Deep learning for light field acquisition and restoration

Brandon Le Bon

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Signal, Image, Vision*

Par
Brandon LE BON

Apprentissage profond pour l'acquisition et la restauration de champs de lumière

Thèse présentée et soutenue à Rennes, le 29 Novembre 2023
Unité de recherche : Centre Inria de l'Université de Rennes

Rapporteurs avant soutenance :

Mårten SJÖSTRÖM Professeur, Mid Sweden University
Frederic DUFAUX Directeur de Recherche, Laboratoire des Signaux et Systèmes Paris

Composition du Jury :

Président :	Franck MULTON	Directeur de recherche, INRIA Rennes
Examinateurs :	Mårten SJÖSTRÖM	Professeur, Mid Sweden University
	Frederic DUFAUX	Directeur de Recherche, Laboratoire des Signaux et Systèmes Paris
	Ioan TABUS	Professeur, Tampere University of Technology
	Franck MULTON	Directeur de recherche, INRIA Rennes
Dir. de thèse :	Christine GUILLEMOT	Directrice de recherche, INRIA Rennes
Co-encadrant. de thèse :	Mikaël LE PENDU	Chercheur, INTERDIGITAL Rennes

Acknowledgement

First of all, I would like to express my deepest gratitude to my PhD advisors Christine Guillemot and Mikaël Le Pendu for offering me the opportunity to be a PhD student in the SIROCCO INRIA team and to be part of the research community for the past three years. Their expertise in terms of research and signal processing have inspired me all along my PhD. I would like to thank them for their support, advice and patience which have play a key role in my development as a PhD student and in the completion of this manuscript.

I would like to thank all the members of my dissertation committee, Mårten Sjöström, Frederic Dufaux, Ioan Tabus and Franck Multon for the interest in my work they demonstrated, and for their time spent to produce interesting and helpful feedback to improve this dissertation.

I am grateful to all my colleagues in the SIROCCO team, for all the good times and interesting discussions we had during the past three years. In particular, I would like to thank Rita Fermanian, Guillaume Le Guludec, Samuel Willingham and Rémy Leroy for all the interesting and helpful discussions, and for their participation in the pleasant working environnement of the SIROCCO team, along with Pascal Bacchus, Arthur Lecert, Rémi Piau, Tom Bordin, Tom Bachard, Sebastien Bellenous and all the other members of the team.

During this PhD, nobody has been more important to me than the members of my family. I would like to express my gratitude to my parents for their endless love and all the support they gave me during the lockdown due to the complicated sanitary situation. I would like to offer my gratitude to my sister for the support and help. Most importantly, I wish to thank my partner for all the love and support without which my PhD would not have been possible.

Table of Contents

Résumé en français

Contexte

Les images sont devenues des outils incontournables de notre société afin de sauvegarder, partager et diffuser de l'information visuelle sur le monde qui nous entoure. Le processus d'acquisition d'une image est un processus complexe, et qui est souvent restreint par les limitations du matériel d'acquisition ou soumis à des perturbations lors de la capture. Par conséquent, l'image capturée est généralement une représentation incomplète ou corrompue de la scène observée. Par exemple, la majeure partie des caméras traditionnelles sont équipées de capteurs photographiques CCD (Charge-coupled Device) traduisant l'intensité lumineuse émise par une scène en un signal électronique. Afin de capturer les différentes composantes de couleur de la lumière incidente, un filtre de couleur (CFA) est généralement placé devant ces capteurs. Cela a pour conséquence de produire une image où chaque pixel ne contient l'information liée qu'à une seule couleur donnée: l'image est alors une mosaïque de couleur, et est donc une mesure incomplète. Le problème de reconstruction de l'image originale à partir de mesures corrompues ou incomplètes est généralement présenté comme un problème inverse, où l'on souhaite inverser le processus d'acquisition. Le problème de reconstruction de l'image attendue à partir de la mosaïque de couleur est connu sous le nom de « dématriçage », et fait partie d'une pléthore de problèmes inverses d'acquisition d'images qui ont pu être adressés ces dernières décennies par la communauté de la recherche du domaine du traitement d'images.

Dans cette thèse, nous nous intéressons plus particulièrement aux problèmes inverses liés à l'acquisition de champs de lumière. Un champ de lumière est une représentation d'une scène contenant à la fois l'intensité et l'orientation de l'ensemble des rayons lumineux parcourant la scène. Par conséquent, la représentation numérique des champs de lumière la plus couramment utilisée est un ensemble de différents points de vue de la scène observée. En comparaison, les images capturées par une caméra conventionnelle sont des projections 2D de la scène observée, perdant alors toute information sur l'orientation des rayons lumineux incidents. Cela implique donc que la majeure partie de l'information 3D soit perdue lors de l'acquisition de l'image. Malheureusement, cette information 3D perdue

est cruciale pour une variété de tâches en imagerie, comme par exemple l'estimation de profondeur et de géométrie de la scène, le changement de point de vue ou de distance de focus. Bien qu'il y ait une vaste littérature scientifique autour de la réalisation de ces tâches à partir de mesures incomplètes, une autre approche vise plutôt à capturer une plus large quantité d'informations via l'acquisition de champs de lumière.

Motivations et objectifs

La reconstruction d'images à partir de mesures dégradées est un vaste domaine de recherche, s'appuyant notamment sur les travaux mathématiques traitant de la résolution de problèmes inverses. Résoudre un problème inverse vise à inverser le processus d'acquisition, afin de retrouver l'image originale à partir des mesures, bien que le processus ne soit généralement pas inversible. On pose alors un problème de minimisation, cherchant une solution en adéquation avec les mesures. Etant donné qu'une partie de l'information est perdue et/ou corrompue lors de l'acquisition, le problème inverse est souvent mal conditionné, et la résolution de celui-ci nécessite des connaissances à priori sur les images à reconstruire. Cependant, les images sont des signaux complexes dont la structure est difficile à décrire parfaitement.

Avec l'émergence récente des modèles d'apprentissage machine, notamment les modèles d'apprentissage profond, de plus en plus de méthodes de reconstruction d'images utilisent ces derniers afin d'apprendre automatiquement des connaissances à priori complexes sur les images. Ces modèles sont optimisés, ou entraînés, pour une tâche donnée en utilisant une banque de données d'entraînement. Le plus souvent, chaque signal d'entrée de la banque de données est associé à son signal cible. Dans le cas de la reconstruction d'images à partir de mesures incomplètes et/ou corrompues, le signal d'entrée correspond généralement à l'image corrompue et le signal cible correspond à l'image cible reconstruite. De cette façon, ces modèles sont entraînés à reconstruire des images, en apprenant des connaissances à priori sur celles-ci.

Les algorithmes d'optimisation déroulés sont des méthodes bénéficiant de l'avantage des méthodes analytiques pour résoudre des problèmes inverses et de l'avantage des techniques d'apprentissage profond pour apprendre des connaissances à priori sur les images. Grâce à cela, les performances de reconstruction de ces méthodes sont l'état de l'art actuelle pour un grand nombre de problèmes inverses en imagerie. Le principe de ces méthodes est que l'entraînement du modèle est réalisé au sein d'un algorithme d'optimisation

itératif, permettant d'apprendre les connaissances à priori sur les images pour un problème et pour un algorithme d'optimisation précis. L'entraînement de ce modèle est cependant contraint à de forts coûts de calcul, notamment en matière de mémoire et de temps de calcul. Le nombre d'itérations considéré dans l'algorithme d'optimisation est alors restreint par les contraintes liées à l'entraînement du modèle d'apprentissage profond. Le premier axe de travail de cette thèse s'articule alors autour de la problématique suivante :

1. Comment peut-on réduire les contraintes liées à l'entraînement des modèles au sein d'un algorithme d'optimisation déroulé ?

Pour répondre à cette problématique, nous proposons une nouvelle méthode pour l'entraînement des algorithmes déroulés afin de pallier ces contraintes. La principale cause de ces problèmes est que l'entraînement du modèle d'apprentissage profond doit se faire de bout en bout au sein d'un algorithme d'optimisation itératif, afin de garantir le meilleur résultat en sortie de l'algorithme. La méthode présentée doit alors être capable de simplifier ce processus d'apprentissage, afin de le rendre moins coûteux et utilisable quel que soit le nombre d'itérations considéré dans l'algorithme d'optimisation déroulé.

Dans la suite de cette thèse, nous nous intéressons en particulier aux problèmes inverses liés à l'acquisition et à la reconstruction de champs de lumière. Nous utilisons alors les travaux construits autour de la précédente problématique dans l'optique de proposer une méthode pour la reconstruction de champs de lumière.

Un champ de lumière décrit la scène comme étant une collection de rayons de lumière émis en tout point de la scène de coordonnées spatiales (x, y, z), dans toutes les directions, représentées par des coordonnées angulaires (u, v), à n'importe quel moment t et pour n'importe quelles longueurs d'onde λ. Un champ de lumière est donc représenté par une fonction en 7 dimensions, connue sous le nom de "fonction plénoptique". Cette fonction est généralement simplifiée en une fonction en 4 dimensions faisant intervenir deux coordonnées spatiales (x, y) et deux coordonnées angulaires (u, v). Le champ de lumière est alors souvent visualisé par une collection de points de vue adjacents, comme illustré dans la figure 1.

La capture d'une telle quantité d'informations n'est cependant pas une tâche aisée. Une première approche vise à capturer les différents points de vue simultanément, par l'intermédiaire d'une matrice de caméras placées sur un plan 2D, comme illustré dans la figure 2a, ou séquentiellement, via l'utilisation d'une caméra en mouvement. Cependant, ces dispositifs sont généralement très coûteux et posent des problèmes de calibration des caméras. Afin de rendre l'acquisition des champs de lumière plus accessible à tout public, il

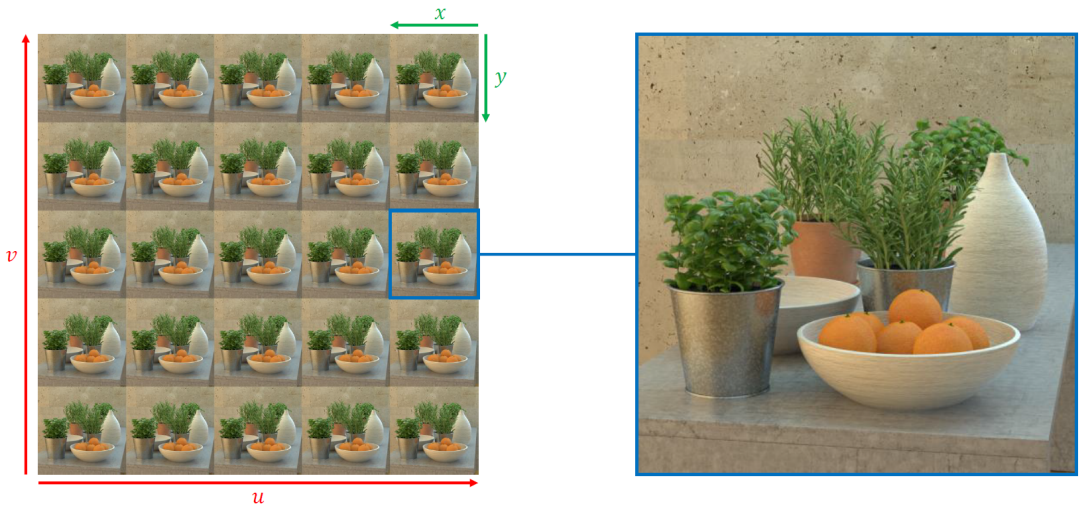


Figure 1 – Visualisation d'un champ de lumière: un ensemble de points de vue adjacents.

est nécessaire de réaliser la capture du champ de lumière par l'intermédiaire d'une unique caméra. Les caméras dites "plénoptiques", utilisant un ensemble de microlentilles placées devant le capteur pour simuler différents points de vue, ont été proposées à cet effet. Les modèles les plus connus de caméra plénoptique sont les caméras Lytro, visibles sur la figure 2b. D'autres approches utilisent des masques codés entre l'ouverture et le capteur d'une caméra traditionnelle, comme illustré sur la figure 2c, permettant d'obtenir une mesure codée de la scène. Ces différents modèles de caméra sont cependant spécifiques à la capture de champs de lumière, ne rendant pas l'accès à l'acquisition de champs de lumière tout public. Une dernière approche vise à capturer un champ de lumière directement avec une unique caméra conventionnelle, via l'acquisition d'un empilement de mises au point, plus connu sous le nom de "focal stack", consistant en une série d'images prises en variant la distance de mise au point.

Il est important de noter que pour toutes les approches utilisant une seule caméra pour capturer un champ de lumière, l'image obtenue reste cependant mesurée sur un plan 2D: un problème inverse de reconstruction du champ de lumière à partir de mesures 2D doit être résolu. Pour résoudre ces problèmes inverses, nous nous intéressons aux algorithmes d'optimisation déroulés évoqués précédemment. Nous nous penchons plus particulièrement au cas d'une acquisition de champs de lumière via la capture d'une focal stack, où il est généralement nécessaire d'acquérir une grande quantité d'images de focal stack pour mesurer suffisamment d'informations sur toutes les profondeurs de la scène.

(a) (b) (c)

Figure 2 – Design de caméras pour la capture de champs de lumière: (a) la matrice de caméras Stanford [1], (b) la caméra plénoptique Lytro [2], (c) une caméra utilisant un masque codé [3].

Notre second objectif de la thèse s'axe autour de ce problème et tente de répondre à la problématique suivante :

2. Est-il possible de construire un algorithme d'optimisation déroulé capable de reconstruire un champ de lumière de haute qualité à partir d'une focal stack composée de très peu de mesures ?

Pour répondre à cette problématique, nous proposons une nouvelle méthode de reconstruction de champs de lumière à partir d'une focal stack composée de seulement quelques images. Afin de reconstruire efficacement le champ de lumière, l'algorithme proposé a besoin (i) de définir un modèle de formation d'une focal stack à partir d'un champ de lumière, qui puisse être utilisé pour trouver une solution cohérente avec les mesures (ii) de connaissances a priori complexes sur les champs de lumière, afin d'inférer toute l'information perdue lors de l'acquisition tout en produisant un champ de lumière cohérent.

Résumé des contributions

Les contributions présentées dans cette thèse sont organisées en deux parties, suivant les problématiques énoncées précédemment.

Dans une première partie, nous abordons la problématique de la résolution de problèmes inverses linéaires en imagerie 2D. Cette partie est divisée en deux chapitres comme suit.

Chapitre 1 : ce chapitre vise à introduire la notion de problèmes inverses linéaires, notamment dans le cadre de l'imagerie, et les approches traditionnelles de la littérature pour les traiter. Ces problèmes étant généralement mal conditionnés, trouver une solution désirable à ces derniers impose la prise en compte de connaissances à priori. Le problème est alors traditionnellement formulé comme un problème de minimisation composé de deux termes: un terme d'attache aux données, visant à mesurer la fidélité de la solution trouvée face aux mesures, et un terme de régularisation, visant à quantifier l'adéquation de la solution aux connaissances à priori. Il existe un grand nombre d'algorithmes utilisés pour la résolution de ce problème de minimisation, utilisant généralement des algorithmes d'optimisations itératifs et/ou des méthodes d'apprentissage profond. L'une des approches les plus prometteuses est l'utilisation d'algorithmes d'optimisation déroulés. Ces derniers sont caractérisés par l'entraînement d'un modèle d'apprentissage profond au sein d'un algorithme d'optimisation itératif, afin d'apprendre les connaissances à priori à la fois pour un problème donné et à la fois pour un algorithme d'optimisation donné. Cet entraînement est cependant coûteux en calcul, et contraint donc le nombre d'itérations utilisées dans l'algorithme d'optimisation.

Chapitre 2 : dans ce chapitre, nous présentons notre première contribution [4], visant à pallier les contraintes d'entraînement des modèles au sein des algorithmes d'optimisation déroulés. L'approche s'appuie sur les propriétés de l'algorithme des directions alternées (ADMM) afin de diviser le problème initial d'entraînement de bout en bout du modèle en un sous-ensemble de problèmes d'entraînement moins coûteux et définis pour chaque itération de l'algorithme d'optimisation. Ces problèmes d'entraînement sont résolus via un processus d'optimisation stochastique, permettant ainsi de considérablement réduire les coûts de calcul, tout en permettant de considérer n'importe quel nombre d'itérations au sein de l'algorithme d'optimisation déroulé.

Dans la seconde partie, nous nous penchons sur l'acquisition et la reconstruction de champs de lumière. De façon similaire à la première partie, celle-ci est aussi divisée en deux chapitres.

Chapitre 3 : ce chapitre présente les fondamentaux de l'acquisition et la reconstruction des champs de lumière. Les champs de lumière représentent toutes les caractéristiques des rayons lumineux présents dans une scène et sont souvent représentés par une image en 4 dimensions. Cette représentation de la scène est très importante pour un grand nombre de problèmes en imagerie nécessitant beaucoup d'informations 3D sur la scène, comme l'estimation de la profondeur ou encore le changement de focus ou de point de vue. Le

champ de lumière est directement capturé par un ensemble de caméras ou une caméra en mouvement, ou alors partiellement capturé puis reconstruit via une caméra modifiée spécialement pour l'acquisition de champs de lumière ou via une caméra traditionnelle capturant un empilement de mises au point.

Chapitre 4 : dans ce dernier chapitre, nous proposons une nouvelle approche [5], [6] pour reconstruire un champ de lumière à partir d'un empilement de mises au point, ou focal stack, capturé avec une caméra traditionnelle. Nous nous focalisons sur cette méthode d'acquisition des champs de lumière afin d'ouvrir l'acquisition de ceux-ci à tout public. Les méthodes de l'état de l'art actuelles nécessitent cependant une grande quantité d'images de focal stack afin d'obtenir suffisamment d'informations 3D sur la scène. Malheureusement, la capture d'une grande quantité d'images de focal stack est difficile, notamment à cause de sa sensibilité aux mouvements de caméra et d'objets dans la scène durant les différentes captures.

L'approche proposée s'intéresse à la reconstruction d'un champ de lumière à partir d'une focal stack contenant très peu d'images. Le problème de reconstruction est posé sous la forme d'un problème inverse linéaire, via l'utilisation de la représentation de champs de lumière "Fourier Disparity Layers (FDL)". Il s'agit d'une représentation compacte du champ de lumière qui permet de synthétiser n'importe quel point de vue de celui-ci. Un algorithme d'optimisation déroulé est présenté afin de résoudre le problème de reconstruction tout en apprenant des connaissances à priori directement dans le domaine des FDL. Les FDL étant théoriquement définis uniquement pour les scènes sans occlusions, synthétiser des vues à partir des FDL peut produire des artéfacts autour des zones d'occlusions dans une scène réelle. Nous proposons alors une nouvelle méthode de synthèse de vues à partir des FDL, basée sur l'utilisation d'un réseau de neurones profond, pour résoudre ces problèmes d'artéfacts. Nous montrons que la méthode proposée permet d'obtenir des champs de lumière de très bonne qualité en utilisant très peu d'images de mesure.

INTRODUCTION

Context

Nowadays, images are unavoidable tools in our society to save, share, and diffuse visual information about our surrounding world. The acquisition process of an image is, however, restricted by the limitations of the acquisition device or subject to perturbations, resulting in corrupted or incomplete measurements. For instance, the majority of traditional cameras are composed of Charge-Coupled Device (CCD), where each sensor captures the intensity of light rays emitted by the scene to produce an electronic signal. In order to capture the different color components of the incident light, a Color Filter Array (CFA) is usually placed in front of the sensors. As a consequence, each pixel of the sensed image only contains the intensity of a specific color: the sensed image is thus a mosaic image, hence an incomplete measurement of the original signal. The problem of reconstructing the original image from corrupted or incomplete measurements is generally presented as an inverse problem, aiming at inverting the acquisition process. In the case of a mosaic measurement, the problem of reconstructing the original image is generally referred to as a "demosaicing" problem, which is part of a plethora of image inverse problems that have been addressed in the last decades by the image processing community.

In this thesis, we take a particular interest in inverse problems for light field imaging. A light field is a representation of a scene that contains both the intensity and the orientation of the light rays traveling through the scene. The most common numerical representation of a light field is thus a set of adjacent views. In comparison, images captured by a conventional camera are 2D projections of the observed scene, thus losing orientation information from the incident light rays. Consequently, a major part of the 3D information is lost during the acquisition of the image. Nevertheless, this kind of information is crucial for a variety of image processing tasks, for instance depth and geometry estimation, view point switching, or image refocusing. While there exists a large scientific literature on methods to perform these tasks using incomplete measurements, other approaches aim instead at capturing more information about the scene via the acquisition of light fields.

Motivations and goals

The problem of reconstructing images from degraded measurements is a vast field of research that usually relies on the mathematical theory of inverse problem solvers. Solving an inverse problem aims at inverting the acquisition process in order to retrieve the original image from its measurements, although the acquisition process is generally not invertible. A common approach is to pose a minimization problem, aiming at finding a solution matching the measurements. Since part of the information is lost or corrupted during the acquisition process, the inverse problem involved is usually ill-conditioned, meaning that prior knowledge on the type of images we try to recover is required. However, images are complex signals whose structure is difficult to describe perfectly.

With the rise of machine learning, especially deep learning, more and more image reconstruction methods take advantage of these models to automatically learn complex image priors. These models are optimized, or learned, for a specific task using a training set of data. Generally, each input signal of the training set of data is associated to its target signal. In the case of image reconstruction from incomplete or corrupted measurements, the input signal usually corresponds to the degraded signal and the target signal to the target reconstructed image. Therefore, these models are trained to reconstruct images by implicitly learning an image prior.

Unrolled optimization algorithms have emerged as a way to take advantage of both deep learning techniques, to learn an image prior, and analytical solutions, to solve inverse problems. Thanks to this, these methods have achieved state-of-the-art results for a variety of image inverse problems. The principle of unrolled optimization algorithms is to train the deep learned model within an iterative optimization algorithm, hence in a way that it is optimized for a specific task and a specific algorithm. However, this training usually suffers from a high computational burden, usually in terms of memory and computation time. As a result, the number of iterations considered in the unrolled optimization algorithm is usually restricted by the limitations of the training. The first issue addressed in this thesis is thus stated as follows:

1. How can we reduce the limitations of the training of an unrolled optimization algorithm ?

To cope with this issue, we propose a novel approach for the training of unrolled optimization algorithms. This issue is mainly due to the end-to-end training of the deep learned model within an iterative optimization algorithm, which is used to guarantee the

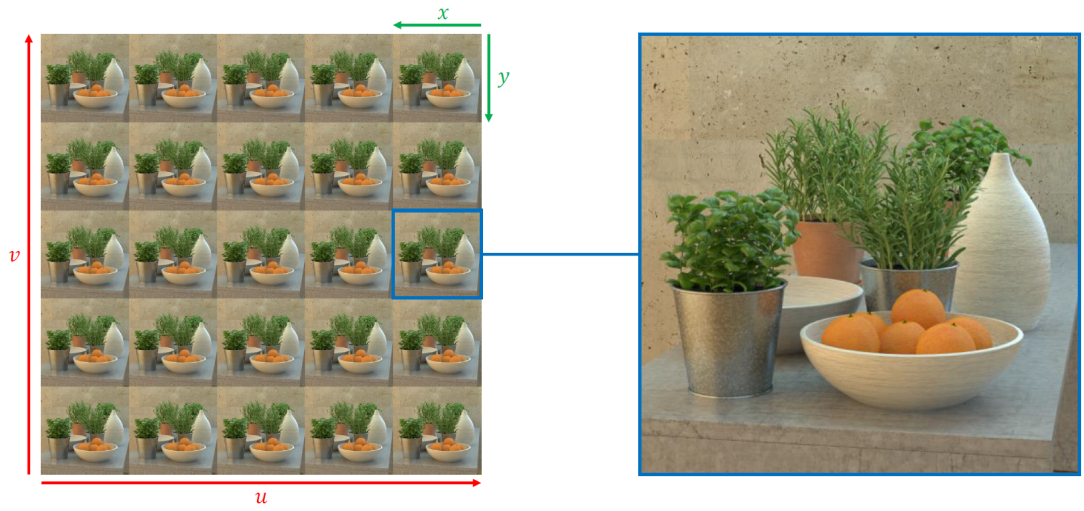


Figure 3 – Visualization of light fields: a set of adjacent views.

best output from the algorithm. The proposed method thus needs to simplify this training in order to reduce its computational burden and to make it applicable for any number of iterations considered in the unrolled optimization algorithm.

In the second part of this thesis, we focus more specifically on inverse problems related to the acquisition and reconstruction of light fields. We thus use the methods constructed around the previous issue in order to propose a novel approach to reconstruct light fields.

A light field describes the scene as a collection of light rays emitted at every point of the scene at spatial coordinates (x, y, z), in every directions represented by angular coordinates (u, v), at any time t and for any wavelength λ. A light field is thus represented by a 7-dimensional function, also known as the "plenoptic function". This function is often reduced to a 4-dimensional function, using two spatial coordinates (x, y) and two angular coordinates (u, v). The light field is thus visualized as a collection of adjacent points of view, as illustrated in figure 3

Capturing such a high quantity of information is, however, a difficult task. A first approach aims at capturing the different viewpoints simultaneously, using a camera array as illustrated in figure 4a, or sequentially, using a camera on a moving gantry. However, these camera architectures are complex and pose calibration issues. In order to make the acquisition of light fields accessible to the general public, it is necessary to capture the light field with only a single camera. Plenoptic cameras have been proposed, using an array of microlenses placed in front of the sensor in order to simulate different viewpoints. The most

(a) (b) (c)

Figure 4 – Camera architectures for light field imaging: (a) Stanford camera array [1], (b) Lytro plenoptic camera [2], (c) Coded-mask camera [3].

known plenoptic camera is the Lytro camera, shown in figure 4b. Other approaches tend to place coded-masks between the aperture plane and the sensor plane of a conventional camera, as illustrated in figure 4c, which captures a coded projection of the scene. All these camera designs are, however, specific to light field imaging, making them not accessible to the general public. A last approach aims at capturing a light field with a single traditional camera via the acquisition of a focal stack, a set of images captured at different focus distances.

It is important to notice that for all of the approaches using a single camera to capture a light field, the sensed image is obtained on a 2D plane: a light field reconstruction inverse problem from 2D measurements needs to be solved. In order to solve these inverse problems, we are interested in the unrolled optimization algorithms presented earlier. We focus on the acquisition of a light field from focal stack measurements, which usually requires performing a large number of shots with slowly varying focus to retrieve all the details at every depth in the scene. Our second objective in this thesis is to address the following issue:

2. Can we produce an unrolled optimization algorithm to reconstruct a high-quality light field from a focal stack composed of very few images ?

To address this issue, we propose a novel method to reconstruct a light field from a focal stack with very few shots. In order to reconstruct the light field efficiently, the proposed algorithm needs (i) a focal stack formation model that can be used to find a

solution that matches the measurements (ii) a complex image prior in order to retrieve all the information lost during the acquisition process.

Thesis structure and contributions

The contributions presented in this thesis are organized in two parts, following the issues stated previously.

In the first part, we address the problem of solving linear image inverse problems. This part is divided into two chapters, as follows:

Chapter 1: this chapter aims at introducing the notion of linear inverse problems, with a focus on 2D imaging applications, and the approaches to solve them. With these problems being generally ill-conditioned, finding a desirable solution requires prior knowledge on the type of image we try to recover. The problem is thus traditionally posed as the minimization of a function composed of two terms: a data-fidelity term, which measures the fidelity of the solution with the measurements, and a regularization term, which quantifies how the solution matches with prior knowledge on the target signal. There exist a variety of algorithms to solve this minimization problem, usually using iterative optimization algorithms and/or deep learning techniques. The most promising approaches used to solve image inverse problems are unrolled optimization algorithms. These methods aim at learning an image prior within an iterative optimization algorithm, such that it best suits a specific problem and a specific optimization algorithm. However, the training of such a learned prior within multiple iterations of an iterative optimization algorithm has a high computational cost, which restricts the number of iterations considered in the unrolled optimization algorithm.

Chapter 2: in this chapter, we present our first contribution [4], aiming to cope with the issues of the training of an unrolled optimization algorithm. Our approach relies on the properties of the Alternating Direction Method of Multipliers (ADMM) in order to divide the end-to-end training of the deep learned prior to a set of smaller optimization problems, defined per unrolled iteration, that require less computational cost. These optimization problems are solved using a stochastic process, allowing a small computational burden while considering any number of iterations in the unrolled optimization algorithm.

