Image Reconstruction from Multiple Shots with Trainable Algorithms

Bruno Lecouat

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL
Préparée à l'Ecole Normale Supérieure

Image Reconstruction from Multiple Shots with Trainable Algorithms

Soutenue par
Bruno Lecouat
Le 15 Novembre 2023

Ecole doctorale n° 386
Sciences Mathématiques de Paris Centre

Spécialité
Informatique

Composition du jury :

Julie, Delon Université Paris Cité	*Présidente*
Thomas, Pock TU Graz	*Rapporteur*
Jean-Michel, Morel Université Paris Saclay	*Rapporteur*
Julien, Mairal INRIA	*Directeur de thèse*
Jean, Ponce Ecole Normale Supérieure	*Directeur de thèse*

Inria

1794
ENS
ÉCOLE NORMALE SUPÉRIEURE

Ah ! Que le monde est grand à la clarté des lampes !
Aux yeux du souvenir que le monde est petit !

Charles Baudelaire, Le Voyage

Abstract

Among the extensive research dedicated to inverse problems for image restoration, multi-frame-based methods have shown much promise. These methods aim to overcome hardware limitations by combining shots taken in rapid succession, possibly with various camera settings, to best exploit the imaging device's capability. Deep learning is another prominent research direction. Still, it faces limitations for real-world image restoration: (1) artifacts arise from disparities between training images (simulated) and real-world data (2) inferred images may feature false details which represents a significant limitation for accurate scientific and medical applications (3) the high computational costs that make challenging deployments on embedded devices. Hybrid methods have gained significant attention for bridging the gap between model-based approaches and machine learning for tackling inverse problems. By embedding the physical models into learning algorithms, it is possible to achieve state-of-the-art performances on various image restoration tasks with compact architectures, on par with state-of-the-art neural networks, with significantly reduced computational cost and improved robustness on real-world images. The present thesis is dedicated to an in-depth exploration of hybrid methods for solving inverse problems, with a specific focus on their pragmatic implementation in burst photography for real-world applications. The first part of this thesis studies hybrid methods for single-image restoration, providing some methodological tools and some tricks for unrolled optimization. We propose a trainable non-local sparse model for image restoration, leveraging a differentiable relaxation of the unrolled group lasso solver. Taking it a step further, we propose a framework providing differentiable relaxations of convex non-smooth optimization solvers for classic image priors and some. These models demonstrate comparable performance to large neural networks but with significantly fewer parameters, increased interpretability, and faster training times, requiring less training data. The second part of the thesis delves into combining hybrid methods with multi-frame image restoration for super-resolution and HDR reconstruction applications. In this section, our primary focus is reconstructing scenes using real-world images rather than relying on experiments conducted with synthetic data. The design of plug-and-play (PnP) algorithms for burst photography is explored, with efforts directed toward practical implementation and optimization for mobile devices. Throughout our investigation, we have consistently identified registration quality as a prominent bottleneck. Finally, we propose a novel, dense multi-frame registration algorithm to tackle this challenge effectively, enabling 3D scene reconstruction from image bursts with tiny baselines.

Résumé

Cette thèse explore les méthodes hybrides pour les problèmes inverses, en se concentrant sur leur mise en œuvre pratique pour la photographie en rafale. Elle est divisée en deux parties principales. La première partie est consacrée à l'étude de méthodes hybrides pour des applications de restauration d'images, en fournissant plusieurs outils méthodologiques. Notamment, un nouveau problème inverse appris régularisé avec un prior parcimonieux non locale est proposé, en tirant parti d'une relaxation différentiable d'un optimiseur du problème d'optimisation du group lasso. Ensuite, un cadre fournissant des relaxations différentiables de solveurs d'optimisation convexes non lisses pour des priors d'images est étudié. Ces modèles présentent des performances comparables à celles de réseaux de neurones état de l'art plus grands, mais avec beaucoup moins de paramètres, une interprétabilité accrue, des temps d'entraînement plus courts et une plus petite quantité de données d'apprentissage. La deuxième partie de la thèse se penche sur l'intégration de l'apprentissage automatique pour les techniques de restauration d'images multi-images, pour des applications sur des images réelles, pour des problèmes comme la super-résolution et la reconstruction HDR. La conception d'algorithmes plug-and-play pour la photographie en rafale est explorée, avec des efforts dirigés vers la mise en œuvre pratique et l'optimisation de la mémoire pour une implémentation sur appareil mobile. Au cours de notre étude, la qualité de l'alignement des images a été identifié comme un élément bloquant. Pour contourner ce problème, nous proposons un nouvel algorithme de recalage multi-images dense, permettant également la reconstruction de scènes 3D à partir de rafales d'images avec de petits déplacements.

Remerciements

Je tiens à remercier sincèrement Dr. Chuan-Sheng Foo et Dr. Vijay Chandrasekhar pour m'avoir initié à la recherche et encouragé à continuer dans cette voie. Au cours des quatre dernières années, j'ai eu l'opportunité de travailler sur des sujets comme le traitement d'images, la vision par ordinateur et l'apprentissage automatique. Je suis reconnaissant d'avoir collaboré avec des personnes exceptionnelles que je tiens à remercier.

Un remerciement sincère à mon co-directeur de thèse, Julien Mairal, qui m'a ouvert les portes de l'équipe Thoth à Grenoble comme ingénieur de recherche pour explorer les représentations parcimonieuses. Sa disponibilité infaillible pendant ces quatre années et demie et son soutien à tous les niveaux, sur tous les sujets, allant de l'optimisation et de l'apprentissage, au c++, ont été inestimables. Ma gratitude s'adresse également à mon co-directeur de thèse, Jean Ponce, pour m'avoir accueilli dans son équipe Willow. Sa disponibilité, ses conseils avisés sur la recherche, la vision, l'image, nos discussions enrichissantes et son engagement à aiguiser ma pensée critique et ma rigueur ont été une source d'inspiration constante.

Mes remerciements se tournent aussi vers les membres du jury, Julie Delon, Thomas Pock, Jean-Michel Morel pour avoir investi leur temps dans l'évaluation de mon travail. Z Je voudrais aussi exprimer également ma gratitude pour les financements et les accès en ressources de calculs dont j'ai pu bénéficier et qui ont rendu cette thèse possible.

Un chaleureux remerciement à Frédéric Guichard, Imène Tarchouna et Balthazar Neveu pour nos échanges dans le cadre d'EnhanceLab. Un merci tout particulier à Maxim Karpushin et Long Nguyen pour leur temps qu'ils ont consacré à partager leur connaissances et leurs conseils précieux. Merci également à Thibaud Briand,, Olivier Duchenne, Florien Denis et Mathieu Toulemont.

J'adresse mes remerciements les plus chaleureux à Etienne, Théo et Thomas, avec qui j'ai eu la chance de pouvoir travailler lors de ces quatre années. J'ai beaucoup appris à vos côtés. Un remerciement tout particulier à Yann pour toutes ces discussions passionnantes, et évidemment à François pour son oreille attentive et toute sa sagacité.

Mes salutations s'adressent aussi à tous les membres de l'équipe Willow à Paris et de l'équipe Thoth à Grenoble. Je salue la "vieille" génération de l'équipe Thoth : Thomas, Mathilde, Alberto, Valentin, Nikita, Grégoire, Vlad. Un salut chaleureux aux membres des équipes Willow/Sierra que j'ai pu côtoyer : Gaspard, Louis, Yann L., Oumayma, Guillaume, Fabian, Elliot, Antoine Y., Antoine B., Ricardo, Justin et tous les autres.

Je saisis l'occasion pour adresser toute ma reconnaissance à mes amis de longue date toulousains, à Paul B. pour ses conseils et sa bienveillance, à Fabien pour sa spontanéité et sa fraîcheur, à l'exceptionelle Clémence, Arthur, Dimitri, Marine et tous les autres ; aux amis de Fermat : Louis, Rémi, Guillaume; aux amis de Telecom : Paul N., Guillaume, Alexis, Mélanie ; aux amis de Singapour : Houssam, Leo, Alex; à Bastien et bien sûr Karl pour tout son soutien.

Enfin, un immense merci à mes parents, à mon frère, et à Ariane pour leur amour et leur support.

Acknowledgements

This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). Julien Mairal and the author were supported by the ERC grant number 714381 (SOLARIS project) and by ANR 3IA MIAI@Grenoble Alpes (ANR-19-P3IA-0003). Jean Ponce was supported in part by the Louis Vuitton/ENS chair in artificial intelligence and the Inria/NYU collaboration. This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011011252R2 made by GENCI.

Table of Contents

Chapter 1

Introduction

Despite their extreme miniaturization, smartphone cameras have led to remarkable performances and have democratized access to photography. Nowadays, users consider the camera module's performance as a top factor when selecting a new smartphone. In the medical field, high-fidelity miniaturized cameras play a crucial role. In endoscopy, they allow exploration of confined anatomical regions -like the gastrointestinal or urinary tract- leading to better diagnoses. They are also heavily used for minimally invasive surgeries reducing bodily harm.

Image quality is a multifaceted concept. It encompasses various aspects such as faithful details reconstruction, reproduction of human visual experience, and artistic expression. In this thesis, we mainly consider the ability to *reconstruct details* faithfully. But quantifying this capability, so simple in appearance, is a real challenge. The literature has widely acknowledged that relying solely on distortion metrics to gauge disparities between the ground truth and captured signals doesn't consistently align with perceived image quality or the perceived amount of conveyed information. This discrepancy arises from several factors intricately tied to the human visual system. Humans possess, for instance, a remarkable ability to discern faint patterns in strong noise with no spatial correlation. Consequently, quantitative evaluation is still an open question, and it should be acknowledged that perceptual evaluations often dictate image rankings.

Accurate details reconstruction relates to at least three technical components of imaging systems: resolution, noise, and dynamic range. *Resolution* is an abstract concept. Here, we follow the standard definition of the optics literature, considering that it refers to the sensor's capability to resolve two punctual light sources. This can be objectified, for example, with the Rayleigh criterion. *Contrast* is also frequently used and can be quantified using modulation transfer functions (MTFs). *Noise* can be measured in signal-to-noise ratio. While *dynamic range*, characterizing the extent of measurements, can be measured in f-stops. The hardware of the camera is optimized to enhance these metrics.

Cameras are made of two essential components: an optical system and a digital sensor. The limitations of cameras can emerge from either of these components. Sensor-limited devices suffer from undersampling resulting in aliasing due to large pixel pitch and limited signal-to-noise ratio. On the other side of the spectrum, optical-limited devices encounter lens aberrations producing various geometric distortions and affecting resolution. They also may be subject to the diffraction barrier,

harming resolutions and contrast. Moreover, miniaturized cameras introduce additional challenges, extending beyond just cost considerations. These constraints include smaller apertures, reducing the amount of gathered light, as well as the need for closer lens-to-sensor distances, resulting in significantly smaller pixel sizes. Achieving satisfactory signal-to-noise ratios and dynamic ranges under these circumstances using current technologies becomes a real challenge.

Computational Methods for Image Restoration

Computational methods that enhance images through software gain particular appeal as they prove effective for both budget-friendly and high-end cameras, especially when physical constraints impede further hardware advancements in compact devices. In this thesis, we focused on several image restoration problems, including denoising, demosaicking, super-resolution, and high-dynamic among others. Such problems are often cast in the family of *inverse problems*, which consists of inferring an underlying clean signal for a given set of observed noisy outcomes, knowing the acquiring system's response —generally, by solving an optimization problem.

Among the vast body of work devoted to inverse problems for image restoration in the literature, *multi-frame* based methods have shown much promise. Multiframe methods aim to overcome hardware limitations by combining shots taken in rapid succession. These shots can be taken with various camera settings such as different exposures, apertures, focus planes, and, of course, a combination of all to exploit best the capacities of the sensor on distinct portions of the signal. Views can be slightly offset by taking advantage of the user's involuntary hand tremors, or they can be with the same viewpoint using a tripod for acquisition. Depending on the applications, displacements may be necessary (super-resolution, 3D reconstruction). For others, such as HDR or denoising, motions are generally unwanted.

Deep learning is, of course, another prominent research direction. However, in the specific context of real-world image restoration, important limitations arise. (**1**) Using deep learning techniques gives rise to artifacts generally attributed to discrepancies between the training images—gathered via camera simulations—and the actual real-world data. (**2**) It is impossible to detect produced artifacts or false details in the produced result. While images enhanced with plausible yet incorrect details, i.e., *hallucinations*, may be acceptable in some scenarios, the accurate reconstruction of patterns holds paramount importance in scientific and medical applications. (**3**) The computational cost of running large neural networks on embedded devices is generally prohibitive.

Hybrid methods, alternatively referred to as trainable algorithms, have gained significant attention in research for bridging the gap between model-based approaches and machine learning for tackling inverse problems. The core idea behind these methods is to incorporate physical models into learning algorithms. It enables trainable models on par with state-of-the-art neural networks, better stability on real data, significantly smaller memory footprints, and computational costs. In the thesis, we focus on two classes of methods belonging to that family of methods:

(**1**) *Plug-and-Play* (PnP) involves solving inverse problems with alternate optimization, with one part dedicated to the minimization of the data-fitting term, usually achieved with classical optimization, while the second part is handled by

a neural network dedicated to solving a more straightforward denoising problem, i.e., projecting current estimate candidate on the natural image manifold. With such an approach, the burden on the neural network component is significantly reduced; therefore, it is possible to compress model size significantly. However, the resulting iterative algorithm no longer has any guarantees to solve an optimization problem.

