorithme 2.2 Plug-and-Play ADMM**tâche** : estimer x qui minimise $f + g$.**paramètres** : ρ, λ, A critère d'arrêt**initialisation** : $x^{(0)}, k \leftarrow 0$ **tant que** non(A) **faire**

$$x^{(k+1)} \leftarrow \underset{x}{\operatorname{argmin}} f(x) + (\rho/2) \|x - (v^{(k)} - u^{(k)})\|_2^2$$

$$v^{(k+1)} \leftarrow \mathcal{D}_\sigma(x^{(k+1)} + u^{(k)})$$

$$u^{(k+1)} \leftarrow u^{(k)} + (x^{(k+1)} - v^{(k+1)})$$

$$k \leftarrow k + 1$$

fin tant que**renvoyer** : $x^{(k)}$

2.4 Conclusion

Dans ce chapitre, nous avons commencé par présenter le formalisme des problèmes inverses pour nous intéresser ensuite au cas spécifique de la déconvolution. Ce dernier est un problème mal posé et mal conditionné. En effet, plus le support de la FEP est grand et plus le problème est mal conditionné. De ce fait, le problème en imagerie radio est davantage mal conditionné que celui en imagerie optique. Ensuite nous avons introduit le concept de régularisation en montrant différents exemples. Enfin, nous avons présenté les méthodes proximales qui permettent de résoudre les problèmes régularisés dans le cas où certaines parties de la fonction de coût n'étaient plus différentiables. Dans le chapitre 4, nous introduisons la contrainte de forme que j'ai développée pour l'utiliser en tant que terme de régularisation avec la parcimonie dans l'algorithme FBS. La parcimonie et la transformée en ondelettes sont souvent utilisées dans la régularisation des problèmes inverses et sont présentées dans le chapitre suivant.

Chapitre 3

Parcimonie et ondelettes

3.1 Parcimonie

La parcimonie caractérise un signal dont la représentation est donnée par une majorité de coefficients nuls. La propriété sous-jacente à un signal parcimonieux est le fait que ses degrés de liberté intrinsèques sont de nombre réduit. Ainsi en faisant l'hypothèse que le signal à reconstruire est parcimonieux, il est possible de résoudre des problèmes dans des domaines variés comme dans l'astrophysique, l'imagerie biomédicale, la sismologie et l'océanographie.

La parcimonie est alors utilisée en tant qu'a priori pour résoudre des problèmes inverses linéaires, c'est-à-dire les problèmes inverses qui sont modélisés par un opérateur linéaire. Dans cette section, nous commencerons par introduire la parcimonie en général pour ensuite nous attarder sur la représentation parcimonieuse des signaux.

3.1.1 Aperçu sur la parcimonie

Le principe de parcimonie se retrouve en philosophie et est également appelé rasoir d'Ockham, il stipule que si deux modèles permettent d'expliquer un phénomène de manière aussi satisfaisante l'une que l'autre alors il faut opter pour celui contenant le moins d'hypothèses, donc le plus simple. De même, en traitement du signal, la parcimonie est un outil qui pour un problème donné permet de choisir la solution contenant le moins d'éléments possible donc la plus simple.

Pour donner une définition formelle d'un signal parcimonieux, considérons x dans \mathbb{R}^n alors nous disons que x est strictement k -parcimonieux si et seulement si x possède exactement k coefficients non-nuls. Dans ce cas, un signal k -parcimonieux possède k degrés de liberté. Pour mesurer la parcimonie d'un signal, il suffit alors de compter le nombre de ses composantes actives, c'est-à-dire non nulles. Pour ce faire, la pseudo-norme ℓ_0 , notée $\|\cdot\|_0$ est utilisée. Elle est définie comme le cardinal du support de ce signal. Considérons un signal $x \in \mathbb{R}^n$, nous avons alors :

$$\text{supp}(x) = \{i \in \{1, \dots, n\} \mid x[i] \neq 0\} \quad , \quad (3.1)$$

et la pseudo-norme ℓ_0 de x s'écrit comme :

$$\|x\|_0 = \text{card}(\text{supp}(x)) \quad , \quad (3.2)$$

avec $\text{card}(U)$ le cardinal de l'ensemble U . Insistons sur le fait que ℓ_0 n'est pas une norme car elle n'est pas homogène, c'est-à-dire que pour un réel λ , $\|\lambda x\|_0 = \|x\|_0$ n'est pas égal à $|\lambda| \cdot \|x\|_0$, en général. Mais par abus de langage, la pseudo-norme ℓ_0 est également appelée norme ℓ_0 , par abus de langage.

Mais généralement, dans le cas des signaux naturels, la plupart des coefficients du signal n'ont pas de valeur parfaitement nulle donc la valeur de leur norme ℓ_0 n'est pas faible. Ce

qui fait que la k -parcimonie stricte n'est pas adaptée à ces signaux. Néanmoins, parmi des signaux naturels, il existe des signaux dont un nombre restreint de coefficients est d'amplitude élevée et les autres coefficients sont d'amplitudes assez faibles, voire quasi-nulles. Ces signaux sont dits *compressibles* ou *quasi-parcimonieux*. Nous pouvons alors approximer ces signaux par des signaux k -parcimonieux en forçant les valeurs faibles à zéro.

Mathématiquement, cette opération se fait à l'aide de l'opérateur de *seuillage dur* (Hard Thresholding), noté HT_λ avec λ un réel positif jouant le rôle de seuil. Donc tout coefficient ayant une valeur inférieure à λ est mis à zéro :

$$\forall i \in \{1, \dots, n\} \quad \text{HT}_\lambda(x)[i] = \begin{cases} x[i] & \text{si } |x[i]| \geq \lambda \\ 0 & \text{sinon.} \end{cases} \quad (3.3)$$

En choisissant une valeur de λ qui garde les k éléments de x avec l'amplitude la plus élevée, l'opération de seuillage dur donne alors une approximation de x qui est non linéaire et k -parcimonieuse. Nous notons \tilde{x} cette approximation avec :

$$\tilde{x} = \text{HT}_\lambda(x) \quad (3.4)$$

Pour étudier quantitativement la qualité de l'approximation \tilde{x} de x , une métrique d est utilisée pour évaluer la distance entre x et \tilde{x} . Suivant les applications, il existe différentes métriques pour ce faire. Dans le cas général, la norme euclidienne est utilisée, celle-ci correspond à la racine carrée de l'énergie de l'erreur d'approximation. La dégradation entre le signal d'origine et son approximation parcimonieuse est alors mesurée par $d(\tilde{x}, x)$.

Désormais, nous avons les outils nécessaires pour introduire la classe des signaux *compressibles*. Un signal appartient à cette classe si et seulement si les amplitudes de ces coefficients rangés par ordre décroissant, décroissent de manière polynomiale. Considérons i l'image de la permutation qui réarrange les indices des éléments d'un signal suivant leurs amplitudes par ordre décroissant, c'est-à-dire :

$$\forall m \in \{1, \dots, n-1\}, \quad |x[i(m)]| \geq |x[i(m+1)]| \quad . \quad (3.5)$$

Alors un signal est compressible si et seulement si il existe deux constantes $c \in \mathbb{R}_+$ et $s > 1/2$ tel que :

$$\forall k \in \{1, \dots, n\}, \quad |x[i(k)]| \leq ck^{-s} \quad . \quad (3.6)$$

Notons donc que s est liée à la décroissance des coefficients de manière polynomiale, c'est-à-dire que plus la valeur de s est élevée et plus la décroissance des coefficients est rapide. Maintenant, prenons le carré de la distance entre les deux signaux qui vaut par construction :

$$d^2(\tilde{x}, x) = \sum_{j=k+1}^n |x[i(j)]|^2 \leq c^2 \sum_{j=k+1}^n j^{-2s} \quad . \quad (3.7)$$

La somme à droite de l'inégalité peut être vue comme une somme partielle de la série de Riemann commençant à $k+1$. Alors d'après le théorème de comparaison série-intégrale pour les fonctions monotones, nous avons :

$$\sum_{j=k+1}^n j^{-2s} \leq \int_k^n t^{-2s} dt = \left[\frac{t^{-2s+1}}{1-2s} \right]_k^n \leq \frac{k^{-2s+1}}{2s-1} \quad (3.8)$$

Nous pouvons alors majorer l'erreur d'approximation par :

$$d^2(\tilde{x}, x) \leq c^2 \frac{k^{-2s+1}}{2s-1} \quad (3.9)$$

Nous déduisons alors que l'erreur d'approximation a la même tendance que la décroissance des coefficients, par exemple si les coefficients décroissent rapidement alors l'erreur est petite. Par conséquent, les résultats présentés dans ce qui suit pour des signaux rigoureusement parcimonieux restent de bonnes approximations pour les signaux compressibles.

Par contre, il n'y a pas de garantie que les signaux naturels soient compressibles ou parcimonieux quand ils sont représentés dans l'espace où ils sont mesurés. Par exemple, si nous considérons la fonction sinus dans l'espace temporel, il existe une infinité de points pour lesquels le sinus ne s'annule pas. De plus, cette fonction n'est pas décroissante donc elle n'est ni parcimonieuse ni compressible. Mais en prenant sa représentation dans le domaine fréquentiel, nous avons uniquement deux points non nuls correspondant à sa fréquence et son opposé comme le montre la figure 3.1. Alors dans l'espace de Fourier, c'est un signal 2-parcimonieux.

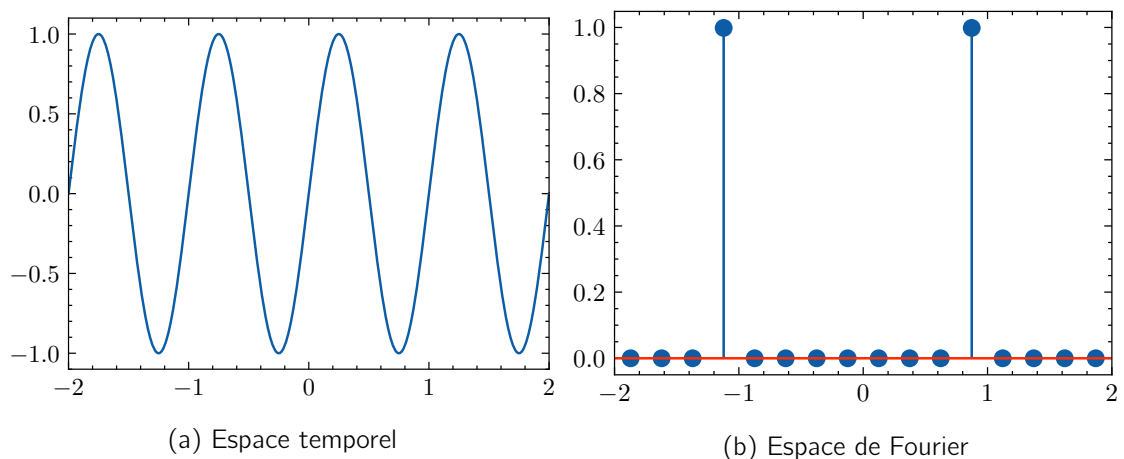


Figure 3.1 – Représentation d'un sinus avec une fréquence de 1 Hz dans l'espace temporel et de son spectre dans l'espace de Fourier. La représentation dans l'espace de Fourier possède uniquement deux coefficients non nuls donc elle est strictement 2-parcimonieuse.

Nous déduisons donc que la parcimonie n'est pas une propriété intrinsèque du signal mais qu'elle est également dépendante de l'espace dans lequel le signal est représenté. Alors pour pouvoir utiliser les méthodes parcimonieuses, il est important tout d'abord de trouver un espace qui optimise la parcimonie des signaux étudiés, ce qui fait de la recherche d'espaces adéquats pour assurer la parcimonie de certaines classes de signaux un domaine actif.

3.1.2 Représentation parcimonieuse des signaux

Pour tirer profit de la parcimonie dans les procédés de reconstruction, il est important de trouver un espace dans lequel nous savons que le signal considéré est parcimonieux. Ainsi suivi les différentes classes de signaux et d'applications, certains espaces sont préférés à d'autres. Par exemple pour les signaux périodiques stationnaires, l'espace de Fourier leur donne une représentation parcimonieuse.

Pour définir les termes utilisés dans le domaine de la parcimonie, un *dictionnaire* est une famille génératrice de l'espace considéré. Les éléments de ce dictionnaire sont appelés *atomes*. Ainsi, considérons un signal x dans \mathbb{R}^n et un dictionnaire $\phi \in \mathbb{R}^{n \times p}$. Comme le dictionnaire est une famille génératrice alors $p \geq n$. Alors x s'écrit comme la combinaison linéaire des p atomes $\phi_i \in \mathbb{R}^n$ tel que :

$$x = \phi\alpha = \sum_{i=1}^p \phi[i]\alpha[i] \quad , \quad (3.10)$$

avec $\alpha \in \mathbb{R}^p$ sont les coefficients de x dans le dictionnaire ϕ . Deux types de dictionnaires existent :

- les *dictionnaires redondants*, ces dictionnaires ont $p > n$ atomes. Cela implique que la décomposition du signal n'est pas unique puisqu'il existe des atomes qui sont linéairement dépendants.
- les *dictionnaires non-redondants*, dans ce cas, il y a n atomes dans un dictionnaire, il s'agit donc d'une base et la décomposition du signal est unique. Ces dictionnaires sont également dits sur-complets.

En utilisant des dictionnaires non-redondants, le problème d'optimisation est plus simple à manipuler. Par contre, un dictionnaire redondant est moins contraignant à développer et offre ainsi plus de possibilité pour trouver une représentation parcimonieuse qui est plus adaptée aux données considérées.

Pour revenir au cas de l'espace de Fourier, le dictionnaire correspondant à la Transformée de Fourier Discrète (Discrete Fourier Transform (DFT), *en anglais*) est non-redondant et nous avons :

$$x = F\hat{x} \quad , \quad (3.11)$$

où \hat{x} est la DFT du signal x et F est dite la matrice de Fourier telle que :

$$F[k, l] = \frac{1}{\sqrt{n}} e^{i2\pi lk/n} \quad . \quad (3.12)$$

D'ailleurs, une implémentation répandue de la DFT est la Transformée de Fourier Rapide (FFT, *en anglais*) qui, comme son nom l'indique, a le mérite d'être rapide, ayant une complexité en $\mathcal{O}(n \log n)$. Il existe aussi d'autres transformations qui sont similaires comme la Transformée en Cosinus Discrète (Discrete Cosine Transform (DCT), *en anglais*) telle que :

$$\alpha[k] = \sum_{i=0}^{n-1} x[n] \cos\left(\frac{\pi}{n} \left(i + \frac{1}{2}\right) k\right) \quad (3.13)$$

et sa transformée inverse donnée par :

$$x[k] = \frac{2}{n} \left(\frac{1}{2} \alpha[0] + \sum_{i=1}^{n-1} \alpha[i] \cos\left(\frac{\pi}{n} \left(k + \frac{1}{2}\right) n\right) \right) \quad . \quad (3.14)$$

La version 2D de la DCT est efficace pour la représentation des textures. Elle est obtenue à partir de la DCT 1D en faisant un produit séparable. Concernant l'implémentation, comme dans le cas de la DFT, elle peut se faire rapidement à l'aide de l'algorithme de la transformée en cosinus rapide avec une complexité en $\mathcal{O}(n \log n)$. Avec ces avantages, la DCT est une transformée utile pour la compression d'images.

Par contre, les atomes de ces dictionnaires contiennent uniquement l'information fréquentielle du signal, sans avoir l'information correspondant à l'espace de départ. Ces atomes

sont dits *non-locaux*. De ce fait, cette représentation n'est pas adaptée pour les signaux non-périodiques avec des détails différents localisés dans des régions différentes. Pour contourner ce problème, un fenêtrage du signal est effectué tel que la fenêtre est suffisamment petite pour supposer que le signal est localement quasi-stationnaire. Cette approche correspond à la Transformée de Fourier à Court-Terme (Short Term Fourier Transform (STFT), en anglais) et également de la Transformée en Cosinus Discrète par bloc. Notons qu'elle est utilisée dans le format JPEG [Wallace 1991].

Une méthode plus élaborée pour contourner ces limitations est le recours aux ondelettes. Les ondelettes sont conçues pour donner une représentation parcimonieuse des signaux lisses par morceaux. Et la représentation qu'elles offrent contient à la fois l'information de l'espace de départ du signal et également l'information fréquentielle. Un autre avantage des ondelettes est leur implémentation encore plus rapide que celles de la DFT et la DCT. En effet, la complexité d'un algorithme optimal est en $\mathcal{O}(n)$. De plus, comme les ondelettes représentent les signaux naturels parcimonieusement, elles ont été utilisées dans le format JPEG 2000 [Skodras, Christopoulos et Ebrahimi 2001].

Depuis l'apparition des premières ondelettes, au cours du temps, d'autres ont été conçues pour servir dans des problèmes spécifiques ou représenter certains types de signal en particulier. Par exemple, la représentation en ondelettes des anisotropies dans les images n'est pas efficace puisque les ondelettes ne captent pas adéquatement la directivité. C'est ainsi que les ridgelets [Candès et Donoho 1999] et des curvelets [Candès et Donoho 2000 ; Starck, Candès et Donoho 2002] ont été développées pour donner une représentation efficace des lignes et des contours, respectivement.

Finalement, au lieu de construire les ondelettes en recherchant des fonctions satisfaisant des propriétés utiles pour le modèle du signal considéré, il est possible de se baser sur les données afin d'apprendre une représentation parcimonieuse de celles-ci [Aharon, Elad et Bruckstein 2006]. Il s'agit de l'Apprentissage de Dictionnaire (AL) (Dictionary Learning, en anglais). Cette approche consiste à trouver le dictionnaire qui offre la représentation la plus parcimonieuse des signaux de l'ensemble d'entraînement, en minimisant, une nouvelle fois, un problème d'optimisation. De ce fait, ces dictionnaires donnent globalement de meilleures performances que les ondelettes dans les problèmes auxquels ils sont dédiés. Dans [Beckouche, Starck et Fadili 2013], nous avons un exemple de AL pour des tâches de débruitage d'image en astrophysique. Actuellement, avec l'émergence de l'apprentissage machine, il est possible d'apprendre des représentations qui couvrent des domaines d'applications et des classes de signaux plus généraux comme avec les learnlets [Ramzi et al. 2021b].

Dans la section suivante, nous allons nous pencher sur la théorie des ondelettes. Nous allons commencer par détailler les fondements de celle-ci pour aboutir à la transformée en starlets qui est adaptée à une représentation parcimonieuse des images de galaxies.

3.2 Transformée en ondelettes

La transformée de Fourier permet de représenter le signal dans le domaine fréquentiel mais en perdant l'information temporelle. Pour s'affranchir de cela, l'analyse en ondelettes a été développée pour étudier les signaux stationnaires en donnant une représentation contenant aussi bien l'information temporelle et fréquentielle. Ainsi il est possible de cerner les motifs d'un signal à différentes échelles et positions suivant ses coefficients en ondelettes.

Les ondelettes permettent de représenter les signaux lisses par morceaux avec parcimonie. Souvent les signaux naturels sont modélisés comme lisses par morceaux. Alors pour utiliser la parcimonie dans la résolution de problèmes inverses, nous pouvons avoir recours

aux ondelettes qui vont être la base de la régularisation parcimonieuse. Par contre la solution finale pour le problème est liée à l'ondelette choisie donc trouver les ondelettes appropriées est un enjeu d'importance.

Ainsi nous dans ce chapitre, nous commencerons tout d'abord par une présentation générale des ondelettes en détaillant les ondelettes continues. Après nous passerons aux ondelettes orthogonales et bi-orthogonales pour finalement introduire les starlets qui nous seront utiles dans les méthodes de reconstructions parcimonieuses développées dans le cadre de notre travail.

3.3 Ondelettes continues

Pour unifier les travaux effectués sur les ondelettes, Morlet et Grossman ont défini, en 1984, la Transformée en Ondelettes Continue (Continuous Wavelet Transform (CWT), en anglais) [Grossmann et Morlet 1984] pour représenter les fonctions de l'espace $\mathbf{L}^2(\mathbb{R})$. Rappelons qu'il s'agit de l'espace des fonctions de carré intégrable. Pour commencer, l'ondelette est une fonction réelle qui doit être inversible. Pour cela, une *condition d'admissibilité* a été définie sur la transformée en ondelette. En notant ψ l'ondelette cette condition s'exprime alors comme :

$$c_\psi = \int_0^\infty |\hat{\psi}(\nu)|^2 \nu^{-1} d\nu < +\infty \quad , \quad (3.15)$$

avec $\hat{\psi}$ est la transformée de Fourier de ψ . Notons alors que les ondelettes doivent avoir une moyenne nulle, c'est-à-dire $\int \psi = 0$ pour que cette condition soit satisfaite. Ainsi par définition, ψ est dit l'ondelette mère. Ensuite en effectuant des translations et des mises à l'échelle de celle-ci, nous obtenons une famille d'ondelettes. Plus concrètement, soit $a \in \mathbb{R}_+$ le facteur de mise à l'échelle et $b \in \mathbb{R}$ le paramètre de translation, dans ce cas, l'ondelette fille que nous notons $\psi_{a,b}$ s'écrit :

$$\forall x \in \mathbb{R}, \quad \psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad . \quad (3.16)$$

Alors en prenant une fonction f dans $\mathbf{L}^2(\mathbb{R})$, nous pouvons écrire sa CWT en projetant la fonction f sur la famille des ondelettes, une à une. Les quantités obtenues alors sont les coefficients d'ondelettes et ont pour formule :

$$\forall a \in \mathbb{R}_+, b \in \mathbb{R}, w_f(a, b) = \langle f, \psi_{a,b} \rangle \quad (3.17)$$

$$= \int_{\mathbb{R}} f(x) \psi_{a,b}^*(x) dx \quad (3.18)$$

$$= \frac{1}{\sqrt{a}} \int_{\mathbb{R}} f(x) \psi^*\left(\frac{x-b}{a}\right) dx \quad (3.19)$$

Remarquons qu'en introduisant $\bar{\phi}_a(x) = \frac{1}{\sqrt{a}} \psi^*\left(\frac{-x}{a}\right)$ la formule précédente se réécrit comme une convolution avec :

$$\forall a \in \mathbb{R}_+, b \in \mathbb{R}, \quad w_f(a, b) = f * \bar{\psi}_a(b) \quad . \quad (3.20)$$

Ainsi nous avons une formulation qui donne accès à une implémentation qui se fait avec des filtres linéaires, d'où son avantage.

Maintenant que nous avons les coefficients d'ondelette, nous pouvons reconstruire le signal avec la formule d'inversion donnée par [Grossmann et Morlet 1984]. En prenant les

coefficients d'ondelettes $(w_f(a, b))_{a>0, b \in \mathbb{R}}$ de la fonction f , celle-ci est reconstruite comme :

$$f(x) = \frac{1}{c_\psi} \int_{\mathbb{R}_+} \int_{\mathbb{R}} w_f(a, b) \psi_{a,b}(x) \frac{da}{a^2} db . \quad (3.21)$$

À titre d'exemple, citons :

- *l'ondelette de Haar* : il s'agit de la première ondelette à être développée [Haar 1910] et s'exprime simplement par :

$$\psi(t) = \begin{cases} 1 & \text{si } 0 \leq t < 0,5 \\ -1 & \text{si } 0,5 \leq t < 1 \\ 0 & \text{sinon.} \end{cases} \quad (3.22)$$

Une représentation de l'ondelette mère est donnée dans la figure 3.2a. Remarquons également qu'avec les coefficients discrets de cette ondelette qui sont donnés par la suivante :

$$\forall (j, n) \in \mathbb{Z}^2, \quad \psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - 2^j n}{2^j}\right) , \quad (3.23)$$

nous avons une base orthonormée de $\mathbf{L}^2(\mathbb{R})$. Pour calculer cette ondelette en pratique, il suffit d'effectuer essentiellement des soustractions. Par contre, son manque de régularité risque d'engendrer des artefacts dans les tâches de restauration, ce qui n'est pas désirable comme effet.

- *l'ondelette chapeau mexicain* : également appelée ondelette de Ricker, cette ondelette est communément utilisée. Elle est définie comme étant la dérivée seconde d'une fonction gaussienne et a pour expression :

$$\psi(t) = \frac{2}{\pi^{\frac{1}{4}} \sqrt{3}\sigma} \left(\frac{t^2}{\sigma^2} - 1 \right) \exp\left(\frac{-t^2}{2\sigma^2}\right) . \quad (3.24)$$

Contrairement au premier exemple, l'ondelette chapeau mexicain est une fonction fortement régulière. Dans le cadre la Vision Artificielle (Computer Vision, en anglais) la version 2D de celle-ci est couramment utilisée et est aussi connue sous le nom de laplacien de gaussienne ou filtre de Marr. Comme cas d'application de celle-ci, nous pouvons évoquer la détection de blob (ensemble de pixels connectés dans une image).

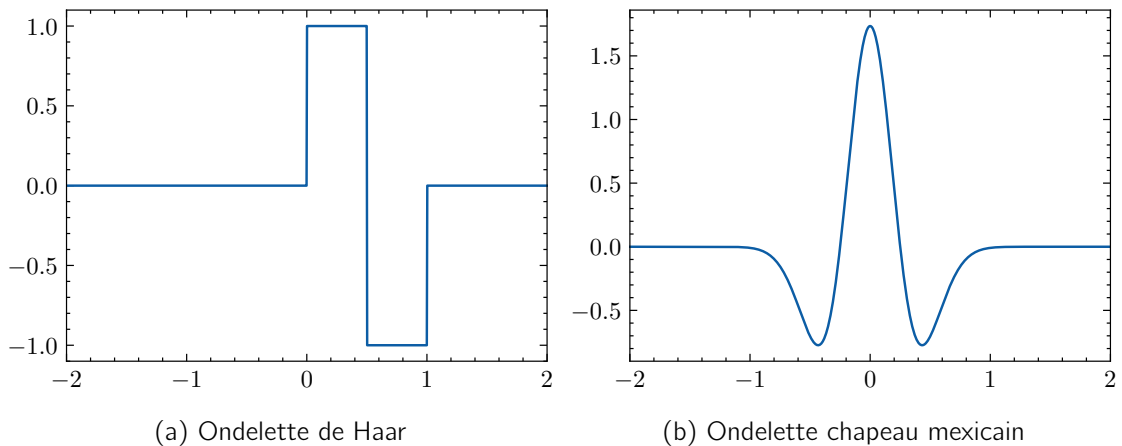


Figure 3.2 – Exemples d'ondelette mère

3.4 Ondelettes orthogonales

Jusque là les ondelettes que nous avons manipulées sont continues mais pour en tirer partie en traitement du signal, nous avons besoin d'utiliser une version discrète de celles-ci. Dans cet élan, l'analyse multi-résolution a été développée par [Mallat 1989 ; Meyer 1993] pour construire des bases d'ondelettes orthogonales. Ce formalisme permet d'avoir des ondelettes discrètes ayant une reconstruction exacte mais également une implémentation qui s'exécute rapidement.

3.4.1 Multi-résolution

L'analyse en multi-résolution est un outil qui a été développé en 1989 par Mallat [Mallat 1989] pour construire des bases d'ondelettes discrètes implémentables avec des convolutions et pouvant reconstruire le signal parfaitement. Les ondelettes obtenues par ce procédé sont appelées ondelettes orthogonales.

Ainsi avec l'analyse multi-résolution, une fonction f dans $\mathbf{L}^2(\mathbb{R})$ est approximée successivement par des fonctions f_j , appelées approximations, en lissant f avec des filtres dont la largeur augmente proportionnellement à 2^j . Pour chaque échelle 2^j , nous notons V_j le sous-espace de $\mathbf{L}^2(\mathbb{R})$ contenant f_j et également toutes les approximations pour cette échelle. Alors f_j est obtenue par la projection de f sur V_j . Or pour préserver l'intégrité du signal contenu dans f , nous définissons également le sous-espace W_j contenant les détails correspondant à la différence entre deux approximations consécutives tel que $f_{j-1} = f_j + w_j$ avec $w_j \in W_j$. Le sous-espace W_j s'écrit donc comme :

$$V_{j-1} = V_j \oplus W_j \quad , \quad (3.25)$$

avec \oplus l'opérateur de somme directe.

En décomposant récursivement les sous-espaces $(V_j)_{j \in \mathbb{Z}}$, l'espace $\mathbf{L}^2(\mathbb{R})$ se réécrit comme la somme directe des $(W_j)_{j \in \mathbb{Z}}$ donnant l'équation :

$$\mathbf{L}^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j \quad . \quad (3.26)$$

Dans [Mallat 1989], pour tout j dans \mathbb{Z} , il est possible de construire une base orthonormée de W_j à partir d'une ondelette ψ en y effectuant des translations et des remises à l'échelle. Cette famille d'ondelette s'exprime comme :

$$\left\{ \psi_{j,n} = \frac{1}{\sqrt{2^j}} \left(\frac{t - 2^j n}{2^j} \right) \right\}_{n \in \mathbb{Z}} \quad . \quad (3.27)$$

Par conséquent, $\{\psi_{j,n}\}_{(j,n) \in \mathbb{Z}^2}$ est une base de l'espace $\mathbf{L}^2(\mathbb{R})$.

Pour une définition formelle de la multi-résolution, soit f une fonction dans $\mathbf{L}^2(\mathbb{R})$ alors une approximation multi-résolution de f est une séquence $\{V_j\}_{j \in \mathbb{Z}}$ de sous-espaces fermés tel que $\{0\} \subset \dots \subset V_1 \subset V_0 \subset V_{-1} \subset \dots \subset \mathbf{L}^2(\mathbb{R})$ et qui vérifie :

- l'invariance par translation : $\forall j \in \mathbb{Z}, f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_{j+1}$
- la causalité : $f(t) \in V_j \Leftrightarrow f(t/2) \in V_{j+1}$
- la condition aux limites : $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$ et $\bigcup_{j \in \mathbb{Z}} V_j = \mathbf{L}^2(\mathbb{R})$
- l'existence d'une fonction θ telle que $\{\theta(t - n)\}_{n \in \mathbb{Z}}$ est une base de Riesz de V_0 .

Avec la base de Riesz nous pouvons alors représenter toute fonction f de V_0 avec des coefficients formant un ensemble discret. À partir de la fonction θ nous pouvons construire la fonction de remise à l'échelle, notée ϕ , telle que pour tout j dans \mathbb{Z} , $\{\phi_{j,n}\}_{n \in \mathbb{Z}}$ est une base orthonormée du sous-espace V_j [Mallat 1989]. Les éléments de cette base s'écrivent comme suit :

$$\forall (j, n) \in \mathbb{Z}^2, \quad \phi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \phi\left(\frac{t-n}{2^j}\right) . \quad (3.28)$$

Ainsi nous pouvons calculer f_j , la fonction d'approximation de f à l'échelle 2^j à l'aide de ses produits scalaires avec la base de fonction $\{\phi_{j,n}\}_{n \in \mathbb{Z}}$ de l'espace d'approximation V_j : Ainsi pour obtenir f_j , l'approximation de f à l'échelle 2^j , nous effectuons la projection de f sur l'espace V_j tel que :

$$f_j = P_{V_j} f = \sum_{n \in \mathbb{Z}} a_j[n] \phi_{j,n} , \quad (3.29)$$

où P_{V_j} est le projecteur orthogonal sur V_j et $a_j[n] = \langle f, \phi_{j,n} \rangle$ sont les coefficients d'approximation discrète de f . Alors pour tout j dans \mathbb{Z} , nous définissons une fonction d'approximation unique f_j de f , avec un ensemble discret de coefficients dans V_j .

De plus, une approximation à une échelle donnée peut être déduite à partir de celle à l'échelle précédente en effectuant une convolution basée sur la fonction de remise à l'échelle. Il suffit de voir que le fait que ϕ appartient à V_0 implique que $\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right)$ appartient à V_1 qui est inclus dans V_0 ayant pour base $(\phi_{0,n})_{n \in \mathbb{Z}}$. Nous écrivons donc la décomposition suivante :

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_{n \in \mathbb{Z}} h[n] \phi(t-n) , \quad (3.30)$$

où $h[n] = \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle$. Plus globalement, nous pouvons récrire toute fonction $\phi_{j+1,n}$ dans la base $\{\phi_{j,p}\}_{p \in \mathbb{Z}}$ de V_j avec le filtre h :

$$\phi_{j+1,p} = \sum_{n \in \mathbb{Z}} h[n-2p] \phi_{j,n} . \quad (3.31)$$

En appliquant le produit scalaire avec f , nous avons :

$$a_{j+1}[p] = \sum_{n \in \mathbb{Z}} h[n-2p] a_j[n] . \quad (3.32)$$

Cette équation permet de calculer de manière récursive les coefficients d'approximation à l'aide de h au lieu d'effectuer des produits scalaires continus. Par ailleurs, il existe des propriétés entre le filtre h et la fonction de remise à l'échelle ϕ qui sont développées dans [Mallat 1989 ; Meyer 1993], comme le théorème du Filtre Miroir Conjugué.

Désormais, nous pouvons approximer toute fonction f de $\mathbf{L}^2(\mathbb{R})$ dans une base orthonormale de V_j en calculant récursivement ses coefficients. Pour aller plus loin, nous pouvons également capturer le signal perdu entre deux échelles d'approximation consécutive. Effectivement puisque pour tout j , V_j est contenu dans V_{j-1} , nous définissons alors l'espace supplémentaire orthogonal de V_j , noté W_j comme :

$$V_{j-1} = V_j \oplus W_j . \quad (3.33)$$

Cet espace possède une base orthonormale $\{\psi_{j,n}\}_{n \in \mathbb{Z}}$ qui est construite à partir d'une ondelette ψ correspondant à une fonction de remise à l'échelle ϕ [Mallat 1989 ; Meyer 1993].

L'ondelette est alors définie par sa transformée de Fourier avec :

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2}} \hat{g}\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right) , \quad (3.34)$$

où g est un filtre discret également défini par sa transformée de Fourier avec :

$$\hat{g}(\omega) = e^{-i\omega} \hat{h}^*(\omega + \pi) . \quad (3.35)$$

Alors pour tout j dans \mathbb{Z} , $\{\psi_{j,n}\}_{n \in \mathbb{Z}}$ est une base orthonormale de W_j avec :

$$\psi_{j,n} = \frac{1}{\sqrt{2}} \psi\left(\frac{t - 2^j n}{2^j}\right) . \quad (3.36)$$

Par conséquent, nous déduisons que $\{\psi_{j,n}\}_{(n,j) \in \mathbb{Z}^2}$ est une base orthonormale de $\mathbf{L}^2(\mathbb{R})$. Alors nous pouvons représenter $\mathbf{L}^2(\mathbb{R})$ comme une somme directe $(W_j)_{j \in \mathbb{Z}}$:

$$\mathbf{L}^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j . \quad (3.37)$$

Donc toute fonction f de $\mathbf{L}^2(\mathbb{R})$ possède une représentation unique en ondelette :

$$f = \sum_{j \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} d_j[n] \psi_{j,n} , \quad (3.38)$$

avec $d_j[n] = \langle f, \psi_{j,n} \rangle$ les coefficients d'ondelette de f à l'échelle 2^j et la position $2^j n$.

D'autre part, par un raisonnement analogue à celui qui a été utilisé sur la fonction de remise à l'échelle pour aboutir à l'équation 3.30, $\psi_{j+1,p}$ s'écrit dans la base $\{\phi_{j,n}\}_{n \in \mathbb{Z}}$ de V_j comme :

$$\psi_{j+1,p} = \sum_{n \in \mathbb{Z}} g[n - 2p] \phi_{j,n} . \quad (3.39)$$

De même en appliquant le produit scalaire avec f , nous avons :

$$d_{j+1}[p] = \sum_{n \in \mathbb{Z}} g[n - 2p] a_j[n] . \quad (3.40)$$

Cette équation est la pierre angulaire des algorithmes de transformée en ondelettes rapide puisqu'elle permet de calculer les coefficients d'ondelette à l'échelle 2^{j+1} à partir des coefficients d'approximation à l'échelle 2^j au lieu de calculer le produit scalaire de f avec l'ondelette $\psi_{j,n}$.

3.4.2 Décomposition pyramidale

Dans la partie précédente, nous avons établi une relation de récurrence sur les échelles à l'aide des filtres h et g et également montré comment reconstruire un signal à partir de sa représentation dans les sous-espaces $\{W_j\}_{j \in \mathbb{Z}}$. Dans cette partie nous allons voir comment les ondelettes orthogonales garantissent une reconstruction parfaite.

Commençons par définir la transformée en ondelette rapide qui est également appelée Transformée en Ondelette Discrète (Discret Wavelet Transform (DWT), en anglais). Soit f une fonction dans $f \in \mathbf{L}^2(\mathbb{R})$ alors sa DWT est définie récursivement d'une échelle 2^j à

une échelle 2^{j+1} par :

$$a_{j+1}[p] = \sum_{n \in \mathbb{Z}} h[n - 2p] a_j[n] = [a_j * \bar{h}]_{\downarrow 2}[p] \quad , \quad (3.41)$$

$$d_{j+1}[p] = \sum_{n \in \mathbb{Z}} g[n - 2p] a_j[n] = [a_j * \bar{g}]_{\downarrow 2}[p] \quad , \quad (3.42)$$

avec $\bar{h}[n] = h[-n]$, $\bar{g}[n] = g[-n]$, $[\cdot]_{\downarrow 2}$ l'opérateur de décimation par un facteur 2 et $a_j[n]$ et $d_j[n]$ sont respectivement le coefficient d'approximation et le coefficient de détail (ou d'ondelette) donnés par :

$$a_j[n] = \langle f, \phi_{j,n} \rangle \quad \text{et} \quad d_j[n] = \langle f, \psi_{j,n} \rangle \quad . \quad (3.43)$$

La définition de la transformée inverse se fait de manière analogue pour remonter d'un échelle 2^{j+1} à partir de l'échelle 2^j avec :

$$a_j[p] = [a_{j+1}]_{\uparrow 2} * h[p] + [d_{j+1}]_{\uparrow 2} * g[p] \quad . \quad (3.44)$$

où $[\cdot]_{\uparrow 2}$ est l'opérateur de sur-échantillonnage de facteur 2. Comme le montre la figure 3.3, l'algorithme de la DWT est donc implémenté avec une cascade de filtres linéaires. La complexité de celui-ci est en $\mathcal{O}(n)$ donc comparé à la FFT qui a une complexité en $\mathcal{O}(n \log(n))$, nous avons un gain en temps de calcul.