In the second part, we address the problem of light field acquisition and reconstruction. Similarly to the first part, this one is divided into two chapters, as follows:

Chapter 3: this chapter presents the fundamentals of the acquisition and reconstruc-

tion of light fields. Light fields describe all the characteristics of the light rays traveling through a scene and are usually represented as a 4D image. This representation is important to solve a variety of image processing tasks requiring complex 3D information about the scene, for instance depth and geometry estimation, viewpoint switching or image refocusing. The light field is usually captured either directly by a camera array or a camera on a moving gantry, or partially captured and then reconstructed using a camera built specifically for light field imaging or using a conventional camera capturing a focal stack.

Chapter 4 : in this last chapter, we propose a novel approach to reconstruct a light field from a focal stack captured with a traditional camera [5], [6]. We address this problem in order to open up light field acquisition to the general public. Most state-of-the-art methods require a focal stack with dense sampling in the focus dimension, so that details can be retrieved at every depth in the scene. However, capturing many shots of a focal stack is a difficult task, especially due to its sensitivity to camera and object movements during the different shots.

The proposed approach aims at reconstructing a light field from a focal stack containing very few images. The reconstruction problem is posed as a linear image inverse problem using the "Fourier Disparty Layers (FDL)" representation of light fields. The FDL model is a compact representation of light fields which decomposes the scene into a few additive layers from which any viewpoint can be reconstructed. A novel unrolled optimization algorithm is presented to solve the reconstruction problem while learning an image prior directly in the FDL domain. The FDL representation being theoretically defined for non-occluded scenes, synthesizing views from a FDL model may bring artifacts around occluded areas in real scenes. Hence, we additionally propose a novel neural network-based process to synthesize views from the FDL model while coping with such artifacts. We show that the proposed method permits to obtain high quality light fields using a focal stack with very few images.

Part I

Deep priors for image inverse problems

Chapter 1

Image inverse problems

1.1 Inverse problems

1.1.1 Problem statement

Inverse problems refer to a broad class of problems that appear in a plethora of image processing and computer vision applications. Due to the limitations of acquisition devices or defective hardware, the captured signal is usually incomplete or corrupted. Inverse problems in imaging can be formalized as the recovery of a target image given its noisy or incomplete observations. Formally, let $\mathbf{x} \in \mathcal{X}$ be the target image and $\mathbf{y} \in \mathcal{Y}$ be the observation. The general formulation of the formation model, representing the acquisition process in digital imaging, is as follows:

$$\mathbf{y} = \mathcal{D}(\mathbf{A}\mathbf{x}), \tag{1.1}$$

where the matrix $\mathbf{A}$ is a linear degradation operator and $\mathcal{D}$ is a non-linear degradation operator, which usually includes randomness such as noise. In many applications, the degradation can be reduced to a linear additive noise formation model, assuming a linear operator $\mathbf{A}$ and an unknown additive noise ϵ modeled by a Gaussian distribution:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{1.2}$$

with σ^2 being the variance of the Gaussian noise. However, for a variety of acquisition systems, other types of noise distributions are more accurate [7]. This is, for instance, the case of photon counting devices, such as Charge-Coupled-Device (CCD) and CMOS cameras, where the noise is modeled by a Poisson distribution [8]–[10]. Multiplicative noises are also considered, such as speckle noise, which generally appears with synthetic aperture radars (SAR) [11]. In the literature, the noise ϵ is generally assumed to follow a Gaussian distribution, which will be the case in the following.

Concerning the linear operator $\mathbf{A}$, its design is specific to the inverse problem involved. There is a plethora of image inverse problems arising from image acquisition system limitations. Examples of well-studied image inverse problems are presented in the following section. Note that there are many more image inverse problems tackled in the literature [12].

1.1.2 Examples of image inverse problems

In this section, a few examples of the multitude of image inverse problems [12] arising from the limitations of acquisition devices are presented, namely: the denoising, the deblurring, the super-resolution, and the demosaicing inverse problems.

Denoising

As mentioned in Section 1.1.1, perturbations such as noise are generally expected when capturing an image. The value of a pixel in the captured image is thus perturbed by additive noise. The expected type of noise differs depends on the kind of camera used, e.g. Poisson noise with Charge-Coupled-Device (CCD) and CMOS cameras or speckle noise with synthetic aperture radars (SAR). Defective sensors are also a common cause of noise in the measurements. An example of a noisy image is presented in Figure 1.1b.

Solving a denoising image inverse problem aims at removing the noise appearing on the original image. The linear degradation $\mathbf{A}$ in (1.2) is thus equal to the identity matrix and the additive noise $\epsilon \neq 0$. Denoising in image processing has been a widely studied problem with continuous progress [13]–[15].

Deblurring

In a wide range of imaging systems, an acquired image results from the convolution of the analog input signal to the sensor along with the Point Spread Function (PSF) characterized by the imaging system. This phenomenon creates a blurring effect on the acquired image. Other types of blurring effects commonly occur, for instance, motion blur, which appears when an object is moving in the scene during the capture, or defocus blur, which occurs when the acquisition device fails to focus an object of interest in the scene. An example of a blurred image is illustrated in Figure 1.1c.

Deblurring inverse problems are special cases of a deconvolution problem, which have been widely studied in a variety of forms [16]–[21]. In a deconvolution inverse problem,

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

Figure 1.1 – Examples of measurements for different image inverse problems: (a) the original image *baby.png* of the Set5 dataset [22], (b) image corrupted with a Gaussian noise with a standard deviation $\sigma = 20$, (c) blurred image with a gaussian kernel of size 5×5 with a standard deviation $\sigma = 20$, (d) downsampled and then upsampled image with a magnification factor of 4 and using a bilinear interpolation, (e) mosaic image obtained using the Bayer kernel.

the linear operator $\mathbf{A}$ in (1.2) is a square matrix modeling a convolution kernel.

Super-resolution

Acquiring a high-resolution image is a challenging task for many devices due to hardware limitations. This is, for instance, the case with small portable devices such as smartphones. The problem of constructing a high-resolution image from a low-resolution image, known as the super-resolution image inverse problem, and illustrated in Figure 1.1d, has received particular attention from the research community, with a variety of proposed methods [23]–[25].

In the case of a super-resolution problem, the degradation operator $\mathbf{A}$ is unknown in real-world applications. However, it can be approximated via a blurring kernel and a downsampling operator with a defined scaling factor.

Demosaicing

Most digital photography cameras are equipped with a single Charge-Coupled-Device (CCD), with each sensor element capturing the intensity of the incident light. To capture color components of the incident light, a Color Filter Array (CFA) is generally placed on top of the sensor, such that each sensor element captures a specific color component. As a result, the acquired image is a mosaic of colors, as illustrated in Figure 1.1e.

The demosaicing inverse problem refers to estimating the original image from the mosaic of colors. In the literature [26]–[28], the problem is presented as in equation (1.2) with $\mathbf{A}$ being a binary mask representing the CFA.

1.1.3 Ill-posed linear inverse problems

A linear inverse problem is formalized by the problem of computing an estimate signal $\hat{\mathbf{x}}$, which approximates the original signal $\mathbf{x} \in \mathcal{X}$, from the measurements $\mathbf{y} \in \mathcal{Y}$ obtained following the formation model detailed in equation (1.2). This type of equation is said to be well-posed, in the sense of Hadamard [29], if it verifies the following three conditions:

1. A solution exists, i.e. $\forall \mathbf{y} \in \mathcal{Y}, \exists \mathbf{x} \in \mathcal{X}, \mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon$
2. The solution is unique, i.e. $ker A = \{0\}$
3. The solution is stable, i.e. $\forall \epsilon \in \mathbb{R}_+, \exists \delta \in \mathbb{R}_+, \forall (\mathbf{y}, \mathbf{y}') \in \mathcal{Y}^2, \|\mathbf{y} - \mathbf{y}'\| < \delta \Rightarrow \|\hat{\mathbf{x}}(\mathbf{y}) - \hat{\mathbf{x}}(\mathbf{y}')\| < \epsilon$

A problem is said to be ill-posed if one of the above conditions is not verified. For instance, the existence of a solution is usually compromised in the case of an overdetermined problem, or the uniqueness of the solution may not be verified when the problem is underdeterminated, which is a common property of many image inverse problems. If the stability property is not verified, the reconstruction of the estimate $\hat{\mathbf{x}}$ increases the noise. This is the case when the linear operator $\mathbf{A}$ is ill-conditioned. If $\mathbf{A}$ is invertible, the estimate is computed using the inverse of the linear degradation operator:

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}(\mathbf{A}\mathbf{x} + \epsilon) = \mathbf{x} + \mathbf{A}^{-1}\epsilon. \tag{1.3}$$

With $\mathbf{A}$ being ill-conditioned, the inverse filtered noise $\mathbf{A}^{-1}\epsilon$ will amplify the noise. In the case where $\mathbf{A}$ is not-invertible, which is the case in many image inverse problems, the classical approach is the least squares method [30] which minimizes the distance between $\mathbf{A}\mathbf{x}$ and the measurements $\mathbf{y}$:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{X}}{\arg\min} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \tag{1.4}$$

Note that the least squares method leads to an irregular image when $\mathbf{A}$ is ill-conditioned, as the inverse of the linear degradation operator does. Furthermore, the uniqueness of the solution is not guaranteed.

1.2 Reconstruction approaches

1.2.1 Variational approaches

A common strategy to deal with ill-posed image inverse problems consists of introducing a regularization term in the minimization problem. This approach was first introduced by Hadamard [31]. The principle of the regularization term is to weight the solution space to promote certain types of solutions, assuming prior knowledge on the kind of typical images we attempt to restore. This leads to the following variational formulation:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{X}}{\arg\min} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\mathcal{R}(\mathbf{x}), \tag{1.5}$$

where $\mathcal{R}$ denotes the regularization term, and $\lambda > 0$ is a parameter that controls the amount of regularization. The function to be minimized is thus composed of two terms: a data-fidelity term $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$ and a regularization term $\lambda\mathcal{R}(x)$. Contrary to the least squares method in equation (1.4), the variational formulation ensures the uniqueness and stability of the solution, hence making the inverse problem well-posed.

The quality of the estimated image $\mathbf{x}$ is, however, highly dependent on the capacity of the regularization term to describe natural image statistics. Designing a regularization term to represent image priors in order to solve image inverse problems has been a major subject of research [32]. We further discuss regularization techniques in Section 1.4.

1.2.2 Bayesian approaches

Another approach to solve inverse problems starts with its Bayesian formulation. In this context, $\mathbf{y}$ and $\mathbf{x}$ in equation (1.1) are realizations of random variables $\mathcal{X}$ and $\mathcal{Y}$. The principle of the bayesian approach is to compute an estimator from the posterior distribution $\mathcal{P}_{\mathcal{X}|\mathcal{Y}}$ which models the information of the original signal assuming the measurements. Let $\mathcal{C}(\hat{\mathbf{x}} - \mathbf{x})$, a cost function that computes the quality of an estimate $\hat{\mathbf{x}}$ assuming the original signal $\mathbf{x}$. With the posterior density $p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|\mathbf{y})$, the Bayesian estimator is obtained as:

$$\hat{\mathbf{x}} = \underset{x\in\mathcal{X}}{\arg\min} \int_{x\in\mathcal{X}} \mathcal{C}(\hat{\mathbf{x}} - \mathbf{x}) p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|\mathbf{y}) d\mathbf{x} \tag{1.6}$$

A common choice for cost function $\mathcal{C}$ is the hit-or-miss function defined as follows:

$$\mathcal{C}(\tau) = \begin{cases} 0 & if \, \|\tau\| < \delta \\ 1 & if \, \|\tau\| \geq \delta \end{cases} \tag{1.7}$$

With $\delta \to 0$, the solution to the equation (1.6) is $\hat{\mathbf{x}}$ which maximizes the posterior distribution, hence the name Maximum A Posteriori (MAP) estimator, defined as:

$$\hat{\mathbf{x}}_{MAP} = \arg\max_{x \in \mathcal{X}} p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|\mathbf{y}). \tag{1.8}$$

Using the Bayes theorem, the posterior density $p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|\mathbf{y})$ is expressed as a function of the likelihood function $p_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}|\mathbf{x})$, and of $p_{\mathcal{X}}(\mathbf{x})$ which models the prior knowledge on $\mathbf{x}$:

$$p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|\mathbf{y}) = \frac{p_{\mathcal{X}}(\mathbf{x}) p_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}|\mathbf{x})}{p_{\mathcal{Y}}(\mathbf{y})}. \tag{1.9}$$

Since $p_{\mathcal{Y}}(\mathbf{y})$ does not depend on $\mathbf{x}$, the MAP estimator can be reduced to the maximization of the numerator in equation (1.9). Assuming the formation model with Gaussian noise in equation (1.1), the MAP estimator is thus equivalent to:

$$\begin{aligned} \hat{\mathbf{x}}_{MAP} &= \arg\min_{x \in \mathcal{X}} \{ -\log p_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}|\mathbf{x}) - \log p_{\mathcal{X}}(\mathbf{x}) \} \\ &= \arg\min_{x \in \mathcal{X}} \{ \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sigma^2 \mathcal{R}(\mathbf{x}) \} \end{aligned} \tag{1.10}$$

One can note that the MAP formulation of the inverse problem in equation (1.10) is closely related to the variational approach described in equation (1.5), with the negative loglikelihood function being the data fidelity-term and the negative log-prior function being the regularization term.

Other cost functions can be considered for the Bayesian estimator. For instance, another typical cost function $\mathcal{C}$ is the quadratic function $\mathcal{C} = \|\tau\|^2$. In this case, the Bayesian estimator is defined as the posterior mean, leading to the Minimum Mean Squared Error (MMSE) estimator:

$$\hat{\mathbf{x}}_{MMSE} = \mathbb{E}[\mathbf{x}|\mathbf{y}] = \int_{x \in \mathcal{X}} \mathbf{x} p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|\mathbf{y}) d\mathbf{x}. \tag{1.11}$$

In the following sections, we consider the problems posed by the variational and MAP approaches. The MMSE estimator and estimators based on other cost functions are out of the scope of this thesis.

1.3 Algorithms

In this section, we introduce the most common classes of algorithms used to solve the minimization problems posed by the variational approach in equation (1.5), and equivalently, by the MAP formulation in equation (1.10).

1.3.1 Iterative optimization algorithms

Since the problem posed in equation (1.5) generally does not admit a closed-form solution, classical approaches to solve the minimization problem use iterative optimization algorithms. The idea behind these algorithms is to generate a sequence of improving estimates, assuming a function f that computes an estimate $\hat{\mathbf{x}}^{k+1}$ from the estimate $\hat{\mathbf{x}}^k$ by minimizing a criteria formulated as a cost function $\mathcal{C}$:

$$\hat{\mathbf{x}}^{k+1} = f(\hat{\mathbf{x}}^k), \quad \text{s.t.} \quad \mathcal{C}(\hat{\mathbf{x}}^{k+1}) \leq \mathcal{C}(\hat{\mathbf{x}}^k) \tag{1.12}$$

The sequence in equation (1.12) converges to a fixed point $\mathbf{x}^*$, defined by $\hat{\mathbf{x}}^* = f(\hat{\mathbf{x}}^*)$, which is expected to well-estimate the original image. In the case of the variational approach, the criteria to be minimized is the sum of the data-fidelity term and the regularization term in equation (1.5).

Several iterative optimization algorithms have been designed in the literature, using different designs of the function f in equation (1.12). These methods can be classified into two main categories: derivative-based methods and proximal methods.

Derivative-based methods

As mentioned previously, the optimization process in (1.12) aims at finding a solution $\hat{\mathbf{x}}$ that minimizes a cost function $\mathcal{C}$. Assuming a differentiable function $\mathcal{C}$, a common strategy is to use its derivatives to find where to move the current estimate in the search space. There are mainly two types of derivative-based methods: first-order (or gradient) algorithms and second-order (or Newton) algorithms.

First-order algorithms use the first derivatives of the cost function $\mathcal{C}$, i.e. the gradient of the function at a point x denoted $\nabla\mathcal{C}(x)$, to estimate the next point. With the gradient representing the direction where the cost function to be minimized is increasing the most, the strategy of first-order algorithms is to move in its opposite direction to find an estimate that reduces the cost function. The most famous first-order algorithm is the gradient

descent algorithm [33], which computes the next estimate as follows:

$$\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k - \gamma^k \nabla \mathcal{C}(\hat{\mathbf{x}}^k), \tag{1.13}$$

with γ^k being the step size, also known as the learning rate, which controls how far to move in the space search at each iteration. Therefore, γ^k needs to be carefully set. A small value of γ^k will result in a long computation time, whereas a large value will result in bouncing around the search space instead of converging to an optimal estimate. Based on the gradient descent in Equation (1.13), several algorithms have been proposed to improve the efficiency of the gradient descent algorithm, mainly for the optimization of deep neural networks. The momentum gradient descent [34] keeps track of a few past descent directions at each iteration, which are used to improve the convergence speed of the algorithm. The Nesterov Accelerated Gradient (NAG) [35] is another momentum gradient descent method, in which the gradient of the approximated next estimate, approximated using the momentum, is used to compute the next estimate. Methods were also proposed to adapt the step size γ^k to eliminate the need to manually tune it. The Adagrad method [36] and the RMSProp method [37] compute a step size per parameter to be optimized, which is adapted depending on the past gradients, similarly to the momentum methods. The Adaptive Moment Estimation (Adam) [38] was then proposed to take advantage of both the momentum and the adaptative step size.

The second-order optimization algorithms use the second derivatives, or the Hessian matrix, of the cost function $\mathcal{C}$ to be minimized in addition to the first derivatives. The idea behind this approach is to approximate the function $\mathcal{C}$ with a quadratic function that is supposed to be easier to minimize. A common strategy to construct this quadratic function is to compute the Taylor approximation of the function $\mathcal{C}$, using the first and second order derivatives. The next estimate $\hat{\mathbf{x}}^{k+1}$ is then computed as follows:

$$\hat{\mathbf{x}}^{k+1} = \underset{\mathbf{x}\in\mathcal{X}}{\arg\min}\, \mathcal{C}(\hat{\mathbf{x}}^k) + \nabla \mathcal{C}(\hat{\mathbf{x}}^k)^\top (\mathbf{x} - \hat{\mathbf{x}}^k) + \frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}^k)^\top \mathcal{H}(\hat{\mathbf{x}}^k)(\mathbf{x} - \hat{\mathbf{x}}^k), \tag{1.14}$$

with $\mathcal{H}$ being the Hessian matrix of $\mathcal{C}$ at $\hat{\mathbf{x}}^k$. This approach is often referred to as the Newton method [39], or quasi-Newton methods when $\mathcal{H}$ is an approximation of the Hessian matrix of $\mathcal{C}$.

Proximal methods

While derivative-based optimization algorithms are well-suited for the minimization of a differentiable cost function, proximal methods have been introduced to solve convex minimization problems of the form:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{C}(\mathbf{x}), \quad \text{s.t.} \quad \mathcal{C}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) + \mathcal{R}(\mathbf{x}), \tag{1.15}$$

where $\mathcal{F}$ is convex and differentiable and $\mathcal{R}$ is convex and possibly non-differentiable. Note that the problem posed in equation (1.5) is a particular case of the variational formulation in equation (1.15). The cost function $\mathcal{C}$ to be minimized is thus composed of a sum of convex possibly non-differentiable functions. Finding $\hat{\mathbf{x}}$ that minimizes $\mathcal{C}$, is equivalent to finding $\hat{\mathbf{x}}$ such that:

$$0 \in \partial\mathcal{C}(\hat{\mathbf{x}}), \tag{1.16}$$

where $\partial\mathcal{C}(\hat{\mathbf{x}})$ denotes the subdifferential of the sum of functions. A key operator to solve this problem is the proximal operator introduced by Moreau [40], noted $\text{prox}_{\mathcal{C}}(.)$ and defined as:

$$\text{prox}_{\mathcal{C}}(\hat{\mathbf{x}}) = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{C}(\mathbf{x}) + \frac{1}{2}\left\|\hat{\mathbf{x}} - \mathbf{x}\right\|_2^2. \tag{1.17}$$

The proximal operator is thus composed of a sum of two functions: the cost function to be minimized, i.e. $\mathcal{C}$ in our case, and a proximity term, i.e. $\frac{1}{2}\left\|\hat{\mathbf{x}} - \mathbf{x}\right\|_2^2$. Intuitively, computing the proximal operator of the function $\mathcal{C}$ at $\hat{\mathbf{x}}$ corresponds to finding the minimum of $\mathcal{C}$ in the neighborhood of $\hat{\mathbf{x}}$. Finding the explicit forms of proximal operators has thus been an attractive subject of research, with several explicit forms, or closed-form solutions, presented in recent works, e.g. in [41]–[44].

A variety of proximal optimization algorithms have been proposed in the literature to solve the minimization problem posed in equation (1.15), and so in equation (1.5). The forward-backward (FB) algorithm by Combettes et al. [41] combines the gradient descent presented in equation (1.13) and the proximal operator in equation (1.17) to deal with the minimization of respectively the differentiable and the non-differentiable terms. The method alternates between computing the gradient descent step over the differentiable function $\mathcal{F}$ and computing the proximal operator of the non-differentiable function $\mathcal{R}$. An iteration of the forward-backward algorithm is thus described as follows:

$$\hat{\mathbf{x}}^{k+1} = \text{prox}_{\mathcal{R}}(\hat{\mathbf{x}}^k - \gamma^k \nabla \mathcal{F}(\hat{\mathbf{x}}^k)) \tag{1.18}$$

Similarly, the iterative shrinkage-thresholding algorithms (ISTA) by Daubechies et al. [45] and its derivatives [35], [46], [47] use a shrinkage operator to model the proximal operator of the ℓ_1-norm and the ℓ_2-norm. The Douglas-Rachford algorithm [48] is a splitting algorithm designed to solve the problem in equation (1.15) with non-differentiable terms. Another well-known proximal algorithm, which will be further used in this thesis, is the Alternating Direction Method of Multipliers (ADMM) by Boyd et al. [49], with its simplified form being the Half-Quadratic Splitting algorithm (HQS) by Geman et al. [50]. It solves the minimization problem in equation (1.5) by introducing a constrained minimization problem by splitting the different terms:

$$\hat{\mathbf{x}} = \underset{\mathbf{x},\mathbf{v}\in\mathcal{X}}{\arg\min} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sigma^2 \mathcal{R}(\mathbf{v}), \quad \text{s.t.} \mathbf{x} = \mathbf{v} \tag{1.19}$$

To incorporate the constraint $\mathbf{x} = \mathbf{v}$ in the minimization problem, we define the augmented Lagrangian function $\mathcal{L}$, in which a dual variable $\mathbf{u}$ and a penalty term ρ are introduced:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{v}, \mathbf{u}) &= \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sigma^2 \mathcal{R}(\mathbf{v}) + \mathbf{u}^\top (\mathbf{x} - \mathbf{v}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{v}\|_2^2, \\ &= \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sigma^2 \mathcal{R}(\mathbf{v}) + \frac{\rho}{2} \left\| \mathbf{x} - \mathbf{v} + \frac{\mathbf{u}}{\rho} \right\|_2^2 - \frac{\|\mathbf{u}\|_2^2}{2\rho} \end{aligned} \tag{1.20}$$

The ADMM optimization algorithm consists of minimizing the augmented Lagrangian function $\mathcal{L}$ for the variables $\mathbf{x}$ and $\mathbf{v}$ alternatively and updating the dual variable $\mathbf{u}$:

$$\hat{\mathbf{x}}^{k+1} = \underset{\mathbf{x}}{\arg\min} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\rho}{2} \left\| \mathbf{x} - \mathbf{v}^k + \mathbf{u}^k \right\|_2^2, \tag{1.21}$$

$$\mathbf{v}^{k+1} = \underset{\mathbf{v}}{\arg\min} \frac{\rho}{2} \left\| \mathbf{v} - (\hat{\mathbf{x}}^{k+1} + \mathbf{u}^k) \right\|_2^2 + \lambda \cdot \mathcal{R}(\mathbf{v}) \tag{1.22}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (\hat{\mathbf{x}}^{k+1} - \mathbf{v}^{k+1}), \tag{1.23}$$

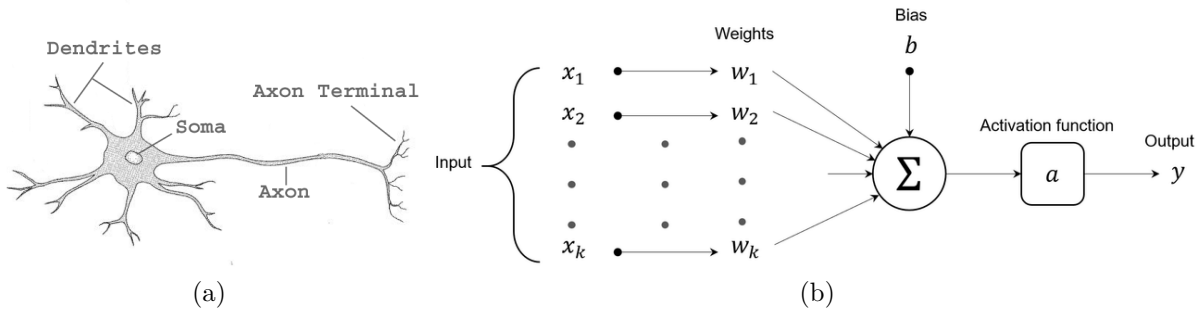


Figure 1.2 – (a) Structure of a neuron [57]: The dentrites deliver the input signals to the soma, which emits an output signal (if the input signals exceed a threshold) to the axon, which delivers it to another neuron (b) Structure of a neural network: the input signals are transformed to compute the output signal using a linear transformation, with learnable weights and biases, and and a non-linear activation function.

1.3.2 Deep learning algorithms

In the last decade, a plethora of deep learning techniques have emerged to deal with many image processing tasks [51]–[54]. The principle of deep learning is to optimize a very large number of parameters of a model in order to minimize a chosen criteria. In the following, we introduce the basics of deep learning for image processing.

Principle of neural networks

Many machine learning models have been designed in the literature [55]. In the context of image processing, deep learning techniques using neural network architectures have been a common choice of algorithms and have yielded a significant performance leap in the last years.

Neural network architectures are inspired by the human brain, especially for its capability of solving a variety of image processing tasks. Our brain contains about 100 billion neurons that communicate with each other with signals passing through thousands of synapses for each neuron. The spiking neuron model [56] specified how neurons share and produce information between each other. In a nutshell, the information signal is represented by a neuron spike, i.e. an action potential. The treatment and emission of signals in a neuron are divided into three main steps, as represented in Figure 1.2a. The dendrites transmit signals coming from other neurons to the soma. The latter processes the input signals and, if they exceed a certain threshold, emits an output signal in the axon, which delivers the output signal to another neuron.

Similarly, a neural network is composed of artificial neurons that gather, treat, and

emit signals. A neural network is an acyclic graph with multiple layers, each containing multiple neurons connected to other neurons in the previous and next layers. The output signal emitted by neurons in a layer is the result of a linear transformation of the input signals passing through a non-linear activation function. The structure of an artificial neuron is presented in Figure 1.2b. There are several neural network layers, with the most common ones for image processing being the fully-connected layers and the convolutional layer. Both network architectures are illustrated in Figure 1.3.