(**2**) *learned inverse problems* (LIP) [1], involves fine-tuning hyperparameters of inverse problems for optimal reconstructions on a training set. Here the hyperparameters include the parameters of the optimization function, such as the regularization parameters, or else parameters controlling optimization, such as gradient steps or the parameters affecting a preconditioner[2]. This concept was initially introduced to accelerate the solution of the Lasso with a trainable ISTA algorithm called LISTA. The training process generally leverages bilevel optimization techniques where one optimization problem is nested within another.

1.1 Contributions of this Thesis

This thesis focuses on exploring hybrid methods and their practical implementation in burst photography. The emphasis lies on the empirical aspect, aiming to develop algorithms that can effectively process real-world data while utilizing minimal computational resources. As far as possible, the limitations impacting the performance of the algorithms in real cases are identified, and solutions are proposed to maintain acceptable performance. The thesis is built in two main parts. The first part study learned inverse problems and proposed some new methodological tools. The second part focuses on the design of multi-frame methods leveraging hybrid methods.

Part I: Learned Inverse Problems

In the first part, we study hybrid methods for single-image restoration tasks with synthetic degradations. That part focuses more on the methodological side: we propose new tools to design LIP.

In chapters 3, we study LIP for inverse problems regularized with a non-local sparse image prior and propose a differentiable relaxation of a group lasso solver to this end. The model we obtained when trained on compact image datasets, performs on par with state-of-the-art attention-based neural networks with 72x less trainable parameters for demosaicking tasks on reference datasets.

In chapter 4, we propose different tools to differentiate through convex non-smooth optimization solvers more systematically. And also presents several tips and tricks for effective unrolled optimization based on empirical observations. Again the proposed models have the advantage of being extremely compact, data-efficient, more interpretable, and very fast to train while on par with the current state-of-the-art for various image restoration problems.

Part II: Burst Photography

The second part of this thesis focuses on the application of hybrid methods to burst photography for real-world scenarios.

In chapter 5, we study burst super-resolution on raw image bursts, leveraging the PnP framework. Central efforts have been conducted on the experimental side, designing systems that worked for real-world images in computationally

[1] also called learned inverse solvers, or trainable priors, in this thesis

[2] Also sometimes referenced as *learned optimization* in the literature.

constrained environments. Notably, (**1**) we addressed generalization issues by improving the simulated raw data pipeline, and with a refined camera model (**2**) we developed a deghosting method for handling instability issues arising from non-rigid motion and misaligned frames (**3**) we improved registration algorithms for our specific use case by proposing a hierarchical implementation of the Lucas-Kanade algorithm. The hybrid approaches allow very compact neural networks, less computationally demanding than concurrent attention-based neural networks but achieving similar reconstruction quality with improved stability. We worked on an efficient implementation to minimize memory footprint, allowing a first implementation on mobile devices. That very first prototype ran on smartphones in approximately 3 seconds, for a small image, using CPU resources.

Chapter 6 tackles two orthogonal problems: HDR, and super-resolution on raw bursts with bracketed exposures. That setting is hard because one must register with high-accuracy frames with heterogeneous content (varying SNRs and saturations). Achieving accurate registration under such diverse conditions poses a significant challenge. To face this technical challenge a differentiation of the Lucas Kanade algorithm to perform registration on a filtered features map. We also propose an accurate simulation of bracketed data

Throughout our investigation, we have consistently identified registration quality as a prominent bottleneck. To effectively tackle this challenge, our focus shifts to Chapter 7, wherein we introduce an approach that performs dense image registration in the multi-frame setting. We directly optimize the depth and surface orientation at every pixel in a reference image and the extrinsic parameters of all other cameras relative to it. The optimization is carried out by minimizing photometric reprojection errors computed via plane-induced homographies. Remarkably, our method enables 3D reconstructions of scenes even when dealing with very small baselines.

Furthermore, within Chapter A located in the appendix, we incorporate unpublished research concerning a related issue, specifically, the multi-frame registration for HDR images. This supplementary study enables the robust fitting of a global transformation by utilizing a collection of pairwise affine transformations that have been individually computed.

Industrialization

Concluding this thesis, we discuss the core limitations of the proposed algorithms within Chapter 8. Subsequently, we provide a brief overview of the industrialization phase of the algorithms presented. Indeed, the algorithms introduced in this thesis have led to the creation of a new startup, "Enhance Lab[3]", involving my research advisors, aiming to provide software solutions for enhancing image quality in various contexts, including but not limited to smartphones and scientific imaging. I have been working on this project for the last two years. At the time of writing this thesis, after some adventures and twists, Enhance Lab has finally succeeded in its first fundraising and has concluded two proof of concept with major companies in the field of smartphones and scientific imaging. Besides the multiple challenges we faced, in order to navigate the transition from academic research to a robust industrial product validating stringent benchmarks, we confronted a series of novel technical challenges. We give an overview of these new challenges in Chapter 8. These challenges compelled us to propose innovative technical solutions; some are briefly mentioned in this concluding chapter. Finally, in conclusion, we present an

[3]https://enhancelab.fr/

exploration of future research directions that we consider promising. Notably, we highlight multiple avenues for enhancing and refining the proposed methods.

1.2 Outline of this Thesis

In the following, we present the structure of this dissertation and the articles upon which it is based. All of them have been the result of collaborative work. For each publication, the contributions of all the authors are listed.

- **Chapter 2: Background.** This chapter gives the image formation model used in the other chapters and explains the main building blocks of imaging systems. It also gives an overview of the state of the art of image processing algorithms.

  The two next chapters focus on LIP for image restoration.

- **Chapter 3: Differentiable Non-Local Sparse Model.** This chapter is based on the paper *Fully Trainable and Interpretable Non-Local Sparse Models for Image Restoration* [1]. B. Lecouat, J. Ponce, J. Mairal, In *ECCV 2020*. Our models are implemented in PyTorch, and our code can be found at `https://github.com/bruno-31/groupsc`. The initial idea of the main algorithm and the experiments were proposed by the author and further developed with the help of Dr. Mairal. All authors contributed to the writing.

- **Chapter 4: A Framework for Designing Trainable Priors.** This work is based on the paper *A Flexible Framework for Designing Trainable Priors with Adaptive Smoothing and Game Encoding* [2] B. Lecouat, J. Ponce, J. Mairal, In *NeurIPS 2020*. Our models are implemented in PyTorch, and our code can be found at `https://github.com/bruno-31/groupsc`. The initial idea and the main algorithms were designed by Dr. Mairal and the author. All authors contributed equally to the writing.

  We present in the next two chapters algorithms for burst photography.

- **Chapter 5: Super-Resolution from Raw Image Bursts.** This work is based on *Lucas Kanade Reloaded : End-to-End Super-Resolution from Raw Image Bursts* [3] B. Lecouat, J. Ponce, J. Mairal, In *ICCV 2021*. Visual results are available at `https://bruno-31.github.io/lkburst2/`. The initial idea and the first version of the code were developed by the author. Dr. Mairal and Prof. Ponce helped to improve the method. Dr. Mairal helped collect real-world bursts. All authors contributed equally to the writing.

- **Chapter 6: Joint HDR and Super-Resolution from Bracketed Raw Bursts.** This work is based on *High Dynamic Range and Super-Resolution From Raw Image Bursts* [4] B. Lecouat, T.Eboli, J. Ponce, J. Mairal, In *SIGGRAPH 2022*. Initial ideas of the deghoster and differentiable LK were proposed and implemented by the author. Dr. Eboli helped to study the HDR literature, point to the right references, proofread the code, benchmarked the method, and helped with many discussions to improve the method. The first draft of the paper was, in large part, written by Dr. Eboli. All authors contributed to the final writing. Dr. Eboli and the author worked on the oral presentation.

- **Chapter 7: Dense Image Registration and 3D Reconstructions from Bursts.** We finally focus on multi-frame dense registration and 3D reconstruction.

This work is based on *Dense Image Registration, Camera Pose and Depth Estimation from Bursts* [5] B. Lecouat*, Y. Dubois de Mont Marin* T. Bodrito*, J. Mairal, J. Ponce. This paper is under review. The first ideas come from Dr. Ponce and the author. The first version of the code was developed by Y. Dubois de Mont Marin and the author. T. Bodrito worked on another implementation in parallel and provided different insights that improved the method. Y. Dubois de Mont Marin designed and implemented the pose solver as well as the final structure solver used in the paper. T. Bodrito focused on data simulation on Blender and on the experimental section by running baselines with the author. Y. Dubois de Mont Marin contributed in large part to the first draft of the paper. All authors contributed to the writing of the final version.

- **Chapter 8: Conclusion, Industrialization, and Perspectives.** This chapter summarizes the contributions of this thesis, describes the industrialization phase of the burst super-resolution method introduced in this thesis, and provides potential directions for future research. Improvements to the burst super-resolution algorithm were developed by the author and discussed with Prof. Ponce and Dr. Mairal. Most of the improvements of the burst super-resolution algorithm were done while working on the Enhance Lab project.

- **Chapter A: Multi frame alignment for HDR images.** This brief additional work proposes an alternative for multi-frame registration by robustly fitting a global transformation given a set of pairwise parametric transformations estimated pairwise. The initial idea was proposed by the author and further refined with the help of Dr. Mairal and Prof. Ponce. Dr. Mairal developed an improved version of the block coordinate descent solver and implemented the code in C++.

Chapter 2

Background

Chapter abstract:
This chapter aims to provide an exploration of the inner workings of digital cameras, focusing on proposing modeling tools that describe the image formation process and basic image processing steps. To accomplish this objective, we begin by delving into the modeling of light propagation and the optical systems utilized to form images on the camera's sensor plane. This topic is addressed in Section 2.1. Then, we review the general operating principle of digital imaging sensors in Section 2.2. To further enhance our understanding of digital cameras, we present a short review of the image signal processing (ISP) pipeline in Section 2.4. This section explores the fundamental processing steps in converting raw measurements into photographs pleasing to the human eye. Finally, in Sections 2.5,2.6, we review algorithms that can be used to improve the quality of photographs. We especially focus on inverse problems while Section 2.6 emphasizes burst photography techniques.

Contents

2.1 Optics

2.1.1 Light

Light is a phenomenon at the origin of visual perception for humans and is an electromagnetic wave according to physics. The visible spectrum is the part of the electromagnetic spectrum to which humans are sensitive [6]. By an adaptation mechanism, like other species, it includes radiations between approximately 380 nm and 780 nm that have the most significant solar irradiance on Earth [6]. In this Section, we attempt to answer from a physical perspective what is light? The presentation of the modeling of light and its propagation in section 2.1.2 follows the structure proposed in [7] and [8]. For a comprehensive understanding of this topic, we highly recommend readers delve into the remarkable book [7] from Eugene Hecht. Alternatively, if you prefer a shorter introduction, we suggest exploring the first chapters of the excellent thesis [7] from Felix Heide. We highly recommend exploring [6], an exceptional resource that provides comprehensive insights into the origins of vision and the human eye.

Electromagnetic radiations. Electrodynamics explains light as an electromagnetic wave. The fundamental Maxwell's equations (1865) introduced by James Clerk Maxwell govern the evolution of the electromagnetic field. These equations may be combined to obtain the wave equation [7] and show that fluctuations in electromagnetic fields propagate at the speed of light $c \approx 3 \times 10^8 \mathrm{m/s}$ in a vacuum [7]. Electromagnetic waves, characterized by their wavelength λ in meters $[\mathrm{m}]$, consist of synchronized oscillating electric $\vec{E}$ in volts per meters $[\mathrm{V/m}]$, and magnetic fields $\vec{B}$ in Teslas $[\mathrm{T}]$ that can propagate without a medium [9]. These fields are orthogonal to each other and the direction of propagation, forming a transverse wave. Electromagnetic radiation (EMR) is the energy emitted or absorbed by charged particles, and transported by electromagnetic waves. Note that the energy carried by the EM wave is continuous with this model. Treating light as an EM wave allows us to model its propagation and many behaviors at a macroscopical level. We can predict diffraction, interference for coherent light sources, refraction, reflection, scattering, or transmission [7].

Optical rays. Light is often modeled using the concept of optical rays, which is a simplification of the EM model [7]. This is a fundamental tool of geometrical optics that allows the simplified analysis of optical systems [9]. Even though light rays do not have a physical reality, it is relevant to represent light with rays for physical systems where the typical dimension D of the system is much larger than the wavelength [9] ($D \gg \lambda$): light rays do not account diffraction. Rays represent light as oriented lines orthogonal to the EMR wavefront and, therefore, are colinear to the Poyinting vector. See Figure 2.8 for an illustration. Light can be modeled with light rays in a vacuum or in dense media such as glass or air. The rays in homogeneous media are straight but may be curved in a medium where the refractive index changes [9]. More generally, at a macroscopic scale, the trajectory of rays is governed by the *Fermat principle* that we will present in the next Section.