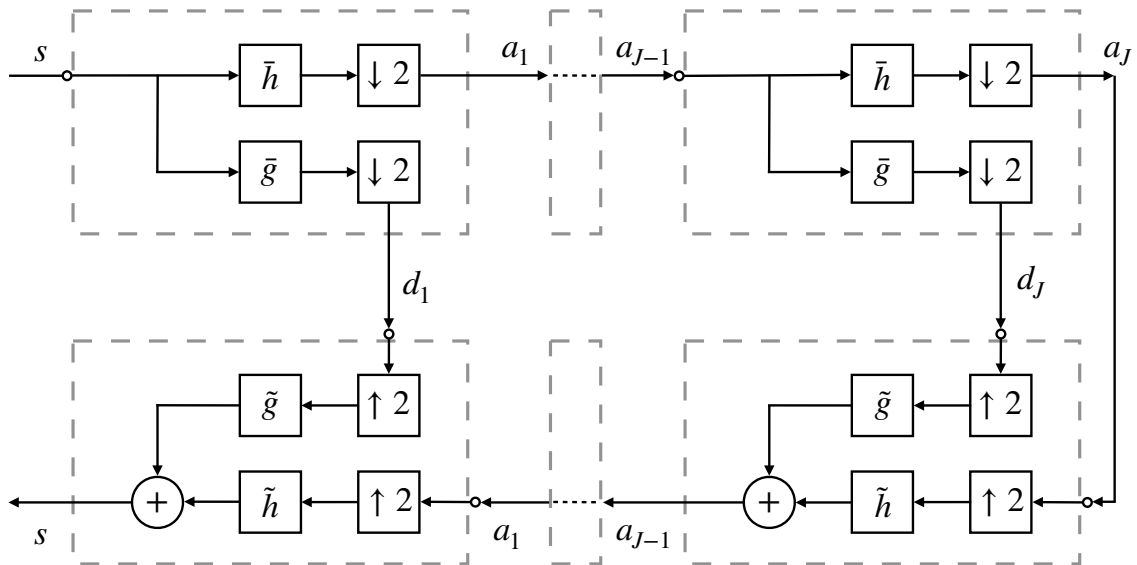


Figure 3.3 – Algorithme de transformée en ondelette rapide pour un banc de filtres $(h, g, \tilde{h}, \tilde{g})$. Dans le cas d'ondelette orthogonale, $\tilde{h} = h$ et $\tilde{g} = g$. L'étage du haut dans le diagramme correspond à l'analyse du signal s . L'étage du bas correspond à la synthèse du signal s .

Pour l'instant, nous avons uniquement considéré une seule ondelette pour les deux étapes d'analyse et de synthèse du signal. Nous introduisons alors les ondelettes bi-orthogonales qui utilisent deux ondelettes différentes, une première ondelette pour l'analyse et une deuxième pour la synthèse. Ces ondelettes donnent une reconstruction exacte du signal si deux propriétés sont satisfaites [Vetterli 1986]. Pour cela, notons (h, g) les filtres correspondant à l'ondelette d'analyse et (\tilde{h}, \tilde{g}) les filtres correspondant à l'ondelette de synthèse. Pour avoir

une reconstruction exacte, il faut que le banc de filtres $(h, g, \tilde{h}, \tilde{g})$ vérifie :

$$\hat{h}^*(\omega + \pi)\hat{h}(\omega) + \hat{g}^*(\omega + \pi)\hat{g} = 0 \quad (3.45)$$

et

$$\hat{h}^*(\omega)\hat{h}(\omega) + \hat{g}^*(\omega)\hat{g} = 2 \quad . \quad (3.46)$$

Dans le cas bi-orthogonal, l'équation de reconstruction s'écrit :

$$a_j[p] = [a_{j+1}]_{\uparrow 2} * \tilde{h}[p] + [d_{j+1}]_{\uparrow 2} * \tilde{g}[p] \quad . \quad (3.47)$$

En passant des ondelettes orthogonales aux ondelettes bi-orthogonales, nous obtenons plus de flexibilité sur le choix des ondelettes. Ainsi pour un couple d'ondelettes bi-orthogonales, les fonctions choisies peuvent avoir des propriétés intéressantes comme la symétrie, la régularité ou la compacité du support [Cohen et Daubechies 1992]. Pour l'implémentation de ces ondelettes, les étapes de l'algorithme demeurent les mêmes que précédemment comme illustré dans la figure 3.3.

3.4.3 Ondelettes bi-orthogonales 2D

Les ondelettes ont été étendues au-delà d'une dimension et dans différents espaces, par exemple nous avons les ondelettes sur la sphère [Starck et al. 2006], les ondelettes en 3D [Woiselle, Starck et Fadili 2010] et également les ondelettes sphériques 3D [Lanusse, Rassat et Starck 2012]. Dans le cadre de notre travail, nous nous intéressons aux ondelettes 2D. Celles-ci peuvent être directement construites à partir des ondelettes à une dimension. En effet les filtres 2D sont obtenus en effectuant des produits séparables de filtres 1D. Pour aller plus loin dans la construction des ondelettes 2D, il est important de tenir compte de la grandeur physique que représente chaque dimension. Par exemple, une dimension peut représenter le temps, l'espace, la fréquence ou l'énergie. Nous avons alors deux cas de figures : les dimensions homogènes comme dans le cas des images où chaque dimension correspond à un axe dans l'espace ; et les dimensions inhomogènes comme le temps et l'énergie. Ainsi dans le cas homogène, comme chaque dimension correspond à la même grandeur physique, il est alors pertinent d'utiliser la même échelle d'ondelette pour les deux dimensions. Au contraire, dans le cas inhomogène, il est préférable d'utiliser des échelles différentes telles que chaque échelle soit adaptée au signal et à la grandeur physique associés à la dimension, par exemple pour l'étude des sources transitoires en radio où l'objet a une énergie élevée et un temps d'apparition faible.

Commençons avec les ondelettes inhomogènes, en considérant la famille d'ondelettes 1D $\{\psi_{j,n}\}_{(j,n) \in \mathbb{Z}^2}$ formant une base de $\mathbf{L}^2(\mathbb{R})$ la famille d'ondelettes 2D

$$\{\psi_{j_1, j_2, n_1, n_2}\}_{(j_1, j_2, n_1, n_2) \in \mathbb{Z}^4} = \{\psi_{j_1, n_1} \cdot \psi_{j_2, n_2}\}_{(j_1, j_2, n_1, n_2) \in \mathbb{Z}^4} \quad (3.48)$$

est une base de $\mathbf{L}^2(\mathbb{R})$ avec j_1 et j_2 non liées ce qui permet d'avoir des échelles différentes pour chaque dimension. Les coefficients dans cette base se divisent en trois catégories :

- coefficients détail-détail : $d_{j_1}^1 d_{j_2}^2$,
- coefficients approximation-détail : $a_{j_1}^1 d_{j_2}^2$ et $d_{j_1}^1 a_{j_2}^2$,
- coefficients approximation-approximation : $a_{j_1}^1 a_{j_2}^2$.

Maintenant passons aux ondelettes homogènes qui sont plus adaptées pour étudier les images. Pour les construire nous utilisons la multi-résolution séparable. Ainsi soit f une fonction de $\mathbf{L}^2(\mathbb{R}^2)$ alors son approximation à l'échelle 2^j correspond à sa projection sur le

sous-espace $V_j^2 = V_j \otimes V_j$. Pour accéder au détail de f à cette échelle, nous commençons par définir l'espace de détail W_j^2 tel que $V_{j-1}^2 = V_j^2 \oplus W_j^2$. Cet espace admet comme base une famille d'ondelettes obtenue par produits séparables de fonction de remise à l'échelle, notée ϕ , et d'une ondelette 1D, notée ψ [Mallat 1999]. Nous avons alors trois cas de figure, pour $x = (x_1, x_2)$ dans \mathbb{R}^2 :

$$\psi^1(x) = \phi(x_1) \psi(x_2), \quad \psi^2(x) = \psi(x_1) \phi(x_2), \quad \psi^3(x) = \psi(x_1) \psi(x_2) \quad . \quad (3.49)$$

Par suite pour tout $k \in \{1, 2, 3\}$, nous définissons :

$$\psi_{j,n}^k(x) = \frac{1}{2^j} \psi^k \left(\frac{x_1 - 2^j n_1}{2^j}, \frac{x_2 - 2^j n_2}{2^j} \right) \quad , \quad (3.50)$$

Et pour tout j dans \mathbb{Z} , la famille d'ondelettes $\{\psi_{j,n}^1, \psi_{j,n}^2, \psi_{j,n}^3\}_{n \in \mathbb{Z}^2}$ est une base orthonormée de W_j^2 . Et la famille $\{\psi_{j,n}^1, \psi_{j,n}^2, \psi_{j,n}^3\}_{(j,n) \in \mathbb{Z} \times \mathbb{Z}^2}$ est une base orthonormée de $\mathbf{L}^2(\mathbb{R}^2)$.

Nous pouvons maintenant donner l'expression des coefficients d'approximation et de détail comme suit :

$$\begin{aligned} a_{j+1} &= [a_j * \bar{h}\bar{h}]_{\downarrow 2} \quad , \\ d_{j+1}^1 &= [a_j * \bar{h}\bar{g}]_{\downarrow 2} \quad , \\ d_{j+1}^2 &= [a_j * \bar{g}\bar{h}]_{\downarrow 2} \quad , \\ d_{j+1}^3 &= [a_j * \bar{g}\bar{g}]_{\downarrow 2} \quad , \end{aligned} \quad (3.51)$$

avec $hg[x_1, x_2] = h[x_1]g[x_2]$. En calculant ces coefficients, nous avons la DWT 2D.

De manière similaire, la transformée inverse est donnée récursivement avec l'équation :

$$a_j = [a_{j+1}]_{\uparrow 2} * hh + [d_{j+1}^1]_{\uparrow 2} * hg + [d_{j+1}^2]_{\uparrow 2} * gh + [d_{j+1}^3]_{\uparrow 2} * gg \quad (3.52)$$

Pour les ondelettes bi-orthogonales 2D, deux ondelettes sont utilisées, une pour la synthèse et une autre pour l'analyse, comme dans le cas 1D. Et la reconstruction exacte d'un signal est garantie si le banc de filtre correspondant, noté (\tilde{h}, \tilde{g}) , vérifie les équations 3.45 et 3.46.

Notons qu'il est courant de représenter les coefficients des ondelettes sur un support de taille identique à l'image auxquelles correspondent ces coefficients. En effet, il suffit de voir que la famille d'ondelettes est une base de l'espace auquel appartient l'image. Cela permet également de dire que cette transformée est non redondante.

3.5 Starlet

Même si les ondelettes introduites jusqu'à présent ont le mérite de donner une reconstruction exacte et une implémentation, elles possèdent tout de même un inconvénient important : elles ne sont pas invariantes par translation. Or cette propriété est utile en traitement d'image si par exemple nous souhaitons étudier ou modifier les propriétés d'un objet dans l'image sans se préoccuper de sa position. Pour palier ce défaut, la transformée en ondelettes redondante a été développée. La redondance correspond au fait que la famille d'ondelettes est toujours génératrice de l'espace dans lequel est l'image mais n'est plus une base de celui-ci. Cette relaxation montre son intérêt dans la conception des ondelettes qui peuvent être choisies avec plus de flexibilité.

3.5.1 Transformée en ondelette à trous

La différence majeure qui distingue la Transformée en Ondelette Non-décimées (Undecimated Wavelet Transform (UWT), en anglais) de la DWT est la déletion de l'étape de décimation. En pratique, cette transformée est implémentée avec un algorithme à trous [Holschneider et al. 1990 ; Shensa 1992] où les coefficients d'ondelette sont obtenus comme suit :

$$a_{j+1}[k] = a_j * \bar{h}^{(j)}[k] = \sum_{l \in \mathbb{Z}} h[l] a_j [l - 2^j k] \quad , \quad (3.53)$$

$$d_{j+1}[k] = a_j * \bar{g}^{(j)}[k] = \sum_{l \in \mathbb{Z}} g[l] a_j [l - 2^j k] \quad . \quad (3.54)$$

Dans le cas de la DWT, les filtres h^j et $g^{(j)}$ sont obtenus en enlevant des composantes des filtres de départ h et g . Par contre, dans le cas de la UWT, les filtres h^j et $g^{(j)}$ sont obtenus rajoutant "des trous" qui correspondent exactement à 2^j zéros entre les composantes consécutives des filtres de départ. Pour donner un exemple, les filtres issus de h pour les deux premières échelles s'écrivent :

$$h^{(1)} = [\dots, h[-2], 0, h[-1], 0, h[0], 0, h[1], 0, h[2], \dots] \quad , \quad (3.55)$$

$$h^{(1)} = [\dots, h[-2], 0, 0, 0, h[-1], 0, 0, 0, h[0], 0, 0, 0, h[1], 0, 0, 0, h[2], \dots] \quad . \quad (3.56)$$

Il a été montré dans Shensa 1992 que les coefficients calculés à la position $2^j n$ utilisant la transformée à trous correspondent toujours aux coefficients de la décomposition en DWT originale. Le signal peut alors toujours être reconstruit à partir de ces coefficients en utilisant le banc de filtres (\tilde{h}, \tilde{g}) avec la formule suivante : Une correspondance existe entre les coefficients obtenus par la transformée à trous et les coefficients obtenus par DWT [Shensa 1992]. Nous déduisons alors que le banc de filtres (\tilde{h}, \tilde{g}) peut également être utilisé pour reconstruire un signal à partir de ses coefficients de la transformée à trous comme suit :

$$a_j[k] = \tilde{h}^{(j)} * a_{j+1}[k] + \tilde{g}^{(j)} * d_{j+1}[k] \quad . \quad (3.57)$$

Dans le cas de la DWT, pour obtenir une reconstruction exacte, deux conditions doivent être remplies par le banc de filtres. Avec la suppression de la décimation dans la UWT, le banc de filtres $(h, g, \tilde{h}, \tilde{g})$ doit seulement remplir la condition suivante :

$$\hat{h}^*(\omega) \hat{h}(\omega) + \hat{g}^* \hat{g}(\omega) = 2 \quad . \quad (3.58)$$

La UWT 2D est déduite du cas 1D de manière analogue à l'extension de la DWT en 2D. Ainsi la UWT d'une image avec J échelles, contient J coefficients d'ondelette et une approximation lisse. Pour chaque échelle, nous avons trois coefficients d'ondelette dont un horizontal, un vertical et un diagonal. Comme chacun de ces coefficients a une taille égale à celle de l'image alors la redondance de la UWT vaut $3J + 1$.

3.5.2 Transformée en starlet

Dans le cas de l'astrophysique, les objets observés sont isotropes de premier abord. Donc pour avoir une transformée en ondelette adaptée à ce problème, il faut que celle-ci soit choisie isotrope. Grâce au formalisme de la UWT, il est possible de faire ce choix et c'est de cet élan que la *Transformée en Ondelette Non-décimée Isotrope* (Isotropic Undecimated

Wavelet Transform (IUWT)) [Starck et Murtagh 2006] a été développée. Cette transformée, également appelée transformée en starlet, est adaptée aux images astrophysiques et possède deux propriétés : la première est que les filtres h et g sont symétriques et la deuxième est l'isotropie des fonctions de remise à l'échelle et de l'ondelette.

Pour la construction de la transformée en starlet, nous commençons tout d'abord par considérer une fonction de remise à l'échelle ϕ qui soit isotrope puis nous définissons l'ondelette comme étant la différence entre les approximations à deux échelles successives :

$$\frac{1}{4}\psi\left(\frac{x_1}{2}, \frac{x_2}{2}\right) = \phi(x_1, x_2) - \frac{1}{4}\phi\left(\frac{x_1}{2}, \frac{x_2}{2}\right) \quad (3.59)$$

Ainsi l'isotropie de ϕ entraîne l'isotropie de ψ comme étant la combinaison linéaire de fonctions isotropes.

Cette formule se répercute sur le couple de filtres (h, g) associés à la transformée comme dans l'équation ci-dessous :

$$g[k, l] = \delta[k, l] - h[k, l] \quad (3.60)$$

où δ est le symbole de Kronecker. Cette relation montre que la somme des coefficients d'ondelette mène à une reconstruction exacte du signal avec :

$$a_0[k, l] = a_J[k, l] + \sum_{j=1}^J d_j[k, l] \quad (3.61)$$

Plus généralement, pour tout filtre à Réponse Impulsionnelle Finie (Finite Impulse Response filter (FIR)) h qui soit paire et symétrique, le banc de filtres associés donne une reconstruction exacte du signal.

D'ailleurs, notons que la transformée en starlet est caractérisée par sa fonction de remise à l'échelle seulement. La fonction de remise à l'échelle qui a été retenue pour cette transformée est alors une fonction B-spline d'ordre 3. Ce choix a été effectué dans le but d'avoir une transformée adaptée aux images en astrophysique. L'équation de la fonction de remise à l'échelle s'écrit :

$$\phi_{1D}(x) = \frac{1}{12} \left(|x-2|^3 - 4|x-1|^3 + 6|x|^3 - 4|x+1|^3 + |x+2|^3 \right) \quad (3.62)$$

La construction de la fonction de remise à l'échelle en dimension n , notée ϕ_{nD} , est obtenue en effectuant un produit séparable de n facteurs égaux à la fonction de remise à l'échelle ϕ_{1D} . En particulier, dans le cas 2D, nous avons :

$$\phi_{2D}(x_1, x_2) = \phi_{1D}(x_1)\phi_{1D}(x_2) \quad (3.63)$$

Une représentation de la fonction de remise à l'échelle et de l'ondelette dans le cas 1D est donnée dans la figure 3.4. De plus, le fait que l'ondelette soit très régulière avec peu d'oscillations et un support compact fait de la transformée en starlet un outil puissant pour les tâches de restauration. Les filtres FIR associés h et g s'écrivent :

$$h_{1D}[k] = \frac{1}{16}[1, 4, 6, 4, 1] \quad (3.64)$$

$$h_{2D}[k, l] = h_{1D}[k]h_{1D}[l] \quad (3.65)$$

$$g_{2D}[k, l] = \delta[k, l] - h_{2D}[k, l] \quad (3.66)$$

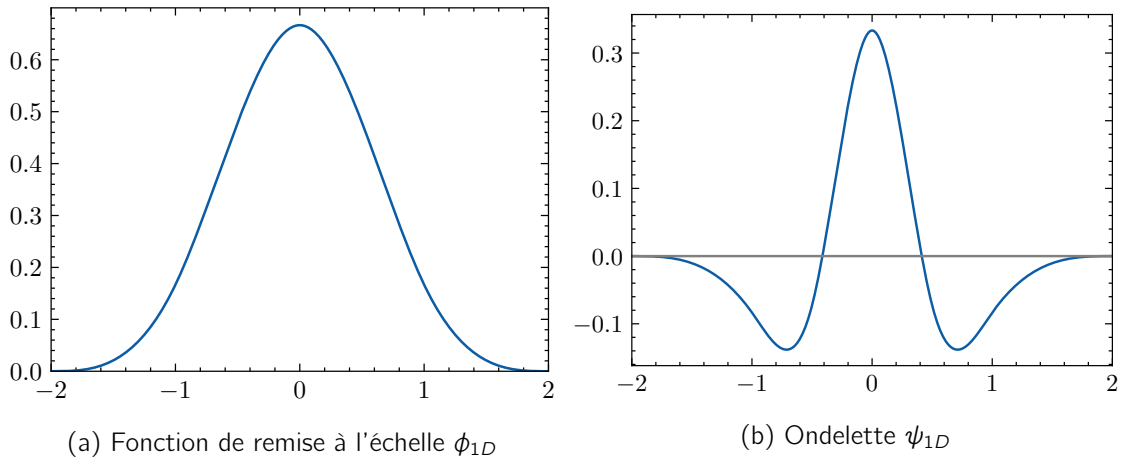


Figure 3.4 – Fonction de remise à l'échelle et l'ondelette de la transformée en starlet 1D..

Concernant l'implémentation de cette transformée, elle est rapide du fait de la séparabilité de la fonction de remise à l'échelle. Cela signifie que les convolutions 2D peuvent être effectuées à l'aide de convolutions 1D consécutives tout au long des lignes et des colonnes.

Pour illustrer, une image et sa transformée en starlet sont données dans la figure 3.5. Vu qu'il s'agit d'une transformée isotrope, à chaque échelle nous avons uniquement un seul coefficient d'ondelette. Donc pour J échelles, nous avons J coefficients d'ondelette et un coefficient d'approximation lisse. Nous déduisons alors que la redondance dans ce cas vaut $J + 1$, ce qui faible en comparaison avec la UWT.

3.5.3 Deuxième Génération de Starlet

Pour l'instant, nous avons défini la transformée en starlet en recherchant des propriétés portant uniquement sur la fonction de remise à l'échelle et l'ondelette dans l'étape d'analyse. Mais à l'étape de synthèse, mise à part la garantie d'une reconstruction exacte, il n'y a pas de propriété particulière qui a été recherchée. Or la qualité de la reconstruction dans le cadre de traitement du signal est affectée par l'opérateur de synthèse. Ainsi pour des images à valeurs positives comme c'est essentiellement le cas en astrophysique, il est préférable d'avoir une reconstruction qui minimise les artefacts de réverbération sur les contours. Alors dans ce cas, nous pouvons chercher des opérateurs pour la synthèse qui assurent que si les coefficients d'ondelette sont positifs alors l'image reconstruite l'est aussi : c'est la propriété de reconstruction positive.

Cette propriété a été greffée à la transformée en starlet pour donner la *deuxième génération de Starlet* [Starck, Fadili et Murtagh 2007]. Le banc de filtre $(h, g, \tilde{h}, \tilde{g})$ associé à la deuxième génération s'exprime comme suit :

$$h[k, l] = h_{1D}[k]h_{1D}[l] \quad , \quad (3.67)$$

$$g[k, l] = \delta[k, l] - h * h[k, l] \quad , \quad (3.68)$$

$$\tilde{h}[k, l] = h[k, l] \quad , \quad (3.69)$$

$$\tilde{g}[k, l] = \delta[k, l] \quad (3.70)$$

où h est le même filtre utilisé dans le cas de la première génération d'ondelette. Par suite, la deuxième génération de starlet telle que définie, donne une reconstruction exacte. Notons

d'ailleurs que même si dans ce cas, le filtre d'analyse g demeure une ondelette en ayant une moyenne nulle, le filtre de reconstruction \tilde{g} n'est plus car il est positif.

3.6 Conclusion

Dans ce chapitre, nous avons commencé par introduire le concept de parcimonie et insisté sur son utilité en tant que terme de régularisation pour la résolution de problèmes inverses. Nous nous sommes ensuite penchés sur la transformée en ondelettes en commençant par leur base jusqu'à aboutir aux starlets. Ces dernières offrent une représentation parcimonieuse adéquate aux images de galaxies en astrophysique. Dans le chapitre suivant, nous introduisons des méthodes de déconvolution parcimonieuses utilisant cette dernière et nous parlerons de la contrainte de forme, portant sur la restauration de la forme des galaxies, altérée par la mesure.

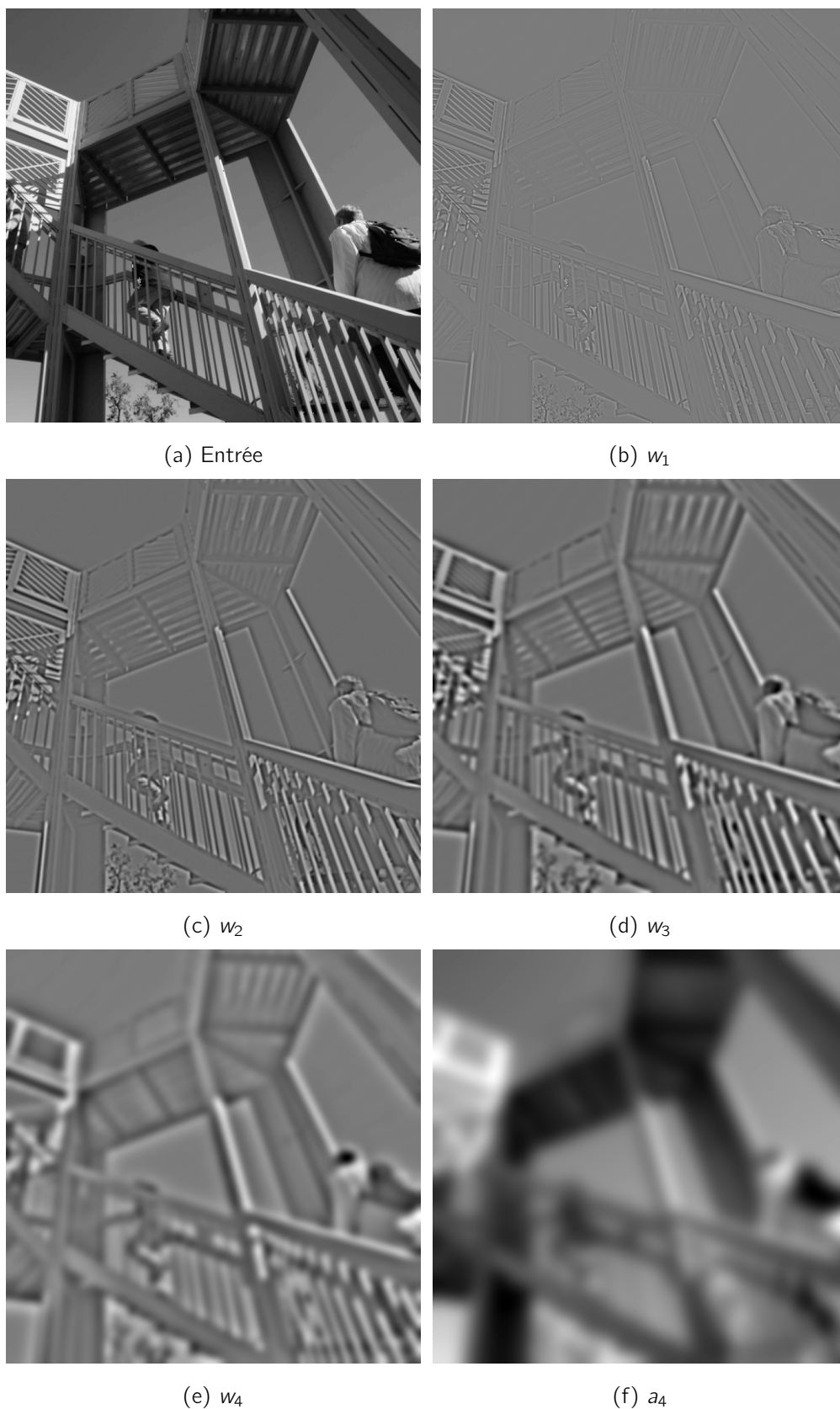


Figure 3.5 – Transformée en Starlet de l'image Ascent. Les images de (b) à (e) représentent les coefficients d'ondelette w_j pour une échelle croissante j et (f) correspond à l'approximation lisse. Source de l'image : SciPy

Chapitre 4

Reconstruction parcimonieuse et contrainte de forme

Les ondelettes font partie de l'état de l'art pour la reconstruction et la compression de données. L'utilisation des ondelettes sous-entend que les données sont parcimonieuses dans l'espace des ondelettes. Ainsi dans ce chapitre, nous allons les utiliser dans un algorithme de reconstruction parcimonieuse avec contrainte de forme : l'algorithme Shape COncstrained REStoration (SCORE). Mais avant nous allons présenter les briques de base de ce dernier qui sont la contrainte de forme que j'ai développée, et un algorithme de déconvolution parcimonieuse.

4.1 Contrainte de forme

Dans plusieurs domaines de l'astrophysique, l'information sur la forme des galaxies est importante mais les algorithmes de reconstruction d'image ne garantissent pas la préservation de celle-ci. Dans ce chapitre nous présentons une contrainte qui permet d'avoir une garantie sur la forme de la solution.

4.1.1 Nécessité de la forme en astrophysique

Dans les problèmes de reconstruction d'images, nous cherchons généralement la solution qui minimise l'Erreur Quadratique Moyenne (EQM)(Mean Square Error (MSE), en anglais). Même si cette erreur est standard, elle n'est pas le critère ciblé dans certains domaines d'application. Notamment, en astrophysique, la forme des galaxies est centrale dans plusieurs objectifs scientifiques comme le lentillage gravitationnel faible (weak gravitational lensing, en anglais) [Mandelbaum 2018] ou l'étude de la formation et de l'évolution des galaxies [Rodríguez, Padilla et García Lambas 2016] (souvent cette information est encodée par l'ellipticité). Néanmoins, il n'est pas de garantie que l'étape de déconvolution préserve la forme des galaxies, surtout si celle-ci est non-linéaire. Pour cette raison, la déconvolution complète d'image de galaxie est rarement utilisée en pratique dans le cas d'étude du lentillage gravitationnel faible. Au lieu de cela, l'effet de la FEP est corrigé soit en utilisant des méthodes basées sur les moments, telle que la méthode Kaiser-Squires-Broadhurst (KSB) [Kaiser, Squires et Broadhurst 1995], soit basées sur l'ajustement de modèle en faisant l'hypothèse que la galaxie a un certain profil analytique (comme par exemple, un cœur gaussien ou un profil Sersic) [Miller et al. 2007]. Notons qu'aucune de ces méthodes ne produit une image déconvoluée de la galaxie originale. Si ce profil est requis pour avancer dans d'autres objectifs scientifiques, une déconvolution explicite est effectuée séparément et l'image résultante ne possède pas l'ellipticité correcte. En radio interférométrie, la situation est encore plus problématique vu que les données sont acquises dans l'espace de Fourier et

que ces méthodes standards de mesure d'ellipticité requièrent la reconstruction de l'image tout d'abord. Dans [Patel 2016 ; Patel et al. 2014], il a été montré que les méthodes standards de reconstruction d'image en radio interférométrie ne peuvent pas être utilisées afin d'obtenir des mesures fiables, ce qui a poussé la communauté à développer des techniques d'ajustement dans l'espace de Fourier [Rivi et al. 2016, 2019]. Puisque chaque composante de Fourier contient de l'information de toutes les galaxies, cela demande alors l'ajustement simultané de l'ellipticité de toutes les galaxies contenue dans l'image observée. Une telle minimisation est plutôt complexe et nécessite l'utilisation de l'algorithme hamiltonien de Monte-Carlo qui est coûteux en ressources de calcul. Une approche alternative et originale est proposée dans [Kumar 2017 ; Shakibaei et Flusser 2014]. Celle-ci utilise la relation entre les moments de l'image dégradée et les moments de l'image originale et ceux de la FEP. La solution est ensuite obtenue en inversant l'équation des moments. Cependant, cette méthode se base sur l'hypothèse que la FEP est une gaussienne elliptique, ce qui n'est pas le cas en pratique, surtout dans le cas des FEP radiointerférométriques obtenues par synthèse d'ouverture. Dans ce chapitre, nous présentons une nouvelle approche fondée sur l'ajout d'une contrainte de forme qui ouvre la voie à des solutions hybrides utilisant les méthodes standards de déconvolution avec régularisation et des apports de la méthode des moments géométriques.

4.1.2 Développement de la contrainte de forme

La contrainte de forme a pour finalité de préserver l'ellipticité d'une galaxie en s'exprimant sur ses moments. Ainsi dans cette section, nous allons voir tout d'abord, ce qu'est l'ellipticité d'une galaxie. Ensuite nous allons voir comment elle peut être estimée à partir des moments de l'image. Enfin, nous construirons la contrainte en nous basant sur ces moments.

Ellipticité

La définition de l'ellipticité est triviale pour les objets ayant un profil lumineux avec des lignes de niveaux elliptiques. Dans le but de généraliser cette définition aux objets ayant des profils quelconques, les moments statistiques du profil lumineux (plutôt de l'image observée en pratique) sont utilisés. Donc pour une image de galaxie, notée $x \in \mathbb{R}^{n \times n}$, nous définissons l'ellipticité complexe $e(x) = e_1(x) + \mathbf{i}e_2(x)$, comme dans [Bartelmann et Schneider 2001], par :

$$e_1(x) = \frac{\mu_{2,0}(x) - \mu_{0,2}(x)}{\mu_{2,0}(x) + \mu_{0,2}(x)} \quad \text{et} \quad e_2(x) = \frac{2\mu_{1,1}(x)}{\mu_{2,0}(x) + \mu_{0,2}(x)} \quad , \quad (4.1)$$

avec $\mu_{s,t}(x)$ les moments centrés. Dans le cas d'une image discrète contenant l'objet d'intérêt, un estimateur non-biaisé de ces quantités sont calculées comme suit :

$$\mu_{s,t}(x) = \sum_{i=1}^n \sum_{j=1}^n x [(i-1)n + j] (i - i_c)^s (j - j_c)^t \quad , \quad (4.2)$$

tel que

$$i_c(x) = \frac{\sum_{i=1}^n \sum_{j=1}^n i \cdot x [(i-1)n + j]}{\sum_{i=1}^n \sum_{j=1}^n x [(i-1)n + j]} \quad \text{et} \quad j_c(x) = \frac{\sum_{i=1}^n \sum_{j=1}^n j \cdot x [(i-1)n + j]}{\sum_{i=1}^n \sum_{j=1}^n x [(i-1)n + j]} \quad (4.3)$$

sont les coordonnées du centroïde de x . Notons également que dans l'équation 4.2, i et j correspondent respectivement à la i^{e} ligne et j^{e} colonne de l'image de galaxie, x .

L'ellipticité est une quantité extrêmement sensible au bruit. L'estimation en pratique de l'ellipticité d'une galaxie est alors un problème mal-posé comme le montre la figure 4.1. Une façon intuitive pour rendre l'estimation plus robuste, c.-à-d. réduire la sensibilité au bruit de fond, se fait à travers l'utilisation d'une fonction fenêtre, notée $g \in \mathbb{R}^{n \times n}$. Typiquement, la fenêtr choisie est une gaussienne 2D. La taille de la fonction fenêtr est soit fixée a priori, ou ajustée sur y , l'image observée [Hirata et Seljak 2003]. Alors un estimateur plus robuste de l'ellipticité à partir de l'image y , noté $e_{\text{int}}(y) = e_{\text{int},1}(y) + i e_{\text{int},2}(y)$ s'écrit :

$$e_{\text{int}}(y) = \frac{\mu_{2,0}(y \odot g) - \mu_{0,2}(y \odot g) + 2i\mu_{1,1}(y \odot g)}{\mu_{2,0}(y \odot g) + \mu_{0,2}(y \odot g)}, \quad (4.4)$$

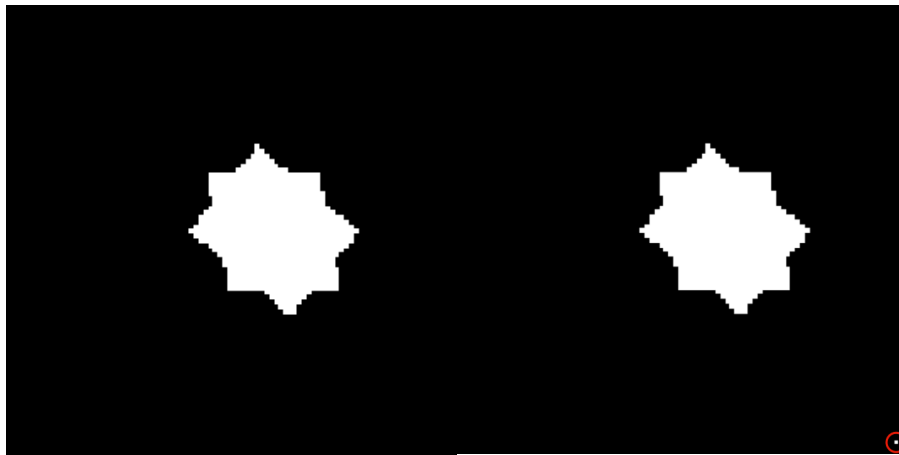


Figure 4.1 – Les deux images binaires diffèrent uniquement d'un seul pixel, encerclé en rouge en bas à droite de l'image droite. L'ellipticité mesurée, en utilisant l'équation 4.1, vaut $0.0016 + i0.2196$ pour l'image de gauche et $-0.0094 + i0.1916$ pour l'image de droite. Cela correspond à une déviation de 14%.

La méthode KSB, que ce soit l'originale [Kaiser, Squires et Broadhurst 1995] ou ses améliorations [Viola, Melchior et Bartelmann 2011], repose sur un estimateur identique à celui vu en équation (4.4). L'ellipticité d'une version pondérée de l'image observée est d'abord calculée pour réduire les effets du bruit. Ensuite, les effets de la FEP sont corrigés dans l'espace des moments. La clef de voûte de cette approche est d'approximer linéairement les effets de la FEP, sous hypothèse que la FEP est légèrement anisotrope.