Fully-connected layer: every neuron of the previous layer is connected to each neuron of the current layer, as illustrated in Figure 1.3a. Formally, let $\mathbf{z}^i \in \mathbb{R}^m$, with m being the number of input signals, be the vector containing the input signals coming from the layer number i. Let $\mathbf{W}^{i+1} \in \mathbb{R}^{n \times m}$ and $\mathbf{b}^{i+1} \in \mathbb{R}^n$ be the weights and the bias associated with the linear transformation, and $\sigma(.)$ be the non-linear activation function of the current layer. The output signal $\mathbf{z}^{i+1} \in \mathbb{R}^n$ is then computed as follows:

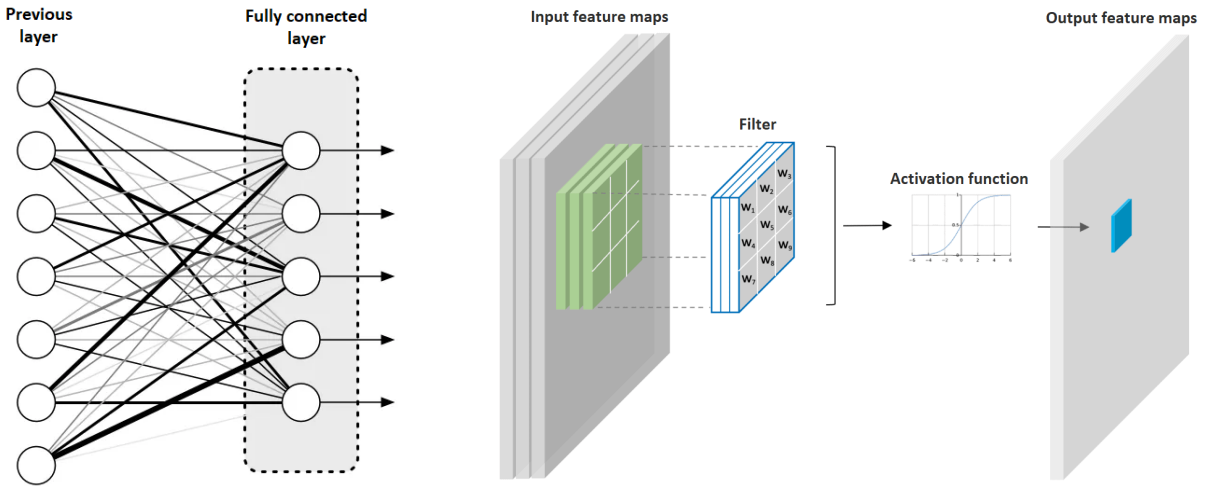
$$\mathbf{z}^{i+1} = \sigma(\mathbf{W}^{i+1}\mathbf{z}^i + \mathbf{b}^{i+1}). \tag{1.24}$$

In the context of image processing, fully-connected layers have the advantage of capturing global information in the image since all the information is passing through every neuron, which comes, however, with a high computational cost.

Convolutional layer: contrary to fully-connected layers, which forces global connections between neurons, convolutional layers compute local analysis of the input signal. They were created, at first, for 2D image processing to capture spatial dependencies in the signal while maintaining a suitable computational cost. At each layer, a set of filters, or kernels, are convolved with the input image, resulting in a set of convolved images, named feature maps, as illustrated in Figure 1.3b. Formally, let $\mathbf{z}^i \in \mathbb{R}^{h \times w \times c}$ be an input image with height h, width w and number of channels c. Let $\mathbf{W}^{i+1} \in \mathbb{R}^{h_k \times w_k \times c \times n}$ and $\mathbf{b}^{i+1} \in \mathbb{R}^n$ be the n bias and convolutional filters, with kernel size $h_k \times w_k$, and $\sigma(.)$ be the non-linear activation function of the current layer. The output feature map $\mathbf{z}^{i+1} \in \mathbb{R}^{h \times w \times n}$ is then computed as follows:

$$\mathbf{z}^{i+1}_{:,:,j} = \sum_{k=0}^{c} \sigma(\mathbf{z}^i_{:,:,k} * \mathbf{W}^{i+1}_{:,:,k,j} + \mathbf{b}^{i+1}_j), \tag{1.25}$$

with $*$ denoting the convolution operator.



(a) (b)

Figure 1.3 – (a) Structure of a fully connected layer: every neuron of the previous layer is connected to each neuron of the current layer (b) Structure of a convolutional layer: the input image is convolved with filters to produce feature maps.

Neural network optimization

A neural network can be seen as a function that maps the input signal space to the output signal space. For instance, in the case of an image reconstruction problem, as posed in the variational approach in equation (1.5), a neural network takes a degraded image $\mathbf{y} \in \mathcal{Y}$ as input and is expected to output the original image $\hat{\mathbf{x}} \in \mathcal{X}$. The output of the neural network is, however, dependent on the values of the weights $\mathbf{W}$ and bias $\mathbf{b}$ in each layer of the network. In the following, we note θ the parameters of the neural network, i.e. the concatenation of all the weights and biases of the neural network, with θ_i being the parameters of the i^{th} layer. These parameters θ thus need to be optimized such that the neural network maps an input signal to its corresponding target output signal. Similarly to an individual who learns by experience, i.e. learning from examples, the parameters of a neural network are optimized, or learned, using a training set of data. In this document, we only consider a supervised learning setup, i.e. where an output target signal is known for each input signal in a training dataset. Note that there exists a large literature on unsupervised learning [58], which is out of the scope of this document.

Let a training set of N paired data $\{(\mathbf{y}^1, \mathbf{x}^1),...,(\mathbf{y}^N, \mathbf{x}^N)\}$, with an input data $\mathbf{y}^n \in \mathcal{Y}$ being associated to its target output data $\mathbf{x}^n \in \mathcal{X}$. The parameters θ of a neural network $\mathcal{N}$ are optimized such that, assuming an input $\mathbf{y}^n$, its output $\hat{\mathbf{x}}^n$ well-estimate the target

output $\mathbf{x}^n$. The criteria to be minimized is thus a function, noted $\mathcal{L}(.)$ and generally referred to as loss function, which computes the error between the target data $\mathbf{x}^n$ and the prediction $\hat{\mathbf{x}}^n$ of the neural network. The optimization of the parameters θ is thus posed as the minimization of the loss function over the whole training dataset:

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} \mathcal{L}(\mathcal{N}(\mathbf{y}^n, \theta), \mathbf{x}^n). \tag{1.26}$$

As mentioned in Section 1.3.1, the common strategy to solve this minimization problem is to use first-order iterative optimization algorithms based on the gradient descent algorithm in equation (1.13):

$$\theta^{k+1} = \theta^k - \gamma^k \frac{\partial \sum_{n=1}^{N} \mathcal{L}(\mathcal{N}(\mathbf{y}^n, \theta^k), \mathbf{x}^n)}{\partial \theta^k}. \tag{1.27}$$

The parameters are generally initialized following specific distributions [59]–[61]. To reduce the computational burden of computing the derivative of the loss function with respect to the parameters θ, the usual strategy is to randomly select mini-batches of paired data at each gradient descent step, instead of using the whole training dataset. This strategy is known in the literature as the mini-batch stochastic gradient descent [62]. Note that a neural network is composed of a set of sequential layers. Therefore, the function $\mathcal{N}$, representing the neural network, is a composite function, with each function computing a layer. The gradients of the weights of each layer are thus computed using the chain rule of derivatives. With $\mathbf{z}^i$ being the output of the i^{th} layer l^i, and I being the number of layers, the chain rule is computed as follows:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^I} \frac{\partial l^I(\mathbf{z}^{I-1}, \theta_I)}{\partial \mathbf{z}^{I-1}} \ldots \frac{\partial l^i(\mathbf{z}^{i-1}, \theta_i)}{\partial \theta_i}, \tag{1.28}$$

where each part of the chain rule, e.g. $\frac{\partial l^i(\mathbf{z}^{i-1}, \theta)}{\partial \mathbf{z}^{i-1}}$, is a Jacobian matrix, and where $\frac{\partial \mathcal{L}}{\partial \mathbf{z}^I}$ denotes the row vector of the gradient of the loss function $\mathcal{L}$ with respect to the output $\mathbf{z}^I$ of the neural network. The backpropagation [63] refers to the algorithm that computes the gradient of the loss function with respect to each parameter to be optimized in the neural network, by propagating backward the error from the loss function following the chain rule in equation (1.28).

1.4 Regularization in inverse problems

In Section 1.3, the traditional algorithms to solve an ill-posed image inverse problem were presented. An important component of the variational and the MAP formulations of the problem in equations (1.5) and (1.10) is the regularization function, which constrains the space of acceptable estimates assuming prior knowledge on the type of signal to be reconstructed. Building an efficient image prior through a regularization function is, however, a challenging task. Several regularization approaches have been proposed in the literature [32], which are presented in this section.

1.4.1 Hand-crafted priors

A first category of regularization approaches is to build a hand-crafted function modeling a specific image prior. A variety of functions have been designed in the literature, mostly relying on the Tikhonov regularization or on sparsity prior.

Tikhonov regularization

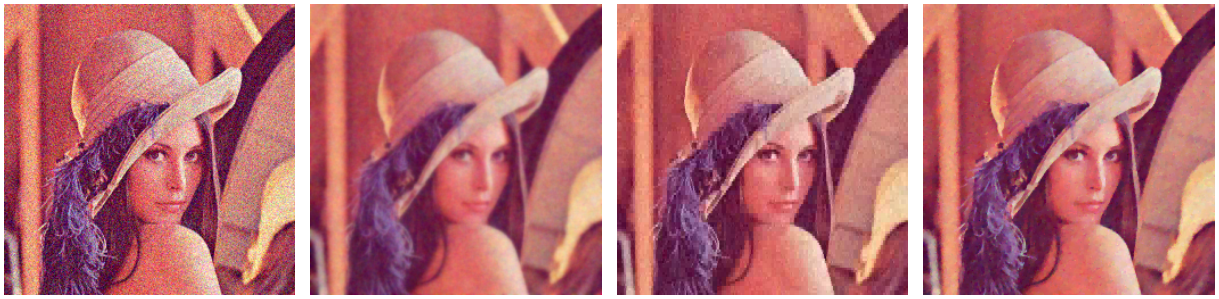
The most well-known first regularization technique is the Tikhonov regularization, proposed by Tikhonov [31]. With this regularization, the minimization problem in equation (1.5) is in the following form:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{X}}{\arg\min} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\Gamma\mathbf{x}\|_2^2 , \tag{1.29}$$

with the Tikhonov matrix Γ being a linear operator that promotes desirable properties on the signal $\mathbf{x}$. A well-known choice for the matrix Γ is the identity matrix $\Gamma = I_d$. The regularization constraint becomes the ℓ_2-norm, promoting solutions with small norms with all coefficients being small. Another approach consists of defining Γ as a laplacien filter that promotes small variations in the image, i.e. assuming a smoothness prior on images. An application of the Tikhonov regularization applied to image denoising is presented in Figure 1.4b.

Sparsity regularization

Sparsity prior has been widely considered for a variety of signal processing tasks, including image restoration [66], feature classification [67], and image compression [68]. A signal is said to be sparse when most of its coefficients are zero under a certain linear



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

Figure 1.4 – Examples of hand-crafted regularization applied to a denoising image inverse problem: (a) Image corrupted with additive Gaussian noise (b) Tikhonov regularization [31] promoting smoothness (c) Sparsity prior using the wavelet regularization with the Bayes Shrink method [64] (d) Total variation regularization proposed by Chambolle et al. [65].

transformation defined by a dictionary, which is a common assumption for natural images. Let a dictionary Φ and $\mathbf{s}$ a vector of coefficients in Φ. The variational problem formulated in equation 1.5 becomes the problem of finding $\hat{\mathbf{s}}$ which minimizes:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \|\mathbf{A}\Phi\mathbf{s} - \mathbf{y}\|_2^2 + \lambda \|s\|_1 , \tag{1.30}$$

with the ℓ_1-norm promoting sparse solutions. Common choices of linear transformations associated with the dictionary Φ are the Fourier transform [69], the cosine transform [70] and the wavelet transform [71], since natural images are generally sparse representations in the frequency domain. A regularization using the wavelet transform is illustrated in Figure 1.4c. A very popular regularization promoting sparsity is the total variation, introduced by Rudin et al. [72]. The principle of total variation relies on the fact that images are mostly composed of homogeneous areas, meaning that variations of intensity are mainly located around the edges. The gradient of an image is thus expected to be a sparse representation. Assuming an image $\mathbf{x} \in \mathbb{R}^{I \times J}$ and $\nabla^h(.)$ and $\nabla^v(.)$ denoting the gradient operators in respectively the horizontal and vertical directions. The total variation $V(\mathbf{x})$ of the image $\mathbf{x}$ is:

$$V(\mathbf{x}) = \sum_{i=1}^{I} \sum_{j=1}^{J} \left(|[\nabla^h(\mathbf{x})]_{i,j}| + |[\nabla^v(\mathbf{x})]_{i,j}| \right) \tag{1.31}$$

While high quality images were obtained in the literature using a total variation regularization, it is also known to produce staircasing effects [73] in areas with smooth variations. Figure 1.4d shows an example of total variation applied to image denoising.

1.4.2 Plug-and-play methods

As illustrated in Figure 1.4, it is a challenging task to obtain high quality reconstructed images using a hand-crafted regularization function. It is mostly due to the complexity of designing a very complex image prior with hand-crafted functions. To cope with this issue, more and more researchers in the image processing community are dedicated to create learned image priors, with a particular interest in deep learning techniques. This section deals with a first category of optimization methods with learned priors, referred to as "Plug-and-play", which has been introduced by Venkatakrishnan et al. [74], and where an off-the-shelf denoiser is used to represent the prior, and is plugged into an iterative optimization algorithm. More precisely, instead of constructing the regularization function explicitly, the plug-and-play approach aims at using a denoising neural network to approximate the gradient of the regularization function or its proximal operator. These approximations are thus plugged into an iterative algorithm to solve any type of image inverse problem.

Image denoising is one of the most well-studied image inverse problems in the literature [13]–[15], with a particular interest in the denoising problem considering an additive white Gaussian noise in the degradation process. A variety of deep learning architectures have been proposed for noise removal, including the well-known DRUnet [75] and DnCNN [76] architectures. Let a Gaussian denoiser $\mathcal{D}$ be parameterized by the parameters θ which are optimized as follows:

$$\begin{aligned} \hat{\theta} &= \arg\min_{\theta} \mathcal{L}(\mathcal{D}(\mathbf{y}, \theta), \mathbf{x}), \\ \text{with} \quad \mathbf{y} &= \mathbf{x} + \epsilon, \quad \text{s.t.} \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \end{aligned} \tag{1.32}$$

where σ^2 is the variance of the Gaussian noise and $\mathcal{L}(.)$ is the loss function computing the error between the denoised image $\mathcal{D}(\mathbf{x} + \epsilon, \theta)$ and the original $\mathbf{x}$. Note that solving an image denoising inverse problem implies the computation of the MAP estimator in equation (1.10) for a denoising problem:

$$\mathcal{D}(\mathbf{y}, \theta) = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \sigma^2 \mathcal{R}(\mathbf{x}). \tag{1.33}$$

One can note that applying the denoiser $\mathcal{D}$ is equivalent to computing the proximal operator of the regularization function $\mathcal{R}$. Therefore, the denoiser $\mathcal{D}$ implicitly learns a

prior on the target image space. Consequently, a first Plug-and-Play approach aims at using an off-the-shelf denoiser in a proximal optimization method to solve any image inverse problem. It was first introduced using the ADMM optimization algorithm by Venkatakrishnan et al. [74]:

$$\hat{\mathbf{x}}^{k+1} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\rho}{2} \left\|\mathbf{x} - \mathbf{v}^k + \mathbf{u}^k\right\|_2^2, \tag{1.34}$$

$$\mathbf{v}^{k+1} = \mathcal{D}(\hat{\mathbf{x}}^{k+1}, \theta), \tag{1.35}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + (\hat{\mathbf{x}}^{k+1} - \mathbf{v}^{k+1}), \tag{1.36}$$

Similarly, the denoiser $\mathcal{D}$ can be considered to compute the MMSE estimator in equation (1.11). Using Tweedie's formula [77], the relationship between the denoiser $\mathcal{D}$ and the probability density $p_{\mathcal{Y}}(\mathbf{y})$ is established as:

$$\mathcal{D}(\mathbf{y}, \theta) = \mathbf{y} - \sigma^2 \nabla \log(p_{\mathcal{Y}}(\mathbf{y})). \tag{1.37}$$

Therefore, the true prior distribution $\mathcal{P}_{\mathcal{X}}$ can be approximated using the corrupted data distribution $\mathcal{P}_{\mathcal{Y}}$, assuming Gaussian noise with a small variance σ^2. It is well-known [78] that the Unadjusted Langevin Algorithm can produce samples from a probability density $p_{\mathcal{X}}(\mathbf{x})$ knowing only $\nabla \log(p_{\mathcal{X}}(\mathbf{x}))$. Samples following the probability density $p_{\mathcal{X}|\mathcal{Y}}$ can thus be computed using a denoiser $\mathcal{D}$ as follows:

$$\begin{aligned}
\hat{\mathbf{x}}^{k+1} &= \hat{\mathbf{x}}^k + \gamma^k \nabla \log(p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}^{\mathbf{k}}|\mathbf{y})) + \sqrt{2\gamma^k}\mathbf{z}^{k+1}, \\
&= \hat{\mathbf{x}}^k + \gamma^k \nabla \log(p_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}|\mathbf{x}^k)) + \gamma^k \nabla \log(p_{\mathcal{X}}(\mathbf{x}^{\mathbf{k}})) + \sqrt{2\gamma^k}\mathbf{z}^{k+1}, \\
&\approx \hat{\mathbf{x}}^k + \gamma^k \nabla \left[\frac{1}{2}\left\|\mathbf{A}\hat{\mathbf{x}}^{\mathbf{k}} - \mathbf{y}\right\|_2^2\right] + \gamma^k \left(\mathbf{x}^{\mathbf{k}} - \frac{\mathcal{D}(\mathbf{x}^{\mathbf{k}}, \theta)}{\theta^2}\right) + \sqrt{2\gamma^k}\mathbf{z}^{k+1},
\end{aligned} \tag{1.38}$$

with $\mathbf{z}^{k+1} \sim \mathcal{N}(0, Id)$ and γ denoting the step size. The MMSE estimate $\hat{\mathbf{x}}_{MMSE}$ is then computed by averaging multiple samples generated using the algorithm described in equation (1.38).

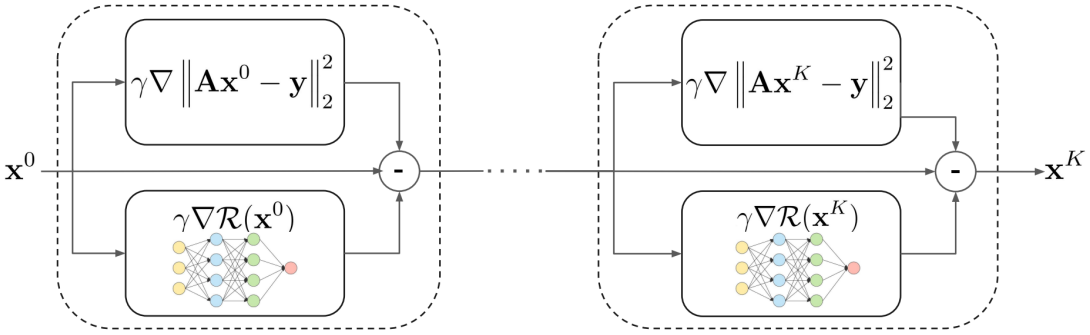


Figure 1.5 – Unrolled gradient descent: the learned neural network acts as the gradient of the regularization function.

1.4.3 Unrolled optimization algorithms

Unrolling a fixed number of iterations of optimization algorithms is another way of coupling iterative optimization and deep learning techniques. Contrary to the plug-and-play approach, where the optimization of the learned prior is decoupled from the iterative optimization algorithm, the learnable network in an unrolled algorithm is trained end-to-end within the iterative algorithm, hence in a way that takes into account the data-fidelity term. Therefore, the prior is learned to obtain optimized results for a given image inverse problem and for a given iterative optimization algorithm. An illustration of an unrolled gradient descent is proposed in Figure 1.5.

Several optimization algorithms have been unrolled in the literature, where a learned regularization network is used at each iteration of the optimization algorithm. The learned network plays a specific role for each unrolled algorithm. Unrolling methods were introduced by Gregor et al. [79], where the Iterative Shrinkage Thresholding Algorithm (ISTA) is unrolled, and where the neural network is learned to give the optimal sparse code. The gradient descent and the proximal gradient algorithms are considered in [80] and [81] respectively. When unrolling the gradient descent algorithm, the network used for regularization is expected to act as the gradient of a regularizer [80]. In the case of the proximal gradient algorithms in [81], the neural network computes a proximal mapping of the regularization, which can be interpreted as a Gaussian denoiser. Unrolled proximal algorithms were also proposed in the literature, following the HQS algorithm [82] and the ADMM optimization method [83]. In the latter methods, a proximal operator of the regularization function is learned.

Thanks to the combination of powerful iterative optimization algorithms and deep

complex image priors, unrolled optimization algorithms have achieved state-of-the-art results for a variety of image inverse problems [4], [5], [84], [85]. It is important to notice that training the neural network within the unrolled optimization algorithm requires, however, high computational resources, especially in terms of memory usage and computation time. Therefore, the number of iterations considered in practice is usually controlled to avoid a high computational burden, which can affect the quality of the reconstructed images. There are mainly two categories of training approaches for unrolled optimization algorithms, respectively based on explicit backpropagation and implicit backpropagation via the deep equilibrium approach.

Unrolling with explicit backpropagation

The neural network involved is optimized by backpropagating the gradients of a loss function $\mathcal{L}$ explicitly from the output of the unrolled optimization algorithm to its input. Formally, let an iterative optimization algorithm be unrolled for K iterations, with f being the function, parameterized with θ, representing an iteration of the unrolled optimization algorithm, and $\hat{\mathbf{x}}^k$ be the output of the k^{th} iteration. For the sake of simplicity, we will consider a recurrent unrolled algorithm, i.e. every unrolled iteration uses the same parameters θ. The differentiation involved in the explicit backpropagation is written using the chain rule as follows:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{k=1}^{K} \left[\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{x}}^K} \frac{\partial f(\hat{\mathbf{x}}^{K-1}, \theta)}{\partial \hat{\mathbf{x}}^{K-1}} \dots \frac{\partial f(\hat{\mathbf{x}}^k, \theta)}{\partial \theta} \right]. \tag{1.39}$$

With the neural network used at each unrolled iteration, one can notice that the computational burden of the explicit backpropagation is linearly dependent on the number of unrolled iterations. Therefore, the optimization of an unrolled optimization algorithm usually suffers from intensive memory usage. To cope with this issue, the number of unrolled iterations considered is usually very small. As a consequence, while iterative optimization algorithms iteratively improve an estimate until convergence, the unrolled optimization algorithm is thus trained to recover the best estimate in a fixed number of iterations, so it loses its convergence properties, as illustrated in figure 1.6. The considered fixed number of iterations thus needs to be carefully tuned for each task, or sometimes for each image, to ensure the best reconstruction performances.

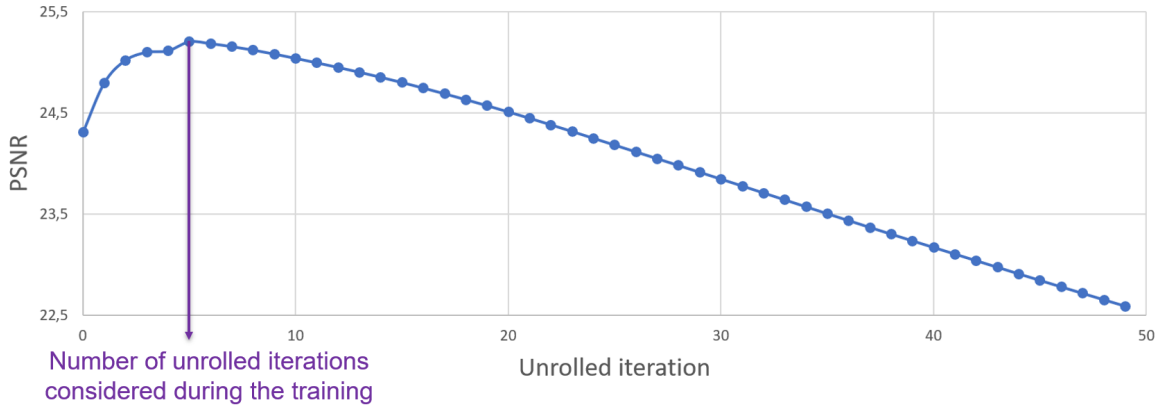


Figure 1.6 – Analysis of the convergence of the unrolled optimization algorithm trained using an explicit backpropagation: the algorithm reconstructs its best estimate after a fixed number of iterations and does not converge to a desired estimate.

Deep equilibrium for unrolled optimization algorithms

While classical iterative methods iterate until convergence, unrolled optimization algorithms consider a fixed number of iterations, which is usually small. Indeed, learning a prior end-to-end while unrolling more iterations would considerably increase the memory usage of the explicit backpropagation.

To overcome this issue, Deep Equilibrium approaches were introduced for unrolled optimizations [86], [87]. The Deep Equilibrium models (DEQ) [86], [88] and the Jacobian-Free Backpropagation Implicit (JFBI) Networks [87], [89] have been introduced using implicit backpropagations, based on the Recurrent Back Propagation (RBP) algorithm [90], [91]. It takes advantage of the implicit function theorem [92] to reduce the memory requirement of the backpropagation.

Let an iterative optimization algorithm be unrolled with a recurrent architecture. Let f be a function, parameterized with θ, representing an iteration of the unrolled optimization algorithm. Assuming that the iterative optimization algorithm converges, f admits a fixed point $\hat{\mathbf{x}}^*$ such that:

$$\hat{\mathbf{x}}^* = f(\hat{\mathbf{x}}^*, \theta). \tag{1.40}$$

This is equivalent to having the following function h:

$$\begin{aligned} h(\hat{\mathbf{x}}, \theta) &= \hat{\mathbf{x}} - f(\hat{\mathbf{x}}, \theta), \\ h(\hat{\mathbf{x}}^*, \theta) &= 0. \end{aligned} \tag{1.41}$$

According to equation (1.41), we can use the implicit function theorem [92] allowing us to define the following differentiable function z:

$$z(\theta) = \hat{\mathbf{x}}^* = f(\hat{\mathbf{x}}^*, \theta) = f(z(\theta), \theta), \tag{1.42}$$

Using equation (1.42), we are able to write the differentiation of $z(\theta)$ w.r.t. θ as follows:

$$\frac{\partial z(\theta)}{\partial \theta} = \frac{\partial f(z(\theta), \theta)}{\partial \theta} = \frac{\partial f(\hat{\mathbf{x}}^*, \theta)}{\partial \hat{\mathbf{x}}^*} \frac{\partial z(\theta)}{\partial \theta} + \frac{\partial f(\hat{\mathbf{x}}^*, \theta)}{\partial \theta}, \tag{1.43}$$

$$\frac{\partial z(\theta)}{\partial \theta} = \left[I - \frac{\partial f(\hat{\mathbf{x}}^*, \theta)}{\partial \hat{\mathbf{x}}^*} \right]^{-1} \frac{\partial f(\hat{\mathbf{x}}^*, \theta)}{\partial \theta}. \tag{1.44}$$

The differentiation in equation (1.44) details the differentiation used in the DEQ methods [86], [88]. It requires the computation of the inverse Jacobian $\left[I - \frac{\partial f(\hat{\mathbf{x}}^*, \theta)}{\partial \hat{\mathbf{x}}^*} \right]^{-1}$, which is the bottleneck of the DEQ method in terms of computation time.

Further developments in [89] have shown that, by omitting the inverse Jacobian in equation (1.44), one still obtains a descent direction of the loss with respect to θ. This leads to the JFBI method, where the differentiation in Eq. (1.44) is rewritten as:

$$\frac{\partial z(\theta)}{\partial \theta} = \frac{\partial f(\hat{\mathbf{x}}^*, \theta)}{\partial \theta}. \tag{1.45}$$

With both differentiations in equations (1.44) and (1.45), we are now able to optimize the network parameters θ in order to fit $z(\theta)$, i.e $\hat{\mathbf{x}}^*$, to the groundtruth. It is important to notice that both differentiations do not depend on how $\hat{\mathbf{x}}^*$ has been computed from $\hat{\mathbf{x}}^0$. Indeed, it only depends on the fixed point $\hat{\mathbf{x}}^*$ and a single application of f. Therefore, contrary to the explicit backpropagation in which the depth scales linearly with the number of unrolled iterations, the depth of implicit backpropagation is independent of the number of unrolled iterations. This explains why implicit methods require low memory usage compared to traditional unrolled methods.

However, implicit methods heavily rely on the assumption that the fixed point exists and is reached in practice in order to compute the gradients accurately. The usual strategy to compute the fixed point is to iterate until idempotence, i.e. the difference between the input and the output of an iteration is lower than an approximation error value ϵ. While having a high value of ϵ may result in instability due to a wrong estimation of the fixed point, a low value of ϵ increases the number of iterations, hence increases the computation time.

In summary, unrolling optimization methods with explicit backpropagation suffer from intensive memory usage. On the other hand, implicit unrolled optimization methods typically use a very large number of iterations to estimate the fixed point of the iterative algorithm, without increasing memory usage of the backpropagation, but suffer from intensive computation time.

1.5 Directions and objectives

In this thesis, we take a particular interest in unrolled optimization algorithms. As mentioned in Section 1.4.3, unrolled optimization algorithms are powerful tools to solve image inverse problems. However, the number of unrolled iterations is generally restricted due to the optimization strategy to avoid a high computation burden.