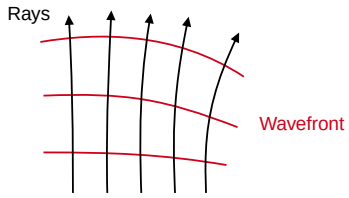


Figure 2.1: A ray serves as a simplified geometric representation of light; it is obtained by choosing a curve perpendicular to the wavefronts that indicate the pathway of energy transmission.

Wave-particle duality. The electromagnetic theory is insufficient in explaining light thoroughly, particularly at the microscopic level. Indeed, it fails to explain some observations. For instance, experiments showed that electrons are ejected from metal surfaces only above a minimum frequency of incident radiations, while no electrons are ejected below this threshold. The energy of the ejected electrons solely depends on the light frequency and not on its intensity [8]. This contradicts Maxwell's theory, which predicts that the energy of light can take any continuous value and depends on the amplitude of the electromagnetic wave and not its frequency [7]. The continuous electromagnetic model also fails at explaining black body radiations [10]. Planck and Einstein introduced the concept of quanta of energy to explain these observations. That approach quantizes energy; materials can receive or emit electromagnetic energy only in specific amounts. Hence, light can exhibit both particle and wave-like behavior. Generally, it is often convenient to consider that light exhibits a particle-like behavior at emission or absorption when interacting with matter and propagates like a wave [9]. By accurately characterizing light as a dual wave-particle that displays both wave-like properties when propagating through space and particle-like behavior during emission and absorption, modern quantum mechanics resolve this conflict [10].

Coherence. Quantum mechanics describe light as a dual wave-particle, where photons are treated as wave packets consisting of a superposition of planar waves [10]. Due to the uncertainty principle [10], the position and speed of photons cannot be determined precisely simultaneously. Consequently, light has a finite frequency bandwidth $\Delta\nu$, which results in a finite length of the wave packet denoted by Δx. Hence, it takes a time Δt for a photon to pass a point in space [9], and this

time satisfies the relation

$$\Delta\nu\Delta t \approx \frac{1}{2\pi}. \tag{2.1}$$

The *coherence length*, denoted by $L = c\Delta t$, refers to the maximum path difference at which light can produce interference (we say that the phase relationship is maintained). A laser can have a coherence length of about 1m [9]. This explains why natural light does not produce interference patterns most of the time. Usually, the luminous intensity of two light beams can be added and produces no interference. Interference is more noticeable with thin layers, such as an iridescent effect on the surface of an oil stain or a pigeon's wing, because the light path difference can be very small in these situations.

Light emission. According to quantum theory, light is emitted when an atom or molecule transitions from a higher to lower energy state [10]. This transition causes the emission of a photon, a quantum of electromagnetic energy. The photon's energy is related to its wavelength or frequency by the equation

$$E = h\nu = \frac{hc}{\lambda}, \tag{2.2}$$

where h is the Planck constant and ν is the frequency of the electromagnetic wave. In most cases, the energy is emitted in a random direction. In some cases, such as in lasers and light-emitting diodes (LEDs), the emission is directed in a specific direction through stimulated emission [9]. While quantum mechanics offer a better explanation of the microscopic behavior of light and its interaction with matter, the electromagnetic model is valid at the macroscopic level [7]. Therefore, we will mainly focus on the electromagnetic model in the next sections.

2.1.2 Propagation of Light

From the wave equation of the electrodynamic model, it follows that, in a vacuum, light travels perpendicularly to the wavefront at the speed of light c. But then a question arises: How does light interact with matter and move in a dense medium such as air? And why does light sometimes travel slower than light speed when photons can only exist at c [7]? These questions are, in fact, crucial for explaining the propagation of light and the phenomena of *transmission*, *reflection*, and *refraction*. These processes look simple at a macroscopical level and can be explained with simple models, but they are, in fact, the result of very complex interactions happening at a microscopic scale [7]. When light encounters matter, countless photons interact with atoms tied together via electromagnetic interactions suspended in the void. A tremendous amount of photons are absorbed and re-emitted by atoms through a process called *scattering*. Transmission, reflection, and refraction can be seen *essentially* as macroscopic manifestations of scattering [7]. We will first explain these phenomena from a sub-microscopic perspective and connect it to the general principles that explain light trajectory at a macroscopical level.

Scattering. For molecules and atoms much smaller than the wavelength, for instance, in media like air, glass, or water, the interaction between light and matter is described with Rayleigh scattering. In that setting, particles act as small oscillators that can be excited by the oscillating electric field of light, causing them to move at the same frequency. The particles, therefore, become small radiating dipoles generating spherical waves at the same frequency (but with different phases). The

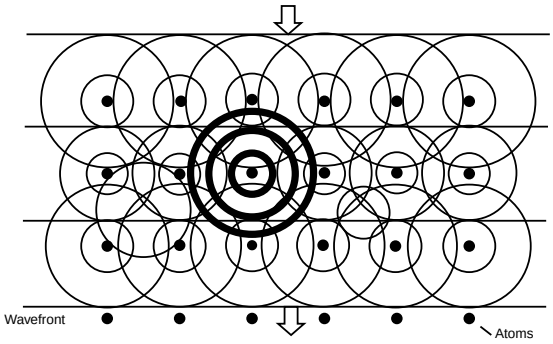


Figure 2.2: Scattering in a dense medium. A planar wave travels downward, encountering an array of atoms. These interactions scatter the wave, resulting in numerous spherical waves with altered phases but matching frequencies. These waves collectively interfere, generating blended patterns that give rise to a secondary downward-traveling plane while exhibiting negative interference in lateral orientations. The phase shift induces a perceived reduction in wave velocity despite electromagnetic waves maintaining a propagation speed of c within the medium. Figure inspired from [7].

closer to the resonance frequency of the molecule, the greater the proportion of power scattered in all directions [7], and for molecules with a resonance far from the radiation frequency, light will not interact.

The sky appears blue due to Rayleigh scattering. Light emitted from the sun passes through atmospheric gas particles whose resonance is close to the UV. Consequently, a large proportion of blue radiations are scattered, while red light is less affected by this phenomenon. Therefore, light rays that have been scattered appear reddish because most of the bluish light has been radiated laterally [7]. This is especially true during sunrise and sunset when light travels through a larger layer of the Earth's atmosphere. Nevertheless, when light passes through a cloudy sky, light is scattered through a different process called *Mie scattering* because the size of the droplets is comparable to the wavelength. In this case, all parts of the light spectrum are scattered equally, causing the clouds to appear white or gray instead of blue [7].

Transmission. In a dense medium, molecules contribute a tremendous number of scattered electromagnetic wavelets that all interfere together. In the case of propagation through a dense medium, due to the geometry, the scattered wavelet mostly cancels each other in all directions except forward, and the beam is sustained. See Figure 2.2 for an illustration of this phenomenon. Refer to [7] for a detailed explanation of this phenomenon. In general, the denser the substance through which light advances and the more ordered the structure of the atom, the less lateral scattering occurs: weak scattering with glass, almost no scattering with quartz, but of course, imperfections aff of sorts such as impurities in a solid cause scattering [9]. On the opposite, randomly and widely spaced scatterers, such as in low-pressure gases, produce wavelets that will mainly not interfere together due to the large path difference except for the forward direction. As a consequence, a part of the radiant energy is scattered laterally [7].

This explains why in glass or in water, the light is not scattered laterally, very little energy is lost in lateral directions, and the propagation can be described with light rays [9]. This is also why, at low altitudes, a dense atmosphere does not pro-

duce Raleigh scattering (otherwise, a far object such as a mountain would appear reddish to the eye). Most of the Rayleigh scattering process occurs at high altitudes where the atmosphere is less dense.

In a dense medium, such as glass, while the propagation direction is the same as in free space, the apparent phase velocity changes, even though photons only exist at speed c [8]. The transmitted light wave moves through the dielectric at a slower speed v with

$$v = \frac{c}{n}, \tag{2.3}$$

where n is the refractive index [8]. This phenomenon arises because when the atoms absorb and re-emit wavelets, it advances or retard the phase of the scattered wave [7]. Interferences with the main wave delayed its apparent velocity even though wavelets *always* travel at speed c. A complete explanation of this phenomenon is well described in [7].

Reflection. Now, let us consider what is happening at a discontinuity between two media with different refractive indices. When a beam of light strikes an interface, some light is scattered forward, but there is always some light scattered backward: this is reflection. In the case of planar wave radiation with an angle of incidence θ_i on a smooth planar surface of a dielectric, a plane wave sweeps in, stimulating atoms across the interface. These radiate and interfere together to give rise to a reflected and a transmitted wave [8]. Due to the geometry of the plane and the wavefront, the incident light beam angle equals the reflected light beam

$$\theta_i = \theta_r, \tag{2.4}$$

and the incident ray (orthogonal to the wavefront) and the reflected ray are in the same plane, see [7] for a more detailed explanation. Note that based on the wave theory, it is possible to calculate the fraction of radiating energy that is reflected and transmitted depending on the two refractive indices using the *Fresnel equations* (1821).

In the case we have just described, when the surface is smooth, this process is called *specular reflection* [11]. A mirror with a polished surface is a good example of specular reflection. Moreover, reflection is enhanced as most of the radiating energy will be reflected due to a metallic coating that suppresses wave transmission [1].

When the surface is rough, it will give a reflection in every direction, and this is called *diffuse reflection* [2] [11]. One particular case commonly used in practice is a *Lambertian surface*, which appears equally bright from all directions and is considered as a perfect diffuser [9]. Most of the objects are modeled as a combination of specular and diffuse reflection [7, 11].

Refraction. Finally, having described the beam reflection at the interface, we now consider the direction of the beam that is transmitted in the medium. When the beam of light hits the object's surface, atoms emit waves in the forward direction, interfering with each other. But because the refractive index changes, this causes the wavefront to change direction [8, 7]. That change of direction depends on the ratio

[1] light does not propagate through metal because the electric field cannot penetrate inside a metal. Otherwise, this would move free electrons till it reaches an equilibrium where the electric field $\vec{E}$ would be null.

[2] In reality, the phenomenon leading to diffuse reflection is slightly more complex as diffuse reflection can occur with a smooth surface as well when light penetrates the material and then bounce back in random directions [11].

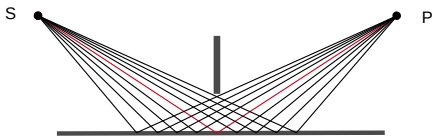


Figure 2.3: Light reflection according to Richard Feynman and QED theory. All paths are explored by light, but only the shortest paths -highlighted in red on this figure- contribute to the final result (as the majority of the paths interfering constructively are close to the shortest path), while the other paths interfere negatively. Figure is inspired from Feynman's lecture [12]

between the refractive index of the two mediums. Macroscopically, that change of direction is described by the *Snell-Descarte* relation

$$n_i \sin\theta_i = n_t \sin\theta_t. \tag{2.5}$$

When light goes from a less dense medium like air into a denser medium like water, it slows down and bends towards the normal of the plane separating the two medium. When light passes from a denser medium to a less dense medium, such as water to air, it speeds up and bends away from the normal. Refraction is responsible for many optical phenomena, such as how lenses bend and focus light to create images on the focal planes in cameras.

Propagation models. We described what happens at a sub-microscopical level, using a wave model, to explain macroscopical observations. Historically, several propagation models have been proposed to explain how light behaves and what path it takes. All these principles effectively predict the propagation of light at a macroscopical scale and have evolved throughout the centuries.

- *Fermat principle* (1662) describes the trajectory of light rays and states that the path taken by a ray between two points is the path that can be traveled in the *least time*. This general principle explains rectilinear propagation in homogeneous media, refraction, and reflection at the interface between two mediums [9].
- *Huygens-Fresnel principle* (1690) states that every point on a wavefront is itself the source of spherical waves. And the emitted waves from different points mutually interfere [7].
- *Electrodynamic theory* (1865). Electromagnetic theory gives a more complete description than the two previous frameworks and can, for example, predict the radiant flux transmitted, refracted, or reflected by a surface with Fresnel equations [7].
- *Quantum electrodynamic-QED (1948).* The Feynman path integral formulation replaces the classical notion of a single trajectory with an integral over an infinity of possible trajectories to compute a complex probability amplitude [12]. Within this framework, light does not take a single path. Instead, light takes all of them at the same time, but only the shortest path contributes to the final result while the other paths interfere negatively and hence do not contribute; see Figure 2.3 for an illustration. So, the system's behavior is once again coherent with the macroscopical observations and previous theories [12].

We showed in this Section that closed-form relations can be obtained for simple settings using the propagation models described above to predict the behavior of light. But for a complex scene, modeling light propagation becomes quickly non-tractable. One must instead rely on simulations to predict light behavior. Computer graphics covers such simulations [13]. A quick overview of these techniques will be discussed in Section 2.1.3.

2.1.3 Photometry

Photometry is the science that studies light measurement. The concepts introduced in this Section will help us understand the basics of ray tracing, numerical simulations, and how cameras gather light to form images.