Des ellipticités aux produits scalaires

Idéalement, la contrainte de forme devrait avoir la forme d'un terme d'attache aux données (voir partie 2.2.1) dans un espace qui correspond à celui de l'ellipticité. Cependant, l'ellipticité $e(x)$ telle que définie dans l'équation 4.1 est une fonction non-linéaire de l'image de galaxie, x . Nous la réexprimons alors comme combinaisons de quantités qui sont linéaires vis-à-vis de x , ces quantités sont plus simples à manipuler mathématiquement. Dans [Bernstein et Armstrong 2014a], il a été montré que l'ellipticité peut être réécrite comme combinaison de produits scalaires. De manière analogue, nous dérivons dans l'appendice A,

les formules suivantes :

$$\begin{aligned} e_1(x) &= \frac{\langle x, u_3 \rangle \langle x, u_5 \rangle - \langle x, u_1 \rangle^2 + \langle x, u_2 \rangle^2}{\langle x, u_3 \rangle \langle x, u_4 \rangle - \langle x, u_1 \rangle^2 - \langle x, u_2 \rangle^2} , \\ e_2(x) &= \frac{2(\langle x, u_3 \rangle \langle x, u_6 \rangle - \langle x, u_1 \rangle \langle x, u_2 \rangle)}{\langle x, u_3 \rangle \langle x, u_4 \rangle - \langle x, u_1 \rangle^2 - \langle x, u_2 \rangle^2} , \end{aligned} \quad (4.5)$$

avec $(u_k)_{k \in \{1, \dots, 6\}}$ dans $\mathbb{R}^{6 \times n \times n}$, définis pour tout i et j dans $\{1, \dots, n\}$ dans

$$\begin{aligned} u_1[(i-1)n+j] &= (i), & u_2[(i-1)n+j] &= (j), \\ u_3[(i-1)n+j] &= (1), & u_4[(i-1)n+j] &= (j^2 + j^2), \\ u_5[(i-1)n+j] &= (i^2 - j^2), & u_6[(i-1)n+j] &= (ij). \end{aligned} \quad (4.6)$$

L'équation 4.5 montre que l'information de l'ellipticité est contenue dans un ensemble de six produits scalaires. De plus, dans l'équation 4.6, nous pouvons voir que les $(u_k)_{k \in \{1, \dots, 6\}}$ sont des vecteurs constants. Ce qui fait que les produits scalaires sont tous des fonctions linéaires de x . Par conséquent, nous les choisissons comme briques de bases pour la contrainte de forme, au lieu d'utiliser directement l'ellipticité (qui n'est pas une fonction linéaire de par rapport à x). Ainsi, une formulation de la contrainte est :

$$M_0(x) = \sum_{i=1}^6 \omega_i \langle x * h - y, u_i \rangle^2 , \quad (4.7)$$

avec $(\omega_i)_{i \in \{1, \dots, 6\}}$ des poids réels positifs.

Fenêtrage

Comme présenté dans la partie 4.1.2, les termes de la somme dans l'équation 4.7 sont extrêmement sensibles au bruit. Une manière naturelle d'augmenter la robustesse de notre contrainte de forme serait, par analogie avec l'estimateur d'ellipticité dans l'équation 4.4, d'appliquer une fonction de pondération g . Cette option vient tout de même avec le dilemme consistant de comment choisir correctement g . D'une part, une fonction de pondération constante n'est pas suffisamment flexible et conduirait à de mauvaises estimations d'ellipticités pour certaines galaxies observées. D'autre part, ajuster g sur y améliorerait la flexibilité mais requiert une étape de prétraitement supplémentaire.

Une approche alternative est d'appliquer plusieurs fenêtres de différentes tailles et orientations, de sorte qu'au moins l'une d'elles soit correctement ajustée à y . Pour trouver un tel ensemble de fenêtres, nous avons considéré des décompositions de la famille des curvelets [Starck, Candes et Donoho 2002 ; Starck, Murtagh et Fadili 2015] où toutes les bandes correspondent à des fenêtres avec différentes orientations, et chaque échelle correspond à une taille différente. Les *shearlets* sont particulièrement appropriées pour cette fin, comme nous pouvons le voir dans la figure 4.2. Ce choix a également été motivé par les deux propriétés suivantes des shearlets :

- *anisotropie* : l'ellipticité est, elle-même, une mesure d'anisotropie et l'utilisation des shearlets qui est une transformation anisotropique devrait permettre une meilleure discrimination de ces objets suivant ce critère.
- *conservation de la grille* : Les opérations de redimensionnement et de cisaillement qui font passer d'une bande de shearlets à une autre, conservent les points sur la

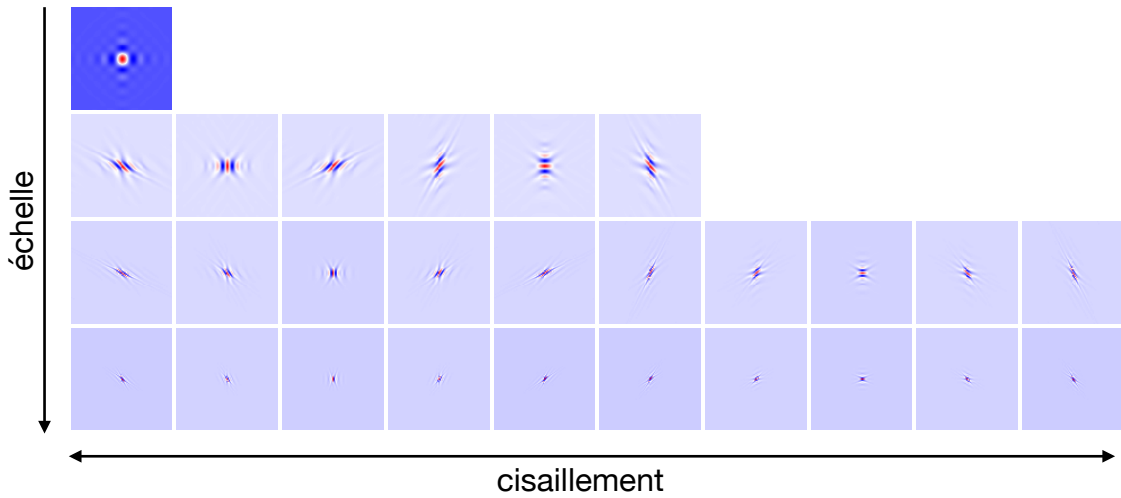


Figure 4.2 – Représentation des bandes de shearlets à trois échelles.

grille, c.-à-d. que les différences entre les anciennes et les nouvelles coordonnées sont entières. Cela offre ainsi une stabilité numérique.

Soit $\psi = (\psi_j)_{j \in \{1, \dots, J\}}$ l'opérateur de transformation en shearlet, avec J le nombre choisi de composantes. Nous formulons alors la contrainte de forme comme suit :

$$M(x) = \sum_{i=1}^6 \sum_{j=1}^J \omega_{ij} \langle \psi_j(x * h) - \psi_j(y), u_i \rangle^2, \quad (4.8)$$

où $(\omega_{ij})_{\substack{i \in \{1, \dots, 6\} \\ j \in \{1, \dots, J\}}}$ sont des réels positifs dont la valeur est déterminée dans la partie suivante.

L'utilisation des moments dans les espaces des ondelettes [Chen et Xie 2011 ; Farokhi et al. 2014] et des curvelets [Dhahbi, Barhoumi et Zagrouba 2015 ; Murtagh et Starck 2008] a déjà été utilisée de manière similaire mais uniquement pour des fins de classifications.

4.1.3 Reformulation de la contrainte

Dans cette section nous effectuons une reformulation judicieuse de la contrainte de forme qui permet de simplifier l'équation, optimiser une éventuelle implémentation et offrir une interprétation élégante de l'ajout de la contrainte de forme à une fonction de coût.

Tout d'abord, en utilisant le fait que la transformée en shearlet est une opération linéaire, nous pouvons commencer par réécrire l'équation 4.8 comme :

$$M(x) = \sum_{i=1}^6 \sum_{j=1}^J \omega_{ij} \langle \psi_j(x * h - y), u_i \rangle^2, \quad (4.9)$$

Ensuite en appliquant la définition de l'opérateur adjoint, noté ψ^* , nous obtenons :

$$M(x) = \sum_{i=1}^6 \sum_{j=1}^J \omega_{ij} \langle x * h - y, \psi_j^*(u_i) \rangle^2, \quad (4.10)$$

En pratique, pour implémenter la contrainte de forme en elle-même, l'équation 4.10 est plus optimale que l'équation 4.8. Effectivement pour calculer les entrées du produit scalaire,

nous effectuons une seule transformation au lieu de deux. De plus, dans un schéma itératif, toujours concernant les entrées du produit scalaire, le fait d'appliquer une transformation sur u_i qui est constante, à la place de $x * h - y$ qui est variable, permet d'appliquer une seule transformation lors de l'étape d'initialisation et d'utiliser la constante $\psi^*(u_i)$ au lieu de calculer $\psi(x * h) - \psi(y)$ à chaque itération.

Pour aller plus loin, reprenons la définition du produit scalaire dans le cas de deux vecteurs, u et v :

$$\langle u, v \rangle = u^\top \cdot v \quad , \quad (4.11)$$

Ainsi $M(x)$ s'exprime :

$$M(x) = \sum_{i,j} \omega_{ij} \langle x * h - y, \psi_j^*(u_i) \rangle^2 \quad , \quad (4.12)$$

$$= \sum_{i,j} \omega_{ij} \left([\psi_j^*(u_i)]^\top [x * h - y] \right)^\top \left([\psi_j^*(u_i)]^\top [x * h - y] \right) \quad , \quad (4.13)$$

$$= (x * h - y)^\top \sum_{i,j} \omega_{ij} \psi_j^*(u_i) [\psi_j^*(u_i)]^\top (x * h - y) \quad (4.14)$$

$$= (x * h - y)^\top \sum_{i,j} Q_{ij} (x * h - y) \quad (4.15)$$

$$= (x * h - y)^\top Q (x * h - y) \quad , \quad (4.16)$$

avec

$$Q = \sum_{i,j} Q_{ij} \quad \text{et} \quad Q_{i,j} = \omega_{ij} \psi_j^*(u_i) [\psi_j^*(u_i)]^\top \quad , \forall (i,j) \in \{1, \dots, 6\} \times \{1, \dots, J\}$$

Par construction, Q est une matrice symétrique semi-définie positive. Sans perte de généralité, considérons maintenant l'ajout de la contrainte de forme à une fonction de coût L , en nous intéressant uniquement à la somme du terme d'attache aux données et de la contrainte de forme pondérée par le réel positif γ :

$$L(x) = \|x * h - y\|_2^2 + \gamma M(x) \quad (4.17)$$

$$= (x * h - y)^\top I_{n^2} (x * h - y) + \gamma (x * h - y)^\top Q (x * h - y) \quad (4.18)$$

$$= (x * h - y)^\top S (x * h - y) \quad , \quad (4.19)$$

avec $S = I_{n^2} + \gamma Q$ donc S est une matrice symétrique définie positive. J'ai ainsi pu définir la norme repondérée par celle-ci et nous avons :

$$L(x) = \|x * h - y\|_S^2 \quad (4.20)$$

L'ajout de la contrainte de forme au terme d'attache aux données est équivalent alors à une modification de la norme dans le terme d'attache aux données. Nous pouvons ainsi l'interpréter comme une extension de ce terme pour tenir compte de l'espace des produits scalaires qui devrait garantir la forme de la galaxie lors de la reconstruction. Ainsi nous pouvons ajouter la contrainte de forme dans un problème d'optimisation avec une fonction de coût. Mais afin que celle-ci soit utilisée dans un algorithme de reconstruction, il est aussi utile d'exprimer deux de ces propriétés : son gradient $\nabla L(x)$ et la constante de Lipschitz, notée α de celui-ci. Ces deux quantités sont directement déduites à partir de l'équation 4.19

et s'expriment comme suit :

$$\nabla L(x) = 2h_\pi * S(x * h - y) \quad (4.21)$$

et

$$\alpha = \rho(2h_\pi * h * S) \quad (4.22)$$

Avec h_π le centro-symétrique de h et ρ la norme spectrale.

Pour déterminer la valeur des pondérations $(\omega_{ij})_{\substack{i \in \{1, \dots, 6\} \\ j \in \{1, \dots, J\}}}$, nous imposons que la contrainte de forme et le terme d'attache aux données non pondéré aient la même influence lorsque γ est égal à 1. De plus, sans a priori additionnel lié au problème, nous imposons également que les composantes de Q aient la même influence l'une vis-à-vis de l'autre. Sans garantie d'orthogonalité, nous écrivons alors les deux conditions comme suit :

$$\begin{cases} \|I_n\|_F = \sum_{i=1}^6 \sum_{j=1}^J \|Q_{ij}\|_F \quad , \\ \|Q_{ij}\|_F = \|Q_{kl}\|_F \quad , \forall i, k \in \{1, \dots, 6\} \quad , \forall j, l \in \{1, \dots, J\}. \end{cases} \quad (4.23)$$

La résolution du système obtenu donne alors les valeurs suivantes aux pondérations :

$$\omega_{ij} = \frac{n}{\|\psi_j^*(u_i)\|_2^2} \quad , \forall i \in \{1, \dots, 6\} \quad , \forall j \in \{1, \dots, J\}. \quad (4.24)$$

Notons alors que la pondération obtenue permet de rendre la contrainte de forme invariante par rapport au nombre de pixels n dans l'image puisque celui est pris en compte dans la pondération. Ainsi le seul paramètre libre dans la contrainte de forme est γ .

Finalement, nous avons construit une contrainte préservant la forme pour être utilisée dans un algorithme de reconstruction.

4.2 Algorithme SCORE

En traitement du signal, la déconvolution est un problème inverse mal posé et en particulier mal conditionné. Ainsi en appliquant un algorithme de descente de gradient sans régularisation sur une image convoluée et bruitée, la reconstruction oscille dans les itérations avancées, principalement à cause des hautes fréquences dégradées par le bruit. Ainsi pour obtenir des solutions plus stables, une façon de faire est de rajouter des a priori sous forme de contraintes pour régulariser le problème. Dans la suite nous introduisons un algorithme de restauration parcimonieuse dans un premier temps pour ensuite l'utiliser pour construire l'algorithme SCORE en y ajoutant la contrainte de forme.

4.2.1 Déconvolution parcimonieuse

L'efficacité de la régularisation parcimonieuse dans les techniques de débruitage a déjà été montrée (voir par exemple dans [Donoho et al. 1995 ; Farrens, Ngolè Mboula et Starck 2017 ; Starck, Murtagh et Fadili 2015]). De plus, dans le cas d'image astrophysique, les pixels correspondent à l'intensité d'un profil de brillance, cela veut dire que les pixels doivent être à valeurs positives. En ajoutant une contrainte de positivité, nous obtenons des résultats satisfaisants vis-à-vis de l'erreur pixellique [Farrens, Ngolè Mboula et Starck 2017]. Pour utiliser adéquatement la parcimonie, il faut trouver une représentation parcimonieuse du

signal considéré. Dans le cas des images de galaxies en astrophysique, il a été montré dans [Starck, Murtagh et Fadili 2015] que la transformée en starlet offre une représentation parcimonieuse.

En astrophysique, que ce soit en imagerie optique ou imagerie radio, la restauration des images de galaxies correspond dans les deux cas à un problème de déconvolution comme le montrent les équations 1.1 et 1.26. Comme les deux problèmes sont modélisés par les mêmes opérations, nous les reformulons en notant l'image acquise par y dans \mathbb{R}^n , le noyau de convolution par h dans \mathbb{R}^n , et le bruit additif par n dans \mathbb{R}^n :

$$y = x_T * h + b \quad (4.25)$$

avec x_T dans \mathbb{R}^n l'image vérité-terrain. Notons que dans le cas optique, le bruit est un bruit de Poisson lié au comptage des photons venant du fond de l'espace et dans le cas radio, le bruit est modélisé par un bruit blanc gaussien masqué dans l'espace de Fourier, ce qui est équivalent à un bruit coloré gaussien dans l'espace direct. Donc dans la suite, nous allons supposer que b est un bruit gaussien avec une variance Σ dans $\mathbb{R}^{n \times n}$. Pour l'imagerie optique, cette approximation est satisfaisante comme le garantit le théorème central limite en probabilité.

Nous pouvons maintenant définir la fonction de coût L_0 que l'Algorithme de Restauration Parcimonieuse (Sparse Restoration Algorithm (SRA), en anglais) minimise :

$$L_0(x) = \underbrace{\frac{1}{2} \|x * h - y\|_2^2}_{\text{attache aux données}, :=L_{0d}(x), \text{ différentiable}} + \underbrace{\|\lambda_0 \odot \phi(x)\|_1 + \iota_+(x)}_{\text{parcimonie} \quad \text{positivité}, :=L_{0p}(x), \text{ non-différentiable}}, \quad (4.26)$$

avec $\|\cdot\|_1$ la norme ℓ_1 , $\phi = \{\phi_i\}_{i \in \{1, \dots, l\}}$ l'opérateur de transformée en starlet tel que l le nombre de composantes choisi, λ_0 la matrice de pondération à coefficients positifs et ι_+ la fonction indicatrice de Moreau sur l'ensemble des vecteurs dans \mathbb{R}^n à coefficients positifs tel que :

$$\iota_+(x) = \begin{cases} 0 & \text{si } x \in \mathbb{R}_+^n \\ +\infty & \text{sinon.} \end{cases} \quad (4.27)$$

Nous notons alors L_{0d} la partie différentiable de la fonction de coût. Cette partie est égale au terme d'attache aux données. Et nous notons L_{0p} la partie non-différentiable qui est égale à la somme des termes de parcimonie et de positivité. La norme ℓ_1 est utilisée pour la parcimonie au lieu de la norme ℓ_0 parce que la première est convexe et satisfait les conditions de convergence de l'algorithme FBS.

Nous pouvons maintenant commencer à donner les propriétés de L_0 qui sont utiles pour la construction de l'algorithme. Ainsi le gradient de sa partie différentiable s'écrit :

$$\nabla L_{0d}(x) = h_\pi * (x * h - y) \quad (4.28)$$

avec h_π le centro-symétrique de h . La constante de Lipschitz du gradient, notée α_0 , est calculée à partir de sa définition telle que :

$$\alpha_0 = \rho(h_\pi * h * I_{n^2}) \quad (4.29)$$

avec ρ l'opérateur de la norme spectrale. La matrice de pondération λ_0 est choisie de sorte à ce qu'elle soit proportionnelle à la carte de bruit de $\phi \nabla L_{0d}(x_T)$ [Starck, Murtagh et Fadili 2015]. Pour la calculer commençons par remarquer que pour $x = x_T$, nous avons $x * h - y$

égale à $-b$ qui est également un bruit gaussien de variance Σ . Par conséquent :

$$\forall i \in \{1, \dots, l\}, \quad \phi_i * \nabla L_{0d}(x_T) = -\phi_i * h_\pi * b \quad . \quad (4.30)$$

Il en découle que $\phi \nabla L_{0d}(x_T)$ est un bruit gaussien coloré de variance Σ_0 égale à :

$$\Sigma_0 = \underbrace{\left\{ (\phi_i * h_\pi * I_{n^2}) \Sigma (\phi_i * h_\pi * I_{n^2})^\top \right\}_{i \in \{1, \dots, l\}}}_{:= (\Sigma_{0i})_{i \in \{1, \dots, l\}}} \quad . \quad (4.31)$$

Nous pouvons alors exprimer λ_0 en prenant la diagonale des $\{\Sigma_{0i}\}_{i \in \{1, \dots, l\}}$ correspondant à la carte de bruit de $\phi_i * \nabla L_{0d}(x_T)$:

$$\lambda_0 = \{\kappa[i] \cdot \text{diag}(\Sigma_{0i})\}_{i \in \{1, \dots, l\}} \quad (4.32)$$

avec κ un vecteur dans \mathbb{R}^l de la forme $(0, q, \dots, q, q+1)$, en supposant que les composantes de ϕ sont rangées par ordre croissant en allant de l'échelle grossière ϕ_1 à l'échelle la plus fine ϕ_l . Dans le cadre de notre travail, nous avons choisi $q = 4$.

De plus, notons que dans le cas où b est un bruit blanc gaussien de variance $\sigma^2 I_n$, l'expression de λ_0 se simplifie et devient :

$$\lambda_0 = \sigma^2 \left\{ \kappa[i] \cdot \|\phi_i * h_\pi\|_2^2 \mathbf{1}_{n^2} \right\}_{i \in \{1, \dots, l\}} \quad . \quad (4.33)$$

Finalement, nous allons donner une approximation de l'opérateur proximal de L_{0p} , la partie non-différentiable de la fonction de coût. Pour ce faire, nous commençons par donner l'expression exacte des opérateurs proximaux de la fonction indicatrice ι_+ et de la norme ℓ_1 repondérée, $\|\lambda \odot \cdot\|_1$ avec $\lambda \in \mathbb{R}_+^{(l+1) \times n \times n}$:

$$\text{prox}_{\iota_+}(x) := (x)_+ = \{\max(x[k], 0)\}_{k \in \{1, \dots, n^2\}} \quad (4.34)$$

et

$$\text{prox}_{\|\lambda \odot \cdot\|_1}(x) = \text{ST}_\lambda(x) \quad , \quad (4.35)$$

où ST_λ est l'opérateur de seuillage doux (Soft Thresholding, en anglais) défini pour tout k dans $\{1, \dots, (l+1)n^2\}$ comme :

$$\text{ST}_\lambda(x)[k] = \begin{cases} x[k] - \text{sgn}(x[k]) \lambda[k] & \text{si } |x[k]| \geq |\lambda[k]| \\ 0 & \text{sinon.} \end{cases} \quad (4.36)$$

Néanmoins en pratique, l'opérateur de seuillage dur, HT_λ est utilisé au lieu de l'opérateur de seuillage doux puisqu'il a été montré que son utilisation mène à un biais plus faible dans l'image restaurée [Blumensath et Davies 2009 ; Starck 2002]. L'expression de l'opérateur de seuillage dur est donnée par l'équation 3.3.

Maintenant nous pouvons définir l'approximation, ρ_{λ_0} , de l'opérateur proximal de L_{0p} , noté $\text{prox}_{L_{0p}}$:

$$\rho_{\lambda_0}(x) = \left[\tilde{\phi}(\text{HT}_{\lambda_0}[\phi(x)]) \right]_+ \quad , \quad (4.37)$$

avec $\tilde{\phi}$ l'opérateur de reconstruction des starlets. Nous utilisons cette approximation dans l'algorithme SRA. Et il s'agit d'une approximation pour deux raisons. La première est que nous avons utilisé $\tilde{\phi}$ au lieu de ϕ^\top or la starlet est une transformée redondante donc non-orthogonale ce qui veut dire que $\tilde{\phi} \neq \phi^\top$ (et $\tilde{\phi}$ n'est pas unique). La deuxième est le fait

de supposer que l'opérateur proximal de la somme est égal à la composition des opérateurs proximaux, ce qui n'est pas vrai en général.

L'implémentation de SRA que nous avons effectuée est celle d'un algorithme de gradient proximal appartenant à la famille des algorithmes proximaux [Starck, Murtagh et Fadili 2015]. Nous obtenons alors l'algorithme 4.1. Mais il faut encore définir le critère d'arrêt A , l'initialisation t dans $\mathbb{R}^{n \times n}$, α_ϵ et $\hat{\lambda}_0$ sont les estimations respectives de α_0 et λ_0 .

Algorithme 4.1 Sparse Restoration Algorithm (SRA)

tâche : reconstruire x_T à partir de y et h .

paramètres : $\epsilon > 0$, A booléen.

initialisation : $x^{(0)} \leftarrow t$, $\beta \leftarrow \alpha_\epsilon^{-1}$, $i \leftarrow 0$.

tant que non(A) **faire**

$$x^{(i+1)} \leftarrow p_{\beta\hat{\lambda}} \left[x^{(i)} - \beta \nabla L_{0d} \left(x^{(i)} \right) \right]$$

$$i \leftarrow i + 1$$

fin tant que

renvoyer : $x^{(i)}$

Nous calculons α_ϵ en utilisant la méthode de la puissance itérée pour obtenir une valeur que nous multiplions ensuite par $(1 + \epsilon)$ pour avoir une estimation de α_0 qui lui soit supérieure pour que l'algorithme ne diverge pas. Dans notre travail, nous avons choisi $\epsilon = 5 \cdot 10^{-2}$. Pour calculer $\hat{\lambda}_0$, nous utilisons la formule dans l'équation 4.32 mais il est également intéressant de noter que nous pouvons directement estimer la carte de bruit en utilisant la méthode de la valeur absolue des écarts (Mean Absolute Deviation (MAD), en anglais). Et l'initialisation t est réglée à $\frac{1}{n^2} \mathbf{1}_{n^2}$. Pour le critère d'arrêt, nous considérons deux cas :

- *débruitage* ($h = \delta$) : dans ce cas, le problème est bien conditionné alors nous nous contentons de définir A comme l'assertion " $i \leq N_i$ " avec N_i le nombre d'itérations. Pour notre travail, nous avons réglé N_i à 40.
- *cas général* : dans ce cas, le problème est mal conditionné, ce qui nous pousse à définir comme la conjonction de " $i \leq N_i$ " et :

$$\left| \frac{[L(x^{(i)}) + L(x^{(i-1)})] - [L(x^{(i-2)}) + L(x^{(i-3)})]}{L(x^{(i-2)}) + L(x^{(i-3)})} \right| \leq c \quad . \quad (4.38)$$

Pour notre travail, nous avons réglé N_i à 150 et c à 10^{-6} , ces valeurs ont été déterminées empiriquement en considérant un sous-ensemble du jeu de données COSMOS paramétrique.

Maintenant nous avons construit l'algorithme SRA qui permet de reconstruire les images avec une erreur pixellique faible mais sans garantie sur la préservation de la forme. Pour cela dans la section suivante nous le modifions en y ajoutant la contrainte de forme pour obtenir l'algorithme SCORE.

4.2.2 SCORE

Pour commencer à construire l'algorithme SCORE, commençons par formuler sa fonction de coût, notée L , en reprenant la fonction de coût L_0 de SRA à l'équation 4.26 et

ajoutons-y la contrainte de forme à l'équation 4.10 :

$$L(x) = \underbrace{\frac{1}{2} \|x * h - y\|_2^2 + \frac{\gamma}{2} M(x)}_{:=L_d(x), \text{ différentiable}} + \underbrace{\|\lambda \odot \phi(x)\|_1 + \iota_+(x)}_{:=L_p(x), \text{ non-différentiable}}, \quad (4.39)$$

contrainte de forme

Nous construisons alors SCORE par analogie avec SRA. Nous avons déjà calculé le gradient de la partie différentiable de la fonction de coût L_d qui correspond à l'équation 4.21 et la constante de Lipschitz du gradient, notée α , dans l'équation 4.22. Il reste alors à déterminer la carte de bruit de $\phi \nabla L_d(x_T)$, nous avons alors :

$$\phi_i * \nabla L_d(x_T) = -\phi * h_\pi * S_b, \quad \forall i \in \{1, \dots, I\}. \quad (4.40)$$

Alors $\phi L_d(x_T)$ est un bruit coloré gaussien de variance Σ_M égale à :

$$\Sigma_M = \underbrace{\left\{ (\phi_i * h_\pi * S) \Sigma (\phi_i * h_\pi * S)^\top \right\}_{i \in \{1, \dots, I\}}}_{:= (\Sigma_{Mi})_{i \in \{1, \dots, I\}}}. \quad (4.41)$$

Dans ce cas, la carte de bruit correspond à la diagonale des matrices de Σ_M et λ est alors exprimée comme :

$$\lambda = [\kappa[i] \cdot \text{diag}(\Sigma_{Mi})]_{i \in \{1, \dots, I\}} \quad (4.42)$$

Comme la partie non différentiable de la fonction de coût est la même que celle de la fonction de coût de SRA, son opérateur proximal et son approximation restent les mêmes. Nous implémentons alors SCORE avec l'algorithme 4.2. Il nous reste encore à déterminer le critère d'arrêt A , l'initialisation t dans $\mathbb{R}^{n \times n}$ et α_ϵ et $\hat{\lambda}$ qui sont respectivement les estimations de α et λ . Identiquement à l'implémentation de l'algorithme SRA, nous calculons α_ϵ en utilisant

Algorithme 4.2 Shape CONstraint REStoration (SCORE)

tâche : Reconstruire x_T à partir de y et h .

paramètres : $\gamma, \epsilon > 0, A$ booléen, $i \leftarrow 0$.

initialisation : $x^{(0)} \leftarrow t, \beta \leftarrow \alpha_\epsilon^{-1}$

tant que not(A) **faire**

$$x^{(i+1)} \leftarrow p_{\beta \hat{\lambda}} \left[x^{(i)} - \beta \nabla L_d(x^{(i)}) \right]$$

$$i \leftarrow i + 1$$

fin tant que

renvoyer : $x^{(i)}$

la méthode de la puissance itérée dont la sortie est multipliée par $(1 + \epsilon)$ pour garantir que l'estimation est supérieure à α afin que l'algorithme ne diverge pas. Dans notre cadre de travail, nous avons choisi $\epsilon = 5 \cdot 10^{-2}$. Pour les autres variables, nous considérons à nouveau les deux cas :

- *débruitage* ($h = \delta$) : dans ce cas, le problème est bien conditionné alors nous nous contentons de définir A comme l'assertion " $i \leq N_i$ " avec N_i le nombre d'itérations. Pour notre travail, nous avons réglé N_i à 40. Et l'initialisation t est égale à la sortie de l'algorithme SRA. Finalement, nous calculons $\hat{\lambda}$ en utilisant directement la formule dans l'équation 4.42.

- *cas général* : dans ce cas, le problème est mal conditionné, ce qui nous pousse à définir comme la conjonction de " $i \leq N_i$ " et :

$$\left| \frac{[L(x^{(i)}) + L(x^{(i-1)})] - [L(x^{(i-2)}) + L(x^{(i-3)})]}{L(x^{(i-2)}) + L(x^{(i-3)})} \right| \leq c \quad . \quad (4.43)$$

Pour calculer $\hat{\lambda}$, nous générons g carte de bruit de gaussien de variance Σ que nous propageons ensuite sur $\phi \nabla L_d$ et estimons ensuite la carte de bruit résultante. L'initialisation t est une constante égale à l'inverse du nombre de pixels dans l'image. Pour notre travail, nous avons réglé N_i à 150, c à 10^{-6} et g à 100.

Nous avons également ajouté une étape de post-traitement optionnelle à SCORE pour le cas où les images à reconstruire possèdent un seul objet tels que son support est connexe. Cette étape consiste alors à supprimer tous les autres pixels qui ne sont pas connectés à l'objet. Pour ce faire, nous créons un masque binaire qui correspond au seuillage de l'image de sortie par le 80^e centile de l'image de sortie. Nous obtenons alors les régions connectées auxquelles nous enlevons la région correspondant à l'objet d'intérêt et nous appliquons ensuite le négatif de ce masque pour masquer les pixels qui ne sont pas connectés à l'objet d'intérêt.

Au final nous avons construit l'agorithme SCORE que nous pouvons comparer ensuite à l'algorithme SRA pour mieux cerner l'effet de l'ajout de la contrainte de forme.

4.3 Expériences numériques

Dans cette section, nous effectuons des expériences numériques sur le jeu de données paramétrique présenté dans la sous-section 1.1.3. Nous commençons tout d'abord par décrire les détails de l'implémentation pour ensuite parler des résultats obtenus en débruitage et en déconvolution.

4.3.1 Implémentation

L'implémentation a été effectuée en utilisant Python 3.6.8, ModOpt 1.3.0¹, Alpha-Transform² et Matplotlib [Hunter 2007]. Dans un esprit de recherche répétable et reproductible, tous les codes et les résultats obtenus sont publiquement disponibles sur GitHub avec le lien suivant : <https://github.com/CosmoStat/score>.

De plus, pour déterminer la valeur du scalaire γ utilisé pour la pondération de la contrainte de forme dans l'équation 4.39, nous avons effectué une recherche linéaire (line search) en deux étapes pour trouver γ_* qui minimise le critère δ_e suivant un niveau de RSB. La première étape consiste à rechercher l'ordre de grandeur optimal et la deuxième consiste à rechercher la valeur optimale dans l'ordre de grandeur trouvé. Nous définissons alors δ_e comme suit :

$$\gamma_* = \underset{\gamma}{\operatorname{argmin}} \delta_e(\gamma) \quad , \quad (4.44)$$

avec δ_e l'erreur moyenne de la MSE de l'ellipticité sur l'ensemble des images :

$$\delta_e(\gamma) = \operatorname{mean}_{1 \leq i \leq 300} [\operatorname{MSE}(e(\hat{x}_{\gamma,i}))] \quad , \quad (4.45)$$

1. <https://github.com/CEA-COSMIC/ModOpt>

2. <https://github.com/dedale-fet/alpha-transform>

tel que $\text{mean}_{i \in K}(u_i)$ est la moyenne des éléments $\{u_i\}_{i \in K}$ et $\hat{x}_{\gamma,i}$ est la reconstruction obtenue par SCORE pour la i^{e} image avec une pondération de γ . Nous obtenons alors les valeurs de γ_* par niveau de RSB comme montré dans le tableau 4.1. Nous remarquons alors que la valeur de γ_{ast} est proche de 1 dans tous les cas considérés.

RSB	Débruitage	Déconvolution
40	1.2	1.2
75	0.8	1.6
150	1.0	1.2
380	0.8	0.6

Table 4.1 – Valeurs de γ_* obtenues par niveau de RSB sur le catalogue COSMOS paramétrique.

4.3.2 Résultats

Vu que le cas particulier de débruitage correspond à un problème dont le conditionnement est meilleur que dans le cas général de déconvolution, nous avons effectué des expériences pour chacun de ces cas.

D'une part, en débruitage, le noyau de convolution se réduit à une fonction de Dirac, $h = \delta$, dans ce cas le problème se simplifie en :

$$y = x_T + b \quad . \quad (4.46)$$

Qualitativement, dans la ligne du haut de la figure 4.3 montre un exemple d'une image de galaxie vérité terrain et son observation bruitée, la ligne du centre montre les images débruitées par SCORE et SRA et la ligne du bas montre les images résiduelles correspondantes. Quantitativement, la figure 4.4 montre les erreurs pixellique et de forme en fonction du RSB. Nous pouvons remarquer que SCORE conduit à une légère augmentation de l'erreur pixellique comparé à SRA. Cet effet n'est pas inattendu puisque le terme d'attache aux données du dernier s'exprime intégralement dans l'espace des pixels tandis que celui de SCORE est partagé avec l'espace des produits scalaires correspondant à la contrainte de forme comme soulevé dans la section 4.1.3. Par contre, en passant de SRA à SCORE, l'erreur d'ellipticité est significativement réduite, d'un facteur de 2 environ. Ainsi dans ce cas, l'ajout de la contrainte de forme crée un compromis entre l'erreur pixellique et l'erreur sur la forme.

D'autre part, en déconvolution, nous montrons également les performances qualitatives en prenant l'exemple d'une galaxie dans la figure 4.5 pour montrer les reconstructions avec les deux approches et les résidus correspondants. Également, dans la figure 4.6, nous montrons la distribution des erreurs pixellique et d'ellipticité suivant le RSB.

Toujours en déconvolution, nous remarquons que les performances de SCORE sont meilleures que celles de SRA pour les deux erreurs pixellique et de forme. En effet, la MSE obtenue par SCORE est plus que celle de SRA d'au moins 16% et d'au plus 36,3%. L'exemple dans la figure 4.5 montre que la reconstruction donnée par SCORE est plus lisse, avec une meilleure reconstruction de la queue de la galaxie en comparaison avec SRA. De plus, ces constats sont corroborés par les résidus puisque dans le cas de SCORE, nous pouvons voir que la région correspondant au centre de l'objet a une intensité plus faible en comparaison avec SRA.

En termes de pixels, nous observons des tendances différentes entre le cas de débruitage et celui de déconvolution. Nous suspectons que cela soit dû à la différence de conditionnement des deux problèmes. Effectivement, comme mentionné précédemment, le problème de déconvolution est plus mal conditionné que le cas plus simple de débruitage. Par conséquent, l'espace des solutions est plus large donc il est plus probable dans ce cas que l'ajout d'une contrainte rapproche la solution trouvée de la vérité terrain.

En termes d'ellipticité, non seulement l'erreur d'ellipticité δ_e de SCORE est plus faible que celle de SRA mais il semblerait également que l'ellipticité mesurée sur les reconstructions de SCORE soit moins biaisée et plus consistante suivant les barres d'erreur. Dans le cas du débruitage, l'erreur est plus faible de 44,1% au moins et de 70,3% au plus et dans le cas de la déconvolution de 49,5% au moins et 62,3% au plus. Les figures 4.3 et 4.6 montrent que le profil de la galaxie et sa forme sont mieux conservés avec SCORE en comparaison avec SRA.

4.3.3 Conclusion

Au final, nous avons proposé un nouveau terme de régularisation fondé sur les moments géométriques d'ordre 2 afin de mieux préserver la forme des galaxies durant l'étape de restauration. Nous avons montré que notre contrainte de forme est peut-être facilement intégrée à un algorithme de restauration parcimonieuse, menant à la nouvelle méthode SCORE. Pour le débruitage, en comparaison avec la reconstruction parcimonieuse, nous avons montré que l'ajout de la contrainte de forme mène à un compromis entre l'erreur pixellique et l'erreur sur la forme de la galaxie où la dernière est réduite par au moins 44,1%, tandis que l'erreur pixellique augmente d'au plus 28,5%. Pour la déconvolution, les deux erreurs baissent, par au moins 49,5% pour l'erreur d'ellipticité et 16,9% pour l'erreur pixellique. Nous pensons que la différence de tendance entre les cas de débruitage et de déconvolution est due au fait que l'espace des solutions potentielles est plus large dans le cas de la déconvolution alors le terme de régularisation de forme ajouté aide à mieux contraindre le problème inverse.

De plus, SCORE contient uniquement un seul paramètre libre qui n'est autre que la pondération entre le terme d'attache aux données et la contrainte de forme, γ .