We seek to address this problem by proposing a novel approach to train an unrolled optimization algorithm, where the computational burden is considerably reduced, and where any number of unrolled iterations can be considered.

Chapter 2

Stochastic Unrolled Proximal Point Algorithm

2.1 Introduction

As mentioned in Section 1.4.3, unrolled optimization algorithms have achieved state-of-the-art results for a plethora of image inverse problems [4], [5], [84], [85], hence being the most promising approach to solve image inverse problems. The idea behind unrolled optimization algorithms is to learn a neural network end-to-end within an iterative optimization algorithm, such that it performs best for a specific image inverse problem and for a specific iterative optimization algorithm. The optimization of the neural network is done by backpropagating the gradients of the loss function from the output to the input of the unrolled optimization algorithm. Since a neural network is used at each unrolled iteration, the computation burden of computing the backpropagation is thus considerable. Indeed, the memory usage scales linearly with the number of unrolled iterations.

In this chapter, we address the problem of reducing the computational burden of training a neural networks in an unrolled optimization algorithm. One of the most promising approaches in the literature is the implicit backpropagation using the Deep Equilibrium approach [86], [88], [89]. This method takes advantage of the implicit function theorem [92] to rewrite the differentiation used in the backpropagation. More precisely, assuming that the iterative optimization algorithm converges to a fixed point, the backpropagation then depends only on the application of the unrolled iteration taking this fixed point as input. Therefore, contrary to the explicit backpropagation classically used in an unrolled optimization algorithm, where the memory usage depends on the number of unrolled iterations, the backpropagation with the deep equilibrium approach depends only on the last iteration, hence making the backpropagation memory-efficient. However, the fixed point of the iteration optimization algorithm needs to be computed to use the implicit backpropagation. In practice, the usual approach to compute this fixed point is to iterate

until convergence, which generally causes a high computational cost.

We thus propose a Stochastic Unrolled Proximal Point Algorithm (SUPPA) to solve linear image inverse problems. The proposed method is a novel memory-efficient optimization method for unrolled optimization algorithms based on the properties of the ADMM optimization algorithm. Each unrolled iteration is re-defined as a proximal mapping, by exploiting the fact that the ADMM is an application of the Proximal Point Algorithm [93], [94]. A stochastic training of the learned weights is performed by considering per-iteration optimization problems. While both explicit and implicit backpropagations aim to optimize the network parameters in order to fit the output of the unrolled iterative algorithm to the ground truth, the backpropagation in our proposed training uses the intermediate unrolled iterations independently of each other. This optimization strategy is thus independent of the number of computed unrolled iterations and is mathematically justified even if the fixed point is not estimated, hence making the training possible for any arbitrary number of unrolled iterations. We assess the method for several image inverse problems against recent unrolled optimization methods and task-specific deep methods. We show that the proposed method achieves state-of-the-art restoration performances for every image inverse problem considered.

2.2 Unrolled Proximal Point Algorithm

The Proximal Point Algorithm (PPA), introduced by Martinet [95], iteratively computes the resolvent of a maximal monotone operator $\mathbf{T}$:

$$\hat{\mathbf{x}}^{k+1} = \mathrm{P}_k(\hat{\mathbf{x}}^k), \tag{2.1}$$

$$\text{with} \quad \mathrm{P}_k(\hat{\mathbf{x}}^k) = (\mathbf{I} + c_k\mathbf{T})^{-1}\hat{\mathbf{x}}^k, \tag{2.2}$$

In the case where the operator $\mathbf{T}$ corresponds to a subdifferential of a convex function g, the proximal point algorithm iteratively computes the proximal point:

$$\hat{\mathbf{x}}^{k+1} = \mathrm{prox}_g(\hat{\mathbf{x}}^k), \tag{2.3}$$

$$\text{with} \quad \mathrm{prox}_g(\hat{\mathbf{x}}^k) = \arg\min_{\mathbf{x}} g(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}^k\|_2^2. \tag{2.4}$$

With g being convex, the proximal point algorithm converges to the minimum of g. Eckstein et al. [93] demonstrated that methods applying the Douglas—Rachford splitting

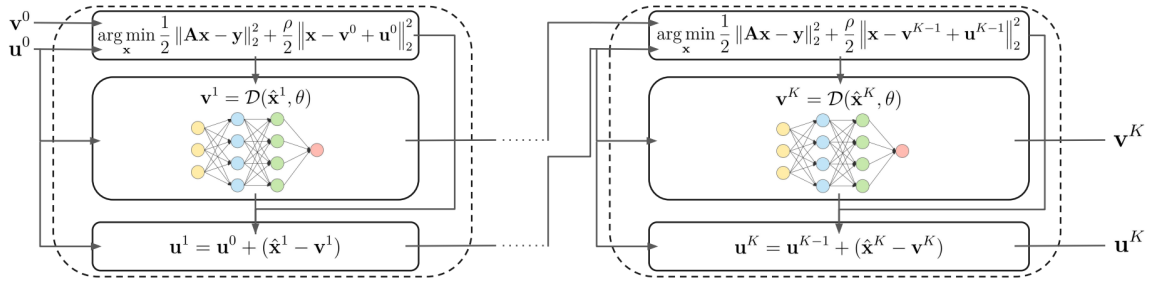


Figure 2.1 – Unrolled ADMM with a learned denoising neural network.

algorithm, such as the ADMM [94], are special cases of the proximal point algorithm described in equation (2.1). Let us now write the ADMM parameterized with θ as the following series:

$$\hat{\mathbf{x}}^{k+1} = f(\hat{\mathbf{x}}^k, \theta), \tag{2.5}$$

where, f represents one iteration of the ADMM algorithm, i.e., f performs equations. (1.34), (1.35), (1.36), with equation (1.35) computed by a trained Gaussian denoiser. With the ADMM being a special case of the proximal point algorithm, each ADMM iteration aims to compute the resolvent of a maximal monotone operator:

$$f(\hat{\mathbf{x}}^k, \theta) = \mathrm{P}_k(\hat{\mathbf{x}}^k). \tag{2.6}$$

To learn the parameters θ of an unrolled ADMM, we propose to define an optimization problem for each unrolled iteration k in order to fit $f(\hat{\mathbf{x}}^k, \theta)$ to a chosen maximal monotone operator $\mathrm{P}_k(\hat{\mathbf{x}}^k)$. Since the subdifferential of a convex function is a maximal monotone operator, we consider a maximal monotone operator defined as the subdifferential of a scalar-valued function g such that each iteration $f(\hat{\mathbf{x}}^k, \theta)$ of the unrolled ADMM, illustrated in Figure 2.1, is expressed as:

$$f(\hat{\mathbf{x}}^k, \theta) = \arg\min_{\mathbf{x}} g(\mathbf{x}, \theta) + \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}^k\|_2^2 = \mathrm{prox}_{g(.,\theta)}(\hat{\mathbf{x}}^k). \tag{2.7}$$

We will show in Section 2.4 that the end-to-end optimization problem of the unrolled ADMM can then be reduced to a set of per-iteration optimization problems.

2.3 Definition of the function g

With g convex, $\text{prox}_{g(.,\theta)}(\hat{\mathbf{x}}^k)$ reduces the distance between the estimate $\hat{\mathbf{x}}^k$ and the fixed point of the ADMM algorithm. Since the fixed point depends on θ, we will note it $\hat{\mathbf{x}}^*_\theta$. Thus, we define $g(\hat{\mathbf{x}}, \theta)$ as the weighted squared ℓ_2-norm between any image $\hat{\mathbf{x}}$ and the fixed point $\hat{\mathbf{x}}^*_\theta$ of the ADMM:

$$g(\hat{\mathbf{x}}, \theta) = \frac{1}{2\lambda}\|\hat{\mathbf{x}}^*_\theta - \hat{\mathbf{x}}\|_2^2, \tag{2.8}$$

with $\lambda > 0$. The proximal operator of g has thus a well-known closed form:

$$\text{prox}_{g(.,\theta)}(\hat{\mathbf{x}}) = \frac{\hat{\mathbf{x}}^*_\theta + \lambda\hat{\mathbf{x}}}{1+\lambda}. \tag{2.9}$$

From this definition, g is convex with respect to $\hat{\mathbf{x}}$ and one can easily verify that iterative applications of $\text{prox}_{g(.,\theta)}$ effectively converge towards the fixed point $\hat{\mathbf{x}}^*_\theta$, with a convergence rate controlled by λ.

2.4 Loss function

Unrolled optimizations aim to optimize θ in order to minimize the difference between the output $\hat{\mathbf{x}}^*_\theta$ and the groundtruth image $\mathbf{x}_{gt}$. In addition, we propose to fit $f(\hat{x}, \theta)$ to $\text{prox}_{g(.,\theta)}(\hat{\mathbf{x}})$ for any $\hat{x}$. We thus want to learn the parameters θ to approximate:

$$\begin{cases} \hat{\mathbf{x}}^*_\theta = \mathbf{x}_{gt}, & (2.10) \\ f(\hat{\mathbf{x}}, \theta) = \text{prox}_{g(.,\theta)}(\hat{\mathbf{x}}), \quad \forall \hat{\mathbf{x}}. & (2.11) \end{cases}$$

According to Eq. (2.9), this is equivalent to:

$$\begin{cases} \hat{\mathbf{x}}^*_\theta = \mathbf{x}_{gt}, & (2.12) \\ \hat{\mathbf{x}}^*_\theta = (1+\lambda) f(\hat{\mathbf{x}}, \theta) - \lambda\hat{\mathbf{x}}, \quad \forall \hat{\mathbf{x}} & (2.13) \end{cases}$$

$$\begin{cases} \hat{\mathbf{x}}^*_\theta = \mathbf{x}_{gt}, & (2.14) \\ \mathbf{x}_{gt} = (1+\lambda) f(\hat{\mathbf{x}}, \theta) - \lambda\hat{\mathbf{x}}, \quad \forall \hat{\mathbf{x}} & (2.15) \end{cases}$$

We can notice that Eq. (2.14) is equivalent to a particular case of Eq. (2.15) where $\hat{\mathbf{x}} = \hat{\mathbf{x}}^*_\theta$, since $f(\hat{\mathbf{x}}^*_\theta, \theta) = \hat{\mathbf{x}}^*_\theta$ by definition of the fixed point. The system can thus be expressed only with Eq. (2.15). The end-to-end optimization of the unrolled ADMM is

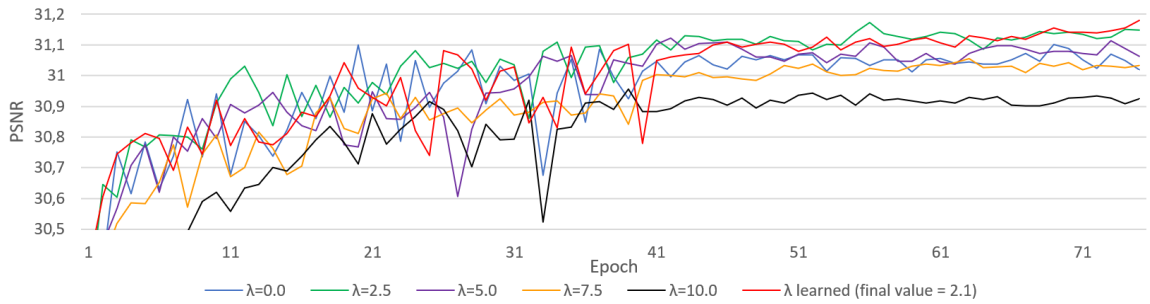


Figure 2.2 – Average validation PSNR per epoch during training with different values of λ in Eq. (2.9), for image super-resolution (bicubic x2 without antialiasing).

thus reduced to a set of independent optimization problems. However, optimizing θ for any possible images $\hat{\mathbf{x}}$ is impractical, since it would require integrating the loss over the space of images. Instead, we consider only the images in the optimization path, i.e., the intermediate estimates of the unrolled algorithm, i.e., $\hat{\mathbf{x}}^k, \forall k \in \{0, ..., K\}$. The full loss function $\mathcal{L}$ is then written as follows:

$$\mathcal{L}(\theta, \lambda, \hat{\mathbf{x}}^0, ..., \hat{\mathbf{x}}^K) = \frac{1}{K} \sum_{k=0}^{K} l(\theta, \lambda, \hat{\mathbf{x}}^k) \tag{2.16}$$

with:

$$l(\theta, \lambda, \hat{\mathbf{x}}^k) = \|(1+\lambda) f(\hat{\mathbf{x}}^k, \theta) - \lambda \hat{\mathbf{x}}^k - \mathbf{x}_{gt}\|_2^2. \tag{2.17}$$

Each term of the sum in Eq. (2.16) considers $\hat{\mathbf{x}}^k$ as an input. The computation of $\hat{\mathbf{x}}^k$ is then not taken into account for the backpropagation, which is essential to keep it shallow. This will be further discussed in Section 2.5.

The latter loss $\mathcal{L}$ is however dependent on the design of the convergence rate λ in Eq. (2.9). Its value needs to be optimized for both the estimation of the proximal mapping and the image restoration problems, i.e., such that it minimizes the loss l in Eq. (2.17) for any input $\hat{\mathbf{x}}^k$. We thus propose to learn λ along with the weights θ. to automatically find its best value. To illustrate the importance of correctly setting λ, we trained the SUPPA with different values of λ on the same task. As shown in Fig. 2.2, the value of λ drastically impacts the image reconstruction quality. Furthermore, learning it offers the best performances.

2.5 Stochastic unrolled iteration learning

Each term of the loss function $\mathcal{L}$ in Eq. (2.16) is associated with a specific independent iteration k, with $\hat{\mathbf{x}}^k$ considered as an input. Instead of considering all the K iterations at each training optimization step, we propose a stochastic selection of a small subset of iterations $\mathcal{I} \subset \{1, .., K\}$. The differentiation of the loss $\mathcal{L}$ w.r.t θ can then be written as follows:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{\text{card}(\mathcal{I})} \sum_{i \in \mathcal{I}} \frac{\partial f(\hat{\mathbf{x}}^i, \theta)}{\partial \theta}^T \frac{\partial l}{\partial f(\hat{\mathbf{x}}^i, \theta)}. \tag{2.18}$$

The differentiation of $\mathcal{L}$ w.r.t. λ can be simplified as:

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \frac{1}{\text{card}(\mathcal{I})} \sum_{i \in \mathcal{I}} \frac{\partial l}{\partial \lambda}. \tag{2.19}$$

The memory usage of the backpropagation is thus independent of the number of unrolled iterations, and is instead only dependent on the size of the subset of selected iterations $\mathcal{I}$. In order to favor high restoration quality, we always include the last computed iteration in the set $\mathcal{I}$. Hence, we propose to use 2 iterations ($\text{card}(\mathcal{I}) = 2$) per backpropagation: the last iteration K and a randomly selected iteration $k < K$. We have not observed any further benefit in increasing the number of randomly selected iterations. The SUPPA has thus the same memory usage advantage as the implicit unrolled methods. Fig. 2.3 gives an overview of the backpropagation used in the different unrolled methods. The explicit backpropagation propagates through every iteration. On the other hand, in the DEQ method, implicit backpropagation is performed in one step, considering the whole unrolled network for the inverse Jacobian matrix computation. By removing the inverse Jacobian, the JFBI method only requires backpropagation through the last iteration. This is similar to the proposed backpropagation scheme, but without the additional randomly selected intermediate iteration.

Algorithm 1 summarizes the different steps of the proposed algorithm. Our PyTorch implementation of the method is available at: `https://github.com/BrandonLeBon/SUPPA`

2.6 Experiments

We evaluate the performances of our method on super-resolution and deblurring inverse problems against state-of-the-art unrolled and task-specific deep methods. Addi-

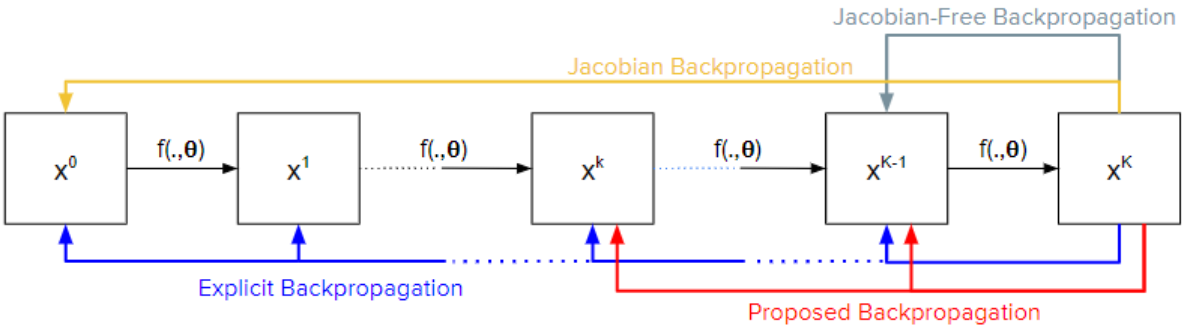


Figure 2.3 – Diagram of backpropagation strategies for the different unrolled methods.

Algorithm 1 Proposed Unrolled Proximal Point algorithm

initialize θ, K, λ, ϵ
for each epoch **do**
  for each batch **do**
    $\hat{\mathbf{x}}^0 \leftarrow$ input batch
    $\mathbf{x}_{gt} \leftarrow$ groundtruth batch
    $\mathcal{I} \leftarrow$ random subset of $\{1, ..., K\}$
    $loss \leftarrow 0$
    $k \leftarrow 1$
    $\hat{\mathbf{x}}^k \leftarrow f(\hat{\mathbf{x}}^0, \theta)$
    while $k < K$ and $\|\hat{\mathbf{x}}^{k-1} - \hat{\mathbf{x}}^k\| > \epsilon$ **do**
      if $k \in \mathcal{I}$ **then**
        update $loss$ with (2.16)
      $\hat{\mathbf{x}}^{k+1} \leftarrow f(\hat{\mathbf{x}}^k, \theta)$
      $k \leftarrow k + 1$
    update θ and λ with (2.18) and (2.19)

tional visual results are reported on the project web page (`http://clim.inria.fr/DeepCIM/SUPPA/index.html`).

Unrolled optimization methods: the selected reference unrolled optimization methods are (i) the explicit unrolled optimization with deep prior [80] (ii) the Deep Equilibrium architectures for inverse problems (DEQ) [86] (iii) an adaptation of the DEQ using the Jacobian-Free differentiation [89], named Jacobian-Free Backpropagation Implicit Unrolled (JFBI). Network parameters are shared between all unrolled iterations and are pre-trained in order to initialize the proximal operator of the regularizer (1.22) for Gaussian noise removal as presented in [75] with the DRUnet denoising architecture. The

explicit unrolled ADMM uses 6 iterations. As stopping criteria for the DEQ, the JFBI, and the SUPPA, we use a maximum of 50 iterations, with an idempotence estimated with a precision of $\epsilon = 10^{-3}$ as in [86].

State-of-the-art task-specific deep methods: the efficient task-specific deep methods considered are (i) the RCAN method [96] and MoG-DUN method [97] for the super-resolution (ii) the method of Dong et al. [18] for deblurring.

Training parameters: all the above methods have been retrained. We used 75 epochs, a batch size of 16, and a learning rate of 10^{-5}. We also used an additional learning rate of 10^{-1} for the convergence rate λ in the proposed method.

Super-resolution: we consider three scales: x2, x3, and x4. The low resolution images are generated with a bicubic downsampling without anti-aliasing. As initialization, we perform a bicubic interpolation on the downsampled image. As in [97], we used the DIV2K dataset [98] [99] with random patches of size 48x48 for training, and Set5 [22], Set14 [100] and BSDS100 [101] datasets for testing. We use the closed-form presented in [75] to solve Eq. (1.21).

Deblurring: we consider a Gaussian blur kernel with parameters $\sigma = 2$, $size = 2 * \sigma$ and a 1% noise level. As in [18], we randomly cropped 256x256 patches from the Waterloo Exploration dataset [102] for training, and we used the BSDS500 dataset [101] for testing. The exact solution $\hat{x}$ in Eq. (1.21) was computed using the same closed-form as for the super-resolution with a scaling factor equal to 1.

2.6.1 Reconstruction performances

Reconstruction performances are evaluated with the mean PSNR per dataset on the RGB channels. Note that the PSNR values for RCAN and MoG-DUN are significantly different from the original papers, since these are computed on the Y channel of the YCrCb color space. Super-resolution and deblurring results are respectively in Table 2.2, and 2.1. As shown in both tables, our method performs as well as the best unrolled methods, and outperforms the task-specific deep methods for the two tested inverse problems. Visual reconstructions are presented in figures 2.4 and 2.5, which show that the proposed SUPPA allows the recovery of fine details.

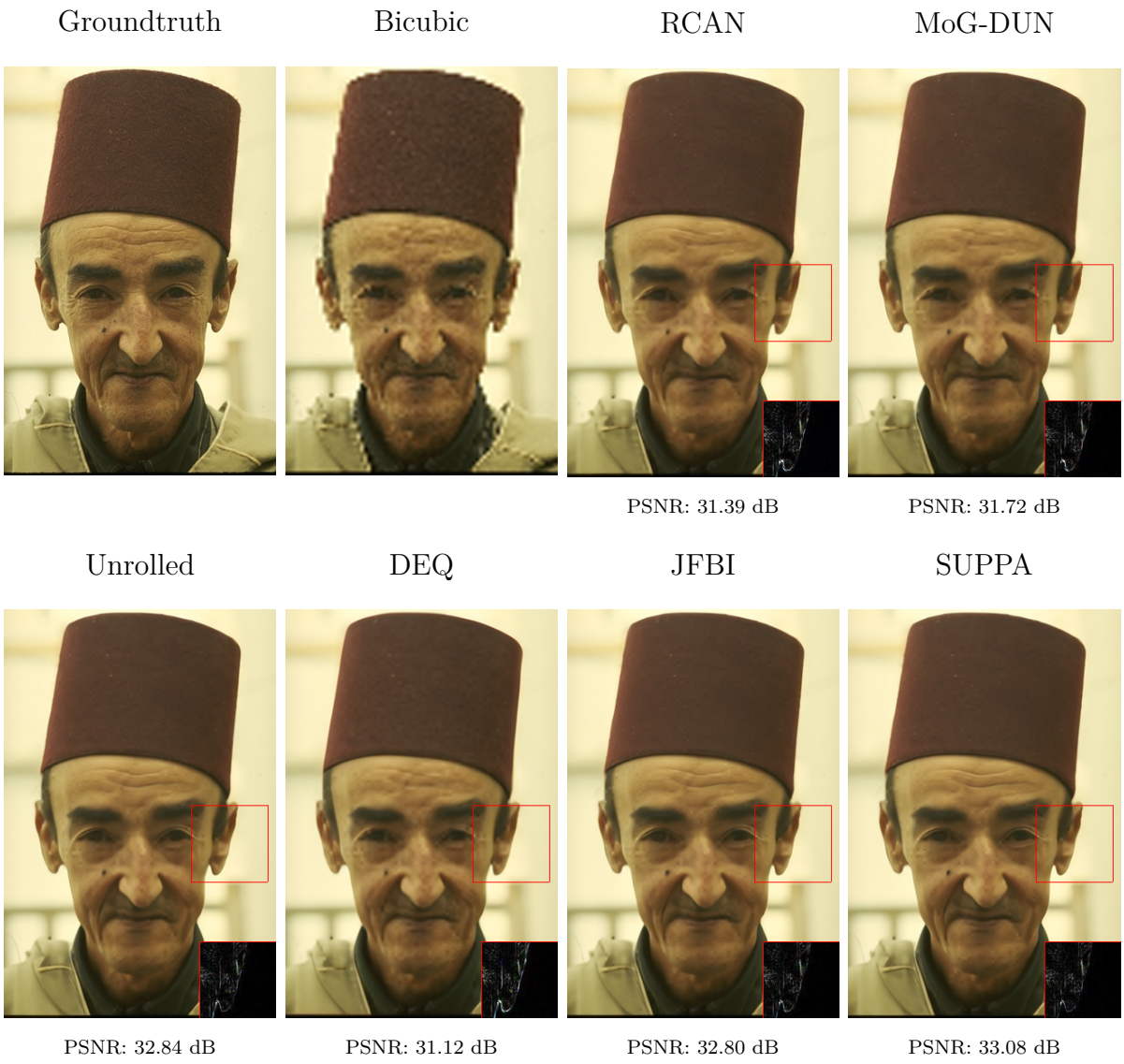


Figure 2.4 – Visual comparisons of image super-resolution of different methods with a bicubic downsampling (x4) without anti-aliasing and with a bicubic interpolation initialization on a sample image from the BSDS100 dataset [101]. A portion of the error map is selected to highlight the error in a specific area. The error is amplified by a factor of 3.

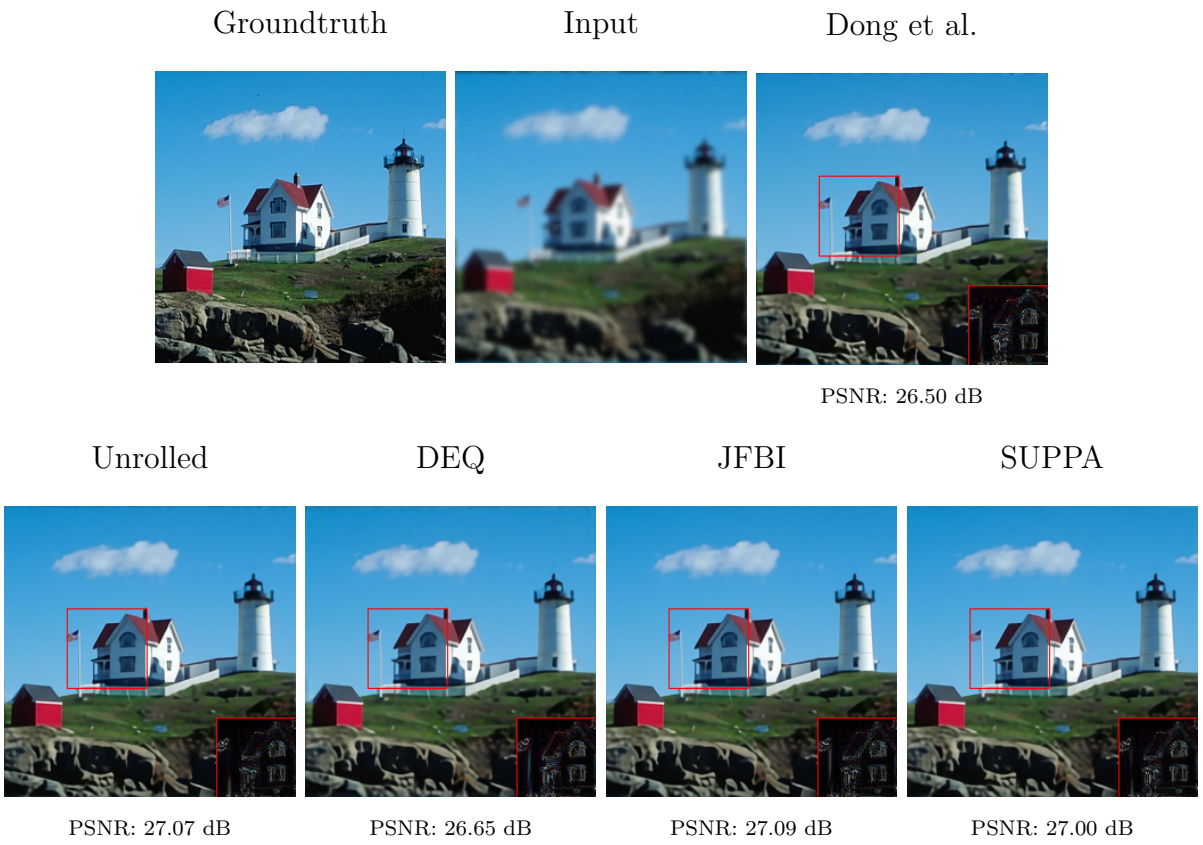


Figure 2.5 – Image deblurring results with different methods, using a Gaussian kernel ($\sigma = 2$, $size = 2 * \sigma$) and an image of the BSDS500 dataset. [101]. A portion of the error map is selected to highlight the error in a specific area. The error is amplified by a factor of 3.