First, we introduce four essential quantities that quantitatively describe light at a macroscopical level: flux, intensity, irradiance, and radiance. We then highlight several crucial light properties and introduce the rendering equation. We refer the reader to [14] for an interactive presentation that helps to grasp these concepts better.

Radiant flux. The radiant energy, denoted by the symbol Q, is the energy carried by EM radiations and is measured in joules $[J]$. The *radiant flux*

$$\Phi = \frac{\mathrm{d}Q}{\mathrm{d}t} \text{ in Watts } [W], \tag{2.6}$$

is the radiating energy emitted, reflected, transmitted, or received per unit of time. Note that these radiometric quantities can be characterized by their spectral power distribution: $\Phi(\lambda)$ is the radiated power spectrum [15]. It is a function of wavelength λ, which describes the amount of power at each wavelength. For the sake of simplicity, we will focus on monochromatic radiations in this Section. We will consider the radiated power spectrum in Section 2.2.5 devoted to colorimetry.

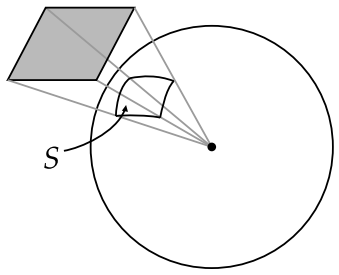


Figure 2.4: Solid angle is the ratio between the surface of a cone (non-necessarily circular) intercepted by a sphere centered at the origin of the cone and the square of the radius of that sphere.

Solid angle. Solid angle, in steradian $[sr]$, generalizes angles in 3 dimensions. It measures the amount of the field of view from some point that a given object covers (which is how large the object appears to an observer looking from that point). This is a useful tool for photometry. Solid angle is the ratio between the surface of a cone (non-necessarily circular) intercepted by a sphere centered at the origin of the cone and the square of the radius of that sphere; see Figure 2.4 for an illustration.

For instance, the solid angle Ω subtended by a hemisphere is 2π. More formally, it is defined as

$$\Omega = \frac{S}{r^2} \text{ in } [sr], \tag{2.7}$$

where r is the radius of the considered sphere and S is the intercepted spherical surface area. For a an infinitesimal surface $\mathrm{d}A$ the elementary solid angle $\mathrm{d}\omega$ subtended from a point at a distance r is:

$$\mathrm{d}\omega = \frac{\mathrm{d}A'}{r^2} = \frac{\mathrm{d}A\cos\theta}{r^2}, \tag{2.8}$$

where θ is the angle between the surface normal and the direction vector, from the point to the surface, $\mathrm{d}A'$ is the *foreshortened* area [11] as shown in Figure 2.5.

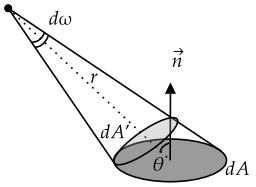


Figure 2.5: Infinitesimal solid angle.

Radiant intensity. The radiant intensity characterizes the amount of light radiation emitted from a source. It is defined as the power per unit solid angle emitted by a light source

$$J = \frac{\mathrm{d}\phi}{\mathrm{d}\omega} \text{ in } [W.sr^{-1}], \tag{2.9}$$

where $\mathrm{d}\phi$ is the infinitesimal radiant flux, $\mathrm{d}\omega$ is the solid angle. Note that radiant intensity can also be defined as the radiant flux reflected, or received by a surface. When computing the radiant intensity emitted by a source, $\mathrm{d}\omega$ represents the solid angle into which the light is emitted, whereas when determining the received radiance, $\mathrm{d}\omega$ corresponds to the solid angle subtended by the source as observed from the perspective of the detector.

Irradiance. Irradiance characterizes the radiant flux falling on a surface. This is a scalar-valued function that measures the amount of radiated power incident on a surface per unit area. It is defined for an elementary surface area $\mathrm{d}A$ at position $\mathbf{x}$ of normal $\mathbf{n}$ as

$$E(\mathbf{x}, \mathbf{n}) = \frac{\mathrm{d}\phi}{\mathrm{d}A} \text{ in } [W.m^{-2}], \tag{2.10}$$

where $\mathrm{d}\phi$ is the radiant flux received by the elementary surface area.

Inverse square law. Assuming light is emitting a flux Φ in a uniform angular distribution. Using the formula of the area of a sphere of radius r, we have $E = \Phi/4\pi r^2$. Irradiance decreases in $\sim 1/r^2$. The furthest from the source, the less power received per unit area. Note that this formula holds for a punctual light source but is not valid when the source cannot be considered as punctual.

Radiance. The radiance is the radiant flux emitted, reflected, or received by a given surface per unit solid angle per unit projected area. Less formally, it quantifies the light power traveling along a ray. Radiance is a scalar-valued function that is defined at a point $\mathbf{x}$ for a direction $\mathbf{r}$ as

$$L(\mathbf{x}, \mathbf{r}) = \frac{\mathrm{d}\phi}{(\mathrm{d}A \cos\theta)\mathrm{d}\omega} \text{ in } [W.m^{-2}.sr^{-1}], \tag{2.11}$$

where $\mathrm{d}\phi$ is the radiant flux emitted, reflected, $\mathrm{d}\omega$ is the solid angle[3], and $\mathrm{d}A\cos\theta$ is the projected area depending on θ, the angle defined between the normal and the ray. Figure 2.6 illustrates these parameters.

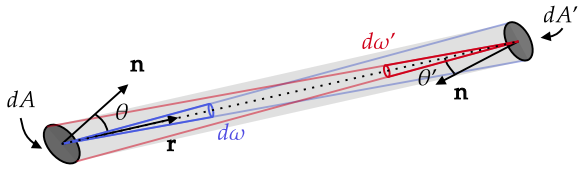


Figure 2.6: Parameters for defining the radiance. Radiance is the power emitted from a unit surface area $\mathrm{d}A$ in a set of directions $\mathrm{d}\omega$ or the power incident on a unit surface $\mathrm{d}A'$ from a set of directions $\mathrm{d}\omega'$. Note that in free space, these two radiances are equal in energy conservation, as we will discuss later.

Radiance integrals. Knowing the radiance distribution through space, we can use it to calculate the irradiance received for an infinitesimal surface patch, defined by its position in space $\mathbf{x}$ and its normal $\mathbf{n}$. The irradiance radiated to the surface is then the following integral [13]

$$E(\mathbf{x}, \mathbf{n}) = \int_{\Omega} L_i(\mathbf{x}, \mathbf{r}_i) |\cos\theta| \mathrm{d}\mathbf{r}_i \tag{2.12}$$

$$= \int_0^{2\pi} \int_0^{\frac{\pi}{2}} L_i(\mathbf{x}, \theta, \phi) \cos\theta \sin\theta \mathrm{d}\theta \mathrm{d}\phi. \tag{2.13}$$

where L_i is the incident radiance, θ is the measured angle between ω and the surface normal $\mathbf{n}$. Here,the $|\cos\theta|$ term is due to the definition of radiance. Note that irradiance is usually computed over the hemisphere of directions. We also wrote in equation 2.13 the integral over spherical coordinates because it is often more convenient to integrate over spherical coordinates instead of solid angle [13]. For integral over spherical (θ, ϕ)coordinates, we use the fact that

$$\mathrm{d}\mathbf{r}_i = \sin\theta \mathrm{d}\theta \mathrm{d}\phi. \tag{2.14}$$

Radiance properties. An essential property of radiance is that it is constant along a light ray in free space. This can be shown easily [11]: given two surfaces along a ray, as illustrated in Figure 2.8, we can write the emitted and received radiant power by two infinitesimal patch surfaces along a ray as $\mathrm{d}\phi_1 = L(\mathbf{x}_1)\mathrm{d}A_1\mathrm{d}\omega_1$

[3]Here, $\mathrm{d}\omega$ denotes the solid angle into which the light is emitted or the solid angle subtended by the receptor as viewed from the receiving surface depending on the context. Refering to Figure 2.6, the transmitted surface radiance is defined as $\mathrm{d}\phi/(\mathrm{d}A' \cos\theta')\mathrm{d}\omega'$

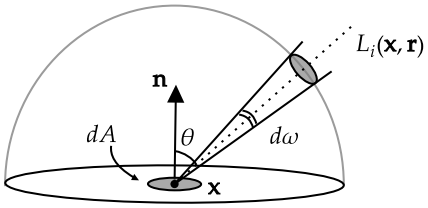


Figure 2.7: Irradiance at a point $\mathbf{x}$ is given by the integral over the hemisphere of radiance multiplied by the cosine of the incident direction angle θ.

and $\mathrm{d}\phi_2 = L(\mathbf{x}_2)\mathrm{d}A_2\mathrm{d}\omega_2$. Here,$\phi_1$ is the radiant power emitted from $\mathrm{d}A_1$ to $\mathbf{x}_2$, and ϕ_2 is the radiant power received by $\mathrm{d}A_2$ from $\mathbf{x}_1$. Since by conservation of energy, the radiance emitted by a source is the same as that received by a detector observing it we have $\mathrm{d}\phi_1 = \mathrm{d}\phi_2$. Furthermore, we can compute the solid angle subtended for each surface element as $\mathrm{d}\omega_1 = \mathrm{d}A_2/r^2$ and $\mathrm{d}\omega_2 = \mathrm{d}A_1/r^2$. Hence we have $\mathrm{d}\omega_1\mathrm{d}A_1 = \mathrm{d}A_1\mathrm{d}A_2/r^2 = \mathrm{d}\omega_1\mathrm{d}A_1$. Therefore we have $L(\mathbf{x}_1) = L(\mathbf{x}_2)$. We have shown that radiance is constant along a ray.

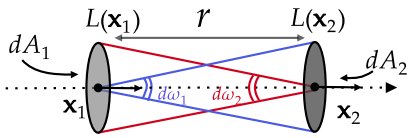


Figure 2.8: Radiance along a ray.

Lambert's cosine law. As mentioned in Section 2.1.2, an ideal Lambertian diffusing surface has equal apparent brightness regardless of the viewing angle. For a small surface relative to the observation distance, it obeys Lambert's cosine law that states that the radiant intensity J is proportional to the cosine of the angle θ between the receptor's direction and the surface normal [11]

$$J = J_0 \cdot \cos\theta. \tag{2.15}$$

Here, the $\cos\theta$ factor enforces a constant radiant flux per solid angle (i.e., radiant intensity) independent of the viewing angle. Figure 2.9 illustrates that principle.

Note that a Lambertian surface appears equally bright from all angles when illuminated by a light source. However, note that the amount of light energy the surface reflects depends on the angle between the surface and the light source because of the same cosine law. When the illuminated surface faces the light source directly, it reflects the most light energy, resulting in maximum brightness. As the angle between the surface and the light source increases, the amount of reflected light energy decreases, causing a decrease in brightness. This phenomenon can be observed with a sheet of paper, often modeled as a first approximation by a Lambertian diffuser. The paper appears equally bright by moving around it; however, brightness varies if the angle of the paper relative to the light source changes [11].

Lambertian diffuser reflected radiance. Note that Lambertian surfaces exhibit *uniform radiance* regardless of the viewing angle. The radiance remains constant as

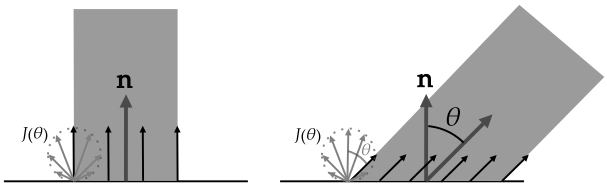


Figure 2.9: Lambert's cosine law: for the same solid angle (assuming here a far observer), a larger portion of the surface is visible for a smaller angle θ. The $\cos\theta$ factor enforces a constant radiated intensity (i.e., radiant flux per solid angle) independent of the viewing angle.

although the radiant power emitted from a particular area element is diminished by the cosine of the emission angle, the solid angle that the visible surface occupies is also decreased by the same factor. This constant ratio preserves the radiance [13]. This concept is depicted in Figure 2.10.

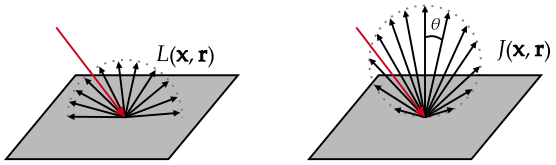


Figure 2.10: Diagram of Lambertian diffuse reflection with different units. The red arrow shows incident radiance. **Left:** The reflected radiance is uniform in all directions because its variation with the viewing angle cancels that of the intensity. **Right**: The black arrows show reflected radiant intensity J in each direction.

Radiance field. The radiance field, also called the plenoptic function, is a useful concept in computational photography and computer graphics. It describes the amount of light that passes through every point in a given direction for a portion of space. For example, views of a scene can be synthesized by sampling that function. In the general case, the radiance field $P(x, y, z, \theta, \phi)$ is a real-valued function $P : \mathbb{R}^5 \to \mathbb{R}$, of five variables. Three spatial coordinates x, y, z, and two angular coordinates θ, ϕ. We can add two more variables: time t to model the time variations and the wavelength λ to characterize color.