Dans ce travail, nous avons utilisé un algorithme de déconvolution parcimonieuse fondé sur la descente de gradient proximal pour sa simplicité mais il est possible d'utiliser des algorithmes plus récents et plus élaborés comme Condat-Vu [Condat 2013; Vũ 2013]. La contrainte de forme peut également être utilisée dans d'autres techniques de déconvolution existantes comme la méthode de minimisation de la variation totale [Weiss, Blanc-Féraud et Aubert 2009] ou d'apprentissage profond [Jeffrey et al. 2020; Ramzi et al. 2021b; Sureau, Lechat et Starck 2020]. Finalement, la contrainte de forme peut aussi être améliorée en utilisant des moments d'ordre supérieur mais également de différents espaces [Flusser, Suk et Zitova 2016; Kostková et al. 2019; Shakibaei et Flusser 2014; Wojak, Angelini et Bloch 2010].

Dans le chapitre suivant, nous comparons SRA et SCORE à des méthodes d'apprentissage profond.

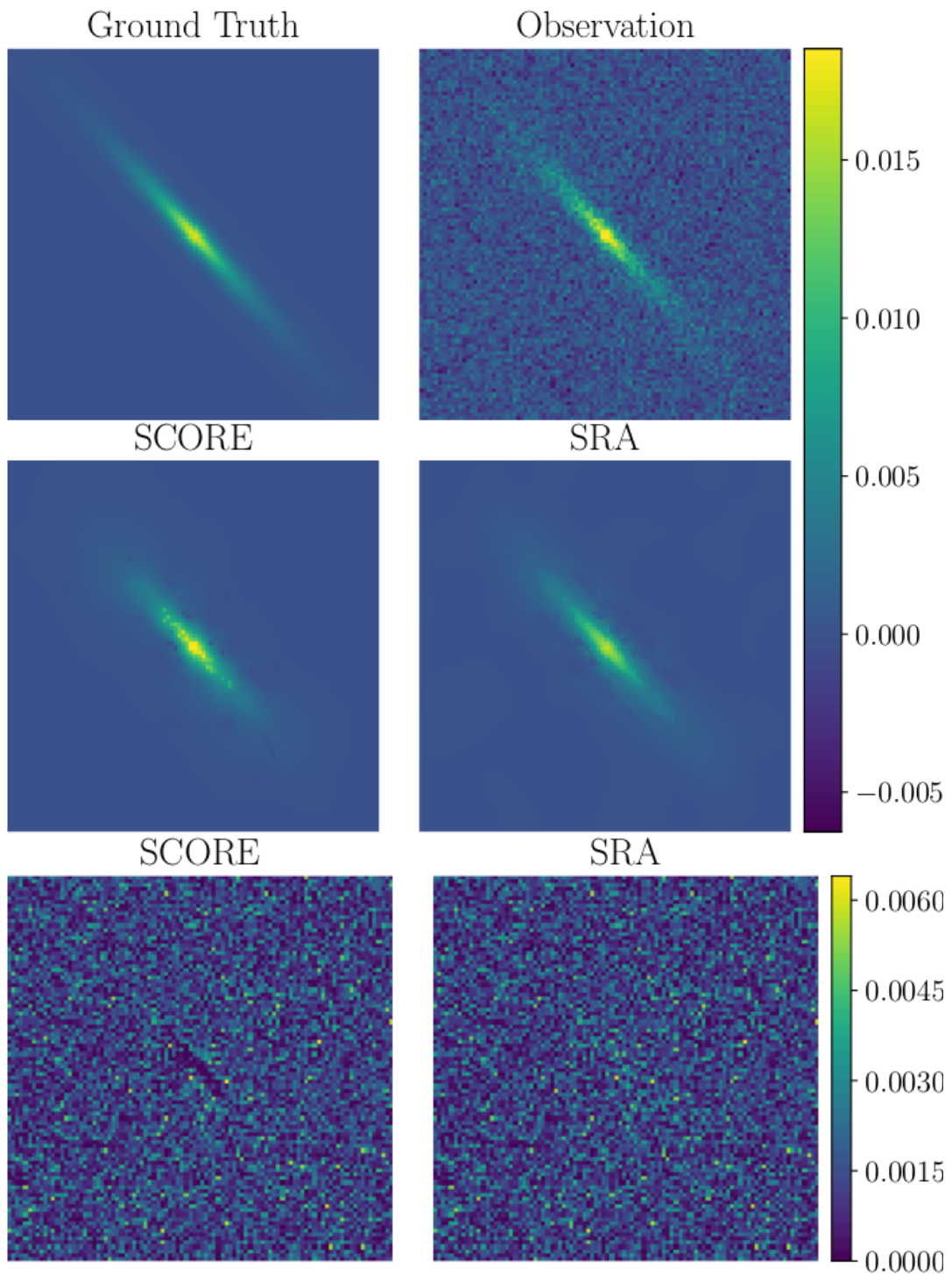


Figure 4.3 – Résultats de débruitage de la galaxie #38 pour un RSB de 75. Haut : image vérité terrain et image observée (image bruitée). Centre : images débruitées avec SCORE et SRA. Bas : images résiduelles avec SCORE et SRA.

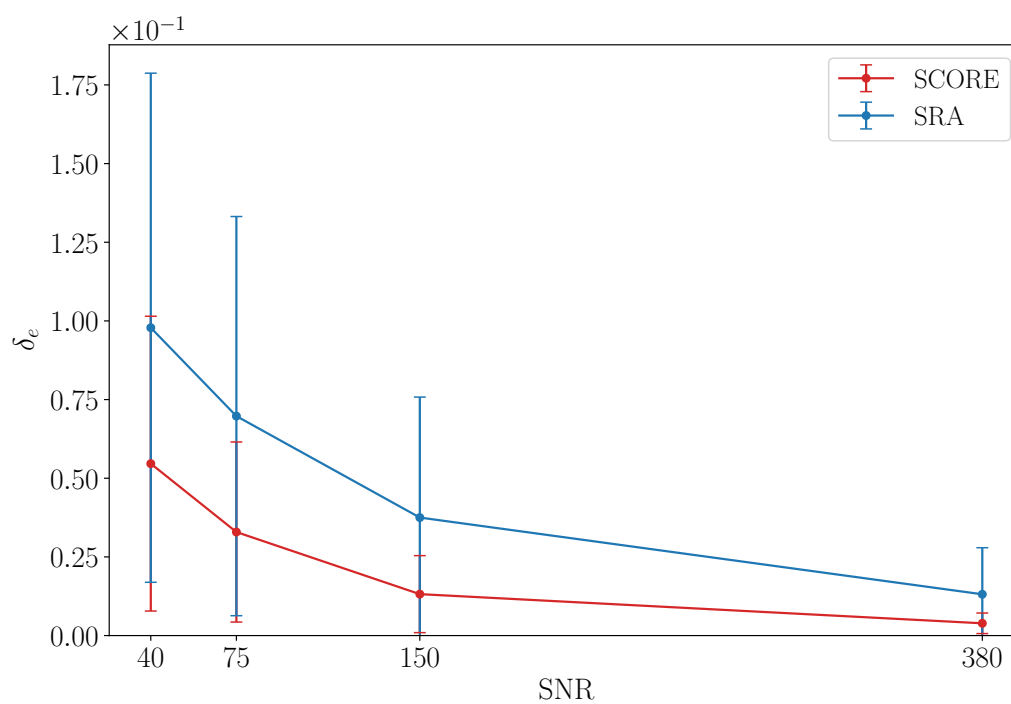
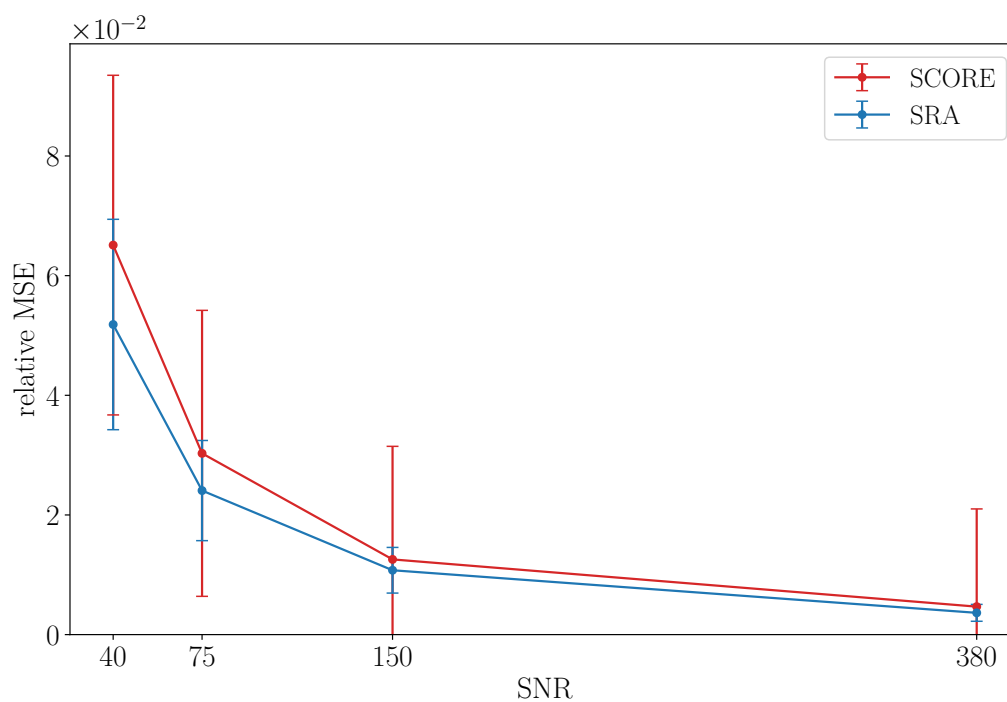


Figure 4.4 – Gauche : la MSE relative des galaxies par niveau de RSB pour l'expérience de débruitage. Droite : l'erreur d'ellipticité δ_e par niveau de RSB. Dans les deux cas, les courbes correspondent à la moyenne par SNR et les barres verticales à l'écart-type.

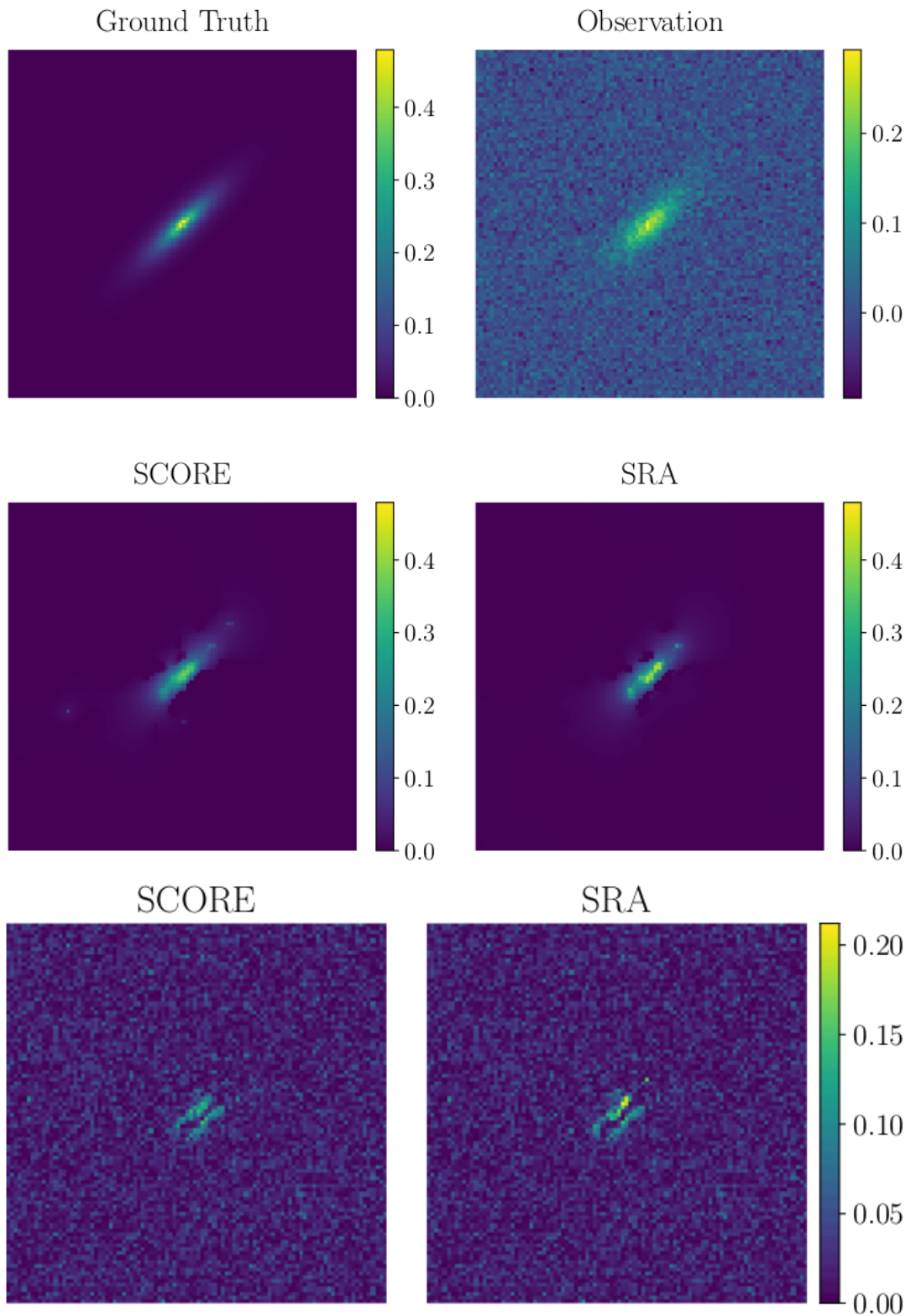


Figure 4.5 – Résultats de déconvolution de la galaxie #16 pour un RSB à 75. Haut : image vérité terrain et image observée (image floutée et bruitée). Centre : images déconvoluées avec SCORE et SRA. Bas : images résiduelles avec SCORE et SRA, utilisant la même barre de couleur.

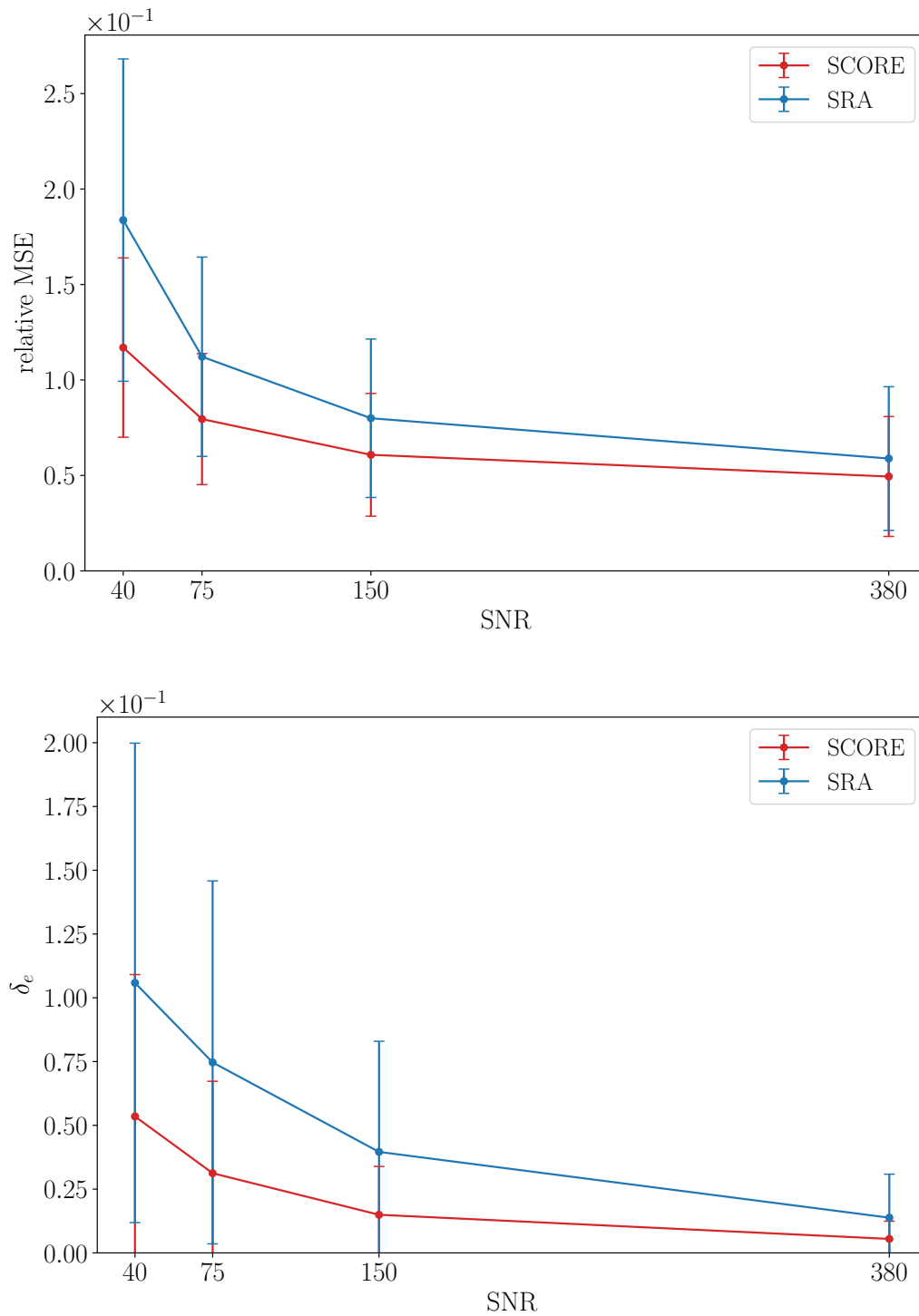


Figure 4.6 – Idem que la figure 4.4, pour l'expérience de déconvolution.

Chapitre 5

Reconstruction avec Apprentissage Profond

Dans ce chapitre nous introduisons l'apprentissage machine pour détailler ensuite l'apprentissage profond représentant tous deux un enjeu majeur du traitement du signal moderne. Nous étudions ensuite des méthodes de reconstruction basées sur cet apprentissage profond.

5.1 Apprentissage Machine

L'Intelligence Artificielle (IA) a été définie par Alan Turing comme la capacité d'une machine à réfléchir par elle-même [Turing 1950]. Il existe deux approches pour cette fin, la première consiste à développer un outil capable de produire de nouvelles informations en utilisant un moteur d'inférence avec une base de règles et une base de faits, il s'agit d'un *système expert* comme DENDRAL [Lindsay et al. 1993]. Cette approche nécessite qu'un opérateur humain développe le système en recouvrant à l'avance tous les cas de figure possibles et qu'il puisse être capable d'expliquer sous forme de code la tâche à effectuer. Ainsi un système expert est adapté aux problèmes ayant un environnement contrôlé où les tâches sont simples à écrire formellement. Par exemple, pour calculer les impôts d'un ménage, un système expert est une solution efficace au problème. Par contre, pour reconnaître une galaxie dans une image, il faut qu'un opérateur humain puisse définir formellement une galaxie et donner explicitement les étapes nécessaires pour reconnaître la galaxie quelque soit sa forme, son orientation, sa taille et sa position dans l'image.

La deuxième approche en IA est plus inspirée du biomimétisme et vise à recopier la faculté du vivant à *apprendre et généraliser* à partir d'un ensemble d'exemples : c'est l'*apprentissage machine* (Machine Learning (ML), en anglais). Cette approche ne requiert pas d'expertise humaine, ni une explication formelle sur le déroulement de la tâche. De plus, elle permet de manipuler des *données massives* (Big Data, en anglais) et d'automatiquement personnaliser la solution si nécessaire. L'apprentissage machine consiste à apprendre à un algorithme à effectuer une tâche donnée suite à une étape *d'entraînement* sur des exemples. Il existe différentes façons d'effectuer l'entraînement, divisées en trois branches majeures :

- *apprentissage supervisé* : dans ce cas les exemples correspondent chacun à une entrée et une sortie attendue. L'algorithme contient des paramètres qui sont ajustés durant la phase d'entraînement grâce à la minimisation d'une fonction de coût qui pénalise l'erreur entre la sortie de l'algorithme et la sortie attendue. Les tâches apprises avec l'apprentissage supervisé sont notamment la classification, la régression et aussi la prédiction avec les séries temporelles. Parmi les méthodes basées sur cet apprentissage citons les réseaux de neurones artificiels [Basheer et Hajmeer 2001], les forêts aléatoires [Breiman 2001], les machines à support de vecteur (Support

Vector Machine (SVM), en anglais) [Hearst et al. 1998] et la programmation logique inductive [Muggleton 1991 ; Plotkin 1971].

- *apprentissage non-supervisé* : dans ce cas les exemples correspondent uniquement aux entrées de l'algorithme. Le but est de pouvoir organiser les données et capturer leur structure. Généralement, l'algorithme minimise une fonction de coût qui pénalise la distance entre les données qui ont des composantes similaires. Les tâches apprises avec l'apprentissage non-supervisé sont notamment le regroupement, l'estimation de densité et la réduction de dimension. Parmi les méthodes basées sur cet apprentissage citons la classification hiérarchique [Patel, Sihmar et Jatain 2015], les k -moyennes (k -means, en anglais) [Steinhaus 1956] et les modèles de mélange (mixture models, en anglais) [McLachlan, Lee et Rathnayake 2019 ; Reynolds 2009].
- *apprentissage par renforcement* : dans ce cas, l'algorithme correspond à un agent qui évolue dans un environnement séquentiel. À chaque étape, l'agent effectue des actions qui sont la sortie d'une fonction appelée *politique* (policy, en anglais), qui prend en entrée son état présent et des états précédents. Les actions prises à chaque étape sont alors évaluées par une métrique appelée *récompense*. Le *gain* pour une étape donnée est défini comme la combinaison de la récompense présente et des récompenses précédentes. L'entraînement de l'agent se fait à partir de plusieurs séries d'actions effectuées, appelées *épisodes*, à partir desquels la politique de l'agent est optimisée en maximisant le gain. L'agent apprend alors des exemples qu'il génère lui-même. La tâche principalement apprise avec l'apprentissage par renforcement est la planification comme la conduite autonome [Sallab et al. 2017], les jeux tels que les échecs [Silver et al. 2018] et les jeux vidéos [Jaderberg et al. 2019 ; Mnih et al. 2013] mais également la locomotion bipède [Benbrahim et Franklin 1997 ; Kidziński et al. 2018]. Les méthodes basées sur cet apprentissage sont notamment le Q-learning [Watkins et Dayan 1992] et le Deep Q-learning [Gu et al. 2016].

Dans le cadre de la thèse, nous nous intéressons à la restauration d'images. Avec le modèle de dégradation, nous pouvons simuler des images vérités terrain et des images observées. Il est alors convenable de considérer les méthodes d'apprentissage supervisées. En particulier, nous allons nous intéresser dans ce chapitre aux réseaux de neurones profonds (Deep Neural Networks (DNN), en anglais). En effet, depuis 2012 les DNN ont marqué leur présence dans l'état de l'art en traitement d'image avec le réseau AlexNet [Krizhevsky, Sutskever et Hinton 2012] en remportant la première classe dans la compétition de reconnaissance d'image ImageNet¹. Dans la suite nous allons présenter plus en détail les DNN.

5.2 Apprentissage Profond

Durant la dernière décennie, la puissance de calcul et la quantité de données massives ont considérablement augmenté [Oussous et al. 2017 ; Sun et al. 2017], ce qui a permis à l'apprentissage profond d'offrir des solutions état de l'art dans plusieurs domaines. Dans cette section, nous allons commencer par définir les réseaux de neurones, ensuite nous allons détailler comment se fait leur apprentissage.

5.2.1 Architecture des réseaux de neurones

Les réseaux de neurones artificiels, également appelés réseaux de neurones, sont des approximateurs universels de fonction comme le garantit le théorème éponyme [Hornik,

1. lien vers la compétition ImageNet 2012 : <https://image-net.org/challenges/LSVRC/2012/>

Stinchcombe et White 1989]. L'idée des réseaux de neurones a émergé au milieu du XX^e siècle en se basant sur des travaux modélisant le fonctionnement du système nerveux [Lettvin et al. 1959]. L'un des premiers réseaux conçus est le *perceptron* [Rosenblatt 1958], dans sa version la plus simple, il s'apparente à un neurone artificiel. Le perceptron est un classifieur linéaire, c'est-à-dire qu'il permet de classifier les ensembles qui sont linéairement séparables. Par exemple, l'opérateur "OU exclusif" qui est non linéaire, ne peut pas être correctement appris par le perceptron simple. L'utilisation de plusieurs neurones organisés en couches permet au perceptron de pallier ce problème [Yanling, Bimin et Zhanrong 2002]. Au niveau théorique, deux autres innovations ont permis d'améliorer la performance des réseaux de neurones et d'obtenir des réseaux profonds : la rétropropagation et l'amélioration de la fonction d'activation. Dans la suite, nous allons introduire l'architecture d'un réseau de neurones en commençant par la base.

Le neurone

Le neurone artificiel est l'élément de base des réseaux de neurones. Pour un signal en entrée $x \in \mathbb{R}^n$, il donne un signal de sortie $y \in \mathbb{R}^m$ comme représenté dans la figure 5.1. Pour chaque composante de y , le neurone possède des poids qui sont multipliés, terme à terme, avec x . Ensuite, la somme de ces produits est effectuée puis seuillée en utilisant un scalaire appelé *biais* comme seuil. Considérons le cas où $m = 1$ et $y \in \mathbb{R}$ alors en notant f

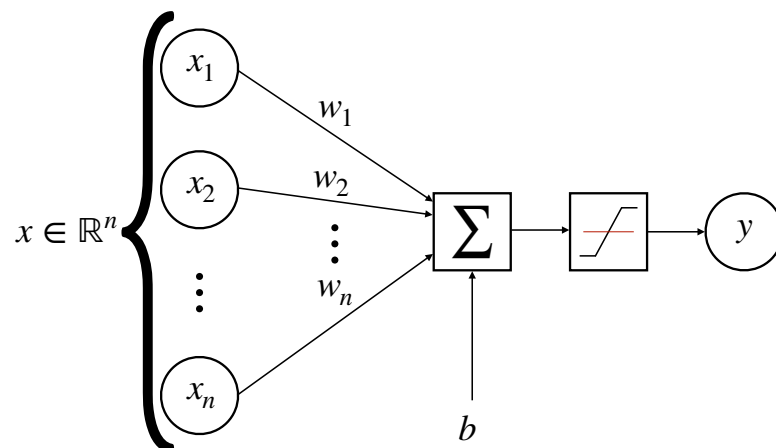


Figure 5.1 – Schéma d'un neurone artificiel. L'entrée x est multipliée par les poids $\{w_i\}_{i \in \{1, \dots, n\}}$ qui sont ensuite sommés et seuillés pour donner la sortie y .

la fonction d'activation qui effectue le seuillage, nous avons :

$$y = f \left(\sum_{i=1}^n x_i w_i + b \right) = f \left(\begin{bmatrix} x \\ 1 \end{bmatrix}^T \begin{bmatrix} w \\ b \end{bmatrix} \right), \quad (5.1)$$

avec $w = \{w_i\}_{i \in \{1, \dots, n\}}$ dans \mathbb{R}^n les poids des entrées du neurone et $b \in \mathbb{R}$ son biais. La fonction d'activation est utilisée pour seuiller la combinaison linéaire de l'entrée et des poids auxquels s'ajoute le biais. Le biais correspond alors au seuil. Il existe plusieurs fonctions d'activation qui sont choisies suivant les propriétés recherchées [Szandala 2020]. Citons trois fonctions qui sont apparues durant les moments clés des réseaux de neurones :

- *Heaviside* : cette fonction a été utilisée dans les premiers réseaux de neurones [Szandala 2020] et s'écrit :

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{sinon,} \end{cases}, \forall x \in \mathbb{R} . \quad (5.2)$$

Cette fonction est non dérivable et l'application de l'algorithme de rétropropagation avec cette fonction est problématique durant l'étape d'apprentissage. D'ailleurs, son utilisation se limite aux problèmes linéairement séparables [Szandala 2020].

- *Sigmoïde* : cette fonction permet d'avoir une sortie non-linéaire et s'écrit :

$$f(x) = \frac{1}{1 + \exp(-x)}, \forall x \in \mathbb{R} . \quad (5.3)$$

De plus, la sigmoïde est dérivable ce qui permet d'utiliser l'algorithme de rétropropagation dans la phase d'apprentissage dans le cas de réseaux de neurones qui ne sont pas profonds. Dans le cas des réseaux de neurones profonds, l'utilisation de la sigmoïde mène souvent au problème de la *disparition du gradient* [Szandala 2020 ; Ven et Lederer 2021]. Ce dernier est un phénomène qui survient lorsque la valeur calculée du gradient est considérablement faible, au point que les valeurs des poids du réseau ne changent plus et risque d'interrompre l'étape d'entraînement (qui est détaillée dans 5.2.2).

- *Rectified Linear Unit (ReLU)* : cette fonction est souvent utilisée en apprentissage profond vu qu'elle permet d'approximer les fonctions linéairement par morceaux, de solliciter la parcimonie dans la représentation des images dans le réseau de neurones et de pallier au problème de disparition du gradient [Szandala 2020]. La fonction s'écrit comme :

$$f(x) = \max(0, x), \forall x \in \mathbb{R} . \quad (5.4)$$

Notons que le fait d'approximer les fonctions linéairement par morceaux lors de l'utilisation de la ReLU comme fonction d'activation, rend cette dernière sous-optimale lorsque le réseau de neurones n'est pas profond et que la fonction à approximer est un sinus par exemple [Szandala 2020]. De plus, au cours des dernières années, plusieurs améliorations ont été apportées à cette fonction comme Leaky ReLU [Maas 2013] et PReLU [He et al. 2015].

Dans la figure 5.2, les trois exemples donnés de fonction d'activation sont représentés. Maintenant passons au cas général avec $y = m$, dans ce cas le schéma dans la figure 5.1 reste valide et la sortie s'écrit en fonction de l'entrée comme :

$$y = f(Wx + b) = f\left([W, b] \begin{bmatrix} x \\ 1 \end{bmatrix}\right), \forall x \in \mathbb{R}^n . \quad (5.5)$$

avec $W \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ et la fonction d'activation f s'obtient à partir de la fonction d'activation 1D, notée f_{1D} , avec :

$$f(x) = \{f_{1D}(x_i)\}_{i \in \{1, \dots, m\}}, \forall x = \{x_i\}_{i \in \{1, \dots, m\}} \in \mathbb{R}^m . \quad (5.6)$$

Sachant que pour tout x dans \mathbb{R}^n l'équation $Wx + b = 0$ représente un sous-espace affine de \mathbb{R}^n donc un neurone permet uniquement de séparer linéairement les entrées x . Un seul neurone permet alors de résoudre des problèmes linéaires. Par contre, une fonction d'activation non linéaire permet d'avoir une sortie non-linéaire en fonction de l'entrée. En utilisant la

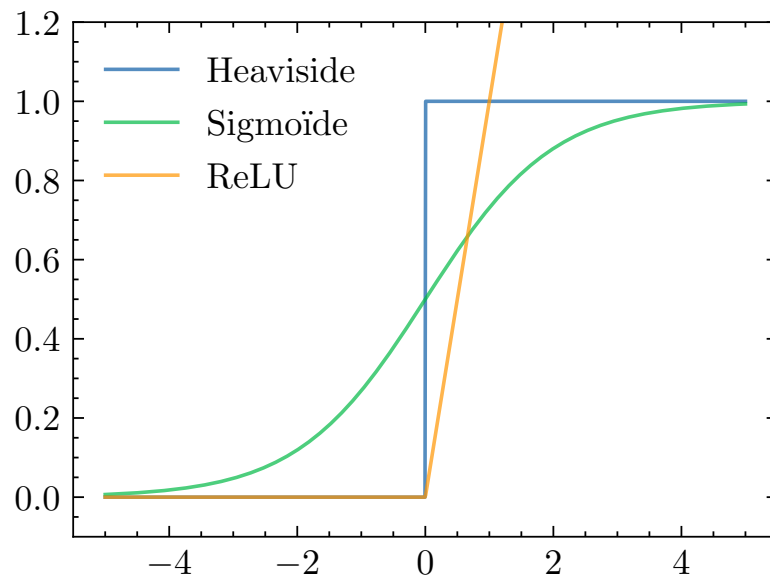


Figure 5.2 – Fonctions d’activation représentées sur l’intervalle [-5;5]. Les fonctions représentées sont la Heaviside, la sigmoïde et la ReLU.

sortie d’un neurone en entrée d’un autre, il devient possible d’avoir des surfaces de séparation non-linéaires des entrées x . Pour reformuler, l’utilisation de plusieurs neurones permet de résoudre les problèmes non-linéaires : c’est le principe d’un réseau de neurones. Dans la suite nous allons nous intéresser à comment les réseaux de neurones sont construits en général.

Réseau de neurones et couches

Il existe deux catégories de réseaux de neurones :

- *Réseaux à propagation avant* : il s’agit de réseaux de neurones qui sont acycliques c’est-à-dire que la sortie d’un neurone n’est pas réinjectée dans l’entrée de celui-ci.
- *Réseaux récurrents* : il s’agit de réseaux de neurones cycliques, par exemple dans la figure 5.3. Ces réseaux sont essentiellement utilisés dans le traitement du langage naturel (Natural Language Processing (NLP), en anglais) [Devlin et al. 2018] et l’apprentissage par renforcement [Li 2017].

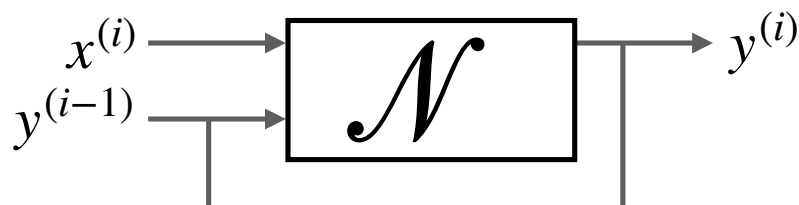


Figure 5.3 – Exemple de réseau cyclique où la sortie est réinjectée en entrée.

Dans la suite de notre étude, nous allons uniquement nous intéresser aux réseaux à propagation avant puisqu’ils sont les réseaux les mieux adaptés au problème de déconvolution. En effet, dans ce dernier, il n’y a pas de relation de récurrence ou de dépendance cyclique entre les variables.

Un réseau de neurones est divisé en *couches*. Une couche correspond généralement à des neurones ayant en entrée la sortie de la couche, voire des couches, précédentes. Les sorties de

chaque couche sont appelées descripteurs ou cartes de descripteurs. Les descripteurs offrent une représentation du signal d'entrée dans un autre espace. La profondeur d'un réseau est alors définie comme le nombre de couches que celui-ci possède. Les couches sont séparées en trois catégories :

- *couche d'entrée* : La couche d'entrée correspond à la couche qui reçoit les données en entrée.
- *couche cachée* : il s'agit des couches entre la couche d'entrée et la couche de sortie. Ces couches contiennent soit des neurones soit des opérateurs appliqués sur la sortie des neurones tels que le sur-échantillonnage ou sous-échantillonnage.
- *couche de sortie* : La couche de sortie est la dernière couche du réseau de neurones et sa sortie correspond à celle du réseau.

D'ailleurs, il existe plusieurs types de couches contenant des neurones. En général, chaque type de couche contient la même structure pour tous les neurones. Ainsi un type de couche est déterminé notamment par les connexions de ses neurones avec les neurones de la couche précédente mais aussi la façon avec laquelle sont combinés les poids du neurone et la fonction d'activation utilisée. Par exemple, une *couche entièrement connectée*, également dite *couche dense*, est une couche où les neurones sont connectés à tous les neurones de la couche précédente. Dans la suite nous allons aborder les réseaux de neurones convolutifs (Convolutional Neural Network (CNN)).

Réseaux de neurones convolutifs

Les CNN sont particulièrement adaptés pour effectuer diverses tâches dans le domaine du traitement de l'image. Ceci s'explique notamment par la couche convolutive utilisée dans ces réseaux. En effet, le signal à l'entrée d'une telle couche est lié aux poids de celle-ci par des opérations de convolution uniquement, ce qui permet l'invariance par rapport à la translation. Cela est avantageux dans le cas de la classification et la détection de motif dans un signal, par exemple. Ainsi avant de donner un exemple d'architecture de CNN, nous allons commencer par décrire la couche convolutive puis les couches de sous-échantillonnage et de sur-échantillonnage.