Table 2.1 – Deblurring results (average PSNR)

	BSDS500
Dong et al. [18]	27.28 dB
Unrolled [80]	27.68 dB
DEQ [86]	27.25 dB
JFBI [89]	27.55 dB
SUPPA	27.60 dB

Table 2.2 – Superresolution results (average PSNR)

	Scale	Set5	Set14	BSDS100
Zhang et al. [96]	x2	34.30 dB	30.13 dB	28.81 dB
Ning et al. [97]	x2	34.59 dB	30.35 dB	30.02 dB
Unrolled [80]	x2	34.87 dB	30.42 dB	30.04 dB
DEQ [86]	x2	34.75 dB	30.28 dB	29.93 dB
JFBI [89]	x2	34.96 dB	30.61 dB	30.09 dB
SUPPA	x2	34.92 dB	30.54 dB	30.08 dB
Zhang et al. [96]	x3	29.60 dB	25.67 dB	25.91 dB
Ning et al. [97]	x3	29.63 dB	25.65 dB	25.93 dB
Unrolled [80]	x3	30.06 dB	26.03 dB	26.33 dB
DEQ [86]	x3	29.94 dB	25.95 dB	26.27 dB
JFBI [89]	x3	29.97 dB	25.90 dB	26.31 dB
SUPPA	x3	29.95 dB	25.90 dB	26.30 dB
Zhang et al. [96]	x4	27.50 dB	23.80 dB	24.67 dB
Ning et al. [97]	x4	27.77 dB	24.28 dB	24.79 dB
Unrolled [80]	x4	28.27 dB	24.63 dB	25.21 dB
DEQ [86]	x4	27.21 dB	24.17 dB	24.44 dB
JFBI [89]	x4	28.34 dB	24.63 dB	25.16 dB
SUPPA	x4	28.20 dB	24.63 dB	25.12 dB

2.6.2 Convergence of the unrolled methods

In this section, we study the convergence of the different considered unrolled optimization methods. Fig. 2.6 illustrates the convergence of the unrolled ADMM algorithms for the deblurring and super-resolution tasks.

First, we can notice that, as expected, the explicit unrolled method does not converge, since it is optimized for a specific number of iterations. Both the implicit backpropagation methods and SUPPA tend to converge, which is an expected behavior of the ADMM algorithm. Furthermore, we can notice that SUPPA converges faster than the JFBI and DEQ methods. A possible explanation for this behavior is that in SUPPA, we explicitly introduce a convergence rate parameter λ allowing control over the convergence speed, while JFBI and DEQ tend to use the maximum number of iterations allowed in the training (50 in our experiments).

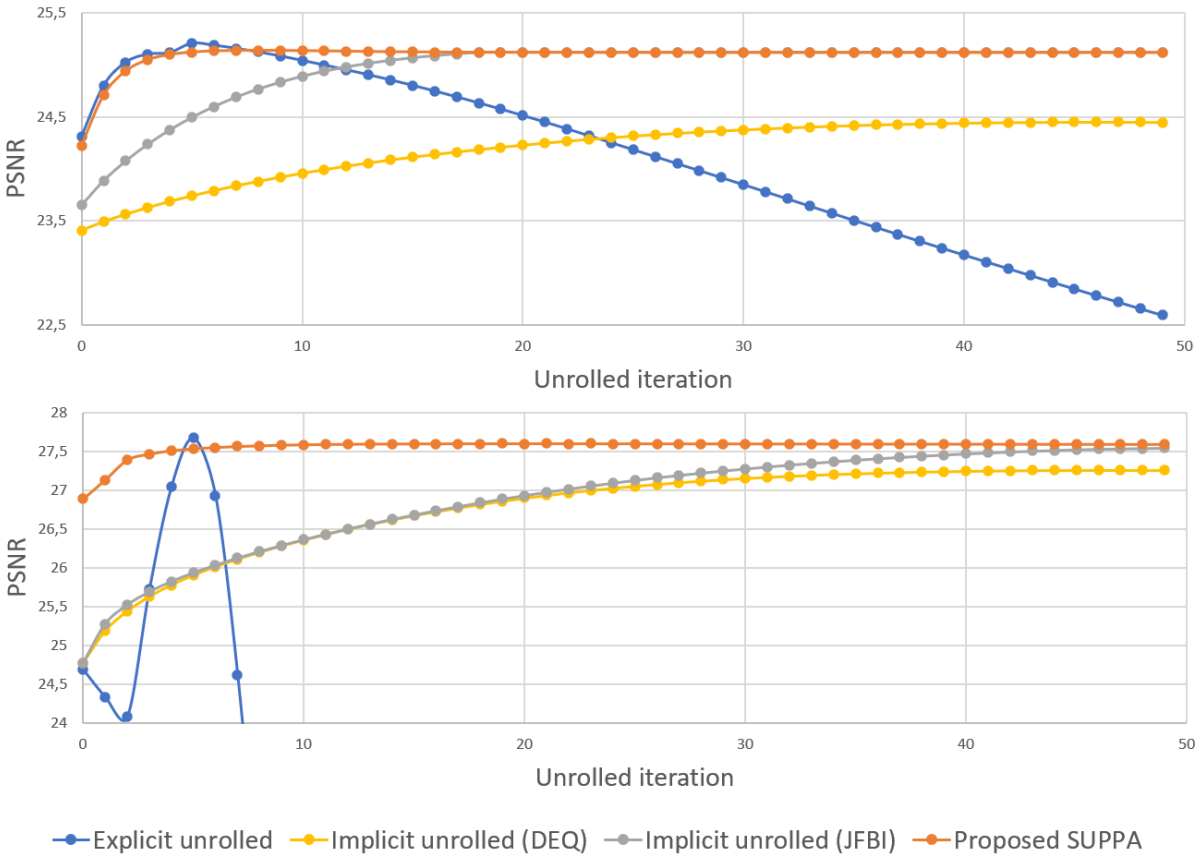


Figure 2.6 – Average PSNR per unrolled iteration for the different unrolled optimization algorithms considered, and for (top) a super-resolution image inverse problem, with a magnification factor of 4, on the BSDS100 dataset [101], (bottom) and a Gaussian deblurring, with a kernel of parameters $\sigma = 2$, $size = 2 * \sigma$ and a noise level of 1%, on the BSDS500 dataset [101].

2.7 Conclusion

In this paper, we addressed the problem of reducing the computational burden of training unrolled optimization algorithms. We proposed a novel training approach to train a deep neural network in an unrolled iterative optimization for solving image restoration problems. Based on the ADMM optimization algorithm, the proposed method re-defines the end-to-end training of the deep neural network as a per-iteration optimization strategy. Along with a stochastic optimization process, this strategy permits to significantly reduce the computational burden of training unrolled optimization algorithms, in terms of both memory usage and computation time, and allows the training to be applicable for any number of unrolled iterations. Furthermore, the image reconstruction quality is on par

with the state-of-the-art for the tested applications (i.e., super-resolution and deblurring), considering both other recent unrolled and task-specific deep methods.

Part II

Light field acquisition and reconstruction

Chapter 3

Light field imaging

3.1 Light field representation

3.1.1 Plenoptic function

Introduced by Gershun [103], in 1939, the 'Light Field' is a representation of 3D scenes, defined as the collection of all light rays passing through every point in the scene and flowing in every direction. A light field is generally described with the 7D plenoptic function ϕ introduced by Adelson et al. [104], which maps the radiance $\mathcal{R}$ of every light ray, at every 3D position (x, y, z) in the scene, in every direction represented by the polar coordinates (θ, ϕ), for every wavelength λ and at any time t:

$$\mathcal{R} = \phi(x, y, z, \theta, \phi, \lambda, t). \tag{3.1}$$

However, capturing such a representation of 3D scenes is a very difficult task. For the sake of simplicity, the light field is usually considered time-invariant and monochrome, leading to the 5D plenoptic function $\phi(x, y, z, \theta, \phi)$.

3.1.2 Lumigraph and two-plane parameterization

Gortler et al. [105] proposed a simplified light field representation, called the 'Lumigraph', assuming that the light rays travel through transparent air, i.e. that the radiance along the rays remains constant. The idea behind the Lumigraph is to capture the radiance of every light ray at the surface of a cube, which encloses the region of interest in the scene. The radiance is determined by tracing back along the light ray through an empty space down to the surface of the cube. A light ray is thus described by its intersection with two parallel planes, defined as the 'Two-plane parameterization' [105], [106], illustrated in Figure 3.1. The light ray intersects the two planes at coordinates (u_i, v_i) and (x_i, y_i), which denote respectively the angular (or view) coordinates on the angular plane and the

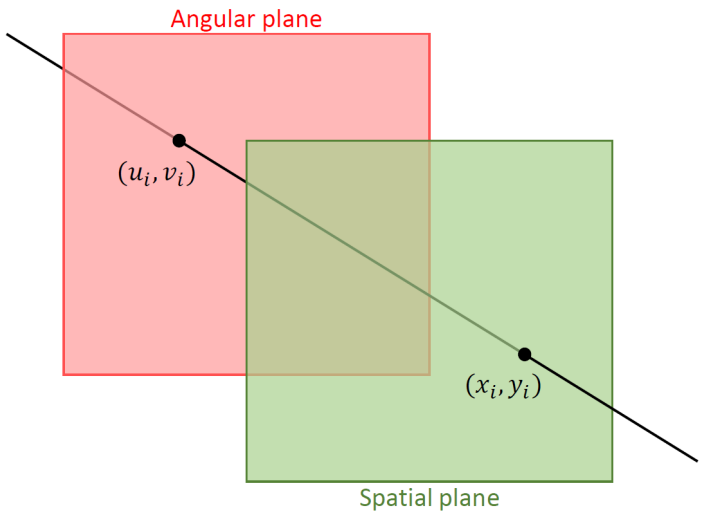


Figure 3.1 – Two-plane parameterization of a light field: a light ray is described by its intersection with the angular plane and the spatial plane.

spatial (or pixel) coordinates on the spatial plane. The light field L is thus a 4D function which maps the radiance $\mathcal{R}$ of every light ray parameterized with two spatial coordinates (x, y) and two angular coordinates (u, v):

$$\mathcal{R} = L(u, v, x, y). \tag{3.2}$$

Assuming this representation, the light field is generally visualized as a set of adjacent viewpoints, as illustrated in Figure 3.2. Each viewpoint of coordinates (u_i, v_i) corresponds to the image formed on the spatial plane with all the light rays intersecting the angular plane at coordinates (u_i, v_i).

3.2 Light field acquisition

In a conventional camera, each sensor element sums all the light rays emitted by one point over the lens aperture. Formally, an image $I(x, y)$ is obtained by integrating the light field over the angular dimensions with an aperture ψ:

$$I(x, y) = \int_{\mathbb{R}} \int_{\mathbb{R}} L(x, y, u, v)\psi(u, v)dudv. \tag{3.3}$$

Therefore, the angular information is mixed up on the sensor, making it hard to recover

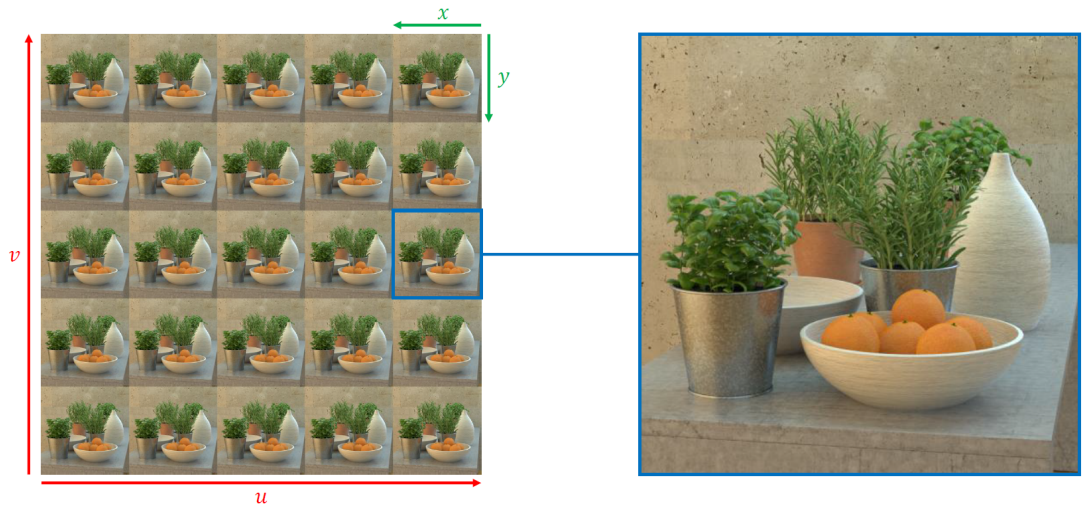


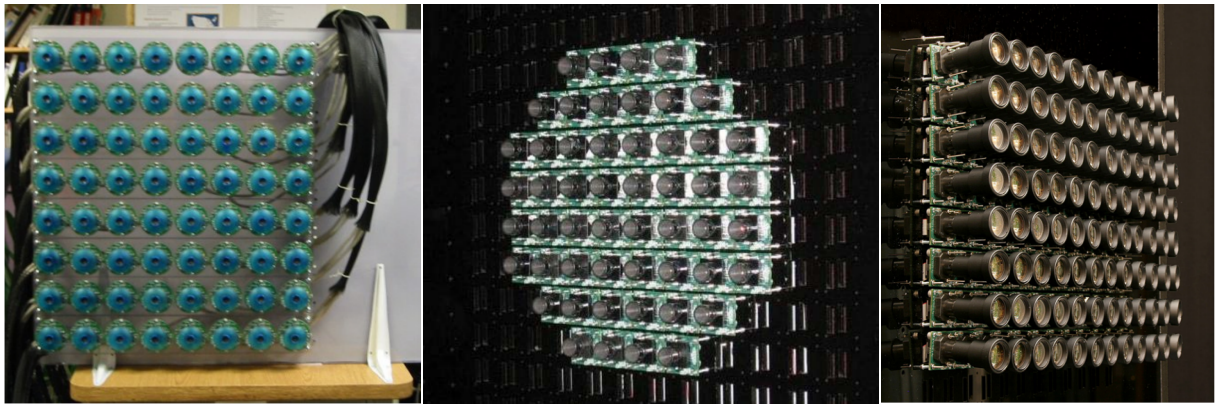
Figure 3.2 – Visualization of light fields: a set of adjacent views.

from the 2D measured image $I(x, y)$. In contrast, light field camera architectures aim at capturing both the light ray intensities and directions. Existing light field acquisition approaches use camera architectures that can be classified into three categories: camera arrays, cameras with additional hardware elements, and conventional cameras capturing a focal stack.

3.2.1 Camera arrays

The principle of camera arrays is to capture the different viewpoints of the light field using multiple cameras placed at different locations on the same plane. The idea emerged from the prior work by Lippmann [107], in 1908, where an imaging device composed of an array of 12 lenses on a photosensitive plate is proposed to capture different viewpoints of the same scene. The first camera array designed for light field imaging, shown in Figure 3.3, was by Yang et al. [108] in 2002, composed of 64 cameras distributed on an 8×8 grid. Several camera arrays were then proposed, e.g. the dense camera array composed of 53 CMOS sensors by Wilburn et al. [109], or the Stanford multi-camera array by Wilburd et al. [1] with 100 VGA video cameras.

A camera array directly captures the 4D Lumigraph. Indeed, the location of each camera corresponds to the angular coordinates (u_i, v_i) and the position of each sensor element in a camera is associated with the spatial coordinates (x_i, y_i). As a result, the angular and spatial dimensions of the captured light field are respectively dependent on



(a) (b) (c)

Figure 3.3 – Camera arrays designed for light field imaging: (a) the camera array proposed by Yang et al. [108], (b) the dense camera array by Wilburn et al. [109], (c) the stanford multi-camera array by [1].

the number of cameras and their resolution. Assuming a sufficient number of cameras with high resolution, a camera array is thus capable of efficiently capturing a light field. However, the construction of a camera array is a challenging task. In addition to being expensive, it requires a perfect alignment of the cameras to accurately measure the light field. In practice, misalignment of the cameras usually occur, involving additional camera calibration problems and image rectifications.

Much cheaper and easier approaches to directly capture different viewpoints of a 3D scene were also considered. One way is to place a conventional camera on a moving gantry [106]. The principle is to capture multiple images of a scene sequentially by moving the camera along a 2D plane. Consequently, this type of device is only limited to the capture of static light fields. Smaller camera arrays for portable devices have also been designed, e.g. the pelican camera array proposed by Venkataraman et al. [110] or the wafer-level-optics camera array by Huang et al. [111], in order to reduce the cost of camera arrays, however, at the cost of reducing the light field resolution.

3.2.2 Cameras with additional hardware elements

Another way to capture light fields is to use a single camera designed for light field imaging, thanks to specific additional hardware elements. There are mainly two types of light field cameras: plenoptic cameras and coded mask-based cameras.

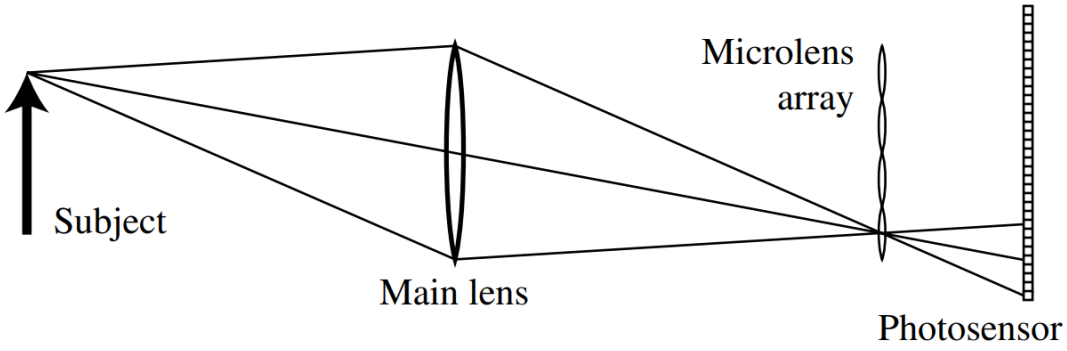


Figure 3.4 – Optical system of the plenoptic camera by Ng et al. [113]: an array of microlenses redirects the incident light rays to form viewpoint images

Plenoptic cameras

Inspired by the 12-lens prototype by Lippmann [107], plenoptic cameras were introduced by Adelson et al. [112], in 1992. The presented camera architecture captures a light field with a single shot using a single main lens and an array of microlenses at the sensor plane. Each microlens redirects the incident light rays, according to their incident angle, to form a microlens image. Each pixel of a microlens image correponds to the same position in the scene, observed at different angles. This design was then further improved by Ng et al. [113], creating the first commercial plenoptic camera. This architecture laid the foundation for the well-known Lytro 1st generation camera and the Lytro illum camera. The optical system of the plenoptic cameras is described in Figure 3.4.

The viewpoints, also called sub-aperture images, are produced by gathering pixels with the same relative position, i.e. with the same incident angle, in the microlens images. One can note that the spatial resolution depends on the number of microlenses, while the angular resolution depends on the number of pixels underneath the microlenses. With the resolution of the sensor being a limiting factor, plenoptic cameras thus impose a spatial-angular trade-off, with dense angular sampling leading to sparse spatial sampling. Several super-resolution methods have been proposed to enlarge the resolution of the captured low-resolution light field. While some of these methods focus on enlarging the spatial resolution of the sub-aperture images [114]–[118], others focus on applying angular super-resolution, i.e. a view synthesis [119]–[121], or both at the same time [122]–[124].

Another type of lenslet cameras referred to as "plenoptic 2.0" or "focused plenotpic cameras" aims at finding a trade-off between angular and spatial resolution. This is for

instance the case of the Raytrix camera [2]. In this case, the camera is designed so that the image formed behind each lenslet is in focus. However depth estimation is required in order to compute the sub-aperture image from the RAW data.

Coded mask cameras

A novel approach, named Coded Aperture Light Field Imaging (CALFI), was introduced by Babacan et al. [125], in 2009. This approach relies on the compressive sensing theory [126]–[128], which states that a sparse signal can be recovered with a sampling rate significantly lower than the sampling rate specified by the Shannon-Nyquist theorem, assuming incoherent measurements. It is a common assumption that natural visual data, such as light fields, have sparse representations. In this framework, a randomly coded attenuation mask is placed in a conventional camera between the aperture plane and the sensor, as illustrated in Figure 3.5. To ensure incoherence in the measurements, the nonzero elements of the coded mask are usually drawn following a specific probability distribution, e.g. a nonzero Gaussian distribution. Formally, using a monochrome-coded attenuation mask ϕ and an aperture ψ respectively placed a distance d_m and d_a from the sensor, the observed image $I(x, y)$ is obtained following the equation:

$$I(x,y) = \int_{\mathbb{R}} \int_{\mathbb{R}} L(x,y,u,v)\phi(x+\sigma(u-x), y+\sigma(v-y))\psi(u,v)dudv, \qquad (3.4)$$
$$\text{with:} \quad \sigma = \frac{d_m}{d_a}.$$

Consequently, the captured image is a 2D-coded projection of the 4D light field. The light field is then reconstructed by solving the compressive sensing inverse problem of recovering the light field $L(x, y, u, v)$ from its coded projection $I(x, y)$. Babacan et al. [125] used a randomly generated mask at the aperture plane and proposed a Bayesian framework to recover the light field. Marwah et al. [129] used a monochrome-coded attuenuation mask, while a color-coded mask is considered by Miandji et al. [130]. A multi-mask camera model is also considered by Nguyen et al. [131]. For all these methods, sparse priors were considered to reconstruct the light field with a dictionary of basis functions, e.g. sinusoids, monomials, or wavelets. Several works also considered using deep learning priors. Light field reconstruction methods with deep prior from monochrome measurements were proposed by Gupta et al. [132] Vadatya et al. [133] and Nabati et al.[134]. Le Guludec et

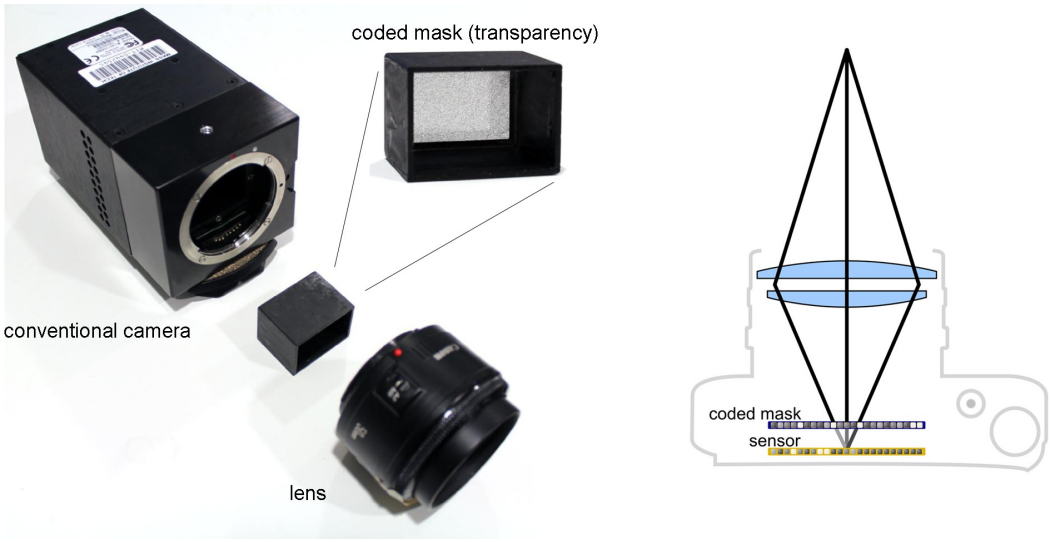


Figure 3.5 – Optical system of the coded mask camera presented in [3]: a coded attenuation mask is placed between the aperture plane and the sensor. The light field is then reconstructed by solving the compressive sensing inverse problem of restoring the light field from the 2D-coded projections.

al. and Guo et al. [135] proposed methods to jointly learn the coded mask along with a deep prior to ensure maximal incoherence in the measurements. An unrolled optimization algorithm was also considered by [85].

The advantage of this approach compared to the system of plenoptic cameras is that it does not come with a spatial-angular trade-off, hence allowing the capture of the light field with both high spatial resolution and high angular resolution.

3.2.3 Conventional cameras capturing a focal stack

All the camera architectures presented in Sections 3.2.1 and 3.2.2 are specific to light field imaging and require either multiple cameras or additional optical elements, hence being not accessible to the general public. Recent works were focused on capturing light fields with a single conventional camera, mainly using a focal stack as measurement of the light field, i.e. a set of images of the scene captured at different focus distances as illustrated in Figure 3.6. Assuming an aperture ψ, a focus image $I^s_{u_0,v_0}(x,y)$ at position (u_0,v_0) on the camera plane is obtained with the following equation:

$$I^s_{u_0,v_0}(x,y) = \int_{\mathbb{R}}\int_{\mathbb{R}} L(x-us, y-vs, u_0+u, v_0+v)\psi(u,v)dudv, \tag{3.5}$$

(a) (b) (c)

Figure 3.6 – Focal stack from [154]: a set of images of the scene captured at different focus distances.

where s is a focus parameter, such that the regions of the scene being at disparity $d = s$ are in focus. With a dense focal stack being 2D projections of the 4D light field [136], [137] that contains rich 3D information, a light field can be efficiently reconstructed. In the literature, there are mainly two types of reconstruction methods: depth-based methods [138]–[142] and deconvolution methods [137], [143]–[153]. In addition to being captured with a conventional camera, a focal stack allows to reconstruct a light field at full sensor resolution. It is important to notice that the current works require a dense focal stack to have sufficient 3D information of the scene. Capturing many focal stack images comes, however, at the cost of losing the instant capture properties. On one hand, the scene needs to remain static since the different shots are processed sequentially. On the other hand, the acquisition device has to stay stable to avoid camera translations between each shot.

Depth-based methods

The different viewpoints of the light field can be retrieved using an estimated depth map. Therefore, several methods in the literature estimate a depth map from the acquired focal stack in order to reconstruct the light field. Mousnier et al. [142] proposed a masked back-projection from an estimated depth map to perform the tomographic reconstruction of epipolar images used to reconstruct the light field. McMillan et al. [139] presented the "Plenoptic Modeling", an image-based rendering system to reconstruct the plenoptic function. Deep learning techniques were also recently considered by Huang et al. [144], with a three sequential convolutional neural networks framework that reconstructs the light field from estimated all-in-focus images, depth maps, and Lambertian light fields.

Deconvolution methods

The decovonlution inverse problem of reconstructing the light field from a set of focal stack images is generally tackled using iterative methods, with or without image priors. Takahashi et al. [147] proposed an iterative method to construct a light field representation named "tensor-display", a few light-attenuating layers. Inspired by the CT image reconstruction tasks, Liu et al. [146] applied the filtered back-projection and the Landweber iterative methods. Other filter-based iterative methods were proposed by Yin et al. [145] and Gao et al. [148]. Additionally to sparsity priors, Blocker et al. [151] and Kamal et al. [150] proposed a low-rank prior to respectively model (i) the low angular variation of light fields (ii) the redundancies of high-dimensional visual signal. Gao et al. [153] proposed the ADMM algorithm with a TV-regularization along with a guided filter. Le Pendu et al. [152] proposed the Fourier Disparity Layers (FDL) representation of light fields to decompose the scene into a set of additive layers. The author used a Tikhonov regularization in the optimization of the FDL.

3.3 Examples of light field applications

3.3.1 Digital refocusing

A major application of light field imaging is post-capture image refocusing. In conventional 2D imaging systems, changing the object in focus in a scene is handled by changing the distance between the sensor and the aperture before the capture. Light field imaging offers the possibility to compute the integration of the light rays with different focus planes and apertures after the capture. Assuming a dense set of captured views V, a refocused image $I^s_{u_0,v_0}(x,y)$, at position (u_0,v_0) on the camera plane, can be obtained with the shift-and-sum method introduced by Ng et al. [113], using an aperture ψ and a refocus parameter s:

$$I^s_{u_0,v_0}(x) = \sum_{(u,v)\in V} L(x-us, y-vs, u_0+u, v_0+v)\psi(u,v) \tag{3.6}$$

3.3.2 Viewpoints switching

Capturing a light field allows to freely change the viewpoint from which a scene is observed, or to simulate the displacement of a moving camera. Since most of the light

field capture systems capture a set of adjacent views, as illustrated in Figure 3.2, the number of available viewpoints is limited. In order to have access to an unlimited number of viewpoints or to allow smooth transitions between viewpoints when simulating a camera movement, many view synthesis frameworks have been developed to generate novel viewpoints from a light field captured with a sparse angular resolution [119]–[121]. Another approach is to use a compact representation of light field, e.g. the "tensor-display" [147], Fourier Disparity Layers [152] and Neural Radiance Fields [155], from which any viewpoint can be reconstructed.