As we saw earlier, radiance along a ray remains constant if there are no blockers. Because the radiance along a ray is constant, the redundant information is one dimension. Therfore the radiance field can be represented with a four-dimensional function. The set of rays in a 4D radiance field can be parameterized in different ways. The two-plane parameterization $L(u, v, s, t)$ being the most common; see Figure 2.11 for an illustration. Refer to [16] for full coverage of this topic.

Reflectance. As we saw in the previous Section 2.1.2, the reflection of a surface is, in general, a combination of diffuse and specular reflection. The bidirectional reflectance distribution function (BRDF) is a standard tool to characterize how light

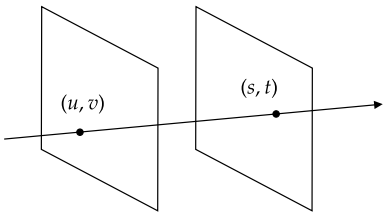


Figure 2.11: Radiance field represented with the two planes parameterization.

is reflected from a surface [13]. It is a positive function with

$$f_r(\mathbf{x}, \mathbf{r}_i, \mathbf{r}_r) = \frac{\mathrm{d}L_r(\mathbf{r}_r)}{\mathrm{d}E_i(\mathbf{r}_i)} = \frac{\mathrm{d}L_r(\mathbf{r}_r)}{L_i(\mathbf{r}_i)\cos(\theta)\mathrm{d}\mathbf{r}_i}. \tag{2.16}$$

The BRDF takes the incoming and outgoing light rays' directions $\mathbf{r}_i$ and $\mathbf{r}_r$ and returns the ratio of the reflected radiance to the irradiance incident on the surface from direction $\mathbf{r}_i$. Note that each direction $\mathbf{r}$ can be parametrized with elevation azimuth angles (θ, ϕ).

Rendering equation. Having defined the BRDF, we have enough tools in hand to introduce the rendering equation. The rendering equation is an integral equation -based on the conservation of energy- to evaluate radiance emitted from a point $\mathbf{x}$. This radiance is calculated as the sum of the emitted and reflected radiance. The full equation is the following

$$L_o(\mathbf{x}, \mathbf{r}_o) = L_e(\mathbf{x}, \mathbf{r}_o) + L_r(\mathbf{x}, \mathbf{r}_o) \tag{2.17}$$

$$= L_e(\mathbf{x}, \mathbf{r}_o) + \int_\Omega f_r(\mathbf{x}, \mathbf{r}_i, \mathbf{r}_o) L_i(\mathbf{x}, \mathbf{r}_i) |\cos\theta| \mathrm{d}\mathbf{r}_i. \tag{2.18}$$

This equation states that the outgoing radiance L_o at a specific position and direction equals the sum of the emitted light L_e by the surface and the total reflected light L_r. The reflected light is computed as the integral of the incoming light L_i from all directions Ω, multiplied by the surface reflection (characterized by the BRDF introduced earlier) and cosine of the incident angle due to the radiance definition [13]. Note that again for simplicity, we discarded the variables λ and t, assuming we consider static scenes and monochromatic radiations[4]. That equation is used in computer graphics to simulate the propagation of light on complex scenes. Although the equation is very general, it does not capture all aspects of light propagation, such as interferences, polarization, transmission, etc.

2.1.4 Pinhole Camera

Pinhole. The first device we introduce to measure light is the pinhole camera. It is a simple type of camera that is made without a lens. The basic design of a pinhole camera is a light-tight box with a small hole (the pinhole) on one side and on the other a light-sensitive material (such as film or a digital sensor)[16] as illustrated in Figure 2.12. Assume a scene is illuminated with a light source, which could be the sun or neon light; each point of the scene emits light to the imaging device. When the camera's shutter is opened, light enters through the pinhole and forms

[4]The full equation considers the spectral quantities indexed by time $L(\mathbf{x}, w, \lambda, t)$ and $f_r(\mathbf{x}, \mathbf{r}_i, \mathbf{r}_o, \lambda, t)$.

an inverted image on the opposite side of the box. A pinhole camera samples the light field through the (θ, ϕ) coordinates at a particular location (x, y). That model is not realistic for a real camera as a real camera needs more light and, hence, a larger aperture to form images with short exposure times. But it still has important interest for its practicality. It also turns out that primitive eyes were pinholes similar to the pinholes we have described [6].

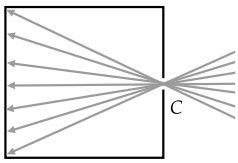


Figure 2.12: A pinhole camera measures radiance for rays with different directions passing through its pinhole.

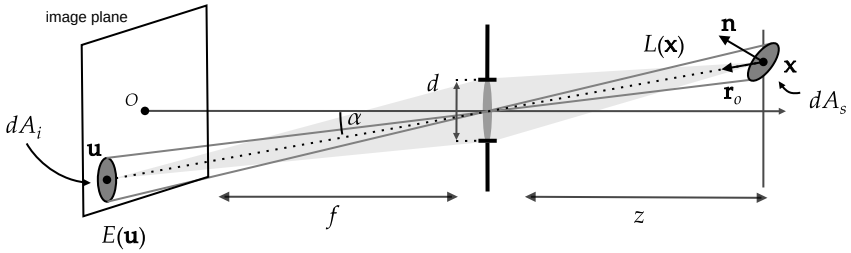


Figure 2.13: Relation between Image Irradiance E and Scene Radiance L for a finite aperture camera.

Measuring radiance. We show how radiance emitted from a scene is related to the irradiance collected on the image plane of the camera. Let us assume that an infinitesimal patch of surface $\mathrm{d}A_s$ is radiating light toward a camera of finite aperture d[5], as shown in Figure 2.13. The radiant flux received by the camera is fully projected onto the image patch $\mathrm{d}A_i$, assuming an infinitesimal aperture or that the camera is equipped with a lens. A fundamental property relates that for a pinhole camera with a finite aperture, the radiance of the ray is proportional to the irradiance on the image plane at the location of the projected ray [11] according to the relation

$$E(\mathbf{u}, \lambda, t) = L(\mathbf{x}, \mathbf{r}_o, \lambda, t) \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha. \tag{2.19}$$

Here, the ray of direction $\mathbf{r}_0$ from the object to the image plane and the optical axis of the camera form an angle α. See [11] for a full demonstration. This equation tells us that the irradiance that is measured by the sensor is directly proportional to the radiance of the scene. Therefore, cameras acquire a radiance map of the scene. We will cover in Section 2.2 how the irradiance is then further converted to an electrical signal and then digital numbers.

[5]Aperture is often measured in f-stops with $N = f/d$

Depth and brightness. Observe that the distance of the object to the camera does not appear in equation 2.19. Even if counterintuitive, it suggests that the image brightness of an object in the image plane does not vary with its distance to the receptor. This is because as the light source moves away, the total power radiated by each point from the object reaching the camera is divided by the square of a distance. However, simultaneously, the amount of visible surface sustained for the same solid angle (or that occupied the same field of view) increases by a square of the distance. Finally, these two effects cancel each other out perfectly, and the perceived brightness remains constant [11]. Figure 2.14 illustrates this phenomenon. This is consistent with observations: the apparent brightness of objects does not decrease when the object gets further away.

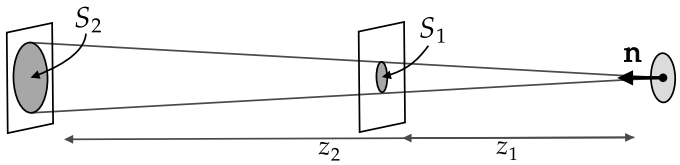


Figure 2.14: The apparent brightness of an object is not dependent on its depth to the observer. As the object moves away, the power radiated by each point of the object reaching the receptor is divided by the square of the distance. However, simultaneously, the amount of visible surface sustained for the same solid angle increases by a square of the distance.

Natural vignetting. The equation 2.19 shows us that image brightness falls off the image center as $\cos^4 \alpha$. We refer to this effect as *natural vignetting* as opposed to the vignetting effect that occurs due to optical systems (refer to Section 2.1.7). Note that for a small field of views, the effect of $\cos^4 \alpha$ is small [11].

2.1.5 Geometrical Model

We show in Section 2.1.4 that a pinhole measures the irradiance of the projected image formed on the camera's image plane. In this Section, we present the pinhole camera model -also known as linear projection- from a geometric perspective. We study the relationship between the position of a point in 3D and the coordinates of its projection on the two-dimensional image plane. We finally highlight several interesting effects in photography induced by this model.

Linear perspective. We assume a world coordinate frame. In this coordinate frame lies our camera with its own coordinate frame, with the z axis being aligned with the camera's optical axis and the origin being the camera center. The relation between a point in the camera coordinate frame $\mathbf{x}_c$ to its corresponding projection pixel coordinates $\mathbf{u}$ in the image plane is

$$\mathbf{u} = \pi(K\mathbf{x}_c). \tag{2.20}$$

Where K is the upper triangular *intrinsic* matrix

$$K = \begin{bmatrix} f_p & 0 & o_x \\ 0 & f_p & o_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.21}$$

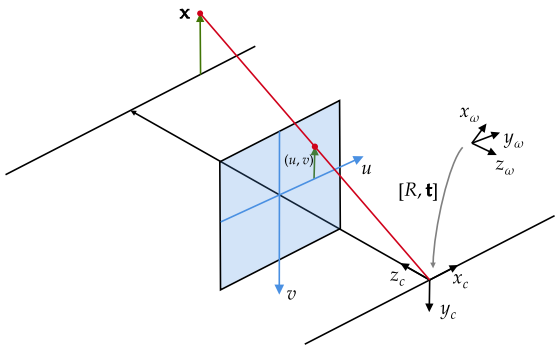


Figure 2.15: Pinhole camera model.

which is composed of the internal parameters of the camera. For a very rudimentary model, we consider the parameters f_p, where $f_p = mf$. Where m is the pixel density in x and y directions in pixels/m, assuming square pixels. And f is the distance of the image plane to the pinhole in meters. (o_x, o_y) is the coordinate of the optical center of the camera in the image plane in pixels. Finally, the perspective projection operator of $\mathbf{x} = \begin{bmatrix} x, y, z \end{bmatrix}^\top$ is written $\pi(\mathbf{x})$ with the relation

$$\pi(\mathbf{x}) = \begin{bmatrix} x/z \\ y/z \end{bmatrix}. \tag{2.22}$$

Now, suppose the position and orientation of the camera are known in the world frame. In that case, we can give a point in the world coordinate $\mathbf{x}_w$ and find its coordinates in the camera coordinate frame using the *extrinsic* matrix $[R|\mathbf{t}]$. We then have $\mathbf{x}_c = R\mathbf{x}_\omega + \mathbf{t}$. In summary, the projection $\mathbf{u}$ in the image of the point $\mathbf{x}_w$ can be found with the formula

$$\mathbf{u} = \pi\Big(K\big[R\mathbf{x}_\omega + \mathbf{t}\big]\Big). \tag{2.23}$$

More elaborate models can include non-linear deformations induced by the camera's optic, such as tangential and radial polynomial deformation models. We will cover lens aberrations in Section (2.1.6).

Perspective projection effects. This model, yet simplistic, has several interesting properties. First, projections of parallel lines in 3D stay parallel in 2D [17]. Second, the magnification factor -which is the ratio between the real object size and the size of its corresponding object formed on the image plane m is given by the formula [17]

$$|m| = \left|\frac{d_i}{d_0}\right| = \left|\frac{f}{z_0}\right|. \tag{2.24}$$

Where f is the focal length in meters, z_0 is the distance of the point to the camera. This can be derived easily derived with the Intercept theorem [17]. This dependence on the magnification and the object distance can introduce some well-known distortion effects among photographers. At large focal lengths (telephoto lenses), or when the scene is very far from the observer, the light rays are parallel, and the object looks flat. On the other hand, when the focal length is small (wide-angle lenses) or if the distance variations are significant, the object can look deformed [17]. As

there are disparities in distance to the camera relative to the focal length, it brings significant differences in magnification across the image. This effect is depicted in Figure 2.16.

Field of view and focal length. The field of view is related to the focal length and the size of the image sensor according to the relation

$$\alpha = 2 \cdot \arctan\left(\frac{h}{2f}\right), \tag{2.25}$$

where α is the field of view angle, h is the sensor size in meters, and f is the focal length in meters. We can see that decreasing the distance of the sensor to the optical center decreases the field of view. This imposes a burden on small camera devices: to keep the standard field of view, the sensor must be chosen smaller (this may alter the sensor's performance due to smaller pixels).

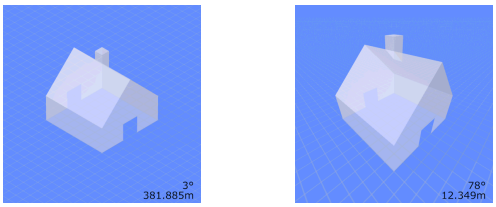


Figure 2.16: Comparison of images taken with different focal lengths. **Left:** Long focal length produces flatter object representation. **Right:** Short focal length causes distortion in the object's appearance. Image credit: Wikipedia.