Une couche convolutive est une couche de neurones où les entrées de chaque neurone partagent les mêmes poids qui correspondent à des filtres de convolution. Dans la figure 5.4, nous donnons le schéma de la première partie d'une couche convolutive avec un seul filtre et le biais associé. Dans le cas général, plusieurs filtres sont utilisés ; pour chaque filtre, nous avons une sortie intermédiaire correspondante identique à celle dans la figure. Un *padding* (rembourrage en français) est également utilisé lors de la convolution pour traiter les bords, le zéro padding est communément utilisé. La sortie de la couche convolutive est un signal possédant autant de canaux que de filtres dans la couche. Soit x dans $\mathbb{R}^{r \times s \times p}$ l'entrée et o dans $\mathbb{R}^{u \times v \times w}$ la sortie d'une couche convolutive alors nous avons :

$$u = \frac{r - u + 2q}{S_x} + 1 \quad , \quad (5.7)$$

$$v = \frac{s - v + 2q}{S_y} + 1 \quad , \quad (5.8)$$

avec q la taille du padding et w est le nombre de filtres. Le nombre de paramètres, noté n_p , dans une couche convolutive est :

$$n_p = (m \cdot n \cdot p + 1) \cdot w \quad . \quad (5.9)$$

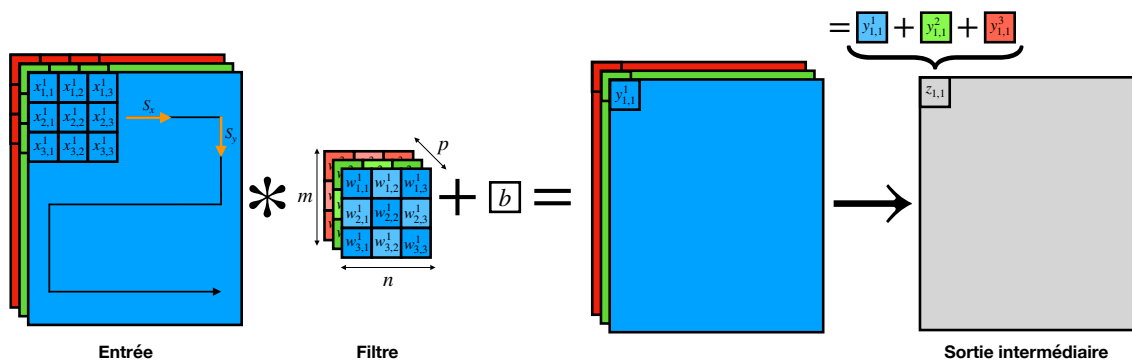


Figure 5.4 – Schéma de la première partie d'une couche convolutive avec un seul filtre. L'entrée dans ce cas est une image composée de $p = 3$ canaux. Le filtre utilisé possède forcément $p = 3$ canaux pour correspondre à l'entrée. Chaque filtre est associé à un biais scalaire et chaque canal du filtre est composé de $m \times n$ poids. S_x et S_y sont respectivement les pas (strides, en anglais) horizontal et vertical avec lequel le filtre parcourt l'entrée. La sortie intermédiaire est obtenue en sommant les valeurs suivant l'axe des canaux. Ensuite le résultat est mis en entrée d'une fonction d'activation. La sortie de la fonction d'activation correspond à la sortie de la couche convolutive.

Pour les signaux de grande taille, une couche convolutive contient considérablement moins de paramètres qu'une couche entièrement connectée. À titre d'exemple, considérons une couche de 5 neurones qui prend en entrée des images de taille 100×100 avec 5 canaux. Dans le cas d'une couche entièrement connectée, nous avons 250,005 paramètres. Dans le cas d'une couche convolutive, en prenant des filtres de taille 5×5 , nous avons 630 paramètres, soit 400 fois moins de paramètres. La présence de moins de paramètres rend l'étape d'apprentissage moins coûteuse en termes de puissance de calcul.

Dans les CNN, l'utilisation des couches de sous-échantillonnage permet d'améliorer le pouvoir de généralisation du réseau et augmente sa robustesse contre la présence de bruit dans le signal d'entrée. Dans la figure 5.5, la fonction f effectue un sous-échantillonnage de l'image a en divisant sa taille par 2 suivant chaque dimension. Il existe plusieurs fonctions pour effectuer le sous-échantillonnage, notamment la moyenne ou le maximum des coefficients en entrée. Si la sortie du réseau est supposée avoir la même taille que l'entrée, alors dans le cas de l'utilisation de couches de sous-échantillonnage, des couches de sur-échantillonnage sont utilisées en contrepartie. Dans la figure 5.5, la fonction g effectue un sur-échantillonnage de l'image b . Parmi les fonctions de sur-échantillonnage, citons la répétition du coefficient et les méthodes d'interpolation.

Nous évoquons succinctement les couches qui sont utilisées pour améliorer la performance du réseau ou pour garantir des propriétés utiles pour le réseau [Ba, Kiros et Hinton 2016; Ioffe et Szegedy 2015; Miyato et al. 2018; Qiao et al. 2019; Salimans et Kingma 2016].

Finalement, nous donnons en exemple, l'architecture du réseau qui sera utilisé dans la suite : le U-net [Ronneberger, Fischer et Brox 2015]. Il s'agit d'un réseau qui est notamment utilisé pour la segmentation et le débruitage. Ce réseau tient son nom de son architecture en forme de "U" comme dans la figure 5.6. Le U-net est divisé en deux parties, la voie contractante et la voie expansive. La voie contractante correspond à la partie du U-net contenant des couches de sous-échantillonnage et la voie expansive correspond à la partie contenant des couches de sur-échantillonnage. Dans un U-net, un étage correspond aux couches convolutives de la voie contractante séparées par une couche de sous-échantillonnage et aux couches

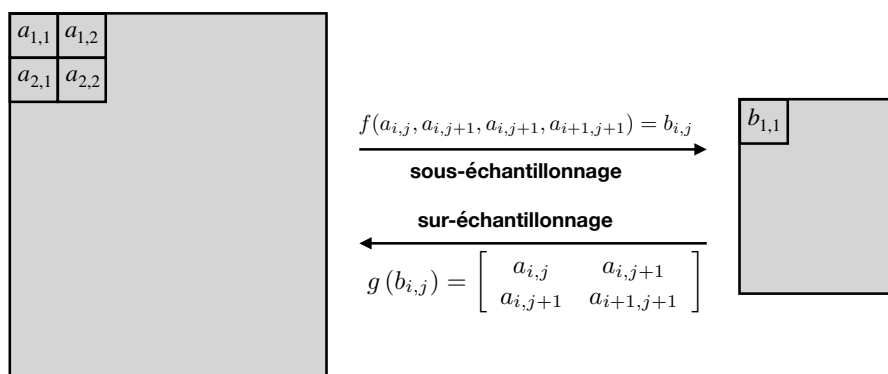


Figure 5.5 – Schéma de sur-échantillonnage et sous-échantillonnage.

convolutives de la voie expansive auxquelles elles sont liées par la connexion skip. Les cartes de descripteurs d'un même étage forment une *échelle*. Par exemple, le U-net dans la figure 5.6 possède 5 étages.

Un CNN profond possède un nombre considérable de paramètres, dans le cas du U-net ce nombre est de l'ordre de 10^7 . Pour régler la valeur de chacun de ces paramètres, une étape d'apprentissage est donc nécessaire.

5.2.2 Entraînement

Les réseaux de neurones profonds contiennent un nombre élevé de paramètres qu'il faut estimer. L'étape d'estimation de ces paramètres est appelée entraînement. Il s'agit dans le cas d'un apprentissage supervisé de donner des couples d'entrée et de sortie au réseau afin d'ajuster la valeur de ses paramètres. Dans la suite, nous allons voir le rôle du jeu de données dans l'entraînement pour ensuite aborder la rétropropagation qui est la méthode utilisée pour effectuer l'entraînement dans le cadre de notre travail.

Jeu de données

Pour l'apprentissage supervisé, chaque élément du jeu de données est constitué d'un couple comprenant le signal d'entrée et d'une cible. En général, dans le cadre de l'apprentissage machine, il est essentiel de noter que les performances de la méthode ne sont pas uniquement liées à l'algorithme utilisé mais qu'elles dépendent également du jeu de données. Plus formellement, le théorème *No Free Lunch* (Pas de Déjeuner Gratuit, en français) [Wolpert et Macready 1997] stipule qu'il n'existe pas d'algorithme ayant une performance supérieure à tout autre algorithme quelque soit le problème et le jeu de données. Ainsi il faut que l'algorithme soit adapté au problème. Puisqu'en apprentissage machine, l'algorithme dépend du jeu de données, il est important que ce dernier soit de qualité. Plus explicitement, les critères recherchés dans un jeu de données sont :

- *la représentativité* : le jeu de données correspond idéalement à un sous-ensemble représentatif des données relatives au problème. Il est important de recouvrir tous les cas envisageables. Effectivement, il n'y a pas de garantie sur la sortie du réseau de neurones si l'entrée n'appartient pas à la distribution sur laquelle il a été entraîné. Inversement, si le jeu de données couvre davantage de cas, c'est-à-dire une distribution plus vaste que la distribution d'intérêt alors d'après le théorème *No Free Lunch*, la performance du réseau risque de se détériorer lorsqu'elle est évaluée

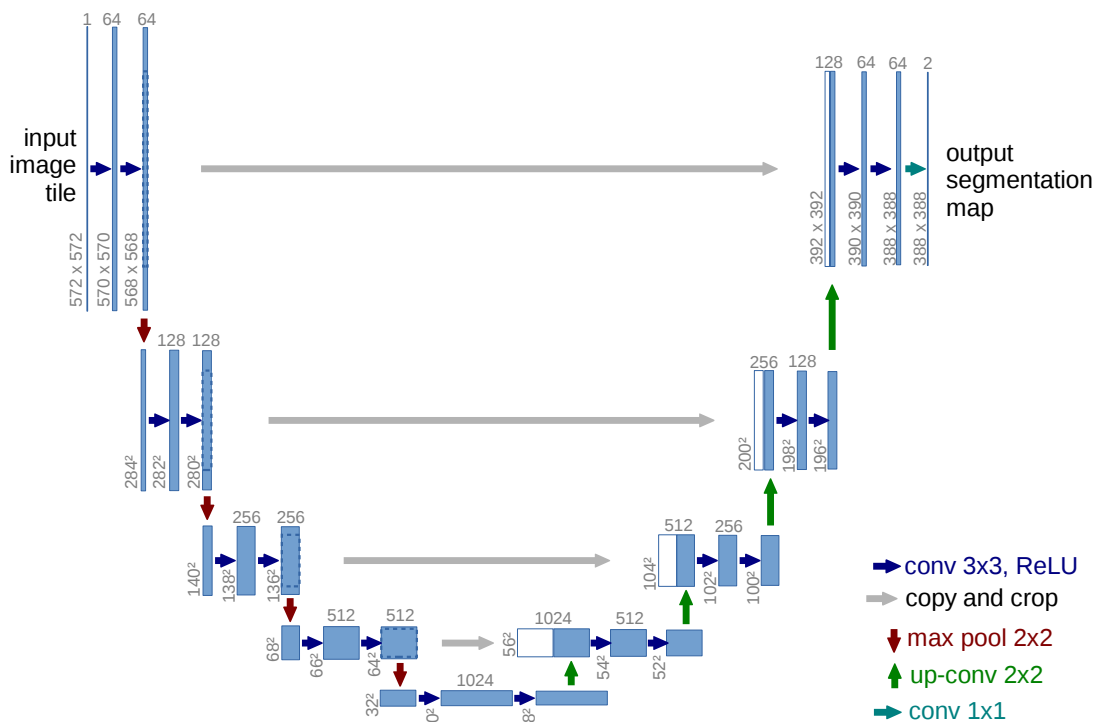


Figure 5.6 – Schéma du U-net. Les rectangles correspondent aux images avec le nombre de canaux indiqué au-dessus et la dimension de l'image indiquée en-dessous. La flèche bleue correspond à une couche convolutive avec des noyaux de taille 3×3 et une fonction d'activation ReLU. La flèche grise correspond à une connexion skip. La flèche rouge correspond à une couche de sous-échantillonnage 2×2 utilisant la fonction qui rend le maximum des coefficients donnés en entrée. La flèche verte correspond à une couche de sur-échantillonnage 2×2 utilisant la convolution (up-convolution, en anglais) [Durall López, Keuper et Keuper 2020]. La flèche en turquoise correspond à une couche de convolution avec un noyau 1×1 . Source [Ronneberger, Fischer et Brox 2015].

sur la distribution d'intérêt, en comparaison avec un réseau entraîné uniquement sur des exemples tirés de cette dernière.

- *la qualité des données* : les cibles servent de référence au réseau pour ajuster ses poids. Si les cibles sont bruitées ou dégradées par un autre processus alors les performances du réseau seront également dégradées. Il est aussi possible d'avoir des données manquantes, par exemple dans le cas des images de galaxies, il se peut que la taille de la galaxie soit supérieure à la taille du support de l'image ainsi l'information sur les bords de la galaxie est incomplète.
- *une plage de valeurs adaptée au réseau* : D'une part, les fonctions d'activation utilisées dans les réseaux de neurones permettent de discriminer efficacement des signaux qui ont une moyenne nulle et un écart-type proche de l'unité. D'autre part, si les données sont constituées de plusieurs grandeurs avec différentes unités alors il est important de normaliser ces grandeurs pour que leurs plages de valeurs soient similaires. En effet, en l'absence de normalisation, l'optimisation des poids du réseau se fera essentiellement sur les grandeurs à plus forte amplitude [Sola et Sevilla 1997].

Concernant le choix de la taille du jeu de données, il existe une règle empirique qui recommande d'utiliser un jeu de données ayant une taille d'un ordre de grandeur de plus que la

dimension de Vapnik-Chervonenkis qui correspond à une mesure de la complexité de la fonction apprise par le réseau de neurones [Hush et Horne 1993]. Une version équivalente de cette règle recommande d'utiliser un jeu de données ayant une taille d'un ordre de grandeur de plus que le nombre de paramètres dans le réseau [Developers 2019]. Il est commun dans le cas réel que la taille maximale du jeu de données soit imposée. Il est alors pertinent d'utiliser des méthodes d'augmentation des données (en effectuant des par exemple des rotations, symétrie et remise à l'échelle, à partir d'un jeu de base) qui soient adaptées au problème [Dyk et Meng 2001 ; Shorten et Khoshgoftaar 2019] et dans certaines situations d'adapter la méthode à la taille du jeu de données pour éviter le sur-entraînement [Hush et Horne 1993].

Dans la suite nous allons voir comment la fonction de coût est choisie pour l'entraînement.

Fonction de coût

En apprentissage, la fonction de coût également appelée *fonction de perte* (loss function, en anglais) est utilisée pour optimiser les poids des réseaux de neurones lors de l'entraînement. En apprentissage machine supervisé, la fonction de coût est utilisée lors de la phase d'entraînement pour régler les poids du réseau de neurones. En notant \mathcal{N}_θ le réseau de neurone, avec θ dans \mathbb{R}^p ses paramètres, y et x dans \mathbb{R}^n respectivement l'observation et la vérité terrain, la fonction de coût, l , s'écrit en général comme :

$$l(\hat{x}) = d(x, \hat{x}) + f(\hat{x}) \quad , \quad (5.10)$$

avec $\hat{x} = \mathcal{N}_\theta(y)$ la solution estimée par le réseau de neurones. La fonction d dans le premier terme, correspond à l'attache aux données qui permet de mesurer l'écart entre \hat{x} et x , il peut s'agir d'une distance, d'une pseudo-norme voire même d'une autre fonction basée sur la différence ou le rapport d'une propriété entre \hat{x} et x . La fonction f dans le deuxième terme correspond aux termes de régularisation comme ceux présentés dans la section 2.2. Notons que la fonction d compare la reconstruction à la vérité terrain et non à l'observation.

Toujours en se basant sur le théorème *No Free Lunch*, il est important de choisir la fonction de coût adaptée au problème afin d'optimiser les performances du réseau. À titre d'exemple, nous donnons les fonctions d suivantes [Janocha et Czarnecki 2017] :

- *l'EQM* : cette fonction pénalise le carré de la distance entre les composantes du signal estimé et la vérité terrain avec :

$$l(\hat{x}) = \frac{1}{n} \sum_{i=1}^n (x[i] - \hat{x}[i])^2 \quad . \quad (5.11)$$

L'EQM est adaptée pour les problèmes de régressions et de reconstruction de signal comme pour le problème de déconvolution.

- *l'entropie croisée* : l'entropie croisée pénalise l'écart entre la distribution estimée et celle de la vérité terrain avec :

$$l(\hat{x}) = - \sum_{i=1}^n \hat{x}[i] \log(x[k]) \quad . \quad (5.12)$$

L'entropie croisée est notamment utilisée pour des tâches de classification et d'estimation de densité de probabilité. En effet, l'inégalité de Gibbs nous garantit que l est minimale lorsque $\hat{x} = x$.

- la fonction de perte de Hinge : elle est utilisée essentiellement pour des problèmes de classification binaire où les deux classes sont codées par $\{-1, 1\}$. Son équation est :

$$l(\hat{x}) = \max_{i \in \{1, \dots, n\}} (0, 1 - x[i] \cdot \hat{x}[i]) \quad (5.13)$$

Notons que cette fonction est également utilisée pour la méthode SVM.

En ce qui concerne le terme de régularisation pour la fonction de coût, les points abordés dans la section 2.2 restent valides. Nous insistons sur le fait que l'utilisation de régularisation durant l'étape d'entraînement permet au réseaux de neurones de mieux généraliser la tâche cible. Par contre, la présence de plusieurs termes de régularisation ou d'une pondération élevée pour ces derniers risque d'augmenter le biais de modélisation.

Lors de l'entraînement du réseau de neurones, les paramètres de celui-ci sont optimisés en minimisant la fonction de coût, cette étape est détaillée dans la partie suivante.

Optimisation

Les réseaux de neurones possèdent un nombre considérable de paramètres à régler. À cette fin, les paramètres sont estimés durant la phase d'entraînement à partir d'un jeu de données. Le jeu de données dédié à l'entraînement est alors partitionné en plusieurs sous-ensembles de taille égale, chaque sous-ensemble est appelé *batch* (lot, en anglais). Généralement, l'entraînement se fait en itérant sur le jeu de données en utilisant deux boucles imbriquées. Dans une boucle dite majeure, chaque itération est appelée *époque* et réutilise l'ensemble du jeu de données. La boucle mineure correspond à une époque et consiste à itérer sur les batches. À chaque itération de la boucle mineure, les paramètres du réseau de neurones sont mis à jour. Dans la littérature il existe deux façons répandues pour ce faire : les algorithmes génétiques [Lam et al. 2001] et les méthodes de descente de gradient.

Dans le cadre de notre travail, nous nous limitons uniquement au cas des méthodes basées sur la descente de gradient puisqu'elles sont le plus communément utilisées. Nous nous intéressons en particulier aux méthodes basées sur la Descente de Gradient Stochastique (Stochastic Gradient Descent (SGD), en anglais) [Bottou, Curtis et Nocedal 2018]. Dans la version originale de cet algorithme, l'ensemble des paramètres θ est mis à jour en permutant les éléments du jeu de données $\{(x_k, y_k)\}_{k \in \{1, \dots, m\}}$ puis en itérant sur chaque paire (x_k, y_k) dans le jeu obtenu comme suit :

$$\theta^{k+1} = \theta^k - \eta \nabla_{\theta} l(y_k) \quad , \quad (5.14)$$

avec η dans \mathbb{R}_+ le pas de descente. Notons que la fonction de coût l prend en entrée l'observation contrairement à l'équation 5.12 où l'estimation est donnée en entrée. Néanmoins cette version est sensible à la présence de bruit dans les données et il n'est pas possible d'avoir une implémentation parallélisée, ce qui empêche d'accélérer cette méthode en pratique. Pour pallier ces problèmes, l'algorithme 5.1 [Bottou, Curtis et Nocedal 2018] a été développé, l'idée est d'itérer sur des mini-batches au lieu de le faire sur un élément à la fois. À chaque itération, la mise à jour se fait en considérant la moyenne des gradients de la fonction de perte pour chaque élément, ce qui permet d'ajouter de la robustesse au bruit dans les données mais également de paralléliser l'implémentation en pratique donc d'avoir une méthode plus rapide. Il existe également d'autres améliorations qui permettent de converger plus rapidement telles que ADAGRAD, en modifiant le pas de descente à chaque itération, et RMSPROP, en ajoutant un terme de moment en plus du terme de gradient [Bottou, Curtis et Nocedal 2018]. Finalement la méthode ADAM est l'une des méthodes les plus utilisées pour entraîner un réseau de neurones [Bock, Goppold et Weiß 2018; Schmidt, Schneider

et Hennig 2020], l'algorithme ADAM combine les avantages de RMSPROP et ADAGRAD [Kingma et Ba 2015]. Pour aller plus loin, une comparaison des méthodes état de l'art en optimisation pour l'apprentissage machine est effectuée dans [Schmidt, Schneider et Hennig 2020].

Algorithme 5.1 Descente de Gradient Stochastique avec Mini-batch

tâche : optimiser θ .

entrées : $\{x_k\}_{k \in \{1, \dots, m\}}$: vérités-terrain, $\{y_k\}_{k \in \{1, \dots, m\}}$: observations.

paramètres : n_e : nombre d'époques, n_b : nombre de mini-batches, η : pas de descente.

initialisation : $\theta^{(0)}$

pour pour i allant de 1 à n_e **faire**

permuter $\{(x_k, y_k)\}_{k \in \{1, \dots, m\}}$

pour pour j allant de 1 à n_b **faire**

$$\theta^{(n_b(i-1)+j)} \leftarrow \theta^{(n_b(i-1)+j-1)} - \eta \frac{n_b}{m} \sum_{k=(j-1)m/n_b+1}^{jm/n_b} \nabla_{\theta} l(y_k)$$

fin pour

fin pour

renvoyer : $\theta^{(n_e n_b)}$

Pour utiliser ces algorithmes, il faut calculer $\nabla_{\theta} l(y_k)$. Pour ce faire, l'algorithme de *rétropropagation* (backpropagation, en anglais) est utilisé [Lecun 2001] : Considérons un réseau de neurones profonds, afin de calculer le gradient de chaque paramètre, le principe de la rétropropagation est de commencer par calculer le gradient des paramètres de la couche de sorties puis de calculer le gradient des paramètres des couches antécédentes en propageant en arrière à l'aide de la règle de la chaîne, jusqu'à arriver aux gradients des paramètres de la couche d'entrée. En pratique, l'implémentation de cet algorithme peut se faire rapidement et efficacement en utilisant la différentiation automatique, également appelée autodiff [Merriënboer et al. 2018].

Dans cette section, nous avons vu comment construire et entraîner un réseau de neurones dans le but de résoudre les problèmes inverses. Dans la suite, nous allons voir comment les réseaux de neurones sont utilisés pour le problème de déconvolution.

5.3 Apprentissage Profond et Déconvolution

Comme précisé dans les chapitres 1 et 2, nous avons établi que dans les décennies à venir un volume considérable de données brutes à haute résolution, que ce soit en imagerie optique ou en radio-interférométrie, allait être produit. Dans les deux cas également, la reconstruction des images de galaxies correspond à un problème inverse de déconvolution avec une FEP spatialement variable. Des travaux préliminaires ont montré l'efficacité de l'apprentissage profond pour la résolution de tels problèmes [Flamary 2017 ; Schawinski et al. 2017]. Dans cette section, nous commençons par présenter deux méthodes de déconvolution que j'ai implémentées et qui sont fondées sur l'apprentissage profond développées dans [Sureau, Lechat et Starck 2020] : une approche directe et une approche modulaire. Puis nous détaillerons comment ajouter la contrainte de forme à ces méthodes pour finir avec les résultats des expériences que j'ai menées sur les jeux de données que j'ai développés.

5.3.1 Approche directe

L'approche directe est appelée *Tikhonet*, il s'agit d'une méthode en deux étapes inspirée de la méthode *ForWaRD* [Neelamani, Choi et Baraniuk 2004]. L'idée est de séparer le

problème de déconvolution en deux problèmes : un problème de déconvolution stricte suivi d'un problème de débruitage. La première étape permet de corriger l'effet de la FEP sans tenir compte de la présence de bruit. La sortie de la première étape est ensuite débruitée dans la deuxième étape à l'aide de l'apprentissage profond pour obtenir la sortie du Tikhonet.

Déconvolution

L'étape de déconvolution se fait en utilisant le filtre de Tikhonov qui est décrit dans la partie 2.2.2. Dans la version originale du Tikhonet, le paramètre de compromis du filtre de Tikhonov est calculé pour chaque galaxie avec l'estimateur SURE [Sureau, Lechat et Starck 2020]. Dans notre travail, nous avons choisi un paramètre de compromis fixe pour toutes les images de galaxies. Effectivement, en considérant l'équation 2.23, changer le paramètre de compromis λ est équivalent à changer le filtre. D'un point de vue fréquentiel, ce changement implique des fréquences de coupure différentes pour différentes valeurs de λ . Ainsi la tâche que le réseau de neurones doit effectuer dans ce cas est de débruiter des images ayant différentes plages de fréquences. Apprendre cette tâche revient à apprendre deux sous-tâches : l'estimation de la plages de fréquences et le débruitage. Si l'estimation de la plage de fréquences n'est pas correcte, pour une fréquence donnée nous avons deux cas de figures : l'information présente à la fréquence est coupée par le réseau ou une information absente à la fréquence est générée par le réseau. Dans le premier cas, nous obtenons des images plus lisses voire floues et dans le second cas, nous risquons de voir apparaître des artefacts dans la sortie du Tikhonet, il s'agit d'un cas d'*hallucination*. En choisissant un paramètre de compromis fixe pour toutes les images, nous restreignons la tâche à apprendre par le réseau au débruitage.

Pour choisir la valeur de λ , nous rappelons, comme argumenté dans la partie 2.2.1, qu'une valeur élevée de λ lisse les détails fins contenu dans l'image. Tandis qu'une valeur faible de λ permet de mieux préserver les détails fins aux dépens de la présence de bruit dans l'image filtrée. Une valeur faible de λ est alors mieux adaptée au Tikhonet puisque les détails sont préservés et le bruit est ensuite traité dans la deuxième étape. En pratique, nous avons effectué une recherche linéaire en utilisant la MSE comme critère et la valeur de $\lambda = 9 \cdot 10^{-3}$ a été obtenue que ce soit pour le jeu de données CFHT2HST ou MeerKAT3600.

Toujours dans la version originale du Tikhonet, Γ dans l'équation 2.21 vaut l'identité [Sureau, Lechat et Starck 2020], ce qui veut dire, dans l'espace de Fourier, que toutes les fréquences du signal à reconstruire sont pénalisées avec le même poids. Or généralement, dans l'espace de Fourier, un signal naturel a un profil décroissant et celui d'un bruit blanc gaussien est constant (dans le cas de l'interférométrie radio, le bruit gaussien corrélé correspond mathématiquement à un bruit blanc gaussien pondéré, voir 1.2.3) donc les hautes fréquences du signal sont plus affectées par le bruit que ses basses fréquences. À cet effet nous avons choisi l'opérateur laplacien pour Γ de sorte que la pénalisation des hautes fréquences soit plus importante que celles des basses fréquences.

Finalement, pour intégrer la contrainte de forme dans l'étape de déconvolution, nous reprenons l'équation 2.21 à laquelle nous ajoutons la contrainte de forme telle que formulée dans l'équation 4.20 :

$$I(x) = \|y - Hx\|_S^2 + \lambda \|\Gamma x\|_2^2 \quad . \quad (5.15)$$

La solution x_S qui minimise I et annule son gradient, vaut :

$$x_S = \left(H^T S H + \lambda \Gamma^T \Gamma \right)^{-1} H^T S y \quad . \quad (5.16)$$

Dans notre travail, nous avons utilisé le filtre de Tikhonov sans l'ajout de la contrainte de forme. En effet, l'implémentation pratique du filtre avec contrainte de forme est inconmode si nous utilisons directement la formule de x_S surtout pour le calcul $H^T S H$ et $H^T S$ dont la dimension est le carré de celle de x_S . À titre de comparaison, nous avons réimplémenté le filtre de Tikhonov sans la contrainte de manière efficace en utilisant la transformée de Fourier et son inverse, ce qui permet de manipuler des opérateurs ayant la même dimension que x_S . En outre, nous avons étudié l'impact de l'ajout de la contrainte de forme à l'étape de débruitage comme nous allons voir dans la partie suivante.

Débruitage

Par rapport au problème de déconvolution, le débruitage est un problème inverse bien conditionné, c'est-à-dire que pour un même niveau de bruit, l'espace des solutions est plus restreint et permet aux méthodes de reconstruction d'offrir des solutions plus stables. Dans la méthode du Tikhonet, à la sortie du filtre du Tikhonov, le bruit contenu dans l'image est un bruit corrélé que ce soit dans le cas optique ou radio. Dans la méthode ForWarD, le débruitage se fait en utilisant les ondelettes [Neelamani, Choi et Baraniuk 2004]. L'apprentissage profond, et notamment les U-nets, font partie de l'état de l'art pour le débruitage [Gurrola-Ramos, Dalmau et Alarcón 2021], ce qui justifie son utilisation dans la méthode Tikhonet. Le U-net utilisé dans le Tikhonet comporte les modifications suivantes par rapport au U-net original [Sureau, Lechat et Starck 2020] :

- *Convolution séparable* : les noyaux de convolution sont remplacés par des noyaux de convolution séparables qui offrent de meilleures performances aux réseaux de neurones [Chollet 2017]. Un noyau de convolution séparable de taille $n \times n$ contient $2n$ paramètres qui correspondent à deux filtres 1D de taille n tel que le filtre séparable est le produit dyadique² des deux filtres 1D. Avec moins de paramètres que dans un noyau de convolution, la capacité à généraliser est plus importante et l'entraînement est moins coûteux en ressources de calcul.
- *Denses blocks* : dans chaque étage du U-net, les couches convolutives sont remplacées par des *denses blocks* [Huang et al. 2017}. Les denses blocks sont un ensemble de couches où sont concaténées, à l'entrée de chaque couche, les cartes de descripteurs des couches précédentes. Cette architecture permet de réutiliser les descripteurs, mieux conserver l'information et limiter la disparition du gradient durant l'entraînement.
- *Average pooling* : c'est la méthode de sous-échantillonnage qui effectue la moyenne. Son utilisation permet d'éviter les effets de sur-segmentation qui apparaissent avec le max pooling en pratique [Sureau, Lechat et Starck 2020].
- *Connexion skip* : la connexion skip entre les couches d'entrée et de sortie du réseau a été retirée. Cette déletion a permis d'avoir de meilleures performances notamment pour les faibles RSB.

2. Soient $u = (u_1, \dots, u_n) \in \mathbb{R}^n$ et $v = (v_1, \dots, v_n) \in \mathbb{R}^n$ alors le produit dyadique de u par v , noté $u \otimes v$, est défini comme une matrice dans $\mathbb{R}^{n \times n}$ telle que :

$$u \otimes v = \begin{bmatrix} u_1 \cdot v_1 & \dots & u_1 \cdot v_n \\ \vdots & \ddots & \vdots \\ u_n \cdot v_1 & \dots & u_n \cdot v_n \end{bmatrix}$$

Le réseau est entraîné avec l'EQM comme fonction de coût. L'ajout de la contrainte de forme dans l'étape de débruitage se fait en modifiant la fonction de coût telle que :

$$I(\hat{x}) = \frac{1}{2} \|x - \hat{x}\|_2^2 \quad . \quad (5.17)$$

Dans la suite notre travail, la méthode Tikhonet avec contrainte de forme correspond à l'ajout de la contrainte dans l'étape de débruitage.

5.3.2 Approche modulaire

L'approche modulaire, également appelée ADMMnet, dans [Sureau, Lechat et Starck 2020] correspond à l'utilisation de l'algorithme ADMM Plug-and-Play tel que dans 2.2 pour reconstruire les images. En effet, il a été montré dans [Meinhardt et al. 2017] qu'un réseau de neurones entraîné à débruiter un bruit blanc gaussien peut remplacer l'opérateur proximal dans les algorithmes proximaux. Dans cette approche itérative, nous pouvons donc utiliser l'apprentissage profond ainsi que la régularisation. Effectivement, dans l'ADMMnet, l'architecture du U-net modifié est utilisée comme débruiteur gaussien pour remplacer l'opérateur proximal. La régularisation utilisée est la positivité telle que définie dans l'équation 4.27.

Dans notre travail, nous avons réimplémenté l'ADMMnet en y ajoutant la contrainte de forme, néanmoins la méthode obtenue débouche sur des résultats préliminaires montrant qu'elle n'est pas encore stable et plusieurs pistes pour stabiliser la méthode sont proposées dans l'appendice C.

5.4 Expériences numériques

Dans cette section, nous présentons les expériences numériques menées afin d'évaluer les méthodes que j'ai développées : SCORE et Tikhonet avec contrainte de forme. Le code a été développé en utilisant le langage Python 3.7.5 et les bibliothèques TensorFlow 1.15.2, Keras 2.3.1 [Chollet 2015], AlphaTransform, Matplotlib 3.1.3 [Hunter 2007], Galaxy2Galaxy et GalFlow [Lanusse et Remy 2019]. Pour l'entraînement du U-net pour les méthodes Tikhonet avec et sans contrainte de forme, nous avons normalisé les valeurs des pixels des images en entrées du réseau, par $4 \cdot 10^3$ dans le cas optique et $2 \cdot 10^{-3}$ dans le cas radio, afin d'avoir des valeurs ayant un ordre de grandeur avoisinant l'unité pour que les fonctions d'activation dans le réseau de neurones puissent mieux discriminer les données.

Les critères de qualité utilisés pour ces expériences sont l'EQM relative moyenne, l'erreur relative moyenne du flux (de la densité de flux en radio) et l'erreur absolue moyenne de chaque composante de l'ellipticité e_1 et e_2 . L'ellipticité dans ce cas a été estimée en utilisant la méthode des moments adaptatifs [Hirata et Seljak 2003] qui consiste à ajuster une fenêtre gaussienne elliptique sur l'image en entrée pour ensuite déduire la mesure à partir de la fenêtre obtenue. Ce choix est justifié par la présence de bruit dans les images vérité terrain du jeu de données optique CFHT2HST, ce qui rend l'utilisation de l'estimateur dans l'équation 4.5 problématique.

5.4.1 En optique

Pour les expériences optiques, nous avons considéré le jeu de données CFHT2HST présenté dans la partie 1.1.3. Dans le même esprit que les expériences menées dans [Sureau, Lechat et Starck 2020], le but est de reconstruire des images à haute résolution à partir d'images à basse résolution, avec chaque résolution correspondant à un télescope. Dans

notre cas, les images hautes résolutions correspondent à des images réelles du télescope HST et les images basses résolution correspondent à des images simulées à la résolution du CFHT obtenues en dégradant les images HST. Ces dernières sont considérées comme les vérités terrain et les images CFHT sont les images observées. Pour mesurer la qualité du signal dans l'image, nous utilisons la magnitude absolue. En effet, le niveau de bruit ajouté à ces dernières est un bruit de fond constant qui est calculé à partir des paramètres du télescope³. De ce fait, la magnitude absolue vaut, à une constante multiplicative près, l'opposé du logarithme du RSB. Nous avons considéré 4 classes de magnitudes, chacune ayant une moyenne de 20,79 ; 22,16 ; 22,83 et 23,30, et contenant chacun un nombre égal d'échantillons. Le jeu de données contient des galaxies quasi-ponctuelles, très petites pour être résolues par le télescope. Comme la mesure la forme de ces galaxies est problématique, nous avons retiré de notre analyse les galaxies où la mesure de forme sur l'image vérité terrain avait échoué. Dans chaque classe de magnitudes, nous avons alors environ 500 galaxies. Les méthodes étudiées sont SRA et SCORE pour les méthodes parcimonieuses et Tikhonet et Tikhonet avec contrainte de forme, appelée *ShapeNet*, pour l'apprentissage profond. Dans ces deux dernières, le U-net utilisé contient 4 échelles et son entraînement s'est déroulé sur 10 époques composées de 625 pas chacune avec des batchs ayant une taille de 128 chacun. Pour le choix de la pondération de la contrainte de forme dans la méthode ShapeNet, nous avons effectué une recherche linéaire et trouvé la valeur $\gamma = 0,5$. Dans le cas de SCORE, la pondération a été fixée à $\gamma = 1$, en se basant sur la partie 4.3.1. De plus, le noyau de convolution utilisé dans SCORE est obtenu en effectuant une division, dans l'espace de Fourier, de la transformée de la FEP d'entrée par celle de sortie. Nous effectuons alors une *déconvolution partielle*.

Qualitativement, dans la figure 5.7b, nous donnons des exemples de galaxies quasi-ponctuelles. Dans le dernier cas, nous pouvons voir qu'en présence d'un niveau de bruit élevé dans l'observation, la reconstruction de la galaxie pour les méthodes SRA et Tikhonet montre un signal très atténué, par contre avec la contrainte de forme, nous pouvons voir que les méthodes SCORE et ShapeNet permettent de reconstruire le signal et son amplitude. Dans le cas des galaxies résolues, nous donnons 5 exemples dans la figure 5.7a. Tout d'abord, en comparant les méthodes parcimonieuses aux méthodes d'apprentissage profond, nous remarquons que les méthodes parcimonieuses ont tendance à rendre l'objet reconstruit plus isotrope et plus lisse, en notant que les détails présents dans la vérité terrain sont perdus dans la reconstruction. Cet effet s'explique par l'utilisation des starlets qui sont des ondelettes isotropes. Les méthodes d'apprentissage profond quant à elles donnent des reconstructions contenant plus de détails et de structures comme dans la première galaxie où la forme du cœur de la galaxie semble se rapprocher davantage de celle de la vérité terrain que celle obtenue par les méthodes parcimonieuses, où le cœur tend à être plus lisse. Étudions maintenant l'effet de l'ajout de la contrainte de forme. Nous remarquons dans les deux cas, que ce soit en passant de SRA à SCORE et de Tikhonet à ShapeNet, une meilleure coïncidence entre l'orientation des galaxies reconstruites et de la vérité terrain en présence de contrainte de forme. Nous remarquons également que l'ajout de la contrainte permet une meilleure reconstruction des galaxies étendues. Cela semble être le cas de la méthode Tikhonet où les galaxies reconstruites sans la contrainte semble être plus petites et moins brillantes que la vérité terrain à partir d'un niveau de bruit non négligeable. Dans les quatre méthodes présentées, les reconstructions contiennent moins de bruits de fond que les vérités terrain, ce qui s'explique par le fait que la représentation des galaxies (via les starlets ou

3. Pour plus de détails sur la génération des galaxies observées et le calcul du niveau de bruit voir : <https://github.com/CosmoStat/ShapeDeconv/blob/master/data/CFHT/HST2CFHT.ipynb>

celle apprise par le U-net) est adaptée aux galaxies. Ce qui permet une meilleure séparation du signal utile (la galaxie) du bruit de fond.