3.3.3 Geometry and depth estimation

Depth estimation is one of the most well-studied computer vision tasks [156]. Many methods have been developed to estimate the depth in a scene using one monocular image [157]–[161] or stereo pairs [156], [162]–[167]. However, estimating the depth and the geometry of the scene using a maximum of two viewpoints usually leads to occlusion issues. Therefore, light field imaging is well-suited for depth and geometry estimation thanks to its high angular resolution. Light fields have thus been considered in several works to capture the depth in the scene, mainly recovered by three types of depth estimation approaches: epipolar plane image (EPI)-based methods [122], [168]–[171], pixel matching methods [172]–[174] and focus-based methods [175]–[177].

3.4 Directions and objectives

As stated in Section 3.2, light fields are generally captured using complex camera designs specific to light field imaging, which are not accessible to the general public. In this thesis, we aim to contribute to the domain by improving the capture of light fields with a single conventional camera, in order to expand light field imaging to the general public. The most promising approach to the acquisition of light fields with a single traditional camera is by capturing a focal stack and reconstructing the light field afterward.

Methods proposed in the literature for light field reconstruction from a focal stack generally require a focal stack with dense sampling in the depth dimension to retrieve all the 3D information from the scene. However, capturing such a focal stack is difficult, specifically due to possible movement in the scene or by the camera between the different shots. Therefore, the capture of a dense focal stack is thus limited to static scenes captured

with a camera equipped with a stabilizer.

We seek to address this problem by proposing a method to reconstruct light fields from a focal stack containing very few images captured at different focus distances. To retrieve all the missing information due to the sparse focal stack, a strong prior on light fields is required. Therefore, the unrolled algorithms are also considered in this work to automatically learn a complex prior on light fields.

Chapter 4

Light field reconstruction from few-shots focal stack

4.1 Introduction

As introduced in chapter 3, light field acquisition by capturing a focal stack, i.e. several images of the scene at different focus distances, is the most promising approach to capture a light field using a single conventional camera. There are mainly two types of methods in the literature to reconstruct a light field from a focal stack: depth-based methods [138]–[142] and deconvolution methods [137], [143]–[153]. However, all existing methods typically require focal stacks with dense sampling in the focus dimension, so that the details can be retrieved at every depth in the scene. Hence, many shots are needed in the capture process. As we mentioned in chapter 3, capturing many sequential shots of a scene restricts the capture to static scenes and is very sensitive to inconsistency in the measurements, e.g. slight camera translation between each shot.

In this chapter, we address the problem of light field reconstruction from a small set of focal stack images. The problem of reconstructing a light field from a focal stack with only a few shots can be seen as a form of compressive sensing, hence posed as an image inverse problem. As discussed in chapters 1 and 2, an efficient strategy to deal with ill-posed image inverse problems consists in learning an image prior as a regularization term in an unrolled iterative optimization algorithm. The optimization problem is then posed as the minimization of a function composed of two terms: a data-fidelity term and a learned regularization term. In the context of light field reconstruction from a set of focal stack images, several iterative optimization algorithms have been designed in the literature [145]–[148], [150]–[153]. However, they only rely on handcrafted priors.

We thus propose a novel unrolled optimization method to solve the inverse problem involved. The Alternating Direction Method of Multipliers (ADMM) [49] is unrolled with a deep prior to optimize Fourier Disparity Layers (FDL), a compact representation of

light fields, introduced by Le Pendu et al. [152], from which any view of the light field can be reconstructed. The problem is solved in the FDL domain, which allows us to derive a closed-form solution for the data-fidelity term of the cost function to be minimized. Unrolling the FDL optimization permits to learn the regularization function directly in the FDL domain. However, the FDL model is known to produce artifacts in occluded non-Lambertian scenes, such as transparency in occluded regions [152], [178]. To cope with this issue, we propose a Deep Convolutional Neural Network (DCNN) within the FDL view synthesis process that is trained to minimize the errors of the reconstructed views. Both the unrolled ADMM FDL and the view synthesis network are jointly optimized to ensure the best reconstruction performances. We show that this proposed framework outperforms recent and efficient state-of-the-art methods for light field reconstruction from a set of focal stack images, and that it significantly improves the FDL model in terms of reconstruction performances and robustness to occluded and non-Lambertian scenes.

4.2 Light field imaging and focal stack formation models

This Section introduces the light field imaging model and the focal stack formation model considered. Let us consider an input light field, represented by a 4D function $L(x, y, u, v)$ describing the radiance along light rays, with the two-plane parameterization [105], [106] presented in chapter 3. The parameters (u, v) denote the angular (view) coordinates and (x, y) the spatial (pixel) coordinates. In this chapter, for notation simplicity and without loss of generality, we consider a 2D light field $L(x, u)$ with one angular dimension and one spatial dimension.

Focal stack images taken at different focus distances can be seen as measurements of the light field to be reconstructed. Let a refocused light field L^s be defined as $L^s(x, u) = L(x - us, u)$, with a refocus parameter s defined such that the regions of disparity $d = s$ in the light field L have a disparity $d = 0$ in the refocused light field L^s. A refocused image $I^s_{u_0}$, at position u_0 on the camera plane, is obtained by integrating the light rays over the angular dimension using the refocused light field and the aperture ψ:

$$I^s_{u_0}(x) = \int_{\mathbb{R}} L(x - us, u_0 + u)\psi(u)du. \tag{4.1}$$

Assuming the light field imaging model in equation (4.1), the angular information of

the light field is lost, for the most part, in the acquisition process. Therefore, the problem of recovering the light field L from a set of focus images is an ill-posed inverse problem.

4.3 Joint Fourier disparity layers unrolling with learned view synthesis

In this section, we present our joint optimization framework, briefly illustrated in Figure 4.1. We first introduce in Section 4.3.1 the Fourier Disparity Layers (FDL) by Le Pendu et al. [152] that will be used in our framework. The proposed method is a joint optimization of two different parts introduced in Sections 4.3.2 and 4.3.3 (i) the parameters θ_1 of a denoiser CNN $\mathcal{D}$ used in an unrolled ADMM FDL optimization (ii) and the parameters θ_2 of a CNN $\mathcal{S}$ of a novel learned view synthesis process trained to adapt the optimized FDL for each novel view to be reconstructed, in order to cope with the issues of the FDL model. Finally, in Section 4.3.4, we present the joint optimization process.

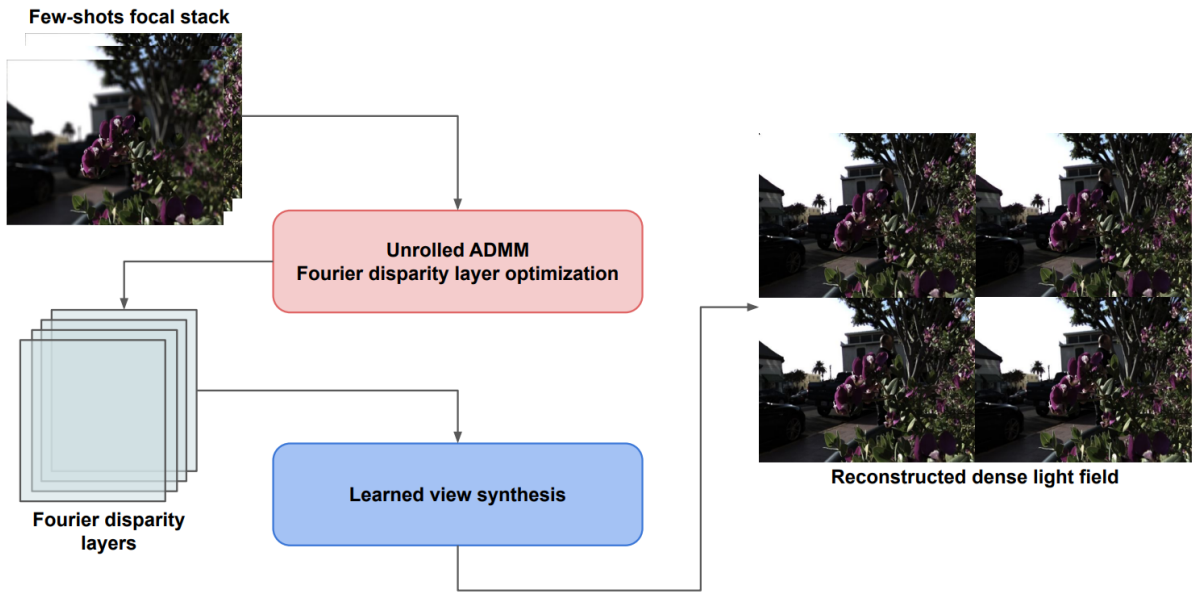
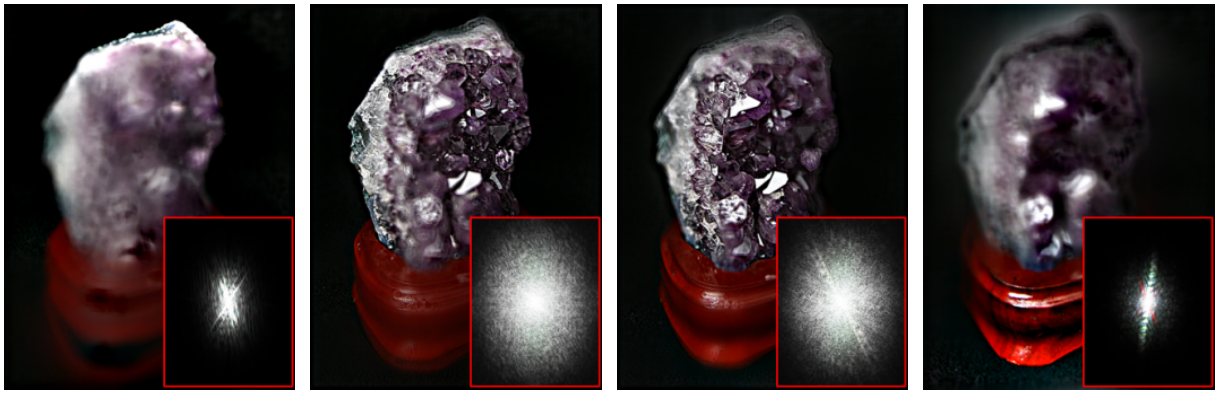


Figure 4.1 – Overview of the proposed joint optimization framework: (i) Fourier disparity layers are optimized from focal stack measurements through an unrolled optimization algorithm (ii) the light field viewpoints are reconstructed with a learned view synthesis process.



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

Figure 4.2 – Examples of the Fourier disparity layers [152] for one scene : each layer contains the information of the scene at a specific disparity.

4.3.1 Fourier Disparity Layers (FDL)

Fourier Disparity Layers (FDL) have been introduced in [152] as a compact representation of dense light fields. The FDL model consists of a set of additive layers $\tilde{L}^k$, each associated with a disparity value d^k, where each layer mostly contains details in the regions of disparity d^k in the scene, as illustrated in figure 4.2. The FDL model is defined such that a sub-aperture view at angular coordinate u_0 is reconstructed by shifting each layer L^k by $d_k u_0$, and by summing the shifted layers.

Formally, let a Lambertian non-occluded scene be divided into n spatial regions Ω_k with constant disparity d_k. The Fourier transform $\tilde{L}(\omega_x, \omega_u)$ of a light field $L(x, u)$ can thus be re-written such that the spatial information remains the same for any view:

$$\tilde{L}(\omega_x, \omega_u) = \sum_k \delta(\omega_u - d_k \omega_x) \tilde{L}^k(\omega_x), \tag{4.2}$$

with $\tilde{L}^k(\omega_x)$, the FDL associated with the disparity d_k, defined by:

$$\tilde{L}^k(\omega_x) = \int_{\Omega_k} e^{-2i\pi x \omega_x} L(x, 0) dx. \tag{4.3}$$

The relation between the Fourier transform $\tilde{I}^s_{u_0}(\omega_x)$ of a refocus image of a focal stack and the FDL is thus etablished in [152] as follows:

$$\tilde{I}^s_{u_0}(\omega_x) = \sum_k e^{+2i\pi u_0 d_k \omega_x} \tilde{\psi}(\omega_x(s - d_k)) \cdot \tilde{L}^k(\omega_x). \tag{4.4}$$

Based on equation (4.4), we can define an optimization algorithm to optimize the FDL from a set of refocused images. Our proposed unrolled optimization algorithm will be further detailed in Section 4.3.2. It is important to notice that equations (4.2), (4.3), and (4.4) are only verified in the case of Lambertian non-occluded scenes. Assuming this model, performing an FDL optimization algorithm will produce light field views with occlusion and reflectance artifacts, e.g. transparency in occluded areas, as illustrated in recent works [152], [178]. We address this problem in Section 4.3.3 by proposing a neural network-based view synthesis process to reconstruct light field views from the optimized FDL.

4.3.2 Unrolled ADMM FDL optimization

In this section, we introduce the proposed unrolled ADMM optimization algorithm used to optimize the FDL.

Let us consider an input focal stack containing images I_j. We note m and n respectively the number of measured focal stack images and the number of considered layers in the FDL model. For each spatial frequency component ω_q of index q in the discrete Fourier transform, we note $\mathbf{b_q} \in \mathbb{C}^m$ a vector with $[\mathbf{b_q}]_j = \tilde{I}_j(\omega_q)$, $\mathbf{x_q} \in \mathbb{C}^n$ a vector with $[\mathbf{x_q}]_k = \tilde{L}^k(\omega_q)$, and $\mathbf{A_q} \in \mathbb{C}^{m \times n}$ a matrix defined as follows:

$$[\mathbf{A_q}]_{j,k} = e^{+2i\pi u_j d_k \omega_x} \tilde{\psi}_j(\omega_x(s_j - d_k)). \tag{4.5}$$

Equation (4.4) is thus reformulated as $\mathbf{A_q x_q} = \mathbf{b_q}$. Therefore, the construction of the FDL spatial frequencies $\mathbf{x_q}$ from measurements $\mathbf{b_q}$ is posed as a linear least squares optimization problem independently for each frequency component ω_q. The matrices $\mathbf{A_q}$ are usually ill-conditioned, making the latter optimization problem ill-posed. To reduce overfitting on the measurements that may cause severe artifacts in the FDL, the authors in [152] include a Tikhonov regularization term, which results in the following per-frequency minimization problem:

$$\hat{\mathbf{x}}_\mathbf{q} = \arg\min_{\mathbf{x_q}} \|\mathbf{A_q x_q} - \mathbf{b_q}\|_2^2 + \lambda \|\mathbf{\Gamma_q x_q}\|_2^2, \tag{4.6}$$

with $\mathbf{\Gamma}$ being the Tikhonov matrix. A calibration method is also proposed in [152] to determine the angular coordinate u_0 of each input view and the disparity values d_k of the layers. However, it only applies in the case of sub-aperture images as measurements. Here,

we consider focal stacks where all the images are taken at the same angular coordinate $u_0 = 0$, assuming a known focus parameter s and aperture ψ. For the disparity values d_k of the FDL model, we use uniformly sampled values over the disparity range of the scene.

While the author in [152] uses a Tikhonov regularization to encourage smooth variations between the light field views generated by the optimized FDL, designing a more complex prior directly in the FDL domain is a challenging task. To cope with this issue, we propose to unroll the FDL optimization, following the ADMM unrolling framework, in order to automatically learn a deep prior in the FDL domain. In order to account for complex image statistics on the FDL model, we consider a regularization of the full layers, rather than a per-frequency regularization, as in equation (4.6). Furthermore, since most neural networks operate on images in the pixel domain, we regularize the images obtained by the inverse Fourier transform of the FDL layers.

Let us define the matrix $\mathbf{X} = [\mathbf{x_1}|...|\mathbf{x_Q}]$ representing the full FDL as a concatenation of the column vectors $\mathbf{x_q}$ for all the frequency components ω_q with $q \in [1..Q]$. The regularized FDL reconstruction problem is then formulated as:

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \left(\lambda \cdot \mathcal{R}(\mathbf{X}\Phi^{-1}) + \sum_q \|\mathbf{A_q}\mathbf{x_q} - \mathbf{b_q}\|_2^2 \right), \tag{4.7}$$

where Φ^{-1} is the inverse 2D Fourier transform, applied to each FDL layer (i.e. rows of $\mathbf{X}$) to regularize the images in the pixel domain. The steps of the ADMM iteration in equations (1.21), (1.22), (1.23), can then be written:

$$\hat{\mathbf{x}}_\mathbf{q}^{i+1} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A_q}\mathbf{x} - \mathbf{b_q}\|_2^2 + \frac{\rho}{2} \left\|\mathbf{x} - \mathbf{y}_\mathbf{q}^i + \mathbf{u}_\mathbf{q}^i\right\|_2^2, \tag{4.8}$$

$$\mathbf{Y}^{i+1} = \mathcal{D}((\hat{\mathbf{X}}^{i+1} + \mathbf{U}^i)\Phi^{-1}; \theta_1)\Phi, \tag{4.9}$$

$$\mathbf{U}^{i+1} = \mathbf{U}^i + (\hat{\mathbf{X}}^{i+1} - \mathbf{Y}^{i+1}), \tag{4.10}$$

where we note $\hat{\mathbf{X}}^i = [\hat{\mathbf{x}}_\mathbf{1}^i|...|\hat{\mathbf{x}}_\mathbf{Q}^i]$ and $\hat{\mathbf{Y}}^i = [\hat{\mathbf{y}}_\mathbf{1}^i|...|\hat{\mathbf{y}}_\mathbf{Q}^i]$. For the regularization, one can see in equation (4.9) that denoising can be applied in the pixel domain by performing the inverse 2D Fourier transform of the denoiser's input layers $(\hat{\mathbf{X}}^{i+1} + \mathbf{U}^i)$, and reapplying the 2D Fourier transform on the denoised output. Instead of using a pre-learned denoiser as in the Plug-and-Play approach [75], [179], the denoiser $\mathcal{D}$ is here trained end-to-end within the unrolled algorithm to better train it for the task of FDL denoising. On the other hand, the data-fidelity subproblem in equation (4.8) can still be solved independently per-frequency

component, and has a well-known closed form solution:

$$\hat{\mathbf{x}}_{\mathbf{q}} = (\mathbf{A}_{\mathbf{q}}^{*}\mathbf{A}_{\mathbf{q}} + \rho\mathbf{I})^{-1}(\mathbf{A}_{\mathbf{q}}^{*}\mathbf{b}_{\mathbf{q}} + \rho(\mathbf{y}_{\mathbf{q}}^{i} - \mathbf{u}_{\mathbf{q}}^{i})), \tag{4.11}$$

where $\mathbf{I}$ is the identity matrix and * is the Hermitian transpose operator. Note that for each frequency component of index q, the matrix inversion $(\mathbf{A}_{\mathbf{q}}^{*}\mathbf{A}_{\mathbf{q}} + \rho\mathbf{I})^{-1}$ in equation (4.11) can be performed efficiently thanks to the small dimensions of the matrix $\mathbf{A}_{\mathbf{q}}$ ($\mathbf{A}_{\mathbf{q}}^{*}\mathbf{A}_{\mathbf{q}} \in \mathbb{C}^{n\times n}$, with n the number of layers). The per-frequency computation of the proximal operator allowed by the FDL model thus significantly reduces the computational burden of computing the estimate $\hat{\mathbf{X}}$.

4.3.3 View synthesis network

In this section, we present our learned view synthesis process to synthesize the light field viewpoints from the optimized FDL.

As derived in [152], the Fourier transform $\tilde{L}_u(\omega_x)$ of a view $L(x, u)$ can be reconstructed by applying a shift-and-sum on the optimized k FDL $\tilde{L}^k(\omega_x)$ as follows:

$$\tilde{L}_u(\omega_x) = \sum_k e^{+2i\pi u d_k \omega_x} \tilde{L}^k(\omega_x). \tag{4.12}$$

However, it is well-known that artifacts will occur in specific areas with this technique, e.g. in occluded regions [152], [178], as mentioned in Section 4.3.1. Since these artifacts are different for each reconstructed light field view, we need to slightly adjust the optimized k FDL $\tilde{L}^k(\omega_x)$ for each novel view. Since the ground truth views $L_{gt}(x, u)$ are known during the training phase, we propose to train the parameters θ_2 of a CNN $\mathcal{S}$ to modify the k optimized FDL $\tilde{L}^k(\omega_x)$ for each view to be reconstructed, such that the reconstructed views well estimate their corresponding ground truth views.

As described in equation (4.3), the FDL contain only the spatial information of the light field within each depth plane. Therefore, we also need to add angular information to the input of the network $\mathcal{S}$ in order to specify which view to reconstruct. We propose to shift the optimized FDL accordingly to the angular coordinates of the view to be reconstructed, as in equation (4.12). However, instead of directly summing the shifted layers, we first concatenate them along with two additional channels $\mathbf{C}$, each containing an angular coordinate of the view. The resulting tensor is fed into a view synthesis CNN $\mathcal{S}$ which computes the modified shifted layers. These modified layers are then summed to

reconstruct the view, as in equation (4.12).

Formally, let u be the coordinates of the view to be reconstructed, $\hat{\mathbf{X}} \in \mathbb{C}^{n \times Q}$ be the matrix representing the concatenation of the optimized FDL as in equation (4.7), and $\mathbf{Z} \in \mathbb{C}^{n \times Q}$ be a matrix with $\mathbf{Z}_{k,q} = e^{+2i\pi u d_k \omega_q}$. The matrix $\hat{\mathbf{X}}_u \in \mathbb{C}^{n \times Q}$, being the concatenation of the shifted FDL associated to the angular coordinates u, is thus computed as follows:

$$\hat{\mathbf{X}}_u = \mathbf{Z} \odot \hat{\mathbf{X}}, \tag{4.13}$$

with $\odot$ being the Hadamard product. With $\mathbf{C}_u$ being a channel filled with the value of the angular coordinates u of the view to be reconstructed, the parameters θ_2 of the network $\mathcal{S}$ are thus optimized as follows:

$$\theta_2 = \arg\min_{\theta_2} \left\| \sum_k \left[\mathcal{S}(\hat{\mathbf{X}}_u \Phi^{-1}, \mathbf{C}_u; \theta_2) \right] - L_{gt}(x, u) \right\|_2^2, \tag{4.14}$$

where we compute the inverse Fourier transform of the shifted layers $\hat{\mathbf{X}}_u \Phi^{-1}$ so that the network $\mathcal{S}$ in the view synthesis process operates in the pixel domain, similarly to the denoising network in equation (4.9).

In practice, we observed that pre-training the framework using only the shifted FDL as the input of the network, and then fine-tuning by adding the coordinate channels to the input offers the best performances. One can notice that several approaches can be used to model the input of the network. We further discuss our choice in comparison with other approaches in Section 4.4.5.

4.3.4 Joint optimization

The proposed framework is composed of two successive optimizations, presented in the previous sections: the unrolled ADMM FDL optimization, with a network $\mathcal{D}$ parameterized with θ_1, described in Section 4.3.2, and the learned view synthesis process, with a network $\mathcal{S}$ parameterized with θ_2, described in Section 4.3.3. Instead of training both networks independently, we propose a joint optimization in an end-to-end framework. A joint optimization of θ_1, θ_2 ensures that both networks are optimized such that the synthesized views well estimate their corresponding ground truths.

Let $\mathcal{F}$ be a function parameterized with θ_1, which computes the application of the whole forward pass of the unrolled ADMM FDL optimization algorithm. The optimization problem of the entire end-to-end framework is:

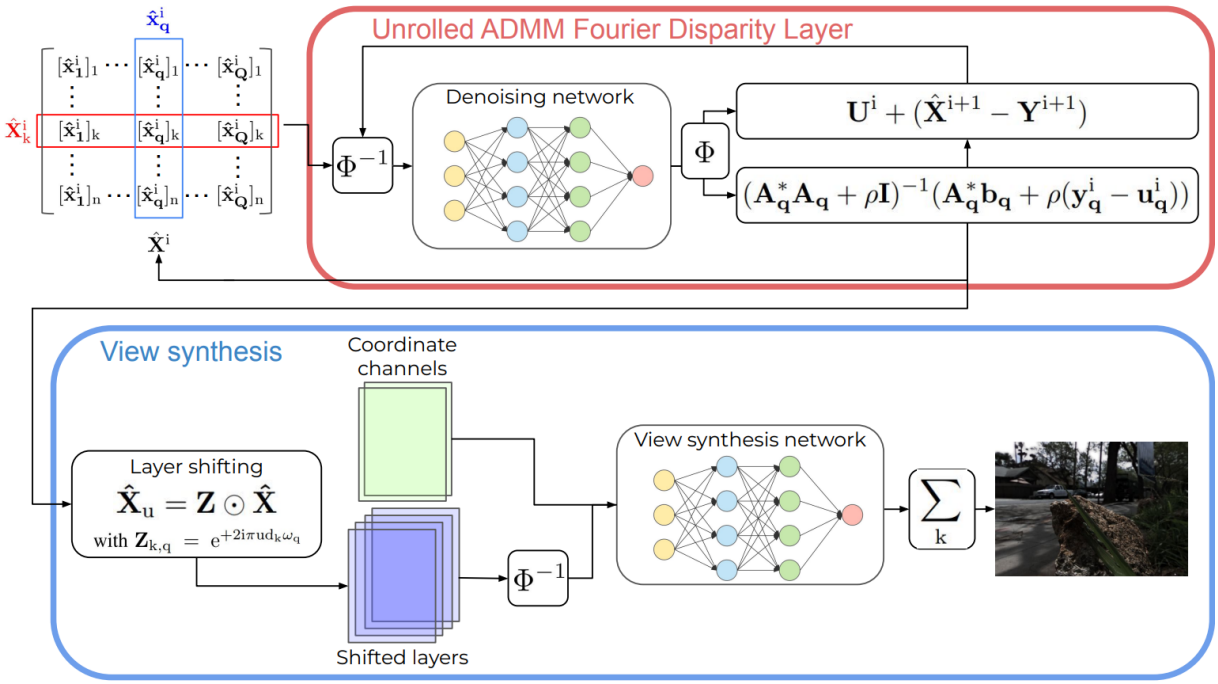


Figure 4.3 – Architecture of the proposed end-to-end framework for light field reconstruction from focal stack measurements. The pipeline is composed of two blocks: (i) an unrolled ADMM FDL optimization (red block) which optimizes a matrix $\hat{\mathbf{X}}^i$, where each row $\hat{\mathbf{X}}_k^i$ corresponds to the vectorized FDL k (ii) a view synthesis (blue block) with a learned network, where the optimized FDL are shifted and concatenated with additional coordinate channels to indicate which view to reconstruct to the network.

$$\begin{aligned} \theta_1, \theta_2 &= \arg\min_{\theta_1,\theta_2} \left\| \sum_k \left[\mathcal{S}(\hat{\mathbf{X}}_u \Phi^{-1}, \mathbf{C}_u; \theta_2) \right] - L_{gt}(x,u) \right\|_2^2, \\ \text{with} \quad & \hat{\mathbf{X}}_u = \mathbf{Z} \odot \hat{\mathbf{X}}, \\ \text{and} \quad & \hat{\mathbf{X}} = \mathcal{F}(\mathbf{b}; \theta_1), \end{aligned} \tag{4.15}$$

where $\mathbf{b}$ is a vector containing the measured focal stack images. The joint optimization algorithm is described in Algorithm 2. The proposed framework is depicted in Fig. 4.3.

4.4 Experiments

We assess our framework for light field reconstruction from focal stacks containing very few shots, i.e. with 2 and 3 shots. We compare the proposed method against the most

Algorithm 2 : Proposed joint optimization

initialize θ_1,θ_2
for each training iteration **do**
    $\hat{\mathbf{X}}^0, \mathbf{Y}^0, \mathbf{U}^0 \leftarrow 0$
    $\mathbf{b} \leftarrow$ input measurements
    $L\_gt \leftarrow$ groundtruth views
    $L\_recons \leftarrow 0$

    for each unrolled iteration i **do**
        for each frequency component $\mathbf{q}$ **do**
            $\hat{\mathbf{x}}_\mathbf{q} \leftarrow (\mathbf{A}_\mathbf{q}^* \mathbf{A}_\mathbf{q} + \rho \mathbf{I})^{-1} (\mathbf{A}_\mathbf{q}^* \mathbf{b}_\mathbf{q} + \rho(\mathbf{y}_\mathbf{q}^i - \mathbf{u}_\mathbf{q}^i))$
        $\mathbf{Y}^{i+1} \leftarrow \mathcal{D}((\hat{\mathbf{X}}^{i+1} + \mathbf{U}^i)\Phi^{-1}; \theta_1)\Phi$
        $\mathbf{U}^{i+1} \leftarrow \mathbf{U}^i + (\hat{\mathbf{X}}^{i+1} - \mathbf{Y}^{i+1})$

    for each view coordinate u **do**
        $\hat{\mathbf{X}}_u^I = \mathbf{Z} \odot \hat{\mathbf{X}}^\mathbf{I}$
        $L\_recons_u \leftarrow \sum_k \left[\mathcal{S}(\hat{\mathbf{X}}_u^I \Phi^{-1}, \mathbf{C}_u; \theta_2)\right]$

    $loss = \|L\_recons - L\_gt\|_2^2$
    $\theta_1 = \theta_1 - \lambda \nabla_{\theta_1}(loss)$
    $\theta_2 = \theta_2 - \lambda \nabla_{\theta_2}(loss)$

recent and efficient state-of-the-art methods for this task: the Fourier Disparity Layers by Le Pendu et al. [152], the TV regularized sparse light field reconstruction model based on guided-filtering recently proposed by Gao et al. [153], and the light field reconstruction and depth estimation using convolutional neural networks proposed by Huang et al. [144]. For fair comparisons, the method of Huang et al. [144] has been re-trained using the datasets listed in Section 4.4.1.