Beyond linear perspective. The linear perspective was discovered in the fifteenth century by artists and architects. It is a straightforward technique for representing three-dimensional space on a two-dimensional plane and forms the basis of many imaging technologies like cameras and computer graphics engines [11]. However, studies have shown that it may fail to reproduce visual experience: artists rarely use it [18]. Alternatively, nonlinear methods have been propose to capture subjective visual experiences more effectively. Promising research directions adapt perspective to the content as it is done for the human eye [19, 20], just like tone mappers adapt contrast locally to the image content [21]. Refer to [22] for an introduction to this topic.

2.1.6 Lenses

Pinhole cameras require small apertures to form sufficiently sharp images. But small apertures limit the amount of light collected [7, 11]. Pinhole cameras, therefore, require long exposure times to capture images [11]. Lens cameras alleviate this problem by leveraging lenses to accumulate more light. To collect more light, lenses in cameras essentially "bend" the rays coming from a diffuser and focus them into one point in the image plane [7]. For this reason, dust on the lens of a camera does not deteriorate the image's quality, as it occludes only fractions of the rays gathered to form a point on the image plane. In this chapter, we explain the working principle of lenses. We then cover the thin lens model, a simple yet convenient model to understand lenses. We will then see that the advantages offered by lenses come at the expense of defocus and optical imperfections known as lens aberrations, which we briefly cover at the end of the chapter.

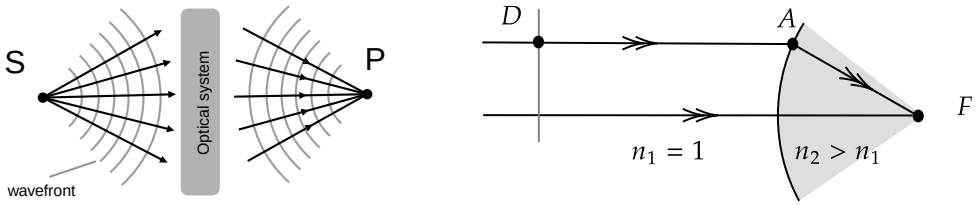


Figure 2.17: **Left:** A diffuser generates (spheric) waves, the optical reshape the wavefront so that they converge to P. **Right:** Surface of a lens with a hyperbolic interface between air and glass. The parallel rays from a planar wave converge to the focal point. The hyperbola is such that the optical path (i.e., taking into account slower speed in materials with large refractive indexes) from D to A to F is the same no matter where D is.

Lens surface equation. Lenses are typically made of glass with a well-chosen shape. When a point source diffuser is placed on an object's surface and is positioned far from the optical system with the sensor beneath the optics, the spherical waves emitted from the diffuser become nearly planar. Lenses are used to reshape this beam of waves, causing the wavefronts to converge at a single focal point F on the image plane (see Figure 2.17). To ensure constructive interference of all incoming waves at this point, the total optical path length through the two mediums must be the same. This imposes that

$$n_1\overline{DA} + n_2\overline{AF} = \text{const}, \tag{2.26}$$

with n_1 and n_2 being the refractive index of air and glass, respectively. By dividing the equation by n_1, we get the equation of a hyperbola [7, 9] of eccentricity e given by $n_2/n_1 > 1$. The greater the eccentricity, the flatter the hyperbola. So, the larger the difference in the refractive index, the less curved the lens. With a similar approach, we can show that for a mirror, the optimal shape is a parabola with eccentricity $e = 1$ as the refractive index does not change. Nevertheless, even though aspherical lenses are sometimes used in optical systems, most of the lenses have a spherical shape as it is way easier to design spherical lenses [7, 8]. Refer to [7] for more details regarding lens shapes.

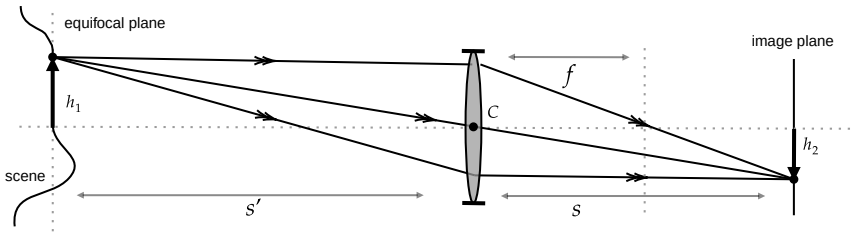


Figure 2.18: The figure illustrates the thin lens model, where s denotes the object distance to the lens, s' represents the image distance to the lens, and f represents the focal length. The sizes of the object and image are denoted by h_1 and h_2, respectively.

Thin lens model. The thin lens model is a simple yet widely used model for lenses. The model consists of a spherical refracting surface with negligible thickness [9] and assumes the parallaxial assumption. The parallaxial assumption states

that rays make a small angle θ to the optical axis and lie close to the axis. It allows the first-order approximation of the trigonometric function [23] such as $\sin(x) \approx x, \quad \cos(x) \approx 1$. With these approximations, we can derive with basic geometry the law governing image formation through a lens, known as the *thin lens law*

$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f}. \tag{2.27}$$

Here s is the object distance, s' is the image distance, and f is the focal length of the lens as illustrated in Figure 2.18. The so-called Lens maker equation can also be easily obtained. It gives the relation between the focal length of a lens to the refractive index n of its material and the radii of curvature of its two surfaces R

$$f = \frac{R}{2(n-1)}. \tag{2.28}$$

The ability of the lens to bend the rays is larger with large radii and with highly refractive glasses [9]. A consequence of the thin lens model is that the surface defining the corresponding set of perfectly focused points is a plane parallel to the sensor plane that we call the *equifocal* plane [11]. If the thickness of a lens is significantly smaller than the radii of curvature of its surfaces, it can be regarded as a thin lens [9]. By neglecting the optical effects resulting from the thickness of lenses, the thin lens approximation simplifies ray tracing calculations. Lenses with noticeable thickness are sometimes referred to as *thick lenses*. The thick lens model is presented in detail in [9].

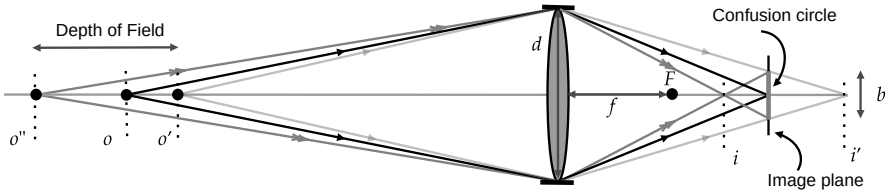


Figure 2.19: o' and o'' are the farthest and nearest distances defocused point for which the confusion circle is maximum, i.e., the formed image point is considered neat.

Defocus. Based on the thin lens models, given an optical system with a lens and an image plane at a distance i of the lens, only one plane at a distance o in front of the lens will be in focus. A point lying outside the image plane at a distance o' will form a blurry contribution to the image plane. If the point is closer to the lens than the focus plane, the light from the point will focus on a point behind the image plane. Following [17] the blur circle diameter b is

$$b = \frac{f^2}{N} \frac{|o - o'|}{o'(o - f)}. \tag{2.29}$$

Here, N is the f-number, which is given by $N = f/d$, where d is the diameter of the entrance pupil (aperture) of the camera. To focus an image plane, the system must move the image plane or the lens position to. the image plane. Therefore, the increase of aperture offered by lenses comes at the cost of a *defocus effect* -producing sharp images only for some portion of space in front of the camera.

Depth of field. DoF is the range of object distances over which the image is sufficiently well focused, i.e., the range over which blur b is less than pixel size. This is illustrated in Figure 2.19 The following simple formula can approximate the depth of field of a camera

$$\text{DoF} \approx \frac{2u^2Nc}{f^2}, \tag{2.30}$$

where c is the circle of confusion, f the focal length, N the f-number and distance to subject u. Note that this is an approximation of the exact depth of field formula, which can be found in [11]. From this equation, we see that three parameters impact the depth of field: aperture, focal length, and distance of the plane in focus. We summarize in the table below their impact on the depth of field.

Depth of field	Aperture	Focal	Distance
Increased (+)	Small	Large	Far
Reduced (-)	Large	Small	Near

2.1.7 Optical Limitations

In practice, real optical systems exhibit deviations from the ideal models discussed in the previous Section, primarily attributed to imperfections in the manufacturing process. This Section will explore the key limitations of optical systems used in cameras.

2.1.7.1 Diffraction

Optical systems have finite apertures, therefore a diffraction phenomenon will appear if the diameter D of the aperture is not much larger than the wavelength ($D \approx \lambda$) [7]. The light wave incoming through the aperture will be diffracted. With the hypothesis that the aperture is circular, a point does not focus on a single point but on a spread-out symmetric Airy disk, which can be derived analytically expressed with the Bessel function [9]. Diffraction is often considered the ultimate limit of an optical system in terms of the resolution of a finite-size imaging system. The theoretical resolution limit can be defined using the Sparrow or Rayleigh criterion. The latter states that for two closely placed punctual sources to be resolved, the central maxima of one's PSF should lie precisely at the first minima of the second one's PSF. If the distance is larger, the two points are resolved while they are considered as not resolved if the distance is smaller. This translates into the equation

$$\theta \approx 1.22\frac{\lambda}{D}. \tag{2.31}$$

where θ is the angular resolution (in radians), λ is the light's wavelength (m), and D (m) is the diameter of the lens' aperture. Factor 1.22 is derived from calculating the position of the first dark circular area of the diffraction pattern.

2.1.7.2 Lens Aberrations

The thin lens model serves as a first-order approximation to the behavior of real lenses. However, the real lens deviates from this model. These deviations from the thin lens model are called aberrations. Modern optical systems are always made of multiple lenses in order to correct these aberrations [9]. We present below the main type of aberration for lenses [8, 23].

Seidel aberrations. Seidel aberrations encompass a set of common lens aberrations that may occur in optical systems, leading to an inaccurate focusing of rays on the sensor plane [23, 7]. These aberrations can be accounted for by employing a third-order approximation of optics[23](whereas thin lens assumes a first-order approximation), which involves adding an extra Taylor series term to approximate trigonometric functions to obtain $\sin(x) \approx x - (1/3!)x^3$. The five types of Seidel aberrations are listed below. For further details on this topic, refer to [8, 23, 24].

- **Spherical aberration.** This distortion occurs when light rays passing through the edges of a lens focus at different points than those passing through the lens center, as shown in Figure 2.34, resulting in a blurred image [7]. This is due to the fact that a spherical lens is not the optimal shape for focusing rays, as shown in Section 2.1.6. This causes blurring and reduces the overall image sharpness.
- **Coma.** Off-axis points appear comet-shaped due to this type of distortion with large apertures.
- **Astigmatism.** Astigmatism results from differences in the focal lengths of a lens or mirror in different meridians (perpendicular planes). Instead of converging to a single focal point, light rays focus at different distances along two perpendicular axes. This leads to stretched or elongated images, particularly when dealing with high-contrast objects.
- **Field curvature.** Field curvature causes the image plane to be curved rather than flat. When a flat object is captured, portions of the image may appear out of focus or blurry because they are not lying on the same curved plane as the center of the image.
- **Radial distortion.** Radial distortion is a type of distortion that causes the magnification or shrinking of the image to vary at different radial distances from the center of the image. This results in image distortion, causing straight lines to appear curved.

Furthermore, we enumerate below an additional pair of aberrations frequently encountered within optical systems.

Chromatic aberrations. Another type of lens aberration is due to the impact of the wavelength of the radiation on the refractive index. Consequently, the optical system may exhibit different responses depending on the wavelength. Hence, the lens may have different responses for different radiation wavelengths [9].

Radiometric distortions. Finally, the last distortion category that we present in this Section alters brightness. The most common radiometric distortion is vignetting, which causes a darkening of the peripherical area of the formed image. There are several causes of vignetting: some light rays with significant incidence are more likely to be blocked by part of the lens system, and there is also a natural vignetting effect: recall the $cos^4\alpha$ fall of irradiance from the equation 2.19.

Optical designers employ various techniques to minimize aberrations, particularly by using different lens elements -compound lenses- made from specific materials to counteract these effects. Alternatively, software algorithms can be employed to correct these defects, as demonstrated in a recent study by Eboli et al. [25].

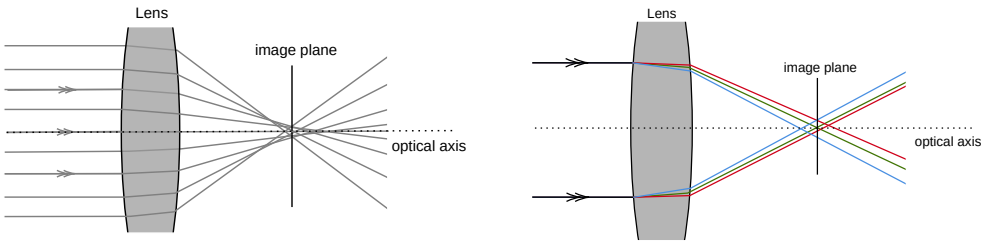


Figure 2.20: **Left:** Illustration of spherical lens aberration. Peripheral rays converge at different points than those passing through the center due to the spherical lens deviating from the optimal hyperbola shape [9]. **Right**: Chromatic aberrations are depicted here. Rays from different wavelengths focus at distinct points, exhibiting here both longitudinal and lateral aberrations [9].