Méthodes	EQM	flux	e_1	e_2
SRA	$4,33.10^{-1} \pm 6,95.10^{-3}$	$2,06.10^0 \pm 5,92.10^{-1}$	$1,28.10^{-1} \pm 2,03.10^{-3}$	$1,54.10^{-1} \pm 2,41.10^{-3}$
SCORE	$2,93.10^{-1} \pm 3,93.10^{-3}$	$1,83.10^0 \pm 5,68.10^{-1}$	$1,18.10^{-1} \pm 1,84.10^{-3}$	$1,45.10^{-1} \pm 2,20.10^{-3}$
Tikhonet	$3,33.10^{-1} \pm 4,45.10^{-3}$	$1,25.10^0 \pm 4,16.10^{-1}$	$1,24.10^{-1} \pm 1,89.10^{-3}$	$1,34.10^{-1} \pm 2,09.10^{-3}$
ShapeNet	$2,45.10^{-1} \pm 3,47.10^{-3}$	$6,49.10^{-1} \pm 6,20.10^{-2}$	$1,16.10^{-1} \pm 1,81.10^{-3}$	$1,27.10^{-1} \pm 2,06.10^{-3}$

Table 5.1 – Résultats du jeu de données CFHT2HST. Les valeurs indiquent l'erreur moyenne de chaque grandeur et l'incertitude correspond à l'erreur type de la valeur donnée.

Pour mesurer l'EQM, nous avons effectué une pondération avec la fenêtre gaussienne elliptique obtenue lors de l'estimation de la forme de la galaxie vérité terrain. Cette pondération réduit le bruit dans les images vérité terrain permettant de diminuer le biais dans l'estimation de l'EQM. Quantitativement, nous commençons également par comparer les méthodes parcimonieuses aux méthodes d'apprentissage profond en prenant SRA et Tikhonet. De manière générale, les tendances obtenues corroborent les résultats de [Sureau, Lechat et Starck 2020]. En effet, dans toutes les mesures effectuées dans le tableau 5.1 et les courbes dans la figure 5.8, les erreurs moyennes des résultats obtenus avec Tikhonet sont inférieures à celles obtenues avec SRA, excepté pour l'erreur moyenne de e_1 pour la plus faible magnitude où l'erreur de Tikhonet est plus élevée de l'erreur de 9%. À partir du tableau 5.1, les reconstructions obtenues avec Tikhonet ont des erreurs inférieures à celles obtenues avec SRA avec des écarts de 23% ; 39% ; 3% et 13% pour l'EQM, le flux, e_1 et e_2 , respectivement. Ensuite, comparons l'ajout de la contrainte de forme à chacune des deux méthodes, nous remarquons dans les deux cas que l'ajout de la contrainte réduit les erreurs que ce soit pour toute magnitude confondue ou par niveau de magnitude, excepté pour l'EQM dans le cas des faibles magnitudes, Tikhonet a la plus faible erreur parmi les quatre méthodes. Il est possible que ce comportement provienne d'un biais de modélisation. Effectivement, une faible valeur de magnitude correspond à un RSB élevé et c'est dans ce régime que le biais de modélisation est le plus apparent, surtout dans le cas de l'apprentissage profond où l'approche largement dépendante des données. Dans ce cas, l'ajout de la contrainte de forme comme terme de régularisation pourrait induire le biais. À partir du tableau 5.1 encore une fois, les reconstructions obtenues avec SCORE ont des erreurs inférieures à celles obtenues avec SRA avec des écarts de 32% ; 13% ; 8% ; 6% pour l'EQM, le flux, e_1 et e_2 , respectivement. De manière analogue, les écarts pour ShapeNet et Tikhonet valent 26% ; 48% ; 6% et 5%. Avec ces expériences, j'ai pu montrer que l'ajout de la contrainte de forme permet donc d'améliorer les performances des méthodes considérées vis-à-vis de tous les critères de qualité étudiés, ce qui corrobore les résultats dans la partie 4.3.2. Notons au passage que l'ajout de la contrainte de forme pour la méthode parcimonieuse permet de la rendre plus compétitive avec la méthode d'apprentissage profond Tikhonet où l'erreur moyenne de la première est plus faible pour l'EQM et la composante e_1 avec des écarts de 12% et 5%, respectivement.

En conclusion, dans le cas optique, les expériences menées sur le jeu de données CFHT2HST montrent que l'ajout de la contrainte de forme et le passage des méthodes parcimonieuses à l'apprentissage profond permettent de réduire les erreurs de reconstruction. Dans la suite, nous allons nous intéresser au cas radio avec le jeu de données MeeKAT3600.

5.4.2 En radio

Pour les expériences radio, nous avons considéré le jeu de données MeerKAT3600 présenté dans la partie 1.2.4. Contrairement aux expériences en optique, les images vérités terrain sont réalistes mais ne sont pas des images réelles. Pour ce cas, la déconvolution effectué est totale. Effectivement, les vérités terrain sont des images simulées à l'aide du catalogue T-RECS et leur résolution n'est pas limitée par la FEP d'un télescope. Par contre les observations sont simulées pour qu'elles soient semblables à des celles du télescope MeerKAT. Pour ce faire, nous avons développé un code de simulation de FEP réaliste. Le niveau de bruit utilisé pour ces expériences est constant et est choisi de sorte à obtenir une variété de niveaux de RSB afin d'avoir une évaluation étendue des différentes méthodes étudiées. Pour quantifier la qualité du signal, nous utilisons un critère communément utilisé en radio-interférométrie, le *Peak Signal to Noise Ratio (PSNR)* qui se définit comme étant le rapport entre le maximum du signal utile sur le l'écart-type du bruit. Nous avons considéré 4 classes de PSNR chacune ayant une moyenne de 4,38 ; 6,92 ; 9,99 et 16,74 et contenant un nombre égal d'échantillons. Les méthodes comparées sont CLEAN, CLEAN isotrope, SRA, SCORE, Tikhonet et ShapeNet. La méthode CLEAN isotrope est une amélioration de CLEAN qui permet d'effectuer une comparaison plus équitable avec les autres méthodes. En effet, par opposition à la méthode originale où la dernière étape dans la reconstruction consiste à convoluer les composantes détectées par le CLEAN beam qui est une fonction gaussienne elliptique ajustée sur la FEP, CLEAN isotrope utilise une FEP gaussienne isotrope avec un écart-type σ valant 2 pixels et un flux normalisé par celui du CLEAN beam. Cette convolution permet ainsi d'adoucir la carte des composantes CLEAN détectées, composée exclusivement de diracs, et d'adapter leur représentation au fait que la galaxie de la vérité terrain est étendue. L'isotropie réduit le biais induit sur la forme de la galaxie reconstruite, idem pour la petite taille du noyau isotrope. Le jeu de données contient des observations avec un PSNR très faible allant en-dessous de 3 qui correspond au seuil de détection de signal utile pour les méthodes CLEAN et CLEAN isotrope. Alors dans le sous-ensemble choisi pour faire les expériences numériques, 72 parmi 3072 galaxies ont été retirées parce que leur PSNR est en-dessous du seuil. Nous avons au final 750 galaxies par classe de PSNR. Pour les méthodes d'apprentissage, le U-net utilisé contient également 4 échelles et son entraînement s'est déroulé sur 10 époques composées de 3125 pas chacune avec des batchs ayant une taille de 32 chacun. Pour le choix de la pondération pour la contrainte de forme, à l'aide de recherches linéaires, nous avons trouvé la valeur $\gamma = 0,5$ pour ShapeNet et $\gamma = 2$ pour SCORE. De plus, pour ces expériences, nous avons également modifié l'initialisation des méthodes parcimonieuses, SRA et SCORE, en remplaçant l'image constante par un filtrage de Tikhonov appliqué à l'observation. Dans les résultats qui sont montrés dans les courbes 5.10, les images 5.7 et le tableau 5.10, nous avons réduit le support des images reconstruites en divisant la taille de chaque dimension par 2, de sorte que la galaxie demeure au centre.

Méthodes	EQM	flux	e_1	e_2
CLEAN	$7,41 \cdot 10^{-1} \pm 2,37 \cdot 10^{-3}$	$4,53 \cdot 10^{-1} \pm 3,62 \cdot 10^{-3}$	$2,53 \cdot 10^{-1} \pm 2,97 \cdot 10^{-3}$	$2,63 \cdot 10^{-1} \pm 3,31 \cdot 10^{-3}$
CLEAN iso	$3,91 \cdot 10^{-1} \pm 3,51 \cdot 10^{-3}$	$5,02 \cdot 10^{-1} \pm 3,47 \cdot 10^{-3}$	$1,30 \cdot 10^{-1} \pm 1,76 \cdot 10^{-3}$	$1,27 \cdot 10^{-1} \pm 1,70 \cdot 10^{-3}$
SRA	$2,81 \cdot 10^{-1} \pm 1,16 \cdot 10^{-2}$	$3,31 \cdot 10^{-1} \pm 9,30 \cdot 10^{-3}$	$1,20 \cdot 10^{-1} \pm 1,87 \cdot 10^{-3}$	$1,26 \cdot 10^{-1} \pm 2,07 \cdot 10^{-3}$
SCORE	$2,69 \cdot 10^{-1} \pm 1,08 \cdot 10^{-2}$	$2,39 \cdot 10^{-1} \pm 7,15 \cdot 10^{-3}$	$1,14 \cdot 10^{-1} \pm 1,78 \cdot 10^{-3}$	$1,21 \cdot 10^{-1} \pm 2,01 \cdot 10^{-3}$
Tikhonet	$5,21 \cdot 10^{-2} \pm 1,47 \cdot 10^{-3}$	$1,58 \cdot 10^{-1} \pm 2,14 \cdot 10^{-3}$	$7,44 \cdot 10^{-2} \pm 1,44 \cdot 10^{-3}$	$7,88 \cdot 10^{-2} \pm 1,53 \cdot 10^{-3}$
ShapeNet	$4,33 \cdot 10^{-2} \pm 1,17 \cdot 10^{-3}$	$1,47 \cdot 10^{-1} \pm 2,48 \cdot 10^{-3}$	$7,34 \cdot 10^{-2} \pm 1,41 \cdot 10^{-3}$	$7,70 \cdot 10^{-2} \pm 1,48 \cdot 10^{-3}$

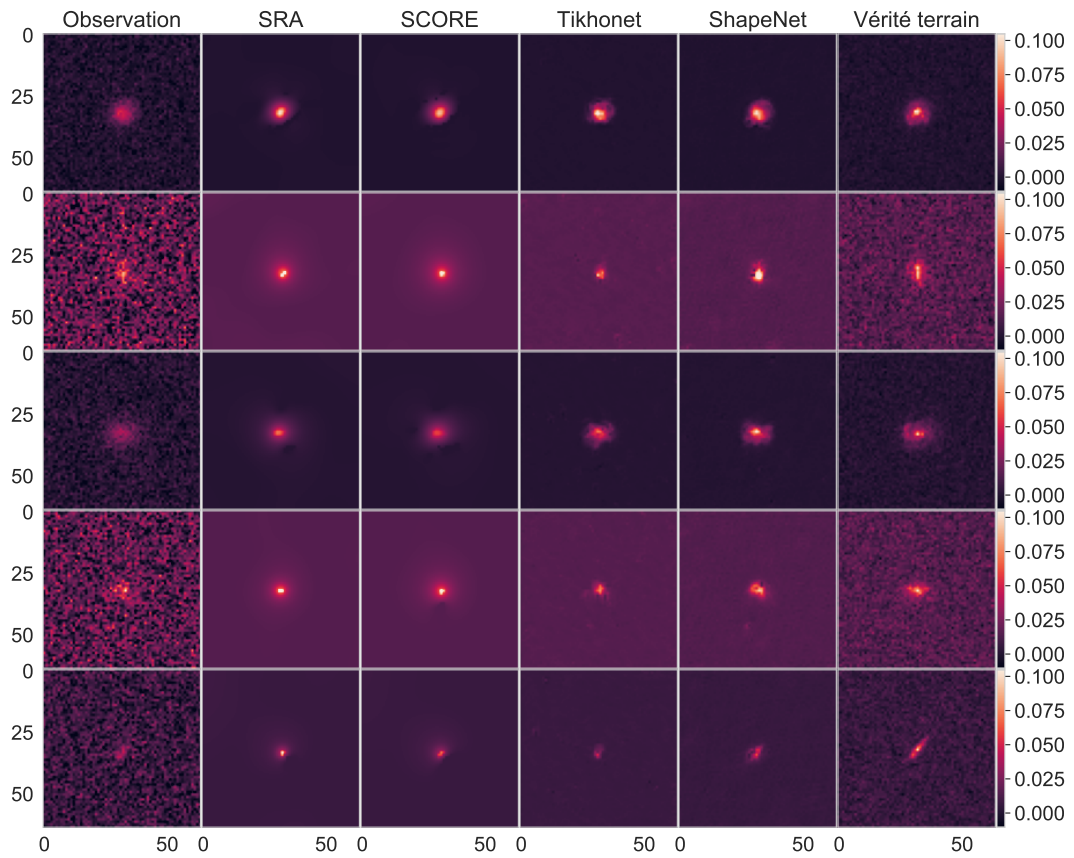
Table 5.2 – Résultats du jeu de données MeerKAT3600. Les valeurs indiquent l'erreur moyenne de chaque grandeur et l'incertitude correspond à l'erreur type de la valeur donnée.

Dans la figure 5.9b, nous donnons des exemples de galaxies où le PSNR est inférieur à 3, nous notons que dans certains cas comme dans les deux premières colonnes, les méthodes parcimonieuses, CLEAN et *CLEAN isotrope* ne détectent pas la galaxie dans leur reconstruction tandis que les méthodes d'apprentissage profond détectent la galaxie mais sans parvenir à reconstruire correctement la forme de la galaxie, qui semble isotrope sur la reconstruction. Dans la figure 5.9a, par opposition dans la figure précédente, nous pouvons mieux distinguer les galaxies du bruit dans les observations. Qualitativement, notons que la méthode CLEAN isotrope offre une meilleure reconstruction de la forme de la galaxie, de son orientation et de sa taille en comparaison avec CLEAN. Dans la première colonne à droite, le PSNR est faible et vaut 2,35, effectivement nous pouvons voir que l'intensité des pixels de la galaxie au centre de l'image avoisine celle du bruit en bas à gauche et en haut à droite. Dans ce cas, nous observons de fausses détections là où l'intensité du bruit est élevée pour les reconstructions obtenues avec CLEAN, CLEAN isotrope et les méthodes parcimonieuses. Dans le cas des méthodes parcimonieuses et de CLEAN, nous remarquons que les reconstructions sont plus lisses que les vérités terrain et que l'orientation des galaxies est fortement biaisée par celle de la FEP. Concernant les méthodes d'apprentissage profond, les reconstructions ont un niveau de détail comparable à celui des vérités terrain tel que la forme, la taille, et l'orientation semblent mieux conservées comparées aux autres méthodes. Concernant la contrainte de forme, il ne semble pas exister de différence notable visuellement que ce soit entre SRA et SCORE ou Tikhonet et ShapeNet. Quantitativement, considérons tout d'abord les méthodes sans contraintes de formes, nous remarquons que pour tous les critères de qualité, CLEAN et CLEAN isotrope ont une erreur supérieure à celle de SRA qui a une erreur supérieure à celle de Tikhonet. Plus en détail, à partir du tableau 5.2, nous mesurons une réduction de 28% de l'EQM en passant de CLEAN isotrope à SRA et de 81% en passant de SRA à Tikhonet. De même, nous mesurons des réductions de l'erreur du flux de 34% et 52%, de l'erreur de e_1 de 8% et 38% et de l'erreur de e_2 de 1% et 37%. Notons que pour les faibles niveaux de PSNR, les performances de CLEAN isotrope et SRA sont semblables à l'exception du flux où l'erreur dans le SRA est de 35% plus faible que celle de CLEAN isotrope. Nous remarquons également que le flux des galaxies est mieux estimé par CLEAN par rapport à CLEAN isotrope avec une erreur de 10% de moins. Concernant la contrainte de forme, son ajout aux méthodes parcimonieuses et d'apprentissage profond améliore les performances dans les deux cas. Ainsi pour l'EQM, nous avons une réduction de 4% en passant de SRA à SCORE et 17% de Tikhonet à ShapeNet. De même, pour l'erreur du flux nous avons 28% et 7%, l'erreur de e_1 5% et 1% et l'erreur de e_2 4% et 2%. Finalement, nous concluons que les méthodes parcimonieuses et d'apprentissage profond ont de meilleures performances que les méthodes CLEAN et CLEAN isotrope, l'ajout de la contrainte de forme apporte un gain dans tous les critères de qualités considérés et l'apprentissage profond offre de meilleures performances que la parcimonie. Il semble donc que la meilleure méthode présentée dans ces expériences soit la méthode ShapeNet que nous avons développée.

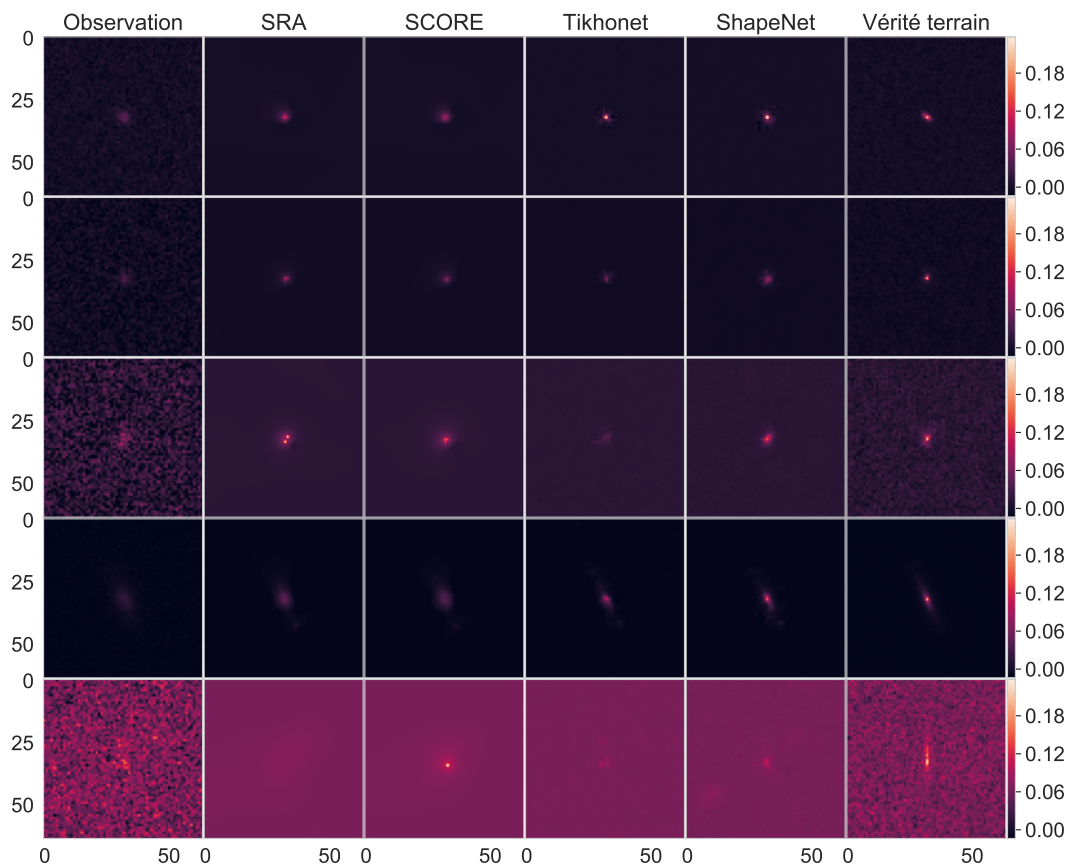
5.5 Conclusion

Dans cette partie, nous avons donné les bases de l'apprentissage machine en se focalisant sur l'apprentissage profond et les réseaux de neurones convolutifs, en particulier. Nous avons également introduit deux méthodes qui ont été développées dans [Sureau, Lechat et Starck 2020] : le Tikhonet qui est non-itérative et l'ADMMnet qui est itérative. Nous avons réimplémenté les deux méthodes, les tests que nous avons menés sur l'ADMMnet montrent

que le procédé de reconstruction est instable. Pour pallier à problème, plusieurs pistes sont envisageables, la première est de "déplier" l'algorithme itératif pour avoir un algorithme *unrolled* [Monga, Li et Eldar 2020]. Effectivement, l'algorithme unrolled possède un nombre fixe d'itérations et les poids du réseau de neurones sont appris avec un apprentissage effectué en utilisant cet algorithme. La convergence de la méthode est alors mieux garantie. Une autre solution est de contraindre le réseau de neurones utilisé dans l'algorithme à être contractant et satisfaire des propriétés qui offrent des garanties de convergence et stabilité [Ryu et al. 2019]. D'autre part, nous avons développé deux jeux de données, un optique et un radio, pour comparer les performances de Tikhonet avec SRA et SCORE, essentiellement, et étudier l'effet de l'ajout de la contrainte de forme. Dans le cas radio, nous avons également réimplémenté la méthode CLEAN avec une amélioration CLEAN isotrope qui permet d'avoir une comparaison plus équitable avec les autres méthodes. Dans les deux cas, optique et radio, nous avons obtenu un gain en performance en passant des méthodes parcimonieuses aux méthodes d'apprentissage profond. Nous avons également observé un gain en ajoutant la contrainte de forme dans ces méthodes. Dans le cas optique, le gain obtenu par le passage des méthodes parcimonieuses à l'apprentissage profond est comparable à celui obtenu avec l'ajout de la contrainte de forme dans le cas de la méthode ShapeNet. Tandis que dans le cas radio, le gain obtenu en passant à l'apprentissage profond est plus important. Finalement, il est possible d'améliorer la méthode ShapeNet en remplaçant le U-net par un autre réseau de neurones tel que Deep Iterative Down-Up CNN for Image Denoising (DIDN) [Yu, Park et Jeong 2019].

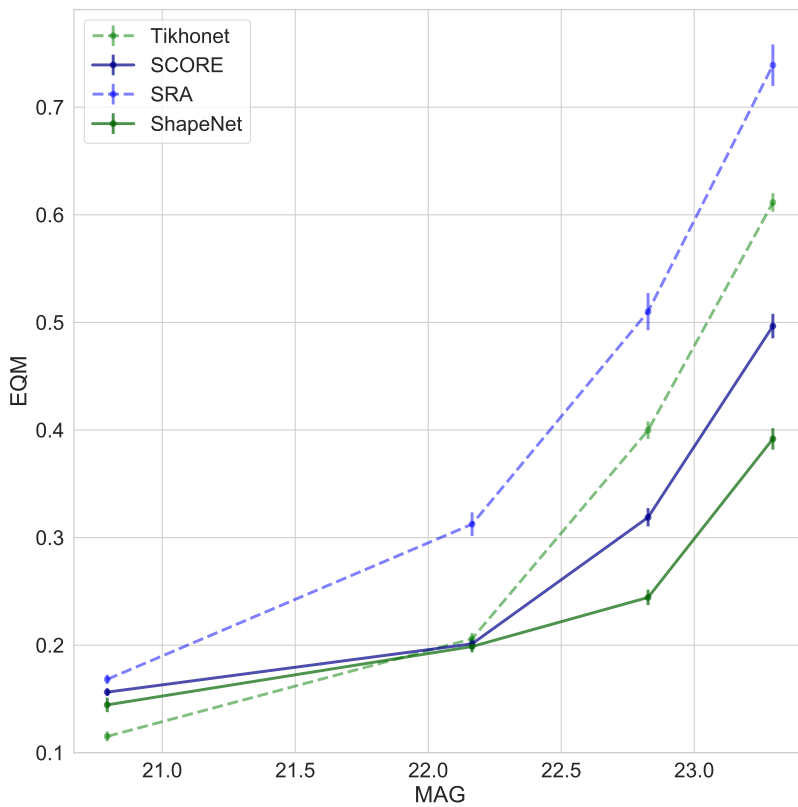


(a) Galaxies optiques avec estimation de forme réussie

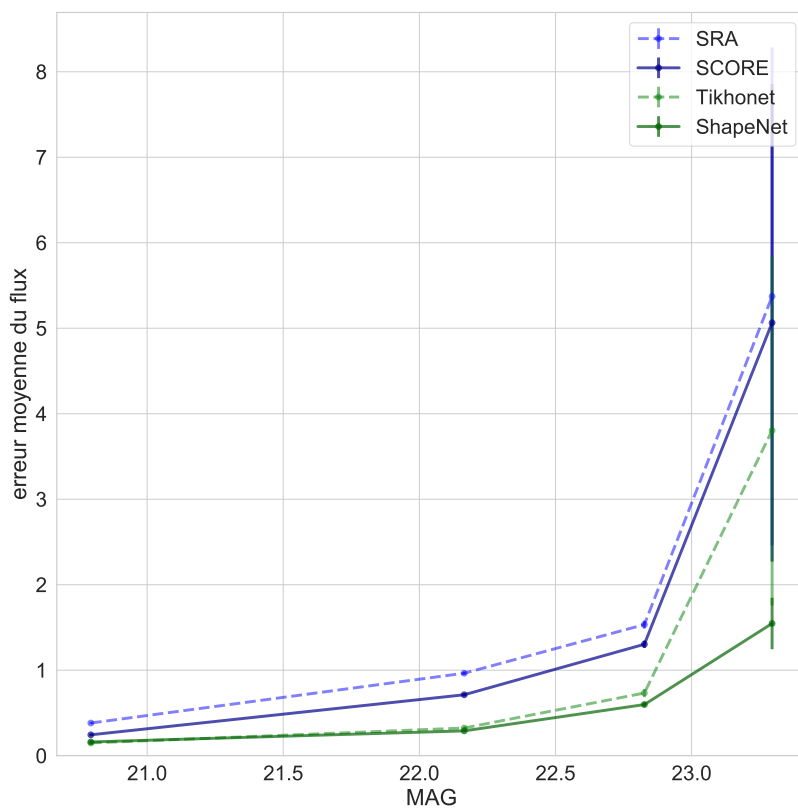


(b) Galaxies optiques avec estimation de forme qui a échouée

Figure 5.7 – Exemples de galaxies reconstruites avec le jeu de données CFHT2HST.



(a) EQM relative moyenne.



(b) Erreur relative moyenne du flux.

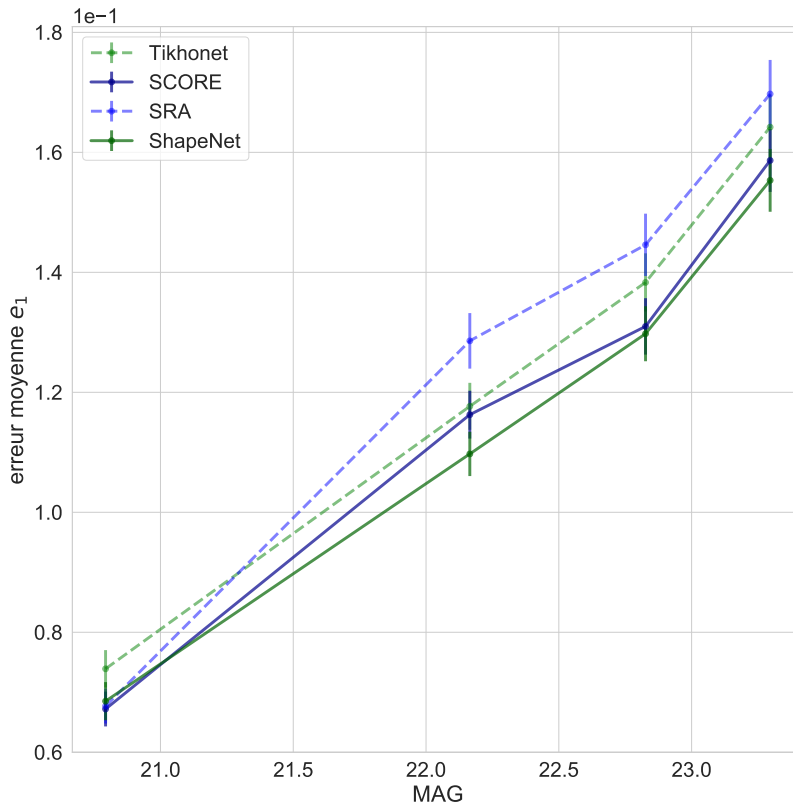
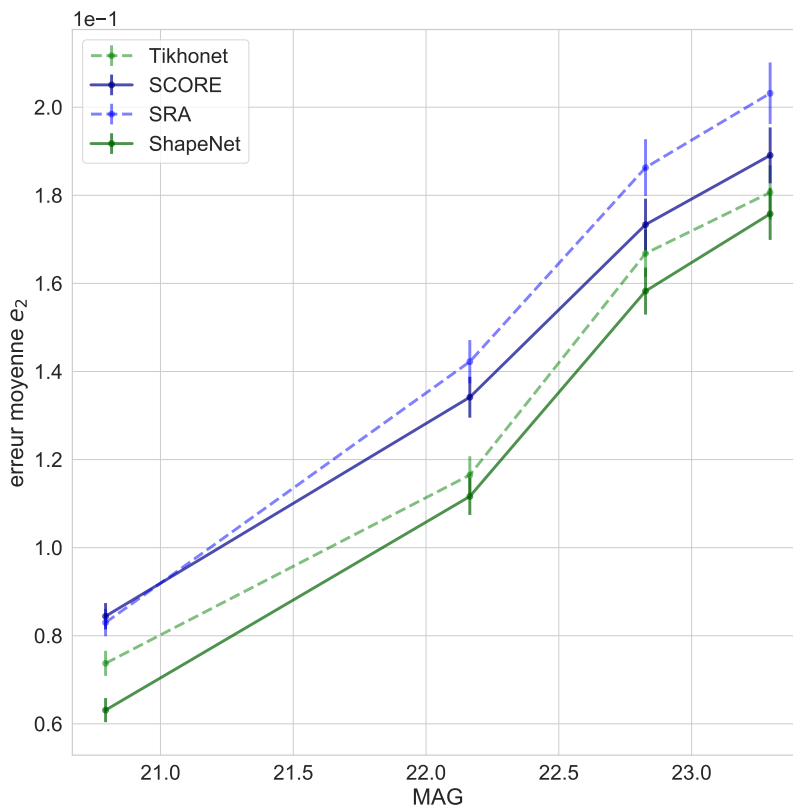
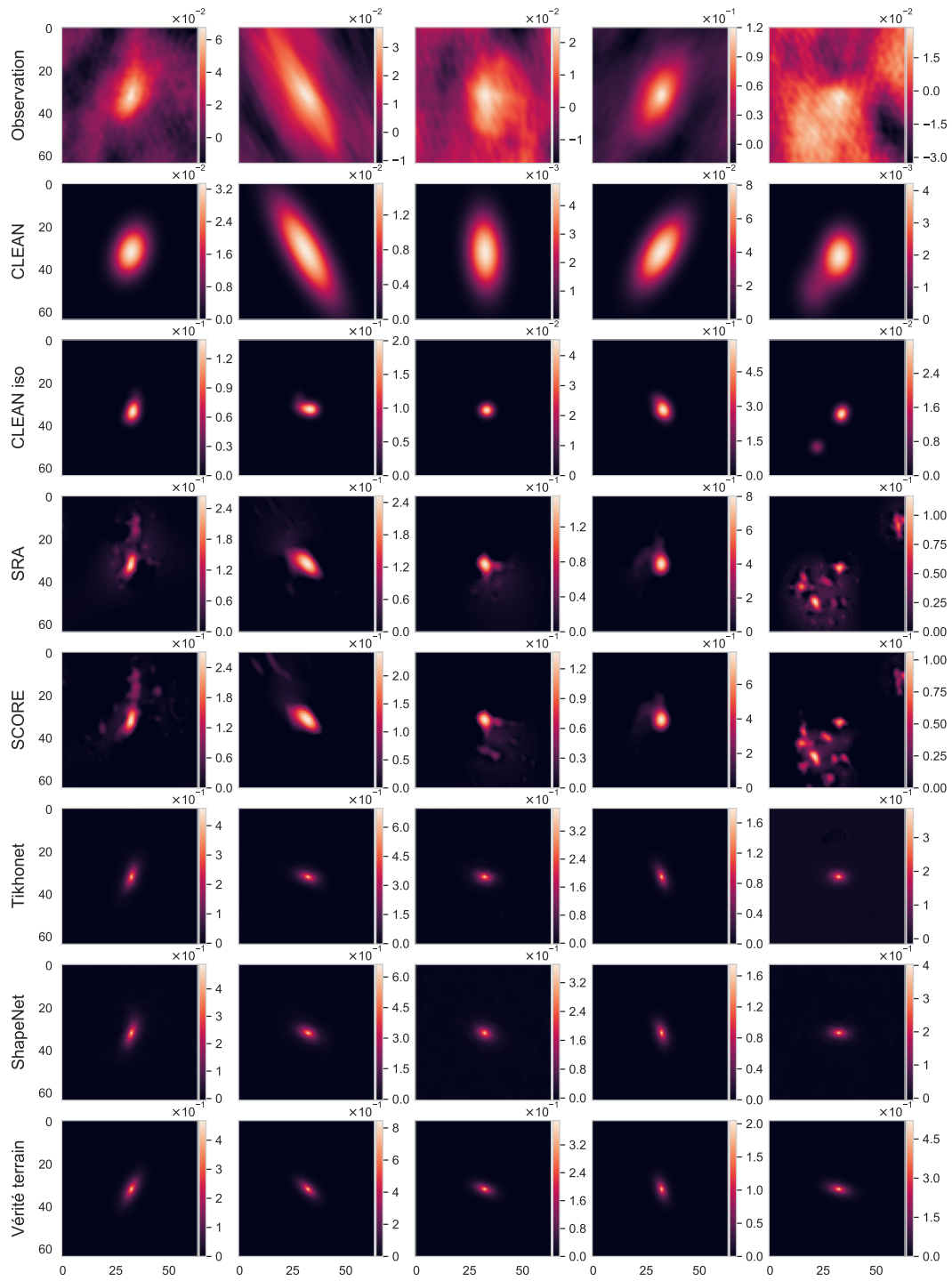
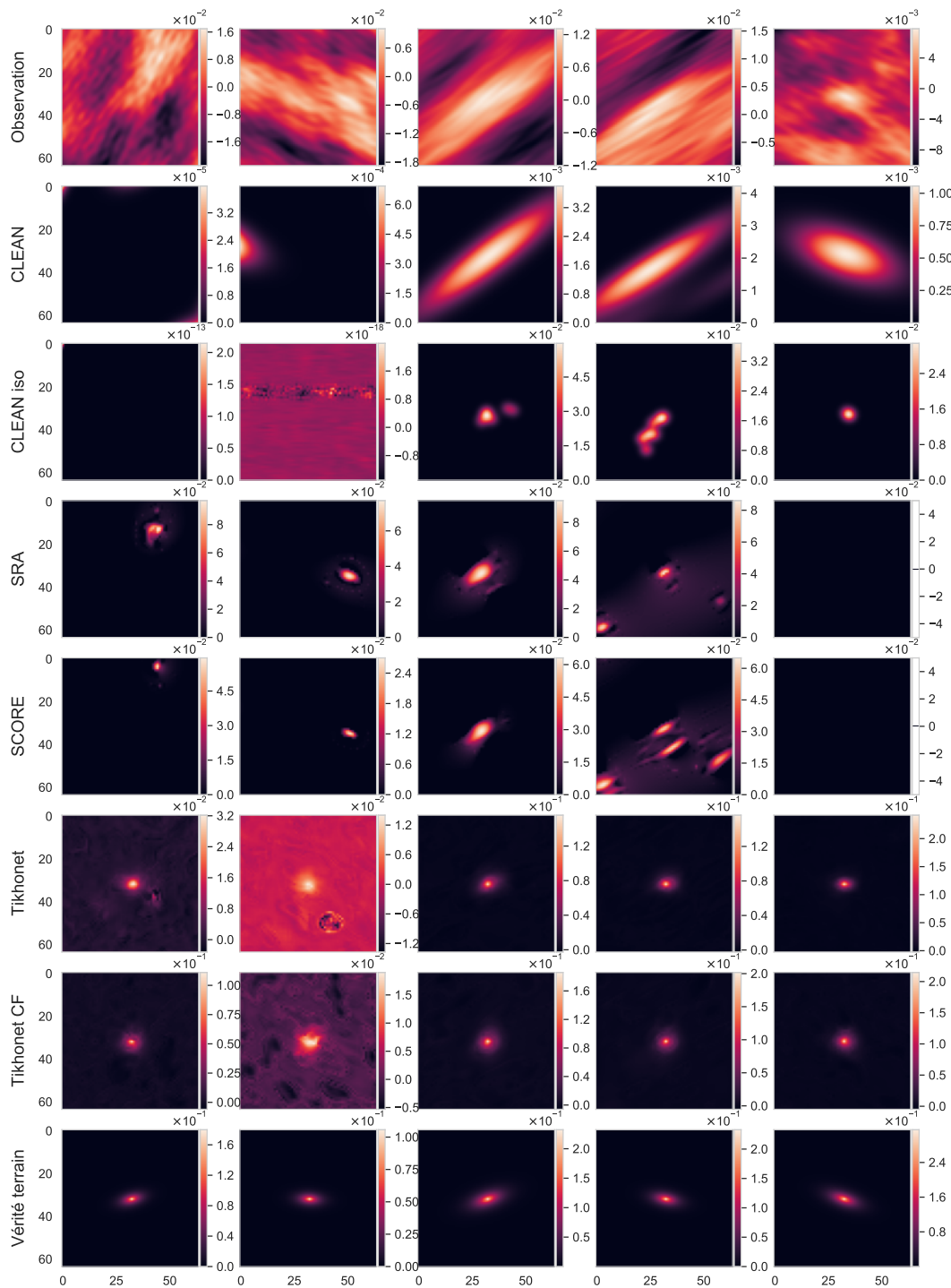
(c) Erreur absolue moyenne de e_1 .(d) Erreur absolue moyenne de e_2 .