Additionally, an ablation study is proposed in Section 4.4.5 to study the importance of (i) using jointly the unrolled ADMM FDL optimization and the learned view synthesis network (ii) using the shifted version of the FDL as well as the angular coordinate channels as network input in the view synthesis process.

4.4.1 Datasets

Two-thirds of both the Stanford Lytro light field archive dataset [180] and the Kalantari dataset [181] were used as training datasets. Reconstruction performances are then

evaluated with the remaining third of both datasets along with the Linköping Light Field dataset [182]. The input measurements consist of focal stacks with 2 or 3 images (i.e. shots) synthesized from ground truth views with the shift-and-add method [113] and with focus parameters s covering the disparity range of the scene. As ground truth, a dense light field with a 7×7 angular resolution is considered.

4.4.2 Architecture and training settings

We used the DRUNet denoising architecture as in [75] for both the denoiser $\mathcal{D}$ in equation (4.9) and the view synthesis network $\mathcal{S}$ in equation (4.14). A total of 30 layers in the FDL model and 12 unrolled iterations have been used. Both networks $\mathcal{D}$ and $\mathcal{S}$ use as input the concatenation of all the layers, in order to treat them jointly. For the input of $\mathcal{S}$, the layers are additionally concatenated with the coordinate channels as described in Section 4.3.3. The penalty term ρ in equation (4.8) is trained along with the weights of the two networks. During training, we used a patch size of 64×64 with an additional padding of size 8. Networks are trained for 1000 epochs with a learning rate of 10^{-5} and a batch size of 1. The networks have been retrained specifically for each number of measurements. The loss function $\mathcal{L}$ used was the squared ℓ_2-norm between the ground truth light field sub-aperture views and the corresponding synthesized views as defined in equation (4.15).

Table 4.1 – Comparisons with efficient state-of-the-art methods: average PSNR and SSIM for light field reconstruction

Datasets	Kalantari [119]		Stanford [180]		Linköping [182]	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Number of shots	2					
Huang et al. [144]	31.44 dB	0.880	31.02 dB	0.875	24.18 dB	0.780
Le Pendu et al. [152]	33.01 dB	0.924	34.76 dB	0.933	26.62 dB	0.861
Gao et al. [153]	35.62 dB	0.936	35.12 dB	0.935	27.19 dB	0.853
Unrolled ADMM FDL	39.82 dB	0.968	37.32 dB	0.955	29.22 dB	0.902
Joint optimization FDL	**40.93 dB**	**0.974**	**38.35 dB**	**0.961**	**29.96 dB**	**0.917**
Number of shots	3					
Huang et al. [144]	31.53 dB	0.895	30.59 dB	0.883	23.62 dB	0.788
Le Pendu et al. [152]	35.47 dB	0.947	36.83 dB	0.953	29.15 dB	0.900
Gao et al. [153]	37.21 dB	0.950	36.38 dB	0.947	28.10 dB	0.872
Unrolled ADMM FDL	40.79 dB	0.974	38.48 dB	0.964	30.75 dB	0.920
Joint optimization FDL	**41.83 dB**	**0.978**	**39.39 dB**	**0.969**	**31.87 dB**	**0.933**

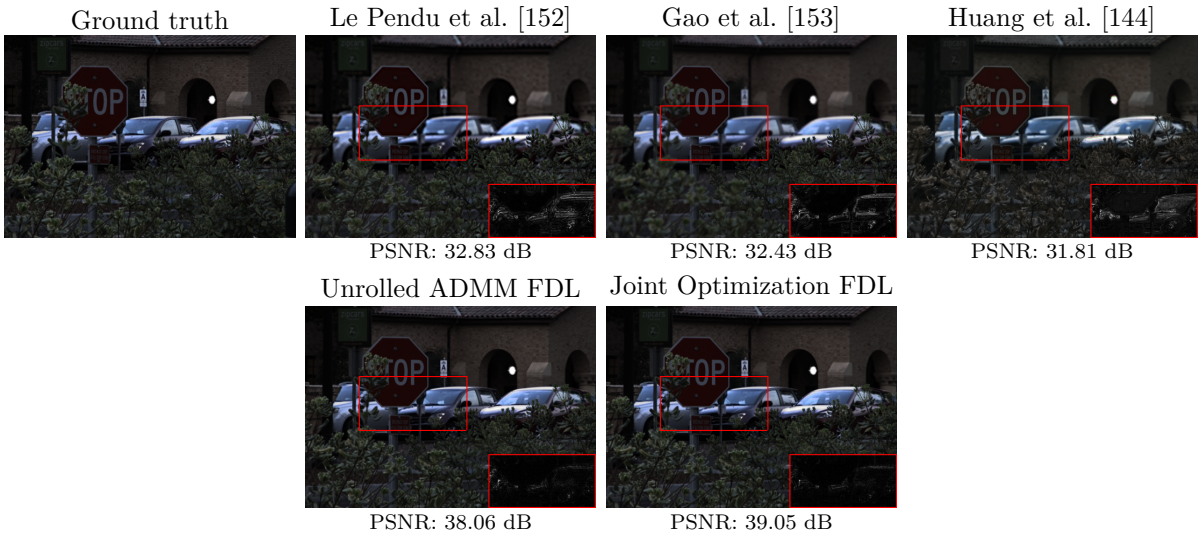


Figure 4.4 – Reconstructed central views for the light field *occlusions_26_eslf* from the Stanford dataset [181] using 2-shots. A portion of the error map is highlighted.

4.4.3 Reconstruction performances

To evaluate the reconstruction performances of the different methods, we measured the quality of the reconstructed light field views using the PSNR and the SSIM metrics, traditionally used by the image processing community. Table 4.1 gives the average PSNRs over the three considered testing datasets for light field reconstruction from 2 and 3 focal stack images as measurements. It shows that the proposed approach significantly outperforms all the state-of-the-art methods on every dataset.

Additionally, Fig. 4.4 shows a reconstructed central view for each evaluated method. As illustrated in the figure, the proposed joint optimization method better reconstructs finer details compared to other approaches.

4.4.4 Algorithm complexity

In this section, we evaluate the complexity of our proposed joint optimization algorithm compared to other state-of-the-art iterative methods. Since the implementation of the method by Gao et al.[153] in CPU only, we present results obtained on both CPU and GPU for fair comparisons. In Table 4.2, we computed the average computation time for the different iterative reconstruction algorithms, and the average computation time for the synthesis of a single view with the FDL-based methods.

On one hand, the obtained computation time shows that the unrolled ADMM FDL

optimization and in the joint optimization method increase the computation time compared to the original FDL reconstruction algorithm presented in [152]. This difference in computation time is mostly due to the computation of the closed-form solution in equation (4.6) and to the application of the denoiser $\mathcal{D}$ in equation (4.9) for several iterations. However, the overall iterative reconstruction algorithm in the FDL domain stays faster to compute than the iterative reconstruction algorithm by Gao et al.[153].

On the other hand, the learned view synthesis presented in Section 4.3.3 increases the computation time of computing a single view from the optimized FDL. Indeed, while each view is computed by a simple shift-and-sum applied to the optimized FDL with the method in [152] and with the unrolled ADMM FDL optimization, the synthesis network $\mathcal{S}$ in (4.14) is applied for each view to be reconstructed in our proposed method. Therefore, the computation time for rendering a dense light field is a limitation of the proposed method. However, it is important to notice that the view synthesis process can be parallelized to synthesize several views simultaneously, which permits us to overcome this computational issue.

Table 4.2 – Algorithm complexity: average computation time (in seconds) (i) for the iterative reconstruction algorithms (ii) for the rendering of a single view.

	Reconstruction algorithm		View synthesis	
	CPU	GPU	CPU	GPU
Gao et al. [153]	443.27 s	-	-	-
Le Pendu et al. [152]	5.56 s	0.22 s	0.02 s	0.002 s
Unrolled ADMM FDL	159.49 s	4.49 s	0.02 s	0.002 s
Joint optimization FDL	159.49 s	4.49 s	5.51 s	0.220 s

4.4.5 Abblation study: the learned view synthesis

In this section, we first study the importance of both the unrolled ADMM FDL optimization and the view synthesis network in the proposed end-to-end framework. We then propose to evaluate different approaches for the input of the network used in the view synthesis process.

End-to-end framework

First of all, we propose to compare the light field reconstruction performances for different frameworks that consider different parts of the proposed joint optimization:

— FDL + view synthesis: this framework uses the FDL optimization proposed in [152], without any learned prior. The learned view synthesis process is trained to reconstruct views from the estimated FDL.

— Unrolled ADMM FDL: the unrolled ADMM FDL optimization without learning the view synthesis process.

— Joint optimization FDL: the proposed joint optimization FDL that considers both the unrolled ADMM FDL optimization and the learned view synthesis parts.

The reconstruction performances are listed in Table 4.3. As shown in the table, having both the unrolled optimization and the view synthesis network offers the best performances by a large margin.

Table 4.3 – Ablation study: average PSNR for light field reconstruction

Datasets	Kalantari [119]		Stanford [180]		Linköping [182]	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Number of shots	2					
FDL + view synthesis	38.75 dB	0.966	36.56 dB	0.954	27.91 dB	0.890
Unrolled ADMM FDL	39.82 dB	0.968	37.32 dB	0.955	29.22 dB	0.902
Joint optimization FDL	**40.93 dB**	**0.974**	**38.35 dB**	**0.961**	**29.96 dB**	**0.917**
Number of shots	3					
FDL + view synthesis	39.77 dB	0.971	38.08 dB	0.963	30.12 dB	0.910
Unrolled ADMM FDL	40.79 dB	0.974	38.48 dB	0.964	30.75 dB	0.920
Joint optimization FDL	**41.83 dB**	**0.978**	**39.39 dB**	**0.969**	**31.87 dB**	**0.933**

To further study this improvement, we propose to empirically verify that the learned view synthesis process is able to reduce the occlusion artifacts not well handled by the FDL model, as explained theoretically in sections 4.3.1 and 4.3.3. Since the FDL are optimized from focal stack measurements captured at angular coordinates $u_0 = 0$, the optimized FDL are then well-defined to reconstruct the central view for any type of scene, while artifacts are expected on the other views in certain areas, e.g. transparency in occluded regions [152], [178]. Therefore, we expect the joint optimization method to reduce these artifacts in order to improve the PSNR of the reconstructed views that are far from the central view.

Fig. 4.6 illustrates a transparency artifact occurring in an occluded region with the unrolled ADMM FDL method. We can visually see that the proposed joint optimization method significantly reduces this artifact. In Fig. 4.5, we computed the average PSNR gain over the Kalantari testing dataset [181] with the end-to-end approach over the unrolled ADMM FDL optimization for several views with different angular coordinates. As shown in Fig. 4.5, the proposed framework always improves the reconstruction quality compared to the unrolled ADMM FDL method, especially for the views that are distant from the central view with an average gain of over 1 dB.

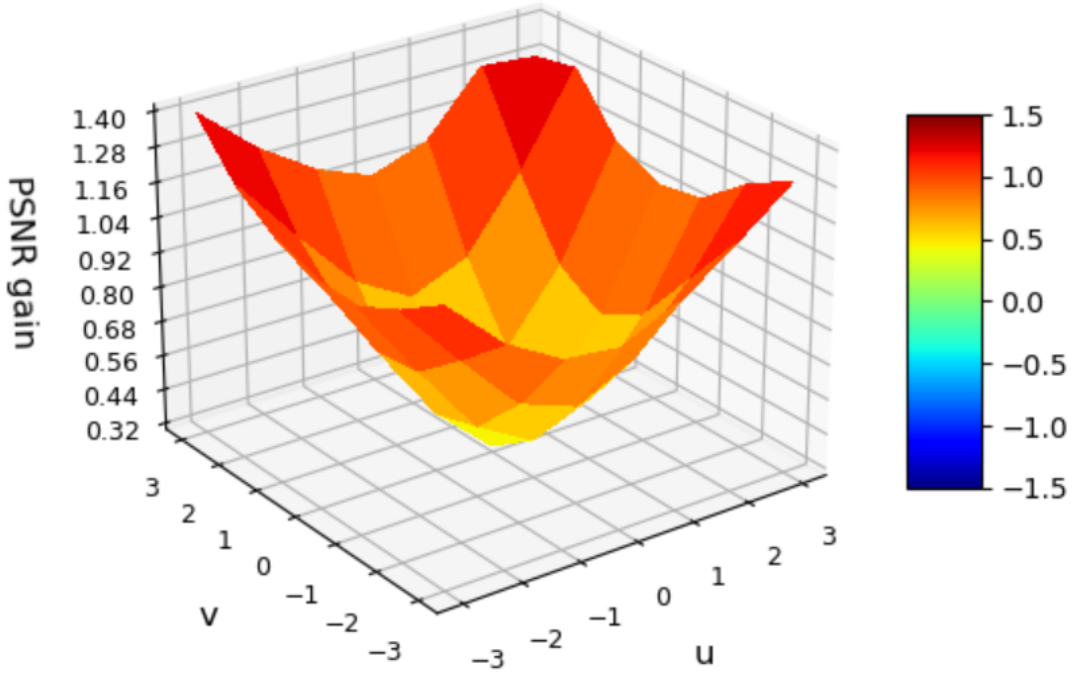


Figure 4.5 – Mean PSNR gain over the test set of the Kalantari dataset [181] for different light field view coordinates with the proposed joint optimization framework compared to the unrolled ADMM FDL optimization using 3-shots focal stacks.

(a) Ground truth

(b) Unrolled ADMM FDL (PSNR: 33.82 dB)

(c) Joint Optimization FDL (PSNR: 39.34 dB)

Figure 4.6 – Example of an occluded region in the light field *occlusion_36_eslf*. The middle row illustrates the transparency artifacts with the FDL model in occluded regions: a building is visible through the grid of a window. These artifacts are reduced in the last row thanks to the learned view synthesis block of the proposed joint optimization.

Network input

In this section, we propose a study of different approaches for the network input in the view synthesis process. To be able to reconstruct any view from the optimized FDL, the network needs an input which contains all the spatial information carried by the optimized FDL, but also angular information so that its output is specific and optimal for each view. A first approach is to concatenate the optimized FDL with additional channels that contain the value of the angular coordinates of the view to be reconstructed. Another approach is to directly incorporate the angular information in the optimized FDL, i.e. by shifting the optimized FDL accordingly to the angular coordinates of the view to be reconstructed, following the view synthesis process of the FDL model in equation (4.12).

In order to select the best approach, we propose to compare the reconstruction performances with different network input configurations, using either additional coordinate channels or the shifted version of the FDL, or both at the same time. In our experiments, when using both approaches at the same time, we obtained better results by first training the joint optimization method by considering only the shifted FDL without any additional coordinate channels as network input, then fine-tuning this pre-trained model by adding the coordinate channels to the network input. We listed the obtained results in Table 4.4 for 2-shots focal stacks as measurements. As shown in the table, the joint optimization method is able to efficiently reconstruct the light fields with all the considered approaches. Additionally, according to these results, both approaches are also complementary, offering the best results when considering both approaches at the same time.

Table 4.4 – Ablation study on view synthesis network input: average PSNR for light field reconstruction

Coordinates	Shift	Kalantari [119]		Stanford [180]		Linköping [182]	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
yes	no	40.77 dB	0.974	38.19 dB	0.960	29.67 dB	0.914
no	yes	40.82 dB	0.974	38.26 dB	0.960	29.93 dB	0.916
yes	yes	**40.93 dB**	**0.974**	**38.35 dB**	**0.961**	**29.96 dB**	**0.917**

4.5 Conclusion

We presented a method to reconstruct a dense light field from a focal stack containing only very few images captured with a single traditional camera. A joint unrolled ADMM

FDL optimization with a learned view synthesis network is presented to extend the Fourier Disparity Layer (FDL) representation of scenes to occluded and non-Lambertian scenes.

The Alternating Direction Method of Multipliers (ADMM) optimization method is unrolled using a deep convolutional denoiser of FDL. Performing the optimization in the FDL domain allows one to derive a closed-form solution for the proximal operator of the data-fit term. Furthermore, unrolling the FDL optimization allows to learn a prior directly in the FDL domain. Additionally, a deep network is trained to adapt the optimized FDL for each view to be reconstructed, in order to minimize the artifacts created with the generation of the views from the FDL model.

Thanks to the capacity of the FDL model to incorporate the light field imaging model in the optimization process, and thanks to deep networks to represent complex priors, the proposed approach significantly outperforms state-of-the-art methods for light field reconstruction from focal stacks with very few shots.

Conclusion

Summary

Image reconstruction from corrupted and incomplete measurements has been a widely studied subject of research, especially in the last decades thanks to the rise of deep learning techniques. The image inverse problem involved, being usually ill-conditioned, is generally posed as the problem of minimizing a function composed of two terms: a data-fidelity term, which measures the fidelity of the solution with the measurements, and a regularization term, which quantifies how the solution matches with prior knowledge on the target image. This minimization problem is usually tackled using iterative optimization algorithms. However, designing a function representing an image prior is a difficult task and has been an attractive subject of research in the domain. Thanks to the power of deep learning techniques to learn complex structures on signals such as images, more complex image priors have been proposed in the literature. One of the most promising approaches uses unrolled optimization algorithms, which have been proposed as a way to automatically learn the image prior for a specific problem and for a specific iterative optimization algorithm. While these methods achieved state-of-the-art reconstruction performances for a variety of image processing tasks, the training of a deep learned prior within an unrolled optimization algorithm poses computational issues, hence limiting the number of iterations considered in the unrolled optimization algorithm.

The first part of this thesis addressed the following issues: *how can we reduce the limitations of the training of an unrolled optimization algorithm ?*

In chapter 2, we addressed this problem and proposed a novel training method for unrolled optimization algorithms, called "Stochastic Unrolled Proximal Point Algorithm" (SUPPA). The proposed approach transforms the training of the whole unrolled optimization algorithm into a set of sub-optimization problems, defined per unrolled iteration. With a stochastic optimization strategy, the computational burden of computing the backpropagation used to train the image prior within the unrolled optimization algorithm is considerably reduced, and the number of unrolled iterations is not restricted by the training process. Consequently, we proposed a method that is efficient in terms of both

the image reconstruction quality and the complexity.

In the second part of this thesis, we focused on inverse problems associated with the acquisition and reconstruction of light fields. A light field describes the scene as a collection of light rays. Each light ray is represented by spatial coordinates, angular coordinates, time, and a wavelength, creating the well-known 7D plenoptic function. The light field is generally considered time-invariant and monochrome, thus reduced to a 4D function with two spatial coordinates and two angular coordinates, thanks to its two-plane parameterization. Therefore, it is generally represented as a set of adjacent views.

To capture such light fields, an intuitive approach consists of taking pictures from several viewpoints, either simultaneously thanks to a large camera array or sequentially with a single camera placed on a moving gantry. Alternatively, more lightweight camera designs have been proposed to capture light fields on a single 2D sensor: the plenoptic cameras, using an array of microlenses placed in front of the photosensor to separate the light rays striking each microlens into a small image, and coded-mask cameras, which modulates the 4D light field into 2D projections. We took a particular interest in an alternative way of capturing a light field which does not require hardware modifications to conventional cameras. It consists in capturing a focal stack, i.e. several images of the scene at different focus distances, in order to reconstruct a light field.

The second part of the thesis thus addressed the following issue: *Can we produce an unrolled optimization algorithm to reconstruct a high-quality light field from a focal stack composed of very few images ?*

To deal with this problem, we presented, in chapter 4, a novel method to reconstruct light fields from a small set of focal stack images. An end-to-end joint optimization framework is proposed, where a novel unrolled optimization method is jointly trained with a view synthesis deep neural network. The proposed unrolled optimization method constructs Fourier Disparity Layers (FDL), a compact representation of light fields which samples Lambertian non-occluded scenes in the depth dimension and from which all the light field viewpoints can be computed. Solving the optimization problem in the FDL domain allows an efficient way to solve the inverse problem efficiently by deriving a closed-form expression of the data-fidelity term of the inverse problem involved. Furthermore, unrolling the FDL optimization allows to learn a prior directly in the FDL domain. In order to widen the FDL representation to more complex scenes, a Deep Convolutional Neural Network (DCNN) is trained to synthesize novel views from the optimized FDL. Thanks to the formulation of the problem in the FDL model and to the efficiency of

deep learning to learn light field priors, the proposed method was able to significantly outperform the state-of-the-art methods for light field reconstruction from focal stack measurements.

Future works and perspectives

We think that this thesis opens interesting research directions and that extensions of the presented works are possible.

The proposed training method for unrolled optimization algorithms, presented in chapter 2, considerably reduces the computational burden of the training. The method relies on the assumption that the Alternating Direction Method of Multipliers (ADMM) optimization method is a special case of the proximal point algorithm in order to write each iteration of the ADMM as a proximal mapping. This allows us to derive small optimization problems, defined per unrolled iteration, used to train the prior of the unrolled ADMM algorithm with a stochastic process. A possible extension of this method would be to widen this approach to other types of iterative optimization algorithms, especially for image inverse problems which cannot be solved using a proximal algorithm.

In chapter 4, we presented an optimization algorithm for the reconstruction of light fields from a focal stack containing only very few images, using the Fourier Disparity Layers (FDL) representation of light fields. Although the method is able to reconstruct high quality light fields, the FDL model is theoretically limited to non-occluded Lambertian light fields. Even though this issue is addressed in chapter 4 by adding a novel learned view synthesis process from the optimized FDL, we think that the method could be further improved by addressing this issue directly in the unrolled optimization algorithm. Furthermore, the addition of a deep view synthesis process increases the computational burden of computing the different viewpoints of a dense light field. Therefore, dealing with the occlusion and reflectance issues of the FDL model directly in the unrolled optimization algorithm could also significantly reduce the computation time of computing the different viewpoints, hence allowing the use of the proposed method in real-time applications.

Another possible shortcoming is the focal stack formation model from a light field presented in chapter 4. While it is classically used in the literature, it is, however, adapted to parfocal lens cameras only. This type of camera is generally used for cinema and broadcasting purposes, while traditional cameras used for photography purposes use varifocal lenses. The main difficulty of our proposed approach with varifocal lens cameras is that

changing the focus also slightly affects the zoom. As a result, the different images of the focal stack have different zoom parameters. Therefore, the focal stack formation model needs to take a zoom factor dependent on the re-focus parameter. A possible improvement of our proposed method would be to integrate this formation model into the unrolled optimization algorithm to open up light field reconstruction from focal stack measurements captured with more types of devices.

LIST OF PUBLICATIONS

[4] B. Le Bon, M. Le Pendu, and C. Guillemot, « Stochastic unrolled proximal point algorithm for linear image inverse problems », *in EUSIPCO 2023-31st European Signal Processing Conference*, 2023.

[5] B. Le Bon, M. Le Pendu, and C. Guillemot, « Unrolled fourier disparity layer optimization for scene reconstruction from few-shots focal stacks », *in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2023, pp. 1–5.

[6] B. L. Bon, M. Le Pendu, and C. Guillemot, « Joint fourier disparity layers unrolling with learned view synthesis for light field reconstruction from few-shots focal stacks », *IEEE Access*, vol. 11, pp. 123 350–123 360, 2023. DOI: 10.1109/ACCESS.2023.3329328.

BIBLIOGRAPHY

[1] B. Wilburn, N. Joshi, V. Vaish, *et al.*, « High performance imaging using large camera arrays », *in ACM SIGGRAPH 2005 Papers*, 2005, pp. 765–776.

[2] *Raytrix*, https://raytrix.de/.

[3] *Lightfieldphotography*, `web.media.mit.edu/~gordonw/CompressiveLightFieldPhotography`

[4] B. Le Bon, M. Le Pendu, and C. Guillemot, « Stochastic unrolled proximal point algorithm for linear image inverse problems », *in EUSIPCO 2023-31st European Signal Processing Conference*, 2023.

[5] B. Le Bon, M. Le Pendu, and C. Guillemot, « Unrolled fourier disparity layer optimization for scene reconstruction from few-shots focal stacks », *in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2023, pp. 1–5.

[6] B. L. Bon, M. Le Pendu, and C. Guillemot, « Joint fourier disparity layers unrolling with learned view synthesis for light field reconstruction from few-shots focal stacks », *IEEE Access*, vol. 11, pp. 123 350–123 360, 2023. DOI: `10.1109/ACCESS.2023.3329328`.

[7] A. K. Boyat and B. K. Joshi, « A review paper: noise models in digital image processing », *arXiv preprint arXiv:1505.03489*, 2015.

[8] D. L. Snyder, A. M. Hammoud, and R. L. White, « Image recovery from data acquired with a charge-coupled-device camera », *JOSA A*, vol. 10, *5*, pp. 1014–1023, 1993.

[9] G. E. Healey and R. Kondepudy, « Radiometric ccd camera calibration and noise estimation », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, *3*, pp. 267–276, 1994.

[10] H. Tian, B. Fowler, and A. E. Gamal, « Analysis of temporal noise in cmos photodiode active pixel sensor », *IEEE Journal of Solid-State Circuits*, vol. 36, *1*, pp. 92–101, 2001.

[11] S. Parrilli, M. Poderico, C. V. Angelino, and L. Verdoliva, « A nonlocal sar image denoising algorithm based on llmmse wavelet shrinkage », *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, *2*, pp. 606–616, 2011.

[12] M. Bertero, P. Boccacci, and C. De Mol, *Introduction to inverse problems in imaging*. CRC press, 2021.

[13] Y. Li, Y. Zhang, R. Timofte, *et al.*, « Ntire 2023 challenge on image denoising: methods and results », *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1904–1920.

[14] A. Kaur and G. Dong, « A complete review on image denoising techniques for medical images », *Neural Processing Letters*, pp. 1–44, 2023.

[15] W. Wu, M. Chen, Y. Xiang, Y. Zhang, and Y. Yang, « Recent progress in image denoising: a training strategy perspective », *IET Image Processing*, 2023.

[16] Y. Tan, D. Zhang, F. Xu, and D. Zhang, « Motion deblurring based on convolutional neural network », *in International Conf. on Bio-Inspired Computing: Theories and Applications. Springer*, 2017, pp. 623–635.

[17] T. Eboli, J. Sun, and J. Ponce, « End-to-end interpretable learning of non-blind image deblurring », *in European Conference on Computer Vision*, Springer, 2020, pp. 314–331.

[18] J. Dong, S. Roth, and B. Schiele, « Deep wiener deconvolution: wiener meets deep learning for image deblurring », *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[19] L. Kong, J. Dong, J. Ge, M. Li, and J. Pan, « Efficient frequency domain-based transformers for high-quality image deblurring », *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5886–5895.

[20] Y. Quan, Z. Wu, and H. Ji, « Neumann network with recursive kernels for single image defocus deblurring », *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5754–5763.

[21] R. Parvaz, « Point spread function estimation for blind image deblurring problems based on framelet transform », *The Visual Computer*, vol. 39, *7*, pp. 2653–2669, 2023.

[22] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, « Low-complexity single-image super-resolution based on nonnegative neighbor embedding », 2012.

[23] S. Anwar, S. Khan, and N. Barnes, « A deep journey into super-resolution: a survey », *ACM Computing Surveys (CSUR)*, vol. 53, *3*, pp. 1–34, 2020.

[24] H. Chen, X. He, L. Qing, *et al.*, « Real-world single image super-resolution: a brief review », *Information Fusion*, vol. 79, pp. 124–145, 2022.

[25] Z. Wang, J. Chen, and S. C. Hoi, « Deep learning for image super-resolution: a survey », *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, *10*, pp. 3365–3387, 2020.