2.1.7.3 Compound Lenses

In cameras, a system of multiple lenses arranged in a specific configuration is preferred over a single lens. This is because lens systems offer the flexibility to correct optical aberrations through careful arrangement [7, 9], and they provide increased options for focal length and zoom capabilities [9, 7]. The modeling of optical systems is typically done using proprietary models [26].

As an illustration, Figure 2.21 depicts a triplet lens system, which serves as a basic example of such compound lenses. These systems offer sufficient degrees of freedom to effectively reduce the Seidel aberrations discussed earlier.

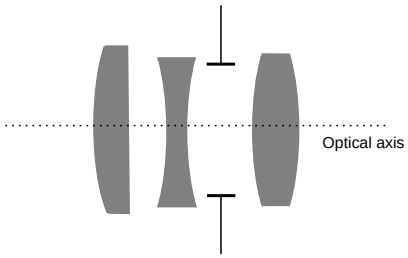


Figure 2.21: A triplet lens is one of the most simple types of compound lenses that consists of three individual lenses. Its design offers the lens designer the necessary degrees of freedom to effectively correct Seidel aberration in the lens [7, 9].

2.1.7.4 Extrinsic Source of Blurs

Other phenomena may alter the quality of the formed image by producing blurred images which are scene-dependent. Depending on the use case, these can become significant and be the main bottleneck that affects the resolution power of the device. We list only a few sources of blurs below for our quick tour.

Motion blur. Taking an image requires a certain exposition time that ranges from a few milliseconds to several minutes, depending on the scene. During this interval, both the camera and objects in the scene can move, resulting in motion blur in the image. Motion blur can be expressed as an integration of different views of the

moving scene over a time τ, as formulated by the authors of [27]:

$$y(\mathbf{u}) = \frac{1}{\tau} \int_{t=0}^{\tau} f(\mathbf{u}, t) dt, \tag{2.32}$$

where f is the image formed on the sensor plane. Motion blur becomes noticeable when the motion of the camera is faster than the camera's shutter speed [27]. This issue is commonly encountered, especially in night photography scenarios. To address this limitation, we introduce burst algorithms in Section 2.6 that help mitigate the effects of motion blur in images.

Atmospheric turbulences. The optical effects caused by atmospheric turbulence stem from variations in temperature and density, leading to fluctuations in the refractive index of the air [28]. These fluctuations can adversely impact long-range imaging systems by introducing random changes in the refractive index along the path of light rays perturbing the wavefront. As a result, these variations cause geometric distortion and result in space and time-varying blur in the captured images [28]. Notably, atmospheric turbulence poses a significant limitation for optical telescopes on Earth.

Having examined the primary sources of limitations for optical systems, the following Section will delve into the common methods used in the literature to model these imperfections.

2.1.8 Modeling

We now aim to model the degradation caused by real optics and establish the relationship between the irradiance image formed on the corresponding high-quality image forming on an idealized camera.

While the image formation model is inherently continuous, we focus solely on digital signals throughout this chapter. This decision arises from the impracticality of numerically reconstructing continuous representations of continuous signals. Hence, we work with high-resolution images sampled on a sufficiently dense grid, respecting the Nyquist criterion. This assumption is commonly employed in the literature (see [29]). For a more detailed discussion and a meticulous discretization of the continuous formation model, we direct interested readers to [30]. Further insights into the sampling process can be found in the dedicated Section 2.2.4. Furthermore, we present a continuous formation model inSection2.3.

Within this Section, we denote the digital signal of the degraded camera as $\mathbf{y}$ and the digital representation of the image captured by the idealized pinhole camera at the same location as $\mathbf{x}$.

2.1.8.1 Point Spread Function

In practice, all intrinsic blurs described in the previous sections co-occur[6], and it is convenient to model these defects with a point spread function (PSF).

[6]Non-ideal sampling blur due to photons integration on finite pixel area of the imaging sensor should also be taken into account, this is discussed in Section 2.2.4

Spatially variant filtering. The PSF describes the response of an imaging system to a point source, and in general, this can be a 4-D function of the spatial position of the source and its wavelength [8, 31]. A common assumption is that the PSF is invariant in the spectral band of each color filter. Therefore, the PSF is invariant for the three RGB channels [8]. Second, standard optical systems' surfaces vary smoothly, so the associated PSFs are also assumed to vary smoothly with the position in the image plane and object distance. Therefore, assuming the scene's depth varies smoothly, a common assumption is that PSFs are *spatially invariant* in a local neighborhood [8]. The formation model, including all intrinsic blurs, can be expressed as a local two-dimensional convolution on image tiles.

We denote by $\mathbf{x}$ the sharp image formed on the image plane of an ideal pinhole camera. We denote by $\mathbf{R}_i$ the linear operator that extracts a patch centered at position i, then $\mathbf{R}_i\mathbf{x} \in \mathbb{R}^{q\times q}$ is the square patch of the image $\mathbf{x}$ centered at position i. For a given tile location i and a given color, the PSF is a 2D convolution kernel of finite support that we denote by $\mathbf{h}_i$. The degraded version of $\mathbf{x}$ is the image $\mathbf{y}$, modeled as the averaging of n overlapping patches blurred with spatially varying PSF constants over the tiles. For a given color channel, the degraded image is

$$\mathbf{y} = \frac{1}{q}\sum_{i=1}^{n} \mathbf{R}_i^\top \left[(\mathbf{R}_i\mathbf{x}) * \mathbf{h}_i\right] = \mathbf{x} \star \mathbf{h}, \tag{2.33}$$

where $\mathbf{R}_i^\top$ is the linear operator that places a patch of size q at position i in an image and the operator $*$ denotes the convolution on 2d images. For clarity, we denote by $\star$ the 2d filtering with a set of spatially varying kernels note abusively $\mathbf{h}$.

Geometrical distortions. We can also model distortions induced by lens imperfections in our model. This effect can be directly modeled within the spatially varying PSFs. Geometrical distortions brought by lens defects lead to a shift of the center of mass of the PSF, especially in the peripherical region of the image [8]. However, it can be more convenient to decouple distortions from PSFs. To this end, geometrical distortions can be represented with a diffeomorphism $F : \mathbb{R}^2 \to \mathbb{R}^2$ of the focal plane into itself [32]. The resulting blurred and distorted version $\mathbf{y}$ of the clean image $\mathbf{x}$ is then expressed as

$$\mathbf{y} = \big(\mathcal{W}_F(\mathbf{x})\big) \star \mathbf{h}, \tag{2.34}$$

where $\mathcal{W}_F$ is a warping operator that deform the image $\mathbf{x}$ based on the mapping F.

Calibrating the PSF. For problems like deblurring or super-resolution, having an accurate estimate of the PSF is often a limiting factor. Relying solely on camera manufacturer-provided information for PSF estimation is often insufficient to create a precise model [33, 31]. PSF calibration involves estimating the PSF of the optical system. One approach is to use a calibration target with known characteristics, such as a grid of small dots [34] or a pseudo-random pattern [35]. The captured image of the target is then analyzed to estimate the PSF. It has been shown that using an appropriate pattern [35, 36] can turn PSF estimation into a well-posed inverse problem. However, estimating spatially varying kernels presents challenges, requiring multiple photographs with the pattern at various locations. Second, the PSFs are not fixed and can vary with many factors such as field heights, object distances, and wavelengths [33].

Limitations of 2d models. As we observed, the Point Spread Function is influenced by both the spatial position of the source and its color. While it is possible to estimate the PSF at different object distances, apertures, and wavelengths [33], certain complex phenomena, such as ray occlusions and multiple inter-reflections (as illustrated in Figure 2.22 and Figure 2.23), cannot be accurately simulated using the PSF alone [26]. Regrettably, these occlusions and inter-reflections have significant visual impact and are common in various scenes [26]. Due to the inherent limitations of 2D models, these phenomena require more faithful modeling of the optical system, necessitating the use of ray tracing simulations.

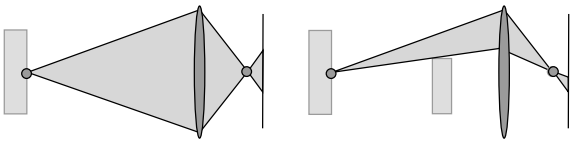


Figure 2.22: When a second object is introduced into the system, the local PSF undergoes changes because the occlusion of rays by the second object modifies the distribution of light received by the camera from the background object.

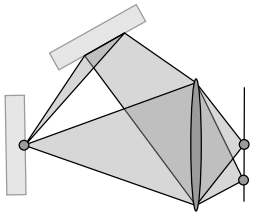


Figure 2.23: The point spread function (PSF) of an optical system is influenced not only by the direct path of light from the diffuser to the optical system but also by the light reflected from objects in the scene. This means that the presence of objects in the scene can alter the PSF, as light from the diffuser may be reflected by these objects and collected by the camera. Hence, the PSF can be influenced by the overall scene configuration.

2.1.8.2 Ray Transfer Matrices

To address the limitations associated with point spread functions (PSFs) and achieve more accurate simulations of optical systems, an alternative approach involves utilizing ray tracing to simulate images formed on the image plane [26]. Thanks to recent advancements in physically-based rendering and enhanced computational capabilities, it is feasible to realistically simulate complex three-dimensional spectral scenes [13].

Matrix ray transfer analysis. Characterizing the transfer response of an optical system can be achieved very effectively through ray transfer matrix analysis [7, 9]. This method utilizes 2x2 matrices to represent optical elements. As illustrated in Figure 2.24, a light ray enters an element crossing its input plane at a distance x_1 from the optical axis, in a direction that makes an angle θ_1 with the optical axis. After propagation to the output plane, that ray is found at a distance x_2 from the optical axis and at an angle θ_2 with respect to it. Under the paraxial approximation

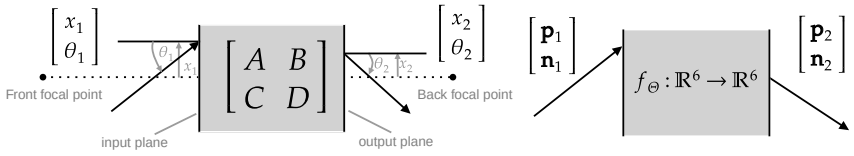


Figure 2.24: Modeling optical systems. **Left:** The ray-transfer matrix considers position x and angle θ relative to input and output planes [7]. **Right:** In contrast, ray tracing extends its capabilities by handling a 6-dimensional input, consisting of a position vector $\mathbf{p}$ and a unit direction vector $\mathbf{d}$, generating a corresponding 6-dimensional output comprising a position and direction vector [26]. Figure inspired from [26].

introduced in Section 2.1.6, these characteristic values of the output ray can be calculated with a 2x2 matrix for many optical elements

$$\begin{bmatrix} x_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_1 \\ \theta_1 \end{bmatrix}. \tag{2.35}$$

For example, in free space between the two input and output planes, the matrix is

$$S = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix}, \tag{2.36}$$

where d is the distance between the input and output plane. Another example is that of a thin lens of focal length f. Its matrix is given by

$$L = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix}. \tag{2.37}$$

A concise ray transfer matrix can be derived by multiplying these matrices, describing the entire optical system. For the example of two thin lenses of focal length f separated by free space of length d

$$LSL = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \tag{2.38}$$

With this technique, it becomes very easy to find the characteristics value of an optical system, such as focal length, front, and output focal point for a compound lens made of three, four, or more thin lenses. However, note that the technique described above uses the parallaxial assumption and requires that all rays have a small angle θ relative to the system's optical axis such that the approximation $\sin\theta \approx \theta$ remains valid. For a complete analysis to evaluate aberrations, full ray tracing should be performed using dedicated software [37].

2.2 Image Sensing

2.2.1 Sensing Irradiance

As we saw in the previous Section, optical modules convert radiance from a scene into an irradiance image at the sensor surface. Once the image is formed on the sensor's plane, the sensor measures the irradiance $E(\mathbf{u}, \lambda, t)$. Camera sensors are composed of millions of light-sensitive cells known as photodiodes arranged in

a grid-like pattern. Each photodiode captures the local intensity of the incident light and generates photo-electron charges through the photoelectric effect, with the number of charges being proportional to the intensity of the received radiation [31].

Formally, following the notation of Garnier et al. [38], the area A_{det} of the photosite detector spatially integrates the spectral irradiance, producing a spectral radiant flux Φ_λ in $W.m^{-1}$, according to the formula

$$\Phi_\lambda(\lambda, t) = \int_{A_{\text{det}}} E(\mathbf{u}, \lambda, t) \mathrm{d}A. \tag{2.39}$$

Finally, the signal for each photosite, after double integration over the time of exposure τ and across the light waveband from λ_{min} to λ_{max}, can be expressed as

$$s = \int_\tau \int_{\lambda_{\text{min}}}^{\lambda_{\text{max}}} \Phi_\lambda(\lambda, t) R(\lambda) \mathrm{d}\lambda \mathrm{d}t, \tag{2.40}$$

where $R(\lambda)$ is the spectral responsivity [38] or *gain*, representing the ratio of the detector output signal (in Volts or Amperes) to the incident spectral flux Φ_λ.