Figure 5.8 – Erreurs moyennes de reconstruction en fonction de la magnitude pour le jeu de données CFHT2HST.

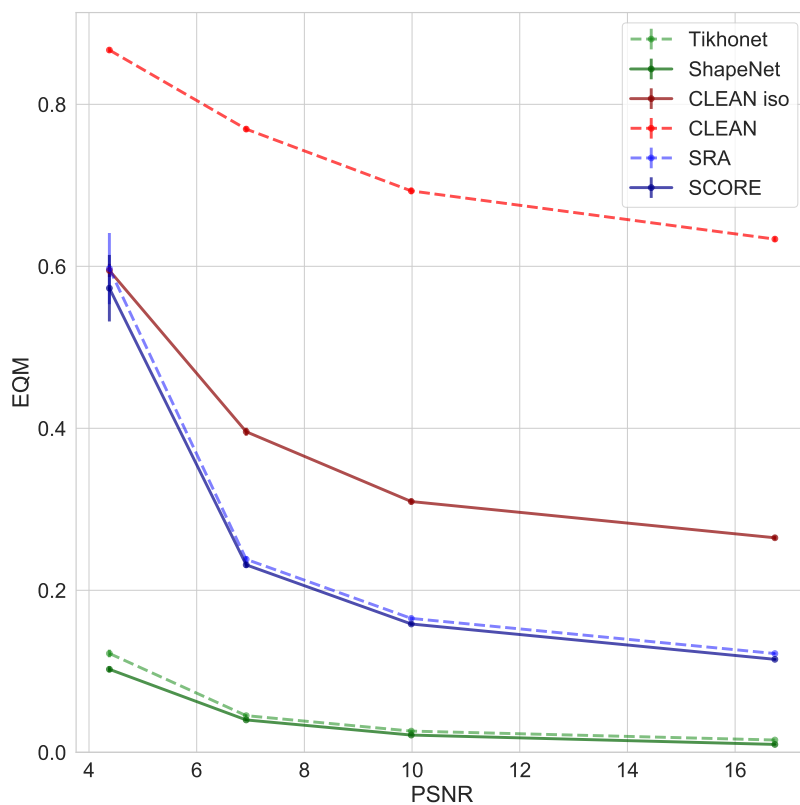


(a) Galaxies radio avec PSNR supérieur à 3.

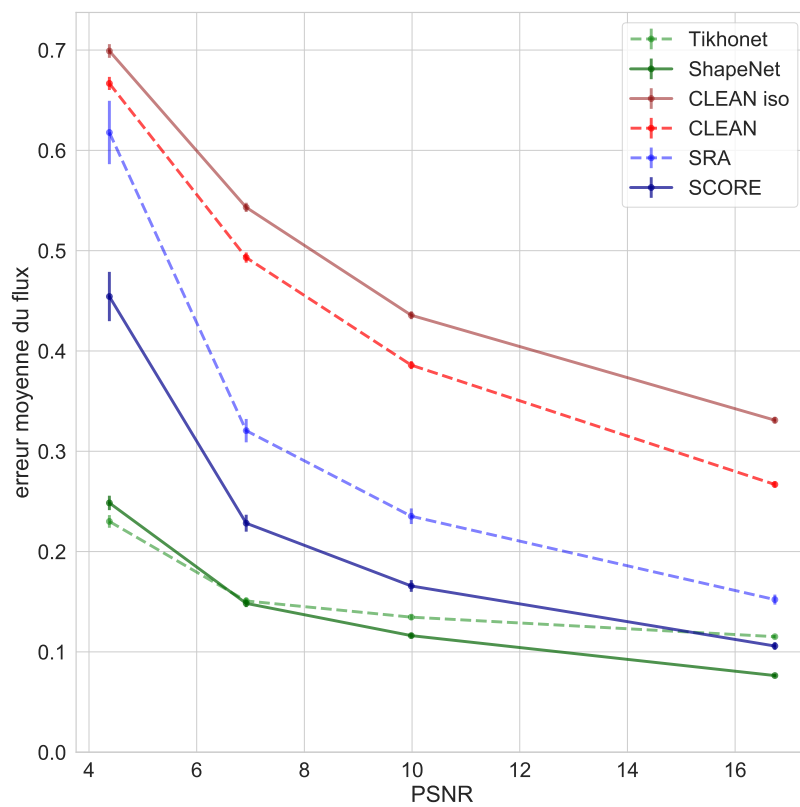


(b) Galaxies radio avec PSNR inférieur à 3.

Figure 5.9 – Exemples de galaxies reconstruites avec le jeu de données MeerKAT3600.



(a) EQM relative moyenne.



(b) Erreur relative moyenne du flux.

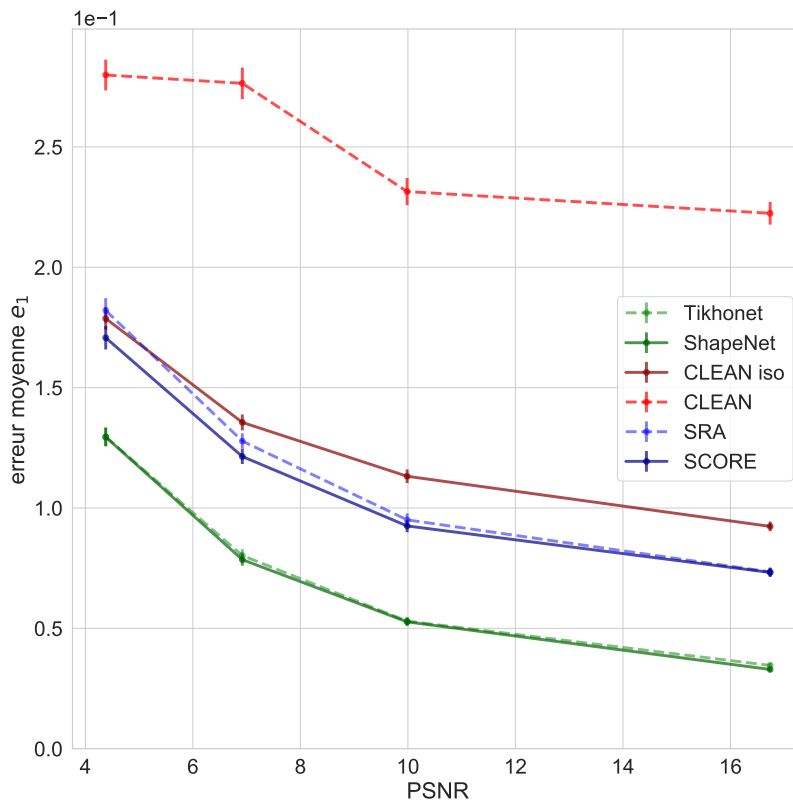
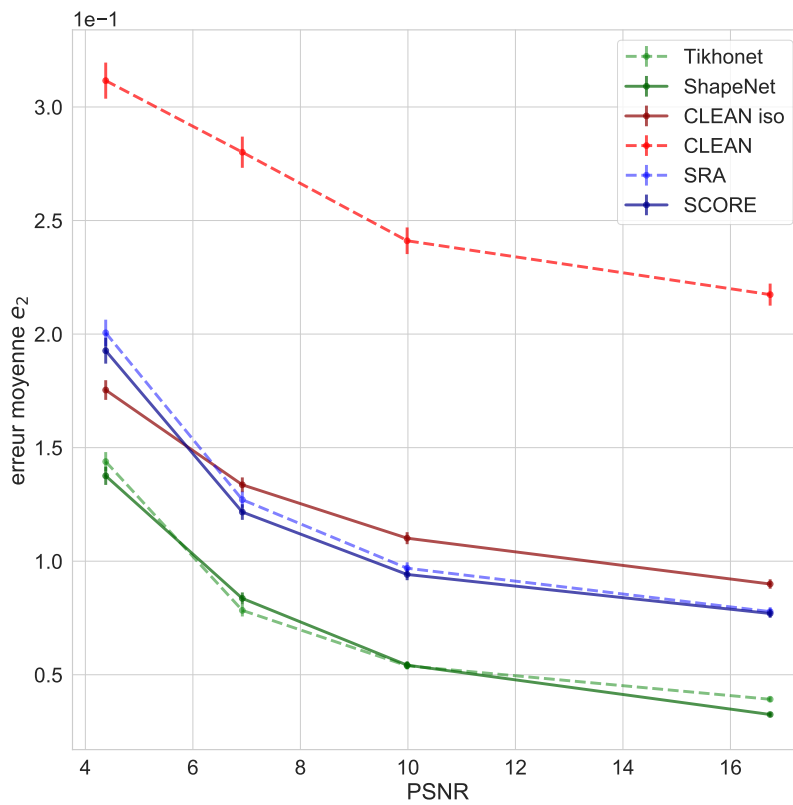
(c) Erreur absolue moyenne de e_1 .(d) Erreur absolue moyenne de e_2 .

Figure 5.10 – Erreurs moyennes de reconstruction en fonction de la magnitude pour le jeu de données MeerKAT3600.

Conclusion

Dans ce manuscrit, j'ai commencé par introduire l'imagerie optique et l'imagerie radio-interférométrique en montrant leur intérêt en astrophysique et en soulignant le problème inverse de déconvolution commun aux deux. Ensuite, j'ai présenté de nouvelles méthodes de reconstruction d'images permettant une meilleure préservation de l'information physique contenue dans ces images, en comparaison avec les méthodes standards. J'ai développé une contrainte de forme spécifique à notre problème de déconvolution, je l'ai implémentée dans un cadre parcimonieux et un cadre d'apprentissage profond et j'ai comparé les méthodes obtenues sur des jeux de données optiques et radio-interférométriques que j'ai créés pour les besoins de l'étude.

Premièrement, j'ai commencé par préciser le cadre de mon étude qui se limite à l'estimation des propriétés physiques des galaxies à partir des données collectées par les télescopes optiques et radio. Ces estimations sont essentielles pour certains domaines en astrophysique tels que l'évolution des galaxies et la cosmologie, et requièrent une haute précision. La présence de déformations (induites par la forme de la FEP) et de bruit dans les données nous mène à étudier un problème inverse de déconvolution commun aux cas optique et radio. À cette fin, j'ai abordé les éléments de base des problèmes inverses et de leur régularisation en me focalisant sur le cas de la déconvolution. Ce dernier est mal conditionné et possède un vaste espace de solutions.

Deuxièmement, j'ai développé la contrainte de forme qui est une régularisation spécifique à notre problème d'intérêt permettant de restreindre l'espace des solutions. J'ai commencé par formuler la contrainte et la réécrire afin d'obtenir une interprétation physique de celle-ci qui correspond à une repondération du terme d'attache aux données. La réécriture du problème m'a également permis d'extraire des propriétés intéressantes telles que celles portant sur le gradient de la contrainte et la constante de Lipschitz de celui-ci. Tous les paramètres utilisés dans la contrainte de forme ont été déterminés avec des considérations liées au problème à l'exception du paramètre de compromis, γ . Il est envisageable d'effectuer une étude dans le but de développer une méthode qui permet de déterminer automatiquement la valeur optimale de γ suivant les cas, en commençant par étudier quantitativement le niveau de bruit dans les images observées et l'amplitude des pixels puis en associant les contraintes supplémentaires induites par la FEP.

Troisièmement, j'ai évalué la contrainte de forme dans un cadre contrôlé pour établir une preuve de concept. En effet, j'ai détaillé la notion de parcimonie pour ensuite introduire l'algorithme de déconvolution parcimonieuse SRA. Puis j'ai développé l'algorithme SCORE en ajoutant la contrainte de forme à SRA. Enfin, j'ai évalué l'effet de l'ajout de la contrainte de forme en considérant le jeu de données paramétrique COSMOS que j'ai généré de sorte que les propriétés des images le composant soient contrôlées. Les expériences numériques ont été séparées en deux, débruitage d'une part et déconvolution d'autre part, pour mieux cerner l'effet de la contrainte dans une approche systématique. Dans les deux expériences, l'ajout de la contrainte de forme réduit l'erreur de la mesure de forme d'au moins 44%. Par contre, l'erreur pixellique augmente de 28,5% dans le cas du débruitage et diminue de 16,9% dans le cas de la déconvolution. Ces résultats sont détaillés dans [Nammour et al. 2021].

Quatrièmement, je me suis penché sur l'apprentissage machine en commençant par donner un aperçu sur ce domaine et en mettant en exergue l'apprentissage profond et les CNN. Ensuite, j'ai réimplémenté les méthodes Tikhonet et ADMMnet, auxquelles j'ai ajouté la contrainte de forme. Comme l'implémentation de la méthode ADMMnet présente des instabilités, j'ai uniquement évalué le Tikhonet dans la suite de mon étude. Pour ce faire, j'ai développé deux jeux de données réalistes, CFHT2HST, pour l'optique, et MeerKAT3600, pour la radio-interférométrie, qui sont adaptés aux besoins de l'apprentissage profond vu leur taille et leur implémentation. Il est possible d'enrichir ces jeux de données en y effectuant une augmentation de données pour améliorer la qualité de l'apprentissage en évitant le sur-entraînement et favorisant l'apprentissage d'invariances par rapport à la translation et la rotation par exemple. Les expériences numériques ont appuyé les résultats énoncés dans le papier original du Tikhonet en montrant que l'approche parcimonieuse est dépassée par l'apprentissage profond. Les expériences étendent également l'étude de la contrainte de forme et du Tikhonet pour la première fois sur le domaine radio. Les résultats montrent que dans les domaines optique et radio, le passage des méthodes de reconstruction parcimonieuse à l'apprentissage profond améliore la qualité de la reconstruction en considérant l'erreur pixellique, l'erreur de forme et l'erreur du flux. De même l'ajout de la contrainte de forme en déconvolution améliore également la qualité de la reconstruction en considérant les mêmes critères. Un papier est en cours de rédaction [Nammour et al., in prep.] pour rendre compte de ces résultats.

Par ailleurs, durant ma thèse, j'ai eu l'occasion d'effectuer différentes activités scientifiques liées à la communication et l'enseignement, celles-ci sont détaillées dans l'annexe D.

Perspectives

Finalement, plusieurs pistes sont à explorer en perspectives. En apprentissage profond, Il est possible d'étudier l'ajout de la contrainte de forme dans l'étape de déconvolution pour améliorer la méthode ShapeNet. Il est également envisageable de remplacer l'architecture U-net par une autre qui soit plus performante telle que DIDN afin d'optimiser l'étape de débruitage. Une étude plus exploratoire consiste à stabiliser notre implémentation de l'ADMMnet en investiguant la normalisation spectrale et l'utilisation des algorithmes unrolled.

D'ailleurs, en astrophysique, il est possible d'étendre le problème de déconvolution de deux manières différentes. La première est de considérer le problème de déconvolution multi-époques qui revient à reconstruire un objet à partir des observations de celui-ci prise à différentes époques avec potentiellement des paramètres différents (télescope, temps de pose, etc.). La deuxième manière est de considérer le problème de déconvolution multi-fréquences. Ainsi, les données manipulées ne sont plus des images 2D mais seront sous la forme de cubes de données où la troisième dimension est la fréquence (ou le temps), pour plus de détails voir [Ammanouil et al. 2019]. Dans ce cas, il est intéressant de développer une méthode qui permettrait d'utiliser explicitement l'information fréquentielle et le lien entre les images d'une fréquence voisine l'une à l'autre. Une approche exploratoire est d'utiliser des méthodes récurrentes comme les Réseaux de Neurones Récursifs (Recursive Neural Networks (RNN), en anglais) ou des *Transformers* [Dosovitskiy et al. 2021 ; Vaswani et al. 2017]. Effectivement, dans le cas des Transformers, le *mécanisme d'attention* pourrait permettre d'apprendre la dépendance fréquentielle entre les différentes images d'un cube, en apprenant les zones d'intérêt dans les images.

Dernièrement, en cosmologie, la méthode de référence actuellement utilisée en estimation du signal de lentillage gravitationnel faible, *Metacalibration* [Sheldon et Huff 2017] effectue une étape de déconvolution à l'aide d'une simple division dans l'espace de Fourier de la transformée de l'image observée par celle de la FEP. Il serait intéressant d'étudier l'effet de l'utilisation du ShapeNet comme méthode de déconvolution dans le schéma numérique de la Metacalibration.

Bibliographie

- Abadi, Martín et al. (2015). *TensorFlow : Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. url : <https://www.tensorflow.org/>.
- Aharon, Michal, Michael Elad et Alfred Bruckstein (2006). « K-SVD : An algorithm for designing overcomplete dictionaries for sparse representation ». In : *IEEE Transactions on signal processing* 54.11, p. 4311-4322.
- Ammanouil, R et al. (août 2019). « A parallel and automatically tuned algorithm for multi-spectral image deconvolution ». In : *Monthly Notices of the Royal Astronomical Society* 490.1, p. 37-49. issn : 0035-8711. doi : 10.1093/mnras/stz2193. eprint : <https://academic.oup.com/mnras/article-pdf/490/1/37/30096950/stz2193.pdf>. url : <https://doi.org/10.1093/mnras/stz2193>.
- Anish Roshi, D et al. (2021). « The Future Of The Arecibo Observatory : The Next Generation Arecibo Telescope ». In : *arXiv e-prints*, arXiv-2103.
- Ba, Lei Jimmy, Jamie Ryan Kiros et Geoffrey E. Hinton (2016). « Layer Normalization ». In : *CoRR* abs/1607.06450. arXiv : 1607.06450. url : <http://arxiv.org/abs/1607.06450>.
- Bartelmann, M. et P. Schneider (jan. 2001). « Weak gravitational lensing ». In : *Phys. Rep.* 340.4-5, p. 291-472. doi : 10.1016/S0370-1573(00)00082-X. arXiv : astro-ph/9912508 [astro-ph].
- Basheer, Imad et M.N. Hajmeer (jan. 2001). « Artificial Neural Networks : Fundamentals, Computing, Design, and Application ». In : *Journal of microbiological methods* 43, p. 3-31. doi : 10.1016/S0167-7012(00)00201-3.
- Beck, Amir et Marc Teboulle (2009). « A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems ». In : *SIAM Journal on Imaging Sciences* 2.1, p. 183-202. doi : 10.1137/080716542. eprint : <https://doi.org/10.1137/080716542>. url : <https://doi.org/10.1137/080716542>.
- Beckouche, S., J. L. Starck et J. Fadili (août 2013). « Astronomical image denoising using dictionary learning ». In : *A&A* 556, A132, A132. doi : 10.1051/0004-6361/201220752. arXiv : 1304.3573 [astro-ph.IM].
- Benbrahim, Hamid et Judy A Franklin (1997). « Biped dynamic walking using reinforcement learning ». In : *Robotics and Autonomous Systems* 22.3-4, p. 283-302.
- Bergeaud, F. et S. Mallat (1995). « Matching pursuit of images ». In : *Proceedings., International Conference on Image Processing*. T. 1, 53-56 vol.1. doi : 10.1109/ICIP.1995.529037.
- Bernstein, Gary M. et Robert Armstrong (jan. 2014a). « Bayesian lensing shear measurement ». In : *Monthly Notices of the Royal Astronomical Society* 438.2, 1880-1893. issn : 1365-2966. doi : 10.1093/mnras/stt2326. url : <http://dx.doi.org/10.1093/mnras/stt2326>.
- Bernstein, Gary M et Robert Armstrong (2014b). « Bayesian lensing shear measurement ». In : *Monthly Notices of the Royal Astronomical Society* 438.2, p. 1880-1893.
- Bertero, Mario et Patrizia Boccacci (2020). *Introduction to inverse problems in imaging*. CRC press.

- Blumensath, Thomas et Mike E Davies (2009). « Iterative hard thresholding for compressed sensing ». In : *Applied and computational harmonic analysis* 27.3, p. 265-274.
- Bock, Sebastian, Josef Goppold et Martin Georg Weiß (2018). « An improvement of the convergence proof of the ADAM-Optimizer ». In : *CoRR* abs/1804.10587. arXiv : 1804.10587. url : <http://arxiv.org/abs/1804.10587>.
- Bonaldi, Anna et al. (juin 2019). *T-RECS : Tiered Radio Extragalactic Continuum Simulation*. ascl : 1906.008.
- Booth, RS et JL Jonas (2012). « An overview of the MeerKAT project ». In : *African Skies* 16, p. 101.
- Bottou, Léon, Frank E Curtis et Jorge Nocedal (2018). « Optimization methods for large-scale machine learning ». In : *Siam Review* 60.2, p. 223-311.
- Boyd, Stephen, Neal Parikh et Eric Chu (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.
- Boyer, Alexandre (2011). *Antennes*.
- Breiman, Leo (2001). « Random forests ». In : *Machine learning* 45.1, p. 5-32.
- Briggs, D. S. (déc. 1995). « High Fidelity Interferometric Imaging : Robust Weighting and NNLS Deconvolution ». In : *American Astronomical Society Meeting Abstracts*. T. 187. American Astronomical Society Meeting Abstracts, p. 112.02.
- Candès, Emmanuel J et David L Donoho (1999). « Ridgelets : A key to higher-dimensional intermittency ? » In : *Philosophical Transactions of the Royal Society of London. Series A : Mathematical, Physical and Engineering Sciences* 357.1760, p. 2495-2509.
- Candes, Emmanuel J et David L Donoho (2000). *Curvelets : A surprisingly effective non-adaptive representation for objects with edges*. Rapp. tech. Stanford Univ Ca Dept of Statistics.
- Carin, Lawrence (2009). « On the Relationship Between Compressive Sensing and Random Sensor Arrays ». In : *IEEE Antennas and Propagation Magazine* 51.5, p. 72-81. doi : 10.1109/MAP.2009.5432044.
- Carrillo, R. E., J. D. McEwen et Y. Wiaux (oct. 2012). « Sparsity Averaging Reweighted Analysis (SARA) : a novel algorithm for radio-interferometric imaging ». In : *Monthly Notices of the Royal Astronomical Society* 426.2, p. 1223-1234. issn : 0035-8711. doi : 10.1111/j.1365-2966.2012.21605.x. eprint : <https://academic.oup.com/mnras/article-pdf/426/2/1223/2965697/426-2-1223.pdf>. url : <https://doi.org/10.1111/j.1365-2966.2012.21605.x>.
- (fév. 2014). « PURIFY : a new approach to radio-interferometric imaging ». In : *Monthly Notices of the Royal Astronomical Society* 439.4, p. 3591-3604. issn : 0035-8711. doi : 10.1093/mnras/stu202. eprint : <https://academic.oup.com/mnras/article-pdf/439/4/3591/3975464/stu202.pdf>. url : <https://doi.org/10.1093/mnras/stu202>.
- Carroll, Bradley W. et Dale A. Ostlie (2007). *An Introduction to Modern Astrophysics*. Sous la dir. de San Francisco : Pearson Addison-Wesley. 2nd (International).
- Chambolle, Antonin et Thomas Pock (juin 2010). « A first-order primal-dual algorithm for convex problems with applications to imaging ». working paper or preprint. url : <https://hal.archives-ouvertes.fr/hal-00490826>.
- Chambolle, Antonin et al. (nov. 2009). « An introduction to Total Variation for Image Analysis ». working paper or preprint. url : <https://hal.archives-ouvertes.fr/hal-00437581>.
- Chan, Stanley H., Xiran Wang et Omar A. Elgendy (mai 2016). « Plug-and-Play ADMM for Image Restoration : Fixed Point Convergence and Applications ». In : *arXiv e-prints*, arXiv :1605.01710, arXiv :1605.01710. arXiv : 1605.01710 [cs.CV].

- Chen, Guangyi et Wenfang Xie (juil. 2011). « Wavelet-based moment invariants for pattern recognition ». In : *Optical Engineering* 50.7, 077205-077205-9, p. 077205-077205-9. doi : 10.1117/1.3597329.
- Chollet, François (2015). *keras*. <https://github.com/keras-team/keras/>.
- (2017). « Xception : Deep Learning with Depthwise Separable Convolutions ». In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1800-1807. doi : 10.1109/CVPR.2017.195.
- Clark, BG (1980). « An efficient implementation of the algorithm 'CLEAN' ». In : *Astronomy and Astrophysics* 89, p. 377.
- Cohen, Albert, Ingrid Daubechies et al. (1992). « A stability criterion for biorthogonal wavelet bases and their related subband coding scheme ». In : *Duke Mathematical Journal* 68.2, p. 313-335.
- Combettes, Patrick et Valérie Wajs (jan. 2005). « Signal Recovery by Proximal Forward-Backward Splitting ». In : *Multiscale Modeling & Simulation - MULTISCALE MODEL SIMUL* 4. doi : 10.1137/050626090.
- Combettes, Patrick Louis et Jean-Christophe Pesquet (2011). « Proximal Splitting Methods in Signal Processing ». In : *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Sous la dir. de Bauschke et al. Springer, p. 185-212. doi : 10.1007/978-1-4419-9569-8. url : <https://hal.inria.fr/hal-00643807>.
- Condat, L. (2013). « A generic first-order primal-dual method for convex optimization involving Lipschitzian, proximable and linear composite terms ». In : *J. Optimization Theory and Applications* 158.2, p. 460-479.
- Developers, Google (2019). *The Size and Quality of a Data Set*. <https://developers.google.com/machine-learning/data-prep/construct/collect/data-size-quality>. [En ligne; consulté le 22-Juillet-2021].
- Devlin, Jacob et al. (2018). « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding ». In : *CoRR* abs/1810.04805. arXiv : 1810.04805. url : <http://arxiv.org/abs/1810.04805>.
- Dhahbi, Sami, Walid Barhoumi et Ezzeddine Zagrouba (2015). « Breast cancer diagnosis in digitized mammograms using curvelet moments ». In : *Computers in biology and medicine* 64, p. 79-90.
- Donoho, David L. et al. (1995). « Wavelet Shrinkage : Asymptopia? » In : *Journal of the Royal Statistical Society : Series B (Methodological)* 57.2, p. 301-337. doi : <https://doi.org/10.1111/j.2517-6161.1995.tb02032.x>. eprint : <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1995.tb02032.x>. url : <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1995.tb02032.x>.
- Dosovitskiy, Alexey et al. (2021). « An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale ». In : *International Conference on Learning Representations*. url : <https://openreview.net/forum?id=YicbFdNTTy>.
- Douglas, Jim et Henry H Rachford (1956). « On the numerical solution of heat conduction problems in two and three space variables ». In : *Transactions of the American mathematical Society* 82.2, p. 421-439.
- Durall López, Ricard, Margret Keuper et Janis Keuper (juin 2020). « Watch Your Up-Convolution : CNN Based Generative Deep Neural Networks Are Failing to Reproduce Spectral Distributions ». In : p. 7887-7896. doi : 10.1109/CVPR42600.2020.00791.
- Dyk, David A van et Xiao-Li Meng (2001). « The Art of Data Augmentation ». In : *Journal of Computational and Graphical Statistics* 10.1, p. 1-50. doi : 10.1198/

10618600152418584. eprint : <https://doi.org/10.1198/10618600152418584>. url : <https://doi.org/10.1198/10618600152418584>.
- Event Horizon Telescope Collaboration et al. (avr. 2019). « First M87 Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole ». In : *ApJ* 875.1, L1, p. L1. doi : 10.3847/2041-8213/ab0ec7. arXiv : 1906.11238 [astro-ph.GA].
- Farokhi, Sajad et al. (2014). « Near infrared face recognition by combining Zernike moments and undecimated discrete wavelet transform ». In : *Digital Signal Processing* 31, p. 13-27.
- Farrens, S., F. M. Ngolè Mboula et J. L. Starck (mai 2017). « Space variant deconvolution of galaxy survey images ». In : *A&A* 601, A66, A66. doi : 10.1051/0004-6361/201629709. arXiv : 1703.02305 [astro-ph.IM].
- Flamary, R. (2017). « Astronomical image reconstruction with convolutional neural networks ». In : *2017 25th European Signal Processing Conference (EUSIPCO)*, p. 2468-2472. doi : 10.23919/EUSIPCO.2017.8081654. url : <https://www.eurasip.org/proceedings/eusipco/eusipco2017/papers/1570340787.pdf>.
- Flusser, Jan, Tomas Suk et Barbara Zitova (2016). *2D and 3D Image Analysis by Moments*. John Wiley & Sons.
- Frieden, B. Roy (avr. 1972). « Restoring with Maximum Likelihood and Maximum Entropy ». In : *Journal of the Optical Society of America (1917-1983)* 62.4, p. 511.
- Garsden, H. et al. (mar. 2015). « LOFAR sparse image reconstruction ». In : *A&A* 575, A90, A90. doi : 10.1051/0004-6361/201424504. arXiv : 1406.7242 [astro-ph.IM].
- Grossmann, Alexander et Jean Morlet (1984). « Decomposition of Hardy functions into square integrable wavelets of constant shape ». In : *SIAM journal on mathematical analysis* 15.4, p. 723-736.
- Gu, Shixiang et al. (2016). « Continuous deep q-learning with model-based acceleration ». In : *International Conference on Machine Learning*. PMLR, p. 2829-2838.
- Gurrola-Ramos, Javier, Oscar Dalmau et Teresa E. Alarcón (2021). « A Residual Dense U-Net Neural Network for Image Denoising ». In : *IEEE Access* 9, p. 31742-31754. doi : 10.1109/ACCESS.2021.3061062.
- Haar, Alfred (1910). « Zur Theorie der orthogonalen Funktionensysteme ». In : *Mathematische Annalen* 69.3, p. 331-371. doi : 10.1007/BF01456326. url : <https://hal.archives-ouvertes.fr/hal-01333722>.
- Haarlem, Michael P van et al. (2013). « LOFAR : The low-frequency array ». In : *Astronomy & astrophysics* 556, A2.
- Hadamard, M. (1907). « Les problèmes aux limites dans la théorie des équations aux dérivées partielles ». In : *J. Phys. Theor. Appl.* 6.1, p. 202-241. doi : 10.1051/jphystap:019070060020200. url : <https://hal.archives-ouvertes.fr/jpa-00241199>.
- He, Kaiming et al. (2015). « Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification ». In : *CoRR* abs/1502.01852. arXiv : 1502.01852. url : <http://arxiv.org/abs/1502.01852>.
- Hearst, Marti A. et al. (1998). « Support vector machines ». In : *IEEE Intelligent Systems and their applications* 13.4, p. 18-28.
- Hirata, Christopher et Uroš Seljak (août 2003). « Shear calibration biases in weak-lensing surveys ». In : *MNRAS* 343.2, p. 459-480. doi : 10.1046/j.1365-8711.2003.06683.x. arXiv : astro-ph/0301054 [astro-ph].
- Högbom, JA (1974). « Aperture synthesis with a non-regular distribution of interferometer baselines ». In : *Astronomy and Astrophysics Supplement Series* 15, p. 417.
- Holschneider, Matthias et al. (1990). « A real-time algorithm for signal analysis with the help of the wavelet transform ». In : *Wavelets*. Springer, p. 286-297.

- Hornik, Kurt, Maxwell Stinchcombe et Halbert White (1989). « Multilayer feedforward networks are universal approximators ». In : *Neural Networks* 2.5, p. 359-366. issn : 0893-6080. doi : [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8). url : <https://www.sciencedirect.com/science/article/pii/0893608089900208>.
- Huang, Gao et al. (2017). « Densely Connected Convolutional Networks ». In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 2261-2269. doi : 10.1109/CVPR.2017.243.
- Hunter, John D (2007). « Matplotlib : A 2D graphics environment ». In : *Computing in science & engineering* 9.3, p. 90.
- Hush, D.R. et B.G. Horne (1993). « Progress in supervised neural networks ». In : *IEEE Signal Processing Magazine* 10.1, p. 8-39. doi : 10.1109/79.180705.
- Ioffe, Sergey et Christian Szegedy (2015). « Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift ». In : *CoRR* abs/1502.03167. arXiv : 1502.03167. url : <http://arxiv.org/abs/1502.03167>.
- Jaderberg, Max et al. (2019). « Human-level performance in 3D multiplayer games with population-based reinforcement learning ». In : *Science* 364.6443, p. 859-865.
- Janocha, Katarzyna et Wojciech Czarnecki (fév. 2017). « On Loss Functions for Deep Neural Networks in Classification ». In : *Schedae Informaticae* 25. doi : 10.4467/20838476SI.16.004.6185.
- Jeffrey, Niall et al. (mar. 2020). « Deep learning dark matter map reconstructions from DES SV weak lensing data ». In : *MNRAS* 492.4, p. 5023-5029. doi : 10.1093/mnras/staa127. arXiv : 1908.00543 [astro-ph.CO].
- Jupyter, Project et al. (2018). « Binder 2.0 - Reproducible, interactive, sharable environments for science at scale ». In : *Proceedings of the 17th Python in Science Conference*. Sous la dir. de Fatih Akici et al., p. 113 -120. doi : 10.25080/Majora-4af1f417-011.
- Kaiser, Nick, Gordon Squires et Tom Broadhurst (août 1995). « A Method for Weak Lensing Observations ». In : *ApJ* 449, p. 460. doi : 10.1086/176071. arXiv : astro-ph/9411005 [astro-ph].
- Kern, Michel (sept. 2002). « Problèmes inverses : aspects numériques ». Engineering school. Lecture. 1999 à 2002, École supérieure d'ingénieurs Léonard de Vinci. url : <https://cel.archives-ouvertes.fr/cel-00168393>.
- Kidziński, Łukasz et al. (2018). « Learning to run challenge solutions : Adapting reinforcement learning methods for neuromusculoskeletal environments ». In : *The NIPS'17 Competition : Building Intelligent Systems*. Springer, p. 121-153.
- Kingma, Diederik P. et Jimmy Ba (2015). « Adam : A Method for Stochastic Optimization ». In : *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Sous la dir. d'Yoshua Bengio et Yann LeCun. url : <http://arxiv.org/abs/1412.6980>.
- Kostková, Jitka et al. (2019). « Image Invariants to Anisotropic Gaussian Blur ». In : *Scandinavian Conference on Image Analysis*. Springer, p. 140-151.
- Krizhevsky, Alex, Ilya Sutskever et Geoffrey E Hinton (2012). « ImageNet Classification with Deep Convolutional Neural Networks ». In : *Advances in Neural Information Processing Systems*. Sous la dir. de F. Pereira et al. T. 25. Curran Associates, Inc. url : <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- Kumar, Ahlad (2017). « Deblurring of motion blurred images using histogram of oriented gradients and geometric moments ». In : *Signal Processing : Image Communication* 55, p. 55-65.

- Lam, Hak-Keung et al. (fév. 2001). « Tuning of the structure and parameters of neural network using an improved genetic algorithm ». In : t. 14, 25 -30 vol.1. isbn : 0-7803-7108-9. doi : 10.1109/IECON.2001.976448.
- Lanusse, François, Anais Rassat et J-L Starck (2012). « Spherical 3D isotropic wavelets ». In : *Astronomy & Astrophysics* 540, A92.
- Lanusse, François et al. (mai 2021). « Deep Generative Models for galaxy image simulations ». In : MNRAS. doi : 10.1093/mnras/stab1214. arXiv : 2008.03833 [astro-ph.IM].
- Lanusse, François et Benjamin Remy (2019). *GalFlow*. <https://github.com/DifferentiableUniverseInitiat> GalFlow.
- Lecun, Yann (août 2001). « A Theoretical Framework for Back-Propagation ». In : Lettvin, J. Y. et al. (1959). « What the Frog's Eye Tells the Frog's Brain ». In : *Proceedings of the IRE* 47.11, p. 1940-1951. doi : 10.1109/JRPROC.1959.287207.
- Li, Yuxi (2017). « Deep Reinforcement Learning : An Overview ». In : *CoRR* abs/1701.07274. arXiv : 1701.07274. url : <http://arxiv.org/abs/1701.07274>.
- Lindsay, R. K. et al. (1993). « DENDRAL : A Case Study of the First Expert System for Scientific Hypothesis Formation ». In : *Artif. Intell.* 61, p. 209-261.
- Maas, Andrew L. (2013). « Rectifier Nonlinearities Improve Neural Network Acoustic Models ». In :
- Mallat, Stéphane (1999). *A wavelet tour of signal processing*. Elsevier.
- Mallat, Stéphane (sept. 1989). « Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(\mathbb{R})$ ». In : *Transactions of the American Mathematical Society* 315, p. 69-87. doi : 10.2307/2001373.
- Mandelbaum, Rachel (sept. 2018). « Weak Lensing for Precision Cosmology ». In : *ARA&A* 56, p. 393-433. doi : 10.1146/annurev-astro-081817-051928. arXiv : 1710.03235 [astro-ph.CO].
- Mandelbaum, Rachel et al. (déc. 2011). « Precision simulation of ground-based lensing data using observations from space ». In : *Monthly Notices of the Royal Astronomical Society* 420.2, p. 1518-1540. issn : 0035-8711. doi : 10.1111/j.1365-2966.2011.20138.x. url : <http://dx.doi.org/10.1111/j.1365-2966.2011.20138.x>.
- Mandelbaum, Rachel et al. (déc. 2018). « Weak lensing shear calibration with simulations of the HSC survey ». In : MNRAS 481.3, p. 3170-3195. doi : 10.1093/mnras/sty2420. arXiv : 1710.00885 [astro-ph.CO].
- McLachlan, Geoffrey J, Sharon X Lee et Suren I Rathnayake (2019). « Finite mixture models ». In : *Annual review of statistics and its application* 6, p. 355-378.
- Meinhardt, Tim et al. (2017). « Learning Proximal Operators : Using Denoising Networks for Regularizing Inverse Imaging Problems ». In : *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Merriënboer, Bart van et al. (2018). « Automatic differentiation in ML : Where we are and where we should be going ». In : *CoRR* abs/1810.11530. arXiv : 1810.11530. url : <http://arxiv.org/abs/1810.11530>.
- Meyer, Yves (1993). « Orthonormal wavelet bases ». In : *Wavelets and Operators*. Sous la dir. de D. H. Translator Salinger. T. 1. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 66-129. doi : 10.1017/CB09780511623820.006.
- Michelson, Albert A. (juin 1891). « Visibility of Interference-Fringes in the Focus of a Telescope ». In : *Publications of the Astronomical Society of the Pacific* 3, p. 217. doi : 10.1086/120291. url : <https://doi.org/10.1086/120291>.