[26] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander III, « Demosaicking methods for bayer color arrays », *Journal of Electronic imaging*, vol. 11, *3*, pp. 306–315, 2002.

[27] D. Menon and G. Calvagno, « Color image demosaicking: an overview », *Signal Processing: Image Communication*, vol. 26, *8-9*, pp. 518–533, 2011.

[28] Y. Wang, R. Cao, Y. Guan, T. Liu, and Z. Yu, « A deep survey in the applications of demosaicking », *in 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, IEEE, 2021, pp. 596–602.

[29] J. Hadamard, « Sur les problèmes aux dérivées partielles et leur signification physique », *Princeton university bulletin*, pp. 49–52, 1902.

[30] B. R. Hunt, « The application of constrained least squares estimation to image restoration by digital computer », *IEEE Transactions on Computers*, vol. 100, *9*, pp. 805–812, 1973.

[31] A. Tikhonov, « Regularization of incorrectly posed problems », *in Soviet Math. Dokl.*, 1963, pp. 1624–1627.

[32] M. Benning and M. Burger, « Modern regularization methods for inverse problems », *Acta numerica*, vol. 27, pp. 1–111, 2018.

[33] C. Lemaréchal, « Cauchy and the gradient method », *Doc Math Extra*, vol. 251, *254*, p. 10, 2012.

[34] N. Qian, « On the momentum term in gradient descent learning algorithms », *Neural networks*, vol. 12, *1*, pp. 145–151, 1999.

[35] Y. Nesterov, « A method for unconstrained convex minimization problem with the rate of convergence o (1/k2) », *in Dokl. Akad. Nauk. SSSR*, vol. 269, 1983, p. 543.

[36] J. Duchi, E. Hazan, and Y. Singer, « Adaptive subgradient methods for online learning and stochastic optimization. », *Journal of machine learning research*, vol. 12, *7*, 2011.

[37] *Rmsprop*, http://www.cs.toronto.edu/ tijmen/csc321/slides/lecture_slides_lec6.pdf/.

[38] D. P. Kingma and J. Ba, « Adam: a method for stochastic optimization », *arXiv preprint arXiv:1412.6980*, 2014.

[39] J. L. Lagrange, *Traité de la résolution des équations numériques de tous les degrés: avec des notes sur plusieurs points de la théorie des équations algébriques*. chez Courcier, 1806.

[40] J.-J. Moreau, « Proximité et dualité dans un espace hilbertien », *Bulletin de la Société mathématique de France*, vol. 93, pp. 273–299, 1965.

[41] P. L. Combettes and V. R. Wajs, « Signal recovery by proximal forward-backward splitting », *Multiscale modeling & simulation*, vol. 4, *4*, pp. 1168–1200, 2005.

[42] C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs, « A variational formulation for frame-based inverse problems », *Inverse Problems*, vol. 23, *4*, p. 1495, 2007.

[43] P. L. Combettes and J.-C. Pesquet, « A douglas–rachford splitting approach to nonsmooth convex variational signal recovery », *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, *4*, pp. 564–574, 2007.

[44] P. L. Combettes and J.-C. Pesquet, « Proximal splitting methods in signal processing », *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212, 2011.

[45] I. Daubechies, M. Defrise, and C. De Mol, « An iterative thresholding algorithm for linear inverse problems with a sparsity constraint », *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, *11*, pp. 1413–1457, 2004.

[46] J. M. Bioucas-Dias and M. A. Figueiredo, « A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration », *IEEE Transactions on Image processing*, vol. 16, *12*, pp. 2992–3004, 2007.

[47] A. Beck and M. Teboulle, « A fast iterative shrinkage-thresholding algorithm for linear inverse problems », *SIAM journal on imaging sciences*, vol. 2, *1*, pp. 183–202, 2009.

[48] J. Douglas and H. H. Rachford, « On the numerical solution of heat conduction problems in two and three space variables », *Transactions of the American mathematical Society*, vol. 82, *2*, pp. 421–439, 1956.

[49] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, « Distributed optimization and statistical learning via the alternating direction method of multipliers », *Foundations and Trends® in Machine learning*, vol. 3, *1*, pp. 1–122, 2011.

[50] D. Geman and C. Yang, « Nonlinear image recovery with half-quadratic regularization », *IEEE transactions on Image Processing*, vol. 4, *7*, pp. 932–946, 1995.

[51] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, « A review on deep learning in medical image analysis », *International Journal of Multimedia Information Retrieval*, vol. 11, *1*, pp. 19–38, 2022.

[52] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, « Image segmentation using deep learning: a survey », *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, *7*, pp. 3523–3542, 2021.

[53] J. Su, B. Xu, and H. Yin, « A survey of deep learning approaches to image restoration », *Neurocomputing*, vol. 487, pp. 46–65, 2022.

[54] Z. Liu, L. Jin, J. Chen, *et al.*, « A survey on applications of deep learning in microscopy image analysis », *Computers in biology and medicine*, vol. 134, p. 104 523, 2021.

[55] B. Mahesh, « Machine learning algorithms-a review », *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, *1*, pp. 381–386, 2020.

[56] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[57] A. M. Salman, A. D. Malony, and M. J. Sottile, « An open domain-extensible environment for simulation-based scientific investigation (odessi) », *in Computational Science–ICCS 2009: 9th International Conference Baton Rouge, LA, USA, May 25-27, 2009 Proceedings, Part I 9*, Springer, 2009, pp. 23–32.

[58] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, « Unsupervised learning based on artificial neural network: a review », *in 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, IEEE, 2018, pp. 322–327.

[59] X. Glorot and Y. Bengio, « Understanding the difficulty of training deep feedforward neural networks », *in Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[60] S. K. Kumar, « On weight initialization in deep neural networks », *arXiv preprint arXiv:1704.08863*, 2017.

[61] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone, « A review on weight initialization strategies for neural networks », *Artificial intelligence review*, vol. 55, *1*, pp. 291–322, 2022.

[62] G. Hinton, N. Srivastava, and K. Swersky, « Neural networks for machine learning lecture 6a overview of mini-batch gradient descent », *Cited on*, vol. 14, *8*, p. 2, 2012.

[63] R. Hecht-Nielsen, « Theory of the backpropagation neural network », *in Neural networks for perception*, Elsevier, 1992, pp. 65–93.

[64] T. D. Bui and G. Chen, « Translation-invariant denoising using multiwavelets », *IEEE transactions on signal processing*, vol. 46, *12*, pp. 3414–3420, 1998.

[65] A. Chambolle, « An algorithm for total variation minimization and applications », *Journal of Mathematical imaging and vision*, vol. 20, pp. 89–97, 2004.

[66] J. Mairal, M. Elad, and G. Sapiro, « Sparse representation for color image restoration », *IEEE Transactions on image processing*, vol. 17, *1*, pp. 53–69, 2007.

[67] M. Yang, D. Dai, L. Shen, and L. Van Gool, « Latent dictionary learning for sparse representation based classification », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 4138–4145.

[68] E. Miandji, J. Kronander, and J. Unger, « Compressive image reconstruction in reduced union of subspaces », *in Computer Graphics Forum*, Wiley Online Library, vol. 34, 2015, pp. 33–44.

[69] I. N. Sneddon, *Fourier transforms*. Courier Corporation, 1995.

[70] N. Ahmed, T. Natarajan, and K. R. Rao, « Discrete cosine transform », *IEEE transactions on Computers*, vol. 100, *1*, pp. 90–93, 1974.

[71] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[72] L. I. Rudin, S. Osher, and E. Fatemi, « Nonlinear total variation based noise removal algorithms », *Physica D: nonlinear phenomena*, vol. 60, *1-4*, pp. 259–268, 1992.

[73] C. Louchet, « Modèles variationnels et bayésiens pour le débruitage d'images: de la variation totale vers les moyennes non-locales », Ph.D. dissertation, Université René Descartes-Paris V, 2008.

[74] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, « Plug-and-play priors for model based reconstruction », *in 2013 IEEE global conference on signal and information processing*, IEEE, 2013, pp. 945–948.

[75] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, « Plug-and-play image restoration with deep denoiser prior », *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2021.

[76] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, « Beyond a gaussian denoiser: residual learning of deep cnn for image denoising », *IEEE transactions on image processing*, vol. 26, *7*, pp. 3142–3155, 2017.

[77] H. E. Robbins, « An empirical bayes approach to statistics », *in Breakthroughs in Statistics: Foundations and basic theory*, Springer, 1992, pp. 388–394.

[78] Y. Song and S. Ermon, « Generative modeling by estimating gradients of the data distribution », *Advances in neural information processing systems*, vol. 32, 2019.

[79] K. Gregor and Y. LeCun, « Learning fast approximations of sparse coding », *in Proceedings of the 27th international conf. on international conf. on machine learning*, 2010, pp. 399–406.

[80] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, « Unrolled optimization with deep priors », *arXiv preprint arXiv:1705.08041*, 2017.

[81] M. Mardani, Q. Sun, S. Vasawanal, *et al.*, « Neural proximal gradient descent for compressive imaging », *arXiv preprintarXiv:1806.03963*, 2018.

[82] U. Schmidt and S. Roth, « Shrinkage fields for effective image restoration », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2774–2781.

[83] Y. Yang, J. Sun, H. Li, and Z. Xu, « Deep admm-net for compressive sensing mri », *in Proceedings of the 30th international conf. on neural information processing systems*, 2016, pp. 10–18.

[84] D. Gilton, G. Ongie, and R. Willett, « Neumann networks for linear inverse problems in imaging », *IEEE Transactions on Computational Imaging*, vol. 6, pp. 328–343, 2019.

[85] G. Le Guludec and C. Guillemot, « Deep unrolling for light field compressed acquisition using coded masks », *IEEE Access*, vol. 10, pp. 42 933–42 948, 2022.

[86] D. Gilton, G. Ongie, and R. Willett, « Deep equilibrium architectures for inverse problems in imaging », *IEEE Transactions on Computational Imaging*, vol. 7, pp. 1123–1133, 2021.

[87] H. Heaton, S. W. Fung, A. Gibali, and W. Yin, « Feasibility-based fixed point networks », *arXiv preprint arXiv:2104.14090*, 2021.

[88] S. Bai, J. Z. Kolter, and V. Koltun, « Deep equilibrium models », *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[89] S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin, « Jfb: jacobian-free backpropagation for implicit networks », *in Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 6648–6656.

[90] L. B. Almeida, « A learning rule for asynchronous perceptrons with feedback in a combinatorial environment », *in Artificial neural networks: concept learning*, 1990, pp. 102–111.

[91] F. J. Pineda, « Generalization of back-propagation to recurrent neural networks », *Physical review letters*, vol. 59, *19*, p. 2229, 1987.

[92] S. G. Krantz and H. R. Parks, *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2012.

[93] J. Eckstein and D. P. Bertsekas, « On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators », *Mathematical Programming*, vol. 55, *1*, pp. 293–318, 1992.

[94] D. Gabay, « Chapter ix applications of the method of multipliers to variational inequalities », *in Studies in mathematics and its applications*, vol. 15, Elsevier, 1983, pp. 299–331.

[95] B. Martinet, « Regularisation, d'inéquations variationelles par approximations succesives », *Revue Francaise d'informatique et de Recherche operationelle*, 1970.

[96] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, « Image super-resolution using very deep residual channel attention networks », *in European conf. on computer vision (ECCV)*, 2018, pp. 286–301.

[97] Q. Ning, W. Dong, G. Shi, L. Li, and X. Li, « Accurate and lightweight image super-resolution with model-guided deep unfolding network », *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[98] E. Agustsson and R. Timofte, « Ntire 2017 challenge on single image super-resolution: dataset and study », *in IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.

[99] A. Ignatov, R. Timofte, *et al.*, « Pirm challenge on perceptual image enhancement on smartphones: report », *in European Conf. on Computer Vision (ECCV) Workshops*, 2019.

[100] R. Zeyde, M. Elad, and M. Protter, « On single image scale-up using sparse-representations », *in International Conf. on curves and surfaces*, Springer, 2010, pp. 711–730.

[101] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, « Contour detection and hierarchical image segmentation », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, *5*, pp. 898–916, May 2011, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.161. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.161.

[102] K. Ma, Z. Duanmu, Q. Wu, *et al.*, « Waterloo Exploration Database: new challenges for image quality assessment models », *IEEE Trans. on Image Processing*, vol. 26, pp. 1004–1016, 2017.

[103] A. Gershun, « The light field », *Journal of Mathematics and Physics*, vol. 18, *1-4*, pp. 51–151, 1939.

[104] E. H. Adelson, J. R. Bergen, *et al.*, « The plenoptic function and the elements of early vision », *Computational models of visual processing*, vol. 1, *2*, pp. 3–20, 1991.

[105] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, « The lumigraph », *in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 43–54.

[106] M. Levoy and P. Hanrahan, « Light field rendering », *in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 31–42.

[107] G. Lippmann, « La photographie integrale », *Comptes-Rendus*, vol. 146, pp. 446–451, 1908.

[108] J. C. Yang, M. Everett, C. Buehler, and L. McMillan, « A real-time distributed light field camera. », *Rendering Techniques*, vol. 2002, *77-86*, p. 2, 2002.

[109] B. Wilburn, N. Joshi, V. Vaish, M. Levoy, and M. Horowitz, « High-speed videography using a dense camera array », *in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, vol. 2, 2004, pp. II–II.

[110] K. Venkataraman, D. Lelescu, J. Duparré, *et al.*, « Picam: an ultra-thin high performance monolithic camera array », *ACM Transactions on Graphics (TOG)*, vol. 32, *6*, pp. 1–13, 2013.

[111] C.-T. Huang, J. Chin, H.-H. Chen, Y.-W. Wang, and L.-G. Chen, « Fast realistic refocusing for sparse light fields », *in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, pp. 1176–1180.

[112] E. H. Adelson and J. Y. Wang, « Single lens stereo with a plenoptic camera », *IEEE transactions on pattern analysis and machine intelligence*, vol. 14, *2*, pp. 99–106, 1992.

[113] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, Ph.D. dissertation, Stanford university, 2005.

[114] R. A. Farrugia, C. Galea, and C. Guillemot, « Super resolution of light field images using linear subspace projection of patch-volumes », *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, *7*, pp. 1058–1071, 2017.

[115] M. Rossi and P. Frossard, « Geometry-consistent light field super-resolution via graph-based regularization », *IEEE Transactions on Image Processing*, vol. 27, *9*, pp. 4207–4218, 2018.

[116] Y. Wang, F. Liu, K. Zhang, G. Hou, Z. Sun, and T. Tan, « Lfnet: a novel bidirectional recurrent convolutional neural network for light-field image super-resolution », *IEEE Transactions on Image Processing*, vol. 27, *9*, pp. 4274–4286, 2018.

[117] R. A. Farrugia and C. Guillemot, « Light field super-resolution using a low-rank prior and deep convolutional neural networks », *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, *5*, pp. 1162–1175, 2019.

[118] T. E. Bishop, S. Zanetti, and P. Favaro, « Light field superresolution », *in 2009 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2009, pp. 1–9.

[119] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, « Learning-based view synthesis for light field cameras », *ACM Transactions on Graphics (TOG)*, vol. 35, *6*, pp. 1–10, 2016.

[120] E. Penner and L. Zhang, « Soft 3d reconstruction for view synthesis », *ACM Transactions on Graphics (TOG)*, vol. 36, *6*, pp. 1–11, 2017.

[121] J. Flynn, M. Broxton, P. Debevec, *et al.*, « Deepview: view synthesis with learned gradient descent », *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2367–2376.

[122] S. Wanner and B. Goldluecke, « Variational light field analysis for disparity estimation and super-resolution », *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, *3*, pp. 606–619, 2013.

[123] Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee, and I. S. Kweon, « Light-field image super-resolution using convolutional neural network », *IEEE Signal Processing Letters*, vol. 24, *6*, pp. 848–852, 2017.

[124] N. Meng, X. Wu, J. Liu, and E. Lam, « High-order residual network for light field super-resolution », *in Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 11 757–11 764.

[125] S. D. Babacan, R. Ansorge, M. Luessi, R. Molina, and A. K. Katsaggelos, « Compressive sensing of light fields », *in 2009 16th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2009, pp. 2337–2340.

[126] E. J. Candès, J. Romberg, and T. Tao, « Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information », *IEEE Transactions on information theory*, vol. 52, *2*, pp. 489–509, 2006.

[127] D. L. Donoho, « Compressed sensing », *IEEE Transactions on information theory*, vol. 52, *4*, pp. 1289–1306, 2006.

[128] R. G. Baraniuk, « Compressive sensing [lecture notes] », *IEEE signal processing magazine*, vol. 24, *4*, pp. 118–121, 2007.

[129] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, « Compressive light field photography using overcomplete dictionaries and optimized projections », *ACM Transactions on Graphics (TOG)*, vol. 32, *4*, pp. 1–12, 2013.

[130] E. Miandji, J. Unger, and C. Guillemot, « Multi-shot single sensor light field camera using a color coded mask », *in 2018 26th European Signal Processing Conference (EUSIPCO)*, IEEE, 2018, pp. 226–230.

[131] H.-N. Nguyen, E. Miandji, and C. Guillemot, « Multi-mask camera model for compressed acquisition of light fields », *IEEE Transactions on Computational Imaging*, vol. 7, pp. 191–208, 2021.

[132] M. Gupta, A. Jauhari, K. Kulkarni, S. Jayasuriya, A. Molnar, and P. Turaga, « Compressive light field reconstructions using deep learning », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 11–20.

[133] A. K. Vadathya, S. Cholleti, G. Ramajayam, V. Kanchana, and K. Mitra, « Learning light field reconstruction from a single coded image », *in 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, IEEE, 2017, pp. 328–333.

[134] O. Nabati, D. Mendlovic, and R. Giryes, « Fast and accurate reconstruction of compressed color light field », *in 2018 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2018, pp. 1–11.

[135] M. Guo, J. Hou, J. Jin, J. Chen, and L.-P. Chau, « Deep spatial-angular regularization for compressive light field reconstruction over coded apertures », *in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, 2020, pp. 278–294.

[136] R. Ng, « Fourier slice photography », *in ACM Siggraph 2005 Papers*, 2005, pp. 735–744.

[137] J. R. Alonso, A. Fernández, and J. A. Ferrari, « Reconstruction of perspective shifts and refocusing of a three-dimensional scene from a multi-focus image stack », *Applied optics*, vol. 55, *9*, pp. 2380–2386, 2016.

[138] S. K. Nayar and Y. Nakagawa, « Shape from focus », *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 16, *8*, pp. 824–831, 1994.

[139] L. McMillan and G. Bishop, « Plenoptic modeling: an image-based rendering system », *in Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 39–46.

[140] N. Asada, H. Fujiwara, and T. Matsuyama, « Edge and depth from focus », *International Journal of Computer Vision*, vol. 26, pp. 153–163, 1998.

[141] A. S. Malik, T. L. Song, and T.-S. Choi, « Depth map estimation based on linear regression using image focus », *International Journal of Imaging Systems and Technology*, vol. 21, *3*, pp. 241–246, 2011.

[142] A. Mousnier, E. Vural, and C. Guillemot, « Partial light field tomographic reconstruction from a fixed-camera focal stack », *arXiv preprint arXiv:1503.01903*, 2015.

[143] A. Levin and F. Durand, « Linear view synthesis using a dimensionality gap light field prior », *in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1831–1838.

[144] Z. Huang, J. A. Fessler, T. B. Norris, and I. Y. Chun, « Light-field reconstruction and depth estimation from focal stack images using convolutional neural networks », *in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 8648–8652.

[145] X. Yin, G. Wang, W. Li, and Q. Liao, « Iteratively reconstructing 4d light fields from focal stacks », *Applied optics*, vol. 55, *30*, pp. 8457–8463, 2016.

[146] C. Liu, J. Qiu, and M. Jiang, « Light field reconstruction from projection modeling of focal stack », *Optics express*, vol. 25, *10*, pp. 11 377–11 388, 2017.

[147] K. Takahashi, Y. Kobayashi, and T. Fujii, « From focal stack to tensor light-field display », *IEEE Trans. on Image Processing*, vol. 27, *9*, pp. 4571–4584, 2018.

[148] S. Gao and G. Qu, « Filter-based landweber iterative method for reconstructing the light field », *IEEE Access*, vol. 8, pp. 138 340–138 349, 2020.

[149] M.-B. Lien, C.-H. Liu, I. Y. Chun, *et al.*, « Ranging and light field imaging with transparent photodetectors », *Nature Photonics*, vol. 14, *3*, pp. 143–148, 2020.

[150] M. H. Kamal, B. Heshmat, R. Raskar, P. Vandergheynst, and G. Wetzstein, « Tensor low-rank and sparse light field photography », *Computer Vision and Image Understanding*, vol. 145, pp. 172–181, 2016.

[151] C. J. Blocker, Y. Chun, and J. A. Fessler, « Low-rank plus sparse tensor models for light-field reconstruction from focal stack data », *in 2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, IEEE, 2018, pp. 1–5.

[152] M. Le Pendu, C. Guillemot, and A. Smolic, « A fourier disparity layer representation for light fields », *IEEE Trans. on Image Processing*, vol. 28, *11*, pp. 5740–5753, 2019.

[153] S. Gao, G. Qu, M. Sjöström, and Y. Liu, « A tv regularisation sparse light field reconstruction model based on guided-filtering », *Signal Processing: Image Communication*, p. 116 852, 2022.

[154] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, « Light field compression with homography-based low-rank approximation », *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, *7*, pp. 1132–1145, 2017.

[155] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, « Nerf: representing scenes as neural radiance fields for view synthesis », *Communications of the ACM*, vol. 65, *1*, pp. 99–106, 2021.

[156] D. Scharstein and R. Szeliski, « A taxonomy and evaluation of dense two-frame stereo correspondence algorithms », *International journal of computer vision*, vol. 47, pp. 7–42, 2002.

[157] A. Saxena, M. Sun, and A. Y. Ng, « Make3d: learning 3d scene structure from a single still image », *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, *5*, pp. 824–840, 2008.

[158] D. Eigen, C. Puhrsch, and R. Fergus, « Depth map prediction from a single image using a multi-scale deep network », *Advances in neural information processing systems*, vol. 27, 2014.

[159] Y. Cao, Z. Wu, and C. Shen, « Estimating depth from monocular images as classification using deep fully convolutional residual networks », *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, *11*, pp. 3174–3182, 2017.

[160] Y. Kuznietsov, J. Stuckler, and B. Leibe, « Semi-supervised deep learning for monocular depth map prediction », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6647–6655.

[161] S. Evain and C. Guillemot, « A lightweight neural network for monocular view generation with occlusion handling », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, *6*, pp. 1832–1844, 2019.

[162] O. Veksler, « Stereo matching by compact windows via minimum ratio cycle », *in Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, IEEE, vol. 1, 2001, pp. 540–547.

[163] C. L. Zitnick and T. Kanade, « A cooperative algorithm for stereo matching and occlusion detection », *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, *7*, pp. 675–684, 2000.

[164] T. Taniai, Y. Matsushita, and T. Naemura, « Graph cut based continuous stereo matching using locally shared labels », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1613–1620.

[165] J. Zbontar and Y. LeCun, « Computing the stereo matching cost with a convolutional neural network », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1592–1599.

[166] Z. Chen, X. Sun, L. Wang, Y. Yu, and C. Huang, « A deep visual correspondence embedding model for stereo matching costs », *in Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 972–980.

[167] A. Kendall, H. Martirosyan, S. Dasgupta, *et al.*, « End-to-end learning of geometry and context for deep stereo regression », *in Proceedings of the IEEE international conference on computer vision*, 2017, pp. 66–75.

[168] R. C. Bolles, H. H. Baker, and D. H. Marimont, « Epipolar-plane image analysis: an approach to determining structure from motion », *International journal of computer vision*, vol. 1, *1*, pp. 7–55, 1987.

[169] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, « Robust depth estimation for light field via spinning parallelogram operator », *Computer Vision and Image Understanding*, vol. 145, pp. 148–159, 2016.

[170] S. Heber, W. Yu, and T. Pock, « Neural epi-volume networks for shape from light field », *in Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2252–2260.

[171] C. Shin, H.-G. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, « Epinet: a fully-convolutional neural network using epipolar geometry for depth from light field images », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4748–4757.

[172] H.-G. Jeon, J. Park, G. Choe, *et al.*, « Accurate depth map estimation from a lenslet light field camera », *in Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1547–1555.

[173] J. Navarro and A. Buades, « Robust and dense depth estimation for light field images », *IEEE Transactions on Image Processing*, vol. 26, *4*, pp. 1873–1886, 2017.

[174] X. Jiang, M. Le Pendu, and C. Guillemot, « Depth estimation with occlusion handling from a sparse set of light field views », *in 2018 25th IEEE international conference on image processing (ICIP)*, IEEE, 2018, pp. 634–638.

[175] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, « Depth from combining defocus and correspondence using light-field cameras », *in Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673–680.

[176] H. Lin, C. Chen, S. B. Kang, and J. Yu, « Depth recovery from light field using focal stack symmetry », *in Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3451–3459.

[177] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, « Depth from shading, defocus, and correspondence using light-field angular coherence », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1940–1948.

[178] T. Herfet, K. Chelli, and M. Le Pendu, « Light field representation: the dimensions in light fields », *in Immersive Video Technologies*, Elsevier, 2023, pp. 173–199.

[179] S. Zheng, Y. Liu, Z. Meng, *et al.*, « Deep plug-and-play priors for spectral snapshot compressive imaging », *Photonics Research*, vol. 9, *2*, B18–B29, 2021.

[180] R. Shah, G. Wetzstein, A. S. Raj, and M. Lowney, *Stanford lytro light field archive*, 2018.

[181] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, « A dataset and evaluation methodology for depth estimation on 4d light fields », *in Asian conf. on computer vision*, Springer, 2016, pp. 19–34.

[182] E. Miandji, H.-N. Nguyen, S. Hajisharif, J. Unger, and C. Guillemot, « Compressive hdr light field imaging using a single multi-iso sensor », *IEEE Transactions on Computational Imaging*, vol. 7, pp. 1369–1384, 2021. DOI: 10.1109/TCI.2021.3132191.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Apprentissage profond pour l'acquisition et la restauration de champs de lumière

Mot clés : Problème inverse, champs de lumière, apprentissage profond, algorithmes déroulés, restauration d'images

Résumé : L'acquisition d'une image est restreinte par les limitations du matériel d'acquisition et est soumise à des perturbations. La reconstruction d'images à partir de mesures dégradées est un problème inverse, souvent mal conditionné et demandant donc la présence d'une connaissance à priori sur les images à reconstruire. Les algorithmes déroulés ont prouvé leur efficacité en matière de résolution de problèmes inverses, mais leur coût en mémoire et en temps de calcul est très élevé.

Notre première contribution est une méthode d'entraînement pour les algorithmes déroulés, permettant de considérablement réduire les coûts et les contraintes liées à l'entraînement de ces méthodes.

Nous nous intéressons ensuite plus particulièrement aux problèmes inverses liés à l'acquisition et à la reconstruction de champs de lumière. Ceux-ci permettent d'obtenir l'information 3D cruciale pour une variété de tâches en imagerie, qui est perdue lors de l'acquisition d'une image avec une caméra traditionnelle. Un champ de lumière est généralement capturé via des appareils coûteux et non accessibles au grand public.

Notre deuxième contribution est une méthode basée sur les algorithmes d'optimisation déroulés, permettant de reconstruire un champ de lumière à partir d'un empilement de mises au point, contenant peu d'images capturées avec une caméra traditionnelle.

Title: Deep learning for light field acquisition and restoration

Keywords: Inverse problem, light fields, deep learning, unrolled algorithms, image reconstruction

Abstract: The acquisition of an image is restricted by the limitations of the acquisition device and subject to perturbations. Image reconstruction from degraded measurements is an inverse problem, usually ill-conditioned, hence requiring the presence of an image prior. Unrolled optimization algorithms have achieved state-of-the-art results for a variety of image reconstruction tasks, but suffer from a high computational burden.

Our first contribution is a training method for unrolled optimization algorithms which considerably reduces the computational burden and constraints of the training.

We then focus on inverse problems related to the acquisition and restoration of light fields. This representation of scenes contains the 3D information that is crucial for a variety of image processing tasks and that is generally lost in the acquisition of an image with a traditional camera. Light fields are generally captured with complex devices that are not accessible to the general public.

Our second contribution is a method, based on the unrolled algorithms, which reconstructs light fields from a focal stack containing very few images captured with a traditional camera.