- Miller, L. et al. (nov. 2007). « Bayesian galaxy shape measurement for weak lensing surveys - I. Methodology and a fast-fitting algorithm ». In : *MNRAS* 382.1, p. 315-324. doi : 10.1111/j.1365-2966.2007.12363.x. arXiv : 0708.2340 [astro-ph].
- Miyato, Takeru et al. (2018). « Spectral Normalization for Generative Adversarial Networks ». In : *International Conference on Learning Representations*. url : <https://openreview.net/forum?id=B1QRgziT->.
- Mnih, Volodymyr et al. (2013). *Playing Atari with Deep Reinforcement Learning*. arXiv : 1312.5602 [cs.LG].
- Moffat, A. F. J. (déc. 1969). « A Theoretical Investigation of Focal Stellar Images in the Photographic Emulsion and Application to Photographic Photometry ». In : *A&A* 3, p. 455.
- Monga, Vishal, Yuelong Li et Yonina C. Eldar (2020). *Algorithm Unrolling : Interpretable, Efficient Deep Learning for Signal and Image Processing*. arXiv : 1912.10557 [eess.IV].
- Moreau, Jean Jacques (1965). « Proximité et dualité dans un espace hilbertien ». fr. In : *Bulletin de la Société Mathématique de France* 93, p. 273-299. doi : 10.24033/bsmf.1625. url : <http://www.numdam.org/articles/10.24033/bsmf.1625/>.
- Muggleton, Stephen (1991). « Inductive logic programming ». In : *New generation computing* 8.4, p. 295-318.
- Murtagh, Fionn et Jean-Luc Starck (2008). « Wavelet and curvelet moments for image classification : application to aggregate mixture grading ». In : *Pattern Recognition Letters* 29.10, p. 1557-1564.
- Nammour, Fadi et al. (2021). « Galaxy Image Restoration with Shape Constraint ». In : *Journal of Fourier Analysis and Applications* 27.6, p. 88. doi : 10.1007/s00041-021-09880-9. url : <https://doi.org/10.1007/s00041-021-09880-9>.
- Narayan, Ramesh et Rajaram Nityananda (jan. 1986). « Maximum entropy image restoration in astronomy. » In : *ARA&A* 24, p. 127-170. doi : 10.1146/annurev.aa.24.090186.001015.
- Neelamani, Ramesh, Hyeokho Choi et Richard Baraniuk (mar. 2004). « ForWaRD : Fourier-Wavelet Regularized Deconvolution for Ill-Conditioned Systems ». In : *Signal Processing, IEEE Transactions on* 52, p. 418 -433. doi : 10.1109/TSP.2003.821103.
- Nesterov, Yu (2013). « Gradient methods for minimizing composite functions ». In : *Mathematical Programming* 140.1, p. 125-161.
- Ng, Kien-Ming (2002). « A continuation approach for solving nonlinear optimization problems with discrete variables ». In : *Doctor Dissertation, Department of Management Science and Engineering of Stanford University*.
- Oussous, Ahmed et al. (juin 2017). « Big Data Technologies : A Survey ». In : *Journal of King Saud University - Computer and Information Sciences* 30. doi : 10.1016/j.jksuci.2017.06.001.
- Patel, Prina (fév. 2016). « Weak Lensing with Radio Continuum Surveys ». In : *arXiv e-prints*, arXiv :1602.07482, arXiv :1602.07482. arXiv : 1602.07482 [astro-ph.CO].
- Patel, Prina et al. (nov. 2014). « Weak lensing measurements in simulations of radio images ». In : *MNRAS* 444.3, p. 2893-2909. doi : 10.1093/mnras/stu1588. arXiv : 1303.4650 [astro-ph.CO].
- Patel, Sakshi, Shivani Sihmar et Aman Jatain (2015). « A study of hierarchical clustering algorithms ». In : *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, p. 537-541.
- Plotkin, G. D. (1971). « Automatic Methods of Inductive Inference ». Thèse de doct. Edinburgh University.

- Qiao, Siyuan et al. (2019). « Weight Standardization ». In : *CoRR* abs/1903.10520. arXiv : 1903.10520. url : <http://arxiv.org/abs/1903.10520>.
- Racine, Rene (1996). « The telescope point spread function ». In : *Publications of the Astronomical Society of the Pacific* 108.726, p. 699.
- Raguét, Hugo, Jalal M. Fadili et Gabriel Peyré (2013). « A Generalized Forward-Backward Splitting ». In : *SIAM Journal on Imaging Sciences* 6.13, p. 1199-1226. doi : 10.1137/120872802. url : <https://hal.archives-ouvertes.fr/hal-00613637>.
- Ramzi, Zaccharie et al. (avr. 2021a). « NC-PDNet : a Density-Compensated Unrolled Network for 2D and 3D non-Cartesian MRI Reconstruction ». This work is an extended version of the work presented at the 2021 ISBI conference. url : <https://hal.inria.fr/hal-03188997>.
- Ramzi, Zaccharie et al. (2021b). « Wavelets in the Deep Learning ERA ». In : *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, p. 1417-1421.
- Reynolds, Douglas A (2009). « Gaussian Mixture Models. » In : *Encyclopedia of biometrics* 741, p. 659-663.
- Rivi, M. et al. (déc. 2016). « Radio weak lensing shear measurement in the visibility domain - I. Methodology ». In : *MNRAS* 463.2, p. 1881-1890. doi : 10.1093/mnras/stw2041. arXiv : 1603.04784 [astro-ph.IM].
- Rivi, M. et al. (jan. 2019). « Radio galaxy shape measurement with Hamiltonian Monte Carlo in the visibility domain ». In : *MNRAS* 482.1, p. 1096-1109. doi : 10.1093/mnras/sty2700. arXiv : 1805.06799 [astro-ph.IM].
- Rodríguez, Silvio, Nelson D. Padilla et Diego García Lambas (fév. 2016). « The effects of environment on the intrinsic shape of galaxies ». In : *MNRAS* 456.1, p. 571-577. doi : 10.1093/mnras/stv2660. arXiv : 1511.03273 [astro-ph.GA].
- Ronneberger, Olaf, Philipp Fischer et Thomas Brox (2015). « U-Net : Convolutional Networks for Biomedical Image Segmentation ». In : *CoRR* abs/1505.04597. arXiv : 1505.04597. url : <http://arxiv.org/abs/1505.04597>.
- Rosenblatt, Frank (1958). « The perceptron : a probabilistic model for information storage and organization in the brain. » In : *Psychological review* 65.6, p. 386.
- Rowe, B. T. P. et al. (avr. 2015). « GALSIM : The modular galaxy image simulation toolkit ». In : *Astronomy and Computing* 10, p. 121-150. doi : 10.1016/j.ascom.2015.02.002. arXiv : 1407.7676 [astro-ph.IM].
- Ryu, Ernest et al. (2019). « Plug-and-Play Methods Provably Converge with Properly Trained Denoisers ». In : *Proceedings of the 36th International Conference on Machine Learning*. Sous la dir. de Kamalika Chaudhuri et Ruslan Salakhutdinov. T. 97. Proceedings of Machine Learning Research. PMLR, p. 5546-5557. url : <https://proceedings.mlr.press/v97/ryu19a.html>.
- Salimans, Tim et Diederik P. Kingma (2016). « Weight Normalization : A Simple Reparameterization to Accelerate Training of Deep Neural Networks ». In : *CoRR* abs/1602.07868. arXiv : 1602.07868. url : <http://arxiv.org/abs/1602.07868>.
- Sallab, Ahmad EL et al. (2017). « Deep reinforcement learning framework for autonomous driving ». In : *Electronic Imaging* 2017.19, p. 70-76.
- Scaife, AMM (2020). « Big telescope, big data : towards exascale with the Square Kilometre Array ». In : *Philosophical Transactions of the Royal Society A* 378.2166, p. 20190060.
- Schawinski, Kevin et al. (mai 2017). « Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit ». In : *MNRAS* 467.1, p. L110-L114. doi : 10.1093/mnrasl/slx008. arXiv : 1702.00403 [astro-ph.IM].

- Schmidt, Robin M., Frank Schneider et Philipp Hennig (2020). « Descending through a Crowded Valley - Benchmarking Deep Learning Optimizers ». In : *CoRR abs/2007.01547*. arXiv : 2007.01547. url : <https://arxiv.org/abs/2007.01547>.
- Schwab, FR (1984). « Relaxing the isoplanatism assumption in self-calibration ; applications to low-frequency radio interferometry ». In : *The Astronomical Journal* 89, p. 1076-1081.
- Sérsic, J. L. (fév. 1963). « Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy ». In : *Boletin de la Asociacion Argentina de Astronomia La Plata Argentina* 6, p. 41-43.
- Shakibaei, Barmak Honarvar et Jan Flusser (2014). « Image deconvolution in the moment domain ». In : *Moments and Moment Invariants—Theory and Applications* 1, p. 111-125.
- Sheldon, Erin S. et Eric M. Huff (mai 2017). « Practical Weak-lensing Shear Measurement with Metacalibration ». In : *ApJ* 841.1, 24, p. 24. doi : 10.3847/1538-4357/aa704b. arXiv : 1702.02601 [astro-ph.CO].
- Shensa, M.J. (1992). « The discrete wavelet transform : wedding the a trous and Mallat algorithms ». In : *IEEE Transactions on Signal Processing* 40.10, p. 2464-2482. doi : 10.1109/78.157290.
- Shorten, Connor et Taghi M. Khoshgoftaar (2019). « A survey on Image Data Augmentation for Deep Learning ». In : *Journal of Big Data* 6.1, p. 60. issn : 2196-1115. doi : 10.1186/s40537-019-0197-0. url : <https://doi.org/10.1186/s40537-019-0197-0>.
- Silver, David et al. (2018). « A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play ». In : *Science* 362.6419, p. 1140-1144.
- Skodras, A., C. Christopoulos et T. Ebrahimi (2001). « The JPEG 2000 still image compression standard ». In : *IEEE Signal Processing Magazine* 18.5, p. 36-58. doi : 10.1109/79.952804.
- Sola, J. et J. Sevilla (1997). « Importance of input data normalization for the application of neural networks to complex industrial problems ». In : *IEEE Transactions on Nuclear Science* 44.3, p. 1464-1468. doi : 10.1109/23.589532.
- Starck, J. L. et al. (fév. 2006). « Wavelets, ridgelets and curvelets on the sphere ». In : *A&A* 446.3, p. 1191-1204. doi : 10.1051/0004-6361:20053246. arXiv : astro-ph/0509883 [astro-ph].
- Starck, Jean-Luc (2002). « Nonlinear multiscale transforms ». In : *Multiscale and multiresolution methods*. Springer, p. 239-278.
- Starck, Jean-Luc, E. J. Candes et D. L. Donoho (juin 2002). « The curvelet transform for image denoising ». In : *IEEE Transactions on Image Processing* 11.6, p. 670-684. doi : 10.1109/TIP.2002.1014998.
- Starck, Jean-Luc, Jalal Fadili et Fionn Murtagh (2007). « The Undecimated Wavelet Decomposition and its Reconstruction ». In : *IEEE Transactions on Image Processing* 16.2, p. 297-309. doi : 10.1109/TIP.2006.887733.
- Starck, Jean-Luc et Fionn Murtagh (2006). *Astronomical Image and Data Analysis*. doi : 10.1007/978-3-540-33025-7.
- Starck, Jean-Luc, Fionn Murtagh et Jalal Fadili (2015). *Sparse image and signal processing : Wavelets and related geometric multiscale analysis*. Cambridge university press.
- Steinhaus, Hugo (1956). « Sur la division des corps matériels en parties ». In : *Bull. Acad. Polon. Sci* 1.804, p. 801.

- Sun, Chen et al. (2017). « Revisiting unreasonable effectiveness of data in deep learning era ». In : *Proceedings of the IEEE international conference on computer vision*, p. 843-852.
- Sureau, F., A. Lechat et J. L. Starck (sept. 2020). « Deep learning for a space-variant deconvolution in galaxy surveys ». In : *A&A* 641, A67, A67. doi : 10.1051/0004-6361/201937039. arXiv : 1911.00443 [astro-ph.IM].
- Szandala, Tomasz (2020). « Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks ». In : *CoRR* abs/2010.09458. arXiv : 2010.09458. url : <https://arxiv.org/abs/2010.09458>.
- Tatarskii, Valerian Ilich (1961). *Wave Propagation in Turbulent Medium*.
- Taylor, G.B. et al. (1999). *Synthesis imaging in radio astronomy II : a collection of lectures from the Sixth NRAO/NMIMT Synthesis Imaging Summer School, held at Socorro, New Mexico, USA, 17-23 June, 1998*. Astronomical Society of the Pacific conference series vol. 2. Astronomical Society of the Pacific. Chap. 1. isbn : 9781583810057. url : <http://books.google.fr/books?id=e23vAAAAMAAJ>.
- Thompson, A Richard, James M Moran et George W Swenson (2017). *Interferometry and synthesis in radio astronomy*. Springer Nature.
- Trujillo, I et al. (2001). « The effects of seeing on Sérsic profiles—II. The Moffat PSF ». In : *Monthly Notices of the Royal Astronomical Society* 328.3, p. 977-985.
- Turing, A. M. (oct. 1950). « I.—COMPUTING MACHINERY AND INTELLIGENCE ». In : *Mind* LIX.236, p. 433-460. issn : 0026-4423. doi : 10.1093/mind/LIX.236.433. eprint : <https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf>. url : <https://doi.org/10.1093/mind/LIX.236.433>.
- Vaswani, Ashish et al. (2017). « Attention is All you Need ». In : *Advances in Neural Information Processing Systems*. Sous la dir. d'I. Guyon et al. T. 30. Curran Associates, Inc. url : <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- Ven, Leni et Johannes Lederer (2021). « Regularization and Reparameterization Avoid Vanishing Gradients in Sigmoid-Type Networks ». In : *CoRR* abs/2106.02260. arXiv : 2106.02260. url : <https://arxiv.org/abs/2106.02260>.
- Venkatakrishnan, Singanallur V., Charles A. Bouman et Brendt Wohlberg (2013). « Plug-and-Play priors for model based reconstruction ». In : *2013 IEEE Global Conference on Signal and Information Processing*, p. 945-948. doi : 10.1109/GlobaISIP.2013.6737048.
- Vetterli, Martin (1986). « Filter banks allowing perfect reconstruction ». In : *Signal processing* 10.3, p. 219-244.
- Viola, M., P. Melchior et M. Bartelmann (fév. 2011). « Biases in, and corrections to, KSB shear measurements ». In : *MNRAS* 410.4, p. 2156-2166. doi : 10.1111/j.1365-2966.2010.17589.x. arXiv : 1006.2470 [astro-ph.CO].
- Vũ, B. (2013). « A splitting algorithm for dual monotone inclusions involving cocoercive operators ». In : *Advances in Computational Mathematics* 38.3, p. 667-681.
- Wallace, Gregory K. (avr. 1991). « The JPEG Still Picture Compression Standard ». In : *Commun. ACM* 34.4, 30-44. issn : 0001-0782. doi : 10.1145/103085.103089. url : <https://doi.org/10.1145/103085.103089>.
- Watkins, Christopher JCH et Peter Dayan (1992). « Q-learning ». In : *Machine learning* 8.3-4, p. 279-292.
- Weiss, P., L. Blanc-Féraud et G. Aubert (2009). « Efficient schemes for total variation minimization under constraints in image processing ». In : *SIAM Journal on Scientific Computing* 31.3. 2047-2080.

- Wernecke, S. J. (oct. 1977). « Two-dimensional maximum entropy reconstruction of radio brightness. » In : *Radio Science* 12, p. 831-844. doi : 10.1029/RS012i005p00831.
- Wiaux, Y. et al. (mai 2009). « Compressed sensing imaging techniques for radio interferometry ». In : *Monthly Notices of the Royal Astronomical Society* 395.3, p. 1733-1742. issn : 0035-8711. doi : 10.1111/j.1365-2966.2009.14665.x. eprint : <https://academic.oup.com/mnras/article-pdf/395/3/1733/18229832/mnras0395-1733.pdf>. url : <https://doi.org/10.1111/j.1365-2966.2009.14665.x>.
- Woiselle, Arnaud, J-L Starck et J Fadili (2010). « 3D curvelet transforms and astronomical data restoration ». In : *Applied and Computational Harmonic Analysis* 28.2, p. 171-188.
- Wojak, Julien, Elsa D Angelini et Isabelle Bloch (2010). « Introducing shape constraint via Legendre moments in a variational framework for cardiac segmentation on non-contrast CT images ». In : *VISAPP*, p. 209-214.
- Wolpert, David H et William G Macready (1997). « No free lunch theorems for optimization ». In : *IEEE transactions on evolutionary computation* 1.1, p. 67-82.
- Yanling, Zhao, Deng Bimin et Wang Zhanrong (2002). « Analysis and study of perceptron to solve XOR problem ». In : *The 2nd International Workshop on Autonomous Decentralized System, 2002*. P. 168-173. doi : 10.1109/IWADS.2002.1194667.
- Yu, Songhyun, Bumjun Park et Jechang Jeong (2019). « Deep iterative down-up CNN for image denoising ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Annexe A

Expression de l'ellipticité des galaxies avec des produits scalaires

Dans cet appendice, nous allons montrer comment nous pouvons réexprimer l'ellipticité d'une galaxie à l'aide de produits scalaires. Pour cela, considérons l'image de galaxie $x \in \mathbb{R}^{n \times n}$ alors son ellipticité s'écrit :

$$e(x) = \frac{\mu_{2,0}(x) - \mu_{0,2}(x) + 2i\mu_{1,1}}{\mu_{2,0}(x) + \mu_{0,2}(x)} , \quad (\text{A.1})$$

avec $\mu_{s,t}$ le moment centré d'ordre $(s + t)$, défini comme suit :

$$\mu_{s,t}(x) = \sum_{i=1}^n \sum_{j=1}^n x [(i-1)n + j] (i - i_c)^s (j - j_c)^t , \quad (\text{A.2})$$

et (i_c, j_c) les coordonnées du centroïde de l'image 2D codée par x :

$$i_c = \frac{\sum_{i=1}^n \sum_{j=1}^n i \cdot x [(i-1)n + j]}{\sum_{i=1}^n \sum_{j=1}^n x [(i-1)n + j]} \text{ and } j_c = \frac{\sum_{i=1}^n \sum_{j=1}^n j \cdot x [(i-1)n + j]}{\sum_{i=1}^n \sum_{j=1}^n x [(i-1)n + j]} . \quad (\text{A.3})$$

Notre but est de montrer que :

$$e(x) = \frac{\langle x, u_3 \rangle \langle x, u_5 \rangle - \langle x, u_1 \rangle^2 + \langle x, u_2 \rangle^2 + 2i (\langle x, u_3 \rangle \langle x, u_6 \rangle - \langle x, u_1 \rangle \langle x, u_2 \rangle)}{\langle x, u_3 \rangle \langle x, u_4 \rangle - \langle x, u_1 \rangle^2 - \langle x, u_2 \rangle^2} , \quad (\text{A.4})$$

avec $\forall i, j \in \{1, \dots, n\}$,

$$\begin{aligned} u_1[(i-1)n + j] &= (i), & u_2[(i-1)n + j] &= (j), \\ u_3[(i-1)n + j] &= (1), & u_4[(i-1)n + j] &= (i^2 + j^2), \\ u_5[(i-1)n + j] &= (i^2 - j^2), & u_6[(i-1)n + j] &= (ij). \end{aligned} \quad (\text{A.5})$$

Pour ce faire, nous devenons exprimer seulement $\mu_{0,2}(x)$; $\mu_{1,1}(x)$ et $\mu_{2,0}(x)$ en fonction de $(\langle x, u_i \rangle)_{1 \leq i \leq 6}$. Nous commençons par introduire $m_{s,t}(x)$, le moment non-centré d'ordre $(s + t)$,

$$m_{s,t} = \sum_{i=1}^n \sum_{j=1}^n x [(i-1)n + j] i^s j^t . \quad (\text{A.6})$$

Nous exprimons ensuite $\mu_{0,2}(x)$; $\mu_{1,1}(x)$ et $\mu_{2,0}(x)$ en fonction des moments non-centrés (d'ordre inférieur ou égal à 2), comme :

$$\begin{aligned}\mu_{0,2}(x) &= m_{0,2}(x) - \frac{m_{0,1}^2(x)}{m_{0,0}(x)} , \\ \mu_{1,1}(x) &= m_{1,1}(x) - \frac{m_{0,1}(x) \cdot m_{1,0}(x)}{m_{0,0}(x)} , \\ \mu_{2,0}(x) &= m_{2,0}(x) - \frac{m_{1,0}^2(x)}{m_{0,0}(x)} .\end{aligned}\tag{A.7}$$

En écrivant ces moments non-centrés à l'aide des produits scalaires $(\langle x, u_i \rangle)_{1 \leq i \leq 6}$, nous obtenons :

$$\begin{aligned}m_{0,0}(x) &= \langle x, u_3 \rangle , & m_{1,0}(x) &= \langle x, u_1 \rangle , \\ m_{0,1}(x) &= \langle x, u_2 \rangle , & m_{1,1}(x) &= \langle x, u_6 \rangle , \\ m_{0,2}(x) &= \frac{1}{2} (\langle x, u_4 \rangle - \langle x, u_5 \rangle) , & m_{2,0}(x) &= \frac{1}{2} (\langle x, u_4 \rangle + \langle x, u_5 \rangle) .\end{aligned}\tag{A.8}$$

En utilisant les équations A.7 et A.8, nous pouvons exprimer $\mu_{0,2}(x)$, $\mu_{1,1}(x)$, et $\mu_{2,0}(x)$ en utilisant $(\langle x, u_i \rangle)_{1 \leq i \leq 6}$:

$$\begin{aligned}\mu_{0,2}(x) &= \frac{1}{2} (\langle x, u_4 \rangle - \langle x, u_5 \rangle) - \frac{\langle x, u_2 \rangle^2}{\langle x, u_3 \rangle} , \\ \mu_{1,1}(x) &= \langle x, u_6 \rangle - \frac{\langle x, u_1 \rangle \langle x, u_2 \rangle}{\langle x, u_3 \rangle} , \\ \mu_{2,0}(x) &= \frac{1}{2} (\langle x, u_4 \rangle + \langle x, u_5 \rangle) - \frac{\langle x, u_1 \rangle^2}{\langle x, u_3 \rangle} .\end{aligned}\tag{A.9}$$

Nous terminons cette démonstration en injectant l'équation A.9 dans l'équation A.1 pour obtenir l'équation A.4.

Annexe B

Jeu de données COSMOS paramétrique

Pour générer les images de galaxie et de FEP, nous choisissons des profils simples et communément utilisés en astrophysique qui sont respectivement les profils de Sersic [Sérsic 1963] et de Moffat [Moffat 1969]. Premièrement, la luminosité de la galaxie, I_G est modélisée par une Sersic en utilisant la formule suivante :

$$I_G(R) = I_e \cdot \exp \left(b_n \left[\left(\frac{R}{R_e} \right)^{\frac{1}{n}} - 1 \right] \right) , \quad (\text{B.1})$$

avec $n \in \mathbb{R}_+$ l'indice de Sersic, R_e le rayon effectif, I_e la luminosité à la valeur R_e et b_n tel que $\gamma(2n; b_n) = \frac{1}{2}\Gamma(2n)$ avec Γ et γ respectivement la fonction gamma et la fonction gamma incomplète. Nous tirons les valeurs des paramètres n , I_e et R_e du catalogue COSMOS [Mandelbaum et al. 2011] pour générer des images de galaxies isotropes auxquelles nous donnons par la suite une ellipticité non nulle.

Deuxièmement, la luminosité de la FEP, I_P , est modélisée par un profil Moffat en utilisant la formule suivante :

$$I_P(R) = 2 \frac{\beta - 1}{\sigma^2} \left(1 + \left[\frac{R}{\sigma} \right]^2 \right)^{-\beta} , \quad (\text{B.2})$$

où β est réglé à 4,765 (cf. [Trujillo et al. 2001]) et σ est calculé en utilisant la relation suivante :

$$F = 2\sigma \sqrt{2^{\frac{1}{\beta}} - 1} , \quad (\text{B.3})$$

où F est la FWHM du profil Moffat. Sa valeur est tirée aléatoirement d'une distribution uniforme entre 0, 1' et 0, 2', ce qui correspond respectivement à des observations des télescopes HST¹ et Euclid². Cela nous donne des images de FEP préliminaires.

Finalement, nous donnons une ellipticité à chacune des images de galaxie et de FEP. Pour ce faire, nous tirons aléatoirement les valeurs des composantes de l'ellipticité à partir d'une distribution normale centrée tronquée entre -1 et 1. L'écart-type est choisi pour être égal à 0,3 pour les galaxies et 0,03 pour les FEP [Bernstein et Armstrong 2014b].

1. https://www.nasa.gov/mission_pages/hubble/main/index.html

2. <https://www.euclid-ec.org>

Annexe C

Implémentation de l'ADMMnet

Nous avons réimplémenté l'algorithme ADMMnet en commençant par utiliser les mêmes paramètres indiqués dans le travail original [Sureau, Lechat et Starck 2020]. Notre implémentation appliquée sur le jeu de données MeerKAT3600 donne des reconstructions instables. Nous avons alors décidé d'étudier le problème en simplifiant la méthode. Pour ce faire, nous avons remplacé l'algorithme ADMM par l'algorithme FBS (voir 2.1), réduisant ainsi le nombre d'hyperparamètres à un. Pour satisfaire les conditions de convergence, cet hyperparamètre a été réglé à $2/(\alpha(1 + \epsilon))$ avec α la constante de Lipschitz du gradient de la partie différentiable de la fonction de coût et $\epsilon = 10^{-5}$. Nous avons utilisé l'algorithme simplifié avec 500 itérations sur un seul exemple où l'observation, la FEP, la vérité terrain et la reconstruction obtenue sont données dans les figures C.1a C.1e, C.1b et C.1c, respectivement. Le terme d'attache aux données dans la fonction de coût est donné dans la figure C.1f, nous remarquons que celui-ci diverge exponentiellement et atteint des valeurs élevées de l'ordre de 10^{12} . L'instabilité de la méthode est corroborée dans la reconstruction où il y a notamment deux points à mettre en avant, le premier est l'amplitude des pixels dans l'image reconstruite qui est à 4 ordres de grandeur au-dessus de celle de l'image vérité terrain. Le deuxième est l'apparition d'artefact au milieu à gauche de l'image reconstruite où nous voyons une autre galaxie apparaître. La piste la plus vraisemblable pour expliquer la divergence de l'algorithme est l'utilisation du réseau de neurones en tant qu'opérateur proximal. En effet, la convergence des algorithmes proximaux est garantie par le fait que les opérateurs proximaux soient des applications non expansives. Cette propriété n'est plus assurée avec l'utilisation du réseau de neurones dans le cas de l'ADMMnet. À titre d'exemple, nous avons considéré le Unet utilisé en tant qu'opérateur proximal dans l'ADMMnet et nous l'avons appliqué sur l'image vérité terrain 5 fois, le résultat est donné dans la figure C.1d. Le résultat est supposé être identique à la vérité terrain qui est supposée être un point fixe dans ce cas mais nous remarquons que la forme et l'amplitude des deux images sont différentes.

Pour stabiliser notre implémentation nous proposons essentiellement deux pistes : La première consiste à utiliser un réseau de neurones qui soit contractant comme expliqué dans [Ryu et al. 2019]. Succinctement, l'idée se base sur la normalisation spectrale des couches du réseau de neurones pour assurer la convergence de l'algorithme. La seconde méthode consiste à utiliser une version *unrolled* (dépliée, en français) de l'algorithme comme proposé dans [Monga, Li et Eldar 2020]. À cette fin, un nombre fixe d'itérations est choisi et les itérations sont écrites l'une à la suite de l'autre, sans utiliser de boucle. Ainsi dans chaque itération se trouve un réseau de neurones avec des poids qui lui sont propres. Au passage, les auteurs du papier mettent en avant que les réseaux utilisés dans ce cas sont plus interprétables. Concernant l'implémentation de ces pistes à partir du réseau de neurones déjà codé, la deuxième piste semble être plus simple à mettre en œuvre mais plus gourmande en ressources. Par contre, la première piste ne requiert pas de sources additionnelles mais l'implémentation de la normalisation spectrale est coûteuse en termes de développement si celle-ci n'est pas déjà présente dans les outils numériques de travail.

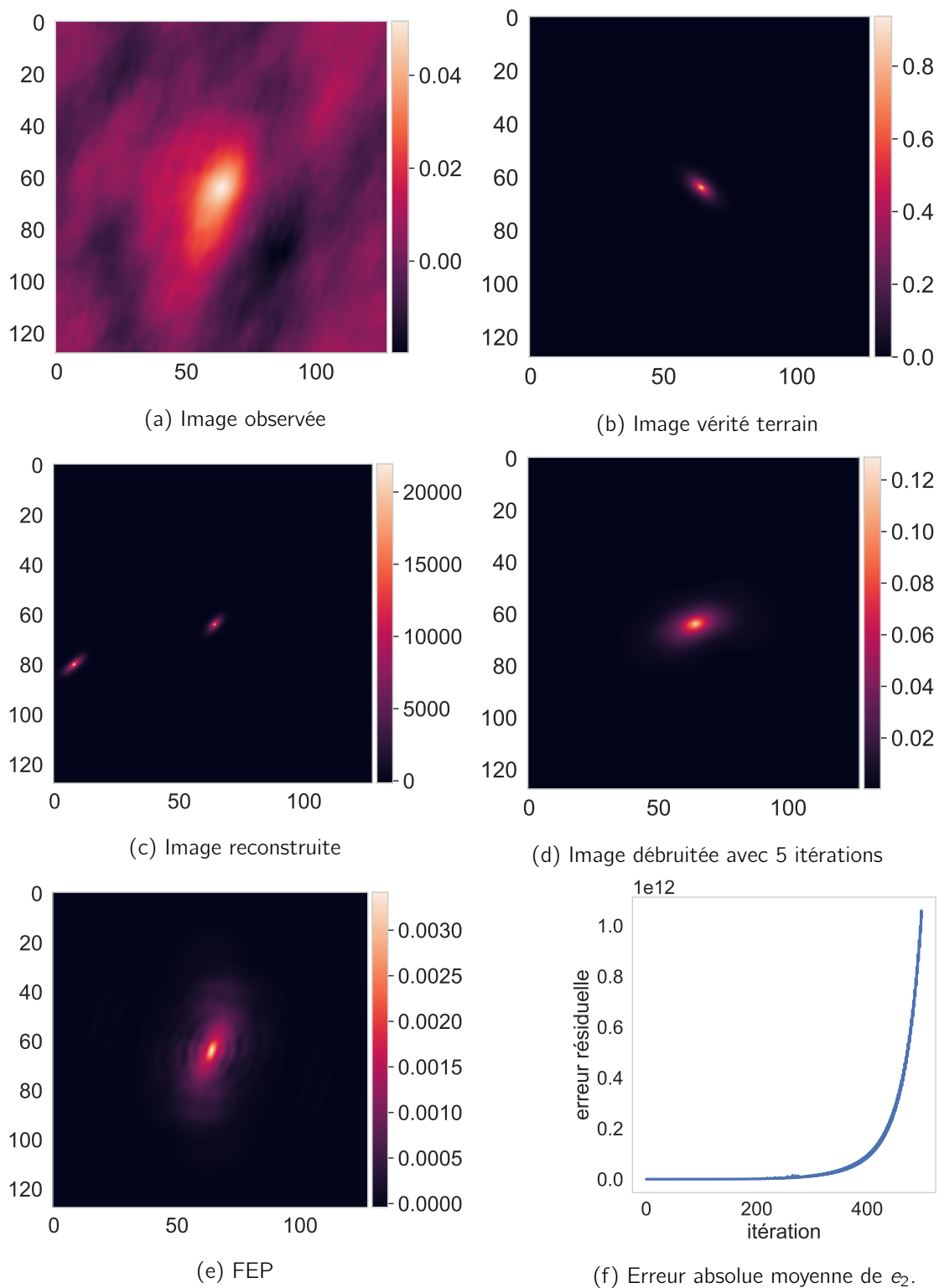


Figure C.1 – Exemple de reconstruction d'une galaxie avec l'ADMMnet simplifié en utilisant 500 itérations.

Annexe D

Communication et activité scientifiques

D.1 Publications

- Nammour, F., Akhaury, U., Lanusse, F., Girard, J. N., Sureau, F., & Starck, J. L. Shapenet : Shape Constraint for Galaxy Image Deconvolution. En préparation.
- Nammour, F., Schmitz, M. A., Mboula, F. M. N., Starck, J. L., & Girard, J. N. (2021). Galaxy Image Restoration with Shape Constraint. arXiv preprint arXiv :2101.10021. Accepté dans Journal of Fourier Analysis and Applications en janvier 2021.

D.2 Conférences

- CosmoStat Machine Learning Day, virtuelle, France, mars 2021 (présentation orale invitée).
- SIAM Conference on Mathematics of Data Science, virtuelle, États-Unis, août 2020 (présentation orale invitée).
- SKA Cosmology SWG, Paris, France, janvier 2020 (présentation orale).
- SPIE Wavelets and Sparsity XVIII, San Diego, États-Unis, août 2019 (présentation orale).
Nammour, F., Schmitz, M. A., Ngolè, F., Starck, J. L., & Girard, J. N. (2019, September). Radio astronomical image restoration with shape constraint. In Wavelets and Sparsity XVIII (Vol. 11138, p. 111380A). International Society for Optics and Photonics. Publié.
- Séminaire sur la Reconstruction d'Images au Département de Mathématiques et Statistiques à SDSU, San Diego, États-Unis, août 2019 (intervenant).
- Wavelets and Beyond, Orsay, France, June 2019.
- SKA General Science Meeting 2019, Cheshire, UK, April 2019 (poster).

D.3 Enseignements

- 2018-2020 : Vacataire, Signaux & Systems, Université de Paris (ex-Paris-Diderot), Paris, France.
- Atelier Bash and Scripting Tutorial for Researchers, virtuel, juin 2020.
- 2019-2020 : Vacataire, Circuits & Architecture, Université de Paris (ex-Paris-Diderot), Paris, France.

D.4 Formations suivies

- Sparsity4PSL, Paris, France, juin 2019 (école d'été).
- Deep Learning in Practice, Gif-sur-Yvette, France, automne 2019 (cours du Master MVA).
- Science Writing EDP Sciences, Gif-sur-Yvette, France, février 2019 (atelier).

D.5 Adaptation de l'enseignement au confinement

Au cours de l'une de mes vacances en 2020, en qualité de chargé de Travaux Pratiques (TP) du cours Signaux & Systèmes, j'ai développé une solution d'apprentissage à distance adaptée au contexte du premier confinement lors de la pandémie de COVID-19.

En effet, les séances de TP pour ce cours sont basées sur MATLAB qui est un logiciel sous licence propriétaire et l'accès à ce logiciel dans le cadre de ce cours est uniquement assuré à partir des salles informatiques au sein des bâtiments de l'université. Ainsi durant le confinement, les étudiants ont dû installer l'environnement de travail par eux-mêmes et sur leur poste personnel. Deux problèmes majeurs se sont présentés, le premier était le fait qu'il fallait que les étudiants paient la licence pour installer MATLAB¹ ou installent un clone du logiciel comme Octave². Le deuxième problème était le fait que l'environnement de travail variait considérablement d'un étudiant à autre, ce qui rendait la tâche d'encadrement plus coûteuse en temps. De ce fait, en comparaison avec les séances en présentiel, les étudiants n'arrivaient pas à avancer dans les exercices qui leurs étaient donnés autant qu'attendu et se retrouvaient souvent bloqués sur des problèmes techniques qui ne sont pas liés aux acquis visés par le cours.

J'ai développé une solution permettant de créer un environnement de travail hébergé sur un serveur distant et qui contient les sujets des TP sous forme de Jupyter Notebooks avec un noyau Octave hébergé par MyBinder [Jupyter et al. 2018]. Cette solution a eu les avantages suivants :

- *open source* permettant une redistribution, modification et accès libres au code,
- portabilité, l'environnement de travail peut être lancé sur les postes personnels et il n'est plus nécessaire d'avoir recourt aux postes de l'université,
- multiplateforme, la solution fonctionne sur tous les systèmes d'exploitations (Linux, Mac, Windows) y compris les téléphones portables,
- simplicité, il n'y a pas besoin d'installer les bibliothèques ni de télécharger les ressources requises pour les TP.

Cette solution a permis aux étudiants de directement commencer à travailler les exercices du TP sans avoir à préparer l'environnement de travail. De plus, avec un environnement contrôlé et identique à tous les étudiants, l'encadrement est devenu plus limpide. Ainsi avec également d'autres approches utilisées pour l'enseignement à distance, la solution a contribué à rendre les conditions de travail à distance plus optimales.

Cette solution a d'ailleurs été transmise à d'autres enseignants pour les aider également dans leurs TP nécessitant Matlab. Cette méthode a également circulé en interne à l'UFR de physique parmi les possibilités de solutions numériques flexibles pendant le confinement.

1. lien vers la page web officielle MATLAB : <https://fr.mathworks.com>

2. lien vers la page web officielle Octave : <https://www.gnu.org/software/octave/>

Nous donnons un exemple de TP sur le lien suivant : https://mybinder.org/v2/gh/CosmoStat/octave_image_processing_example/master. Il faut attendre quelques minutes pour que l'environnement de travail soit chargé et ensuite ouvrir le fichier `example.ipynb`.