The obtained signal is then amplified by an amplifier, converted to a digital signal using an analog-to-digital converter (ADC), and further processed by the ISP chain we will cover in Section 2.4. Digital sensors have replaced photographic films due to their higher sensitivity and efficiency in collecting light. When photons strike the sensor's atoms, they create a local charge of electrons, with digital sensors converting approximately 50% of photons to electrons compared to only 5% collected by photographic films. Two technologies, CCD and CMOS, are used to read this local charge, with CMOS being the dominant technology in prosumer cameras.

In the following Section, we will explore the working principle of digital imaging sensors in more detail.

Photoelectric effect. The photoelectric effect occurs when a material emits electrons upon exposure to electromagnetic radiation, see Figurefig:photo, such as visible light or infrared. Electrons are emitted by the material at a specific frequency of light, with no emission occurring for radiation frequencies below a certain threshold [10]. The energy of emitted electrons is influenced solely by the wavelength of the radiation, not its intensity [39]. The wave interpretation of light does not account for these observations, and Einstein proposed an explanation by postulating that light consists of discrete particles called photons whose energy is directly related to their frequency. That earned him the Nobel Prize in 1921. This effect highlights the particle-like nature of light and serves as a fundamental principle underlying various applications, including photodetectors. Imaging sensors rely on the photoelectric effect to detect the flux of photons using *photodiodes*.

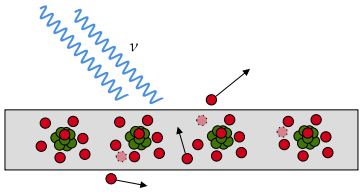


Figure 2.25: The photoelectric effect refers to the ejection of electrons from a material surface upon interaction with light. When incident light hits the material, electrons absorb energy from the photons and can be emitted if it has the minimum energy required [10].

In the upcoming paragraphs, we will explain the working principle of photodiodes. To begin, we will review fundamental concepts from the physics of semiconductors.

Semiconductors. In an isolated atom, the electron's energy is discrete and can only take predefined values [10]. In a solid, electrons' energy can take any value among certain intervals called *bands* separated by forbidden bands [10]. Repartition of electrons in the bands depends on temperature and obeys Fermi-Dirac law [10]. At a temperature of 0 Kelvin, the last fully occupied band is called the *valence band*. The next band above, which can be empty or partially occupied, is called the *conduction band*. The energy difference between the valence and conduction band is called the *gap*. Electrons in the valence band contribute to the local cohesion of the solid, while electrons with a higher energy state in the conduction band can move freely in the solid and be put in motion with electrical fields to generate current [10].

For conductors, these two bands overlap [39], whereas, for isolators and semiconductors, there is a band gap between these two bands. For semiconductors, such as silicon (Si), the gap is smaller, and some electrons can transition to the conduction band with additional energy from radiations or heat [39]. As we saw in the previous paragraph, when light strikes a material, it can stimulate electrons to transition to a higher energy level and make them go in the conduction band, where they can flow in the material. The freed electron will also leave behind a *hole*: a virtual positively charged particle. In semiconductor physics, we say it creates electron/hole pairs [39].

Photodiodes. Without anything else done, the electrons/hole will eventually recombine and therefore cannot be detected [8]. To detect photo-generated electron holes pair a solution is to generate an electric field, to move the charge carriers, and hence create a current that can be further detected. This could be done simply by polarizing a semiconductor (see photoresistors [39, 40]), but this is most efficiently done with a PN junction [39].

PN junction. PN junction is the combination of two types of semiconductors obtained by locally *doping* the semiconductor (a silicon crystal, for example). Doping refers to introducing different atoms in the silicon crystal. Two types of dopants exist: n-type dopant and p-type dopant. N-type dopant -for example, phosphor atoms- increases the electron density in the semiconductor, and p-type dopants -bore atoms, for example- increase hole density. It is crucial to note that after doping, semiconductors stay neutral [39]. When a p-region is put in contact with an

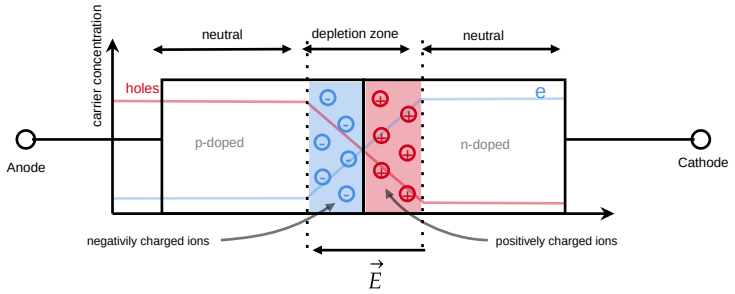


Figure 2.26: PN junction schematic view.

n-region, magic can happen: at the interface between the two regions, holes and electrons recombine thanks to a diffusion process [40]. This locally exposes positively and negatively charged ions on both sides of the region. This creates an equilibrium as the formed ions prevent electrons and holes from moving beyond a *depletion zone* [39].

Now, when a photon is absorbed in the depletion region, it stimulates an electron, and the freed electrons are pulled to the cathode due to the electric field created by the ions on both sides [40]. Hence, it creates a current in the circuit. The larger the radiant flux, the more current will flow through the circuit. This process is illustrated in Figure 2.26

It should be understood that the number of created electron/hole pairs is a nondeterministic process influenced by the material properties and the photon's frequency. The quantum efficiency η -note that η depends on the wavelength [8]- models its relation and is defined as the ratio between the ratio of electron-hole pairs created e to the number of received photons i

$$\eta(\lambda) = \frac{e(\lambda)}{i}. \tag{2.41}$$

For instance, photodiodes in CCD sensors can exhibit a quantum efficiency of well over 90% at specific wavelengths [39].

It is worth noting that it is possible to detect light with a semiconductor without a PN junction. If the light is emitted on a polarized semiconductor, the material's conductivity will increase, and a current variation can be detected. This kind of device is known as a photoresistor, and it is utilized when cost must be kept low and high sensitivity is not a primary requirement.

Full well capacity. The number of photons that can be detected depends on the size of the depletion zone and is called the *full well capacity*, typically about 10^5 electrons per pixel [39]. When high irradiance exceeds this capacity, the exceeding photons cannot be detected, and information is lost due to sensor saturation. The larger the photosite surface, the larger the capacity. Hence, larger pixels generally have a larger capacity than smaller pixels. Furthermore, to enhance the capacity, the photodiodes are *reversely biased* as the potential applied to the semiconductor will change the equilibrium and increase the surface of the depletion region.

Dark current: It is worth mentioning that electron/hole pairs can also be thermally generated within the sensor. These thermally generated electrons cannot be distinguished from the electrons converted from photons and will add to the overall generated current. This component of current is often referred to as the *dark current* of the sensor. Reducing the dark current can be achieved by cooling down the sensor [8].

Both of these aspects play a crucial role in defining the dynamic range of the sensor, which represents the ratio between the largest and smallest values that the sensor can measure. The dynamic range will be further explored in detail in Section 2.2.2.

CCD & CMOS sensors. Having described the working principle of photodiodes, we now review the two imaging sensors: CCD and CMOS. A minimalist imaging sensor consists of photodiodes connected to an amplification and conversion stage. However, using long wires -relative to photosite dimensions- in this setup leads to

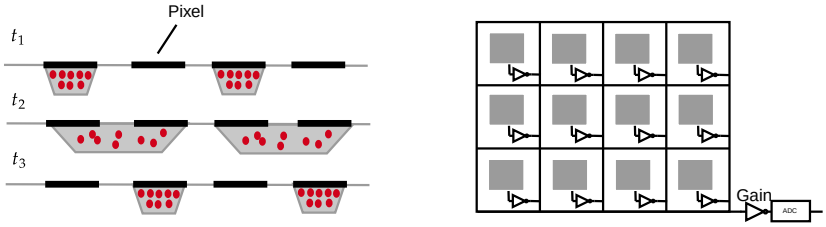


Figure 2.27: CCD and CMOS sensor working principle. **Left:** CCD sensor, charges accumulated on each of the pixels are moved step by step by passing through the neighboring pixels in order to be read sequentially by a single amplifier. **Right:** CMOS sensor locally amplifies the signal from each pixel with a dedicated amplifier. The signal is then read out by addressing the lines and columns. Figures inspired from [8].

low SNR and slow readout speed [8]. CCD and CMOS sensors are designed to overcome this challenge.

CCD sensors, In CCD sensors, the charges accumulate on a collection site for each pixel during the exposition phase (most CCD sensors use global shutters). Then, to be read out, charges are sequentially transferred to neighboring pixels and sent to a single output amplifier. CCD usually have better SNR and dynamic range due to large photon collection sites. But they generally suffer from slow reading time and large power consumption. See Figure 2.27 for an example.

CMOS sensors. CMOS sensors propose a different solution. A local amplifier is integrated into each pixel's collection site. Then, the amplified charges are read out by column and row addressing sequentially. CMOS is cheaper than CCD to produce, but CMOS may suffer from worst SNRs and dynamic range. Additionally, due to manufacturing variations in the semiconductors and local circuitry, every pixel has different gain and noise characteristics, resulting in fixed pattern noise. On the other hand, the addressing mechanism allows reading out sub-regions at high speed. See Figure 2.27 for an illustration of that type of sensor.

Rolling shutter. CMOS sensors typically capture one row at a time. The acquisition process looks like the following: a row at a time in sequence, pixels' charges are reset to zero, and photodiodes are biased and made sensitive. One exposure time later, those first sensitized pixels are read while others are exposed. This process is sometimes called a *rolling shutter*. The main advantage of rolling shutters is cost: they are much cheaper to manufacture because they need smaller memory buffers. But rolling shutters can cause problems when taking pictures of fast-moving objects or when the camera moves rapidly, as the resulting images can appear distorted. CCD sensor or global shutter CMOS does not have this problem. Instead, they accumulate charges for all photo sites of the image all at once and then read charges in a second phase.

Micro optics. Finally, another drawback of CMOS sensors is that the active circuitry for each pixel takes up some area on the surface of the photo site. This reduces the fill factor –the fraction of the pixel surface that converts received light– and then reduces the photon-detection efficiency of the device. To mitigate this problem, a layer of micro-lenses can be placed on top of the photodiodes to focus

the light flux onto the photosensitive area. Back-side illuminated (BSI) sensors can also mitigate this problem. This new generation of sensors places the circuitry to the back of the sensor, allowing more light to reach the photodiodes. These improvements not only improve photon-detection efficiency but also mitigate the occurrence of unwanted inter-reflections, also known as *cross talks*, within the sensor.

2.2.2 Noise Models

In the following sections, we will present a comprehensive sensor model that incorporates noise modeling, accounting for both saturation and noise effects induced by the sensor. Our model draws inspiration from the work of Emil Martinec [41] and the noise model introduced in [42].

Sensor model. Assuming a static scene[7] when a pixel is exposed to a photon flux Φ[8], in photons per second -also known as radiant flux- it accumulates $\Phi\Delta t$ photons during the exposure time Δt. The raw value s returned by the sensor is a linear function of the number of photons collected. The measured raw value s can be expressed as

$$s(\Phi) = \min\left(\frac{\Phi\Delta t}{g} + s_{\text{b}} + N, s_{\max}\right), \tag{2.42}$$

where s is expressed as a digital number (DN) [9]. Here, g is the sensor gain in photons/DN, which is inversely related to the ISO setting G[10] with the relation $G = U/g$, where U is a camera constant. s_{b} represents the black level of the camera, which is the data number at which the photosite receives no light [42]. The saturation value $s_{\max}$ (in DN) is reached when more than $(s_{\max} - s_0)g$ photons are collected. Note that the saturation level depends on the full well capacity of the pixel and the gain. It is important to note that the maximal digital value is usually set to be reached at a lower value than the analog saturation point of the sensor [43].

Lastly, the variable N represents a zero-mean random variable that captures noise introduced throughout the entire acquisition chain. In the following Section, we will review the main sources of noise for imaging sensors and present a widely used noise model.

Shot noise. The more intense the photon flux Φ received by a pixel and the larger the exposition time Δt, the more photons hit the sensor. For a scene with constant brightness, the number of photons arriving at a given sensor area will be random and fluctuate around an average value [44]. We model this random process with the random variable P expressed in DN. The number of arriving photons is then gP. The statistical modeling of such a random process is generally chosen as a Poisson law, which is considered a reasonable statistical model for most light sources [17].

An important characteristic of a Poisson law is that the variance is equal to the mean of the distribution. In other words, if, on average, 1000 photons reach the sensor for a shot, the count of photons will typically be in the range $\pm$ 100 for each

[7] The spectral radiant flux is assumed to be constant through time, i.e., $\Phi_\lambda(\lambda, t) = \Phi_\lambda(\lambda)$

[8] Here, the radiant flux is the integral over the color band B of the spectral radiant flux, $\Phi = \int_{\lambda\in B} F(\lambda)\Phi_\lambda(\lambda)\mathrm{d}\lambda$, where $F(\lambda)$ is the response of the colored filter. See Section 2.2.5 for more details on color sensing.

[9] digital number (DN) is the unit in which raw values are measured. It is also sometimes called ADU (analog-to-digital units).

[10] ISO is a camera setting which refers to the sensitivity of the camera's image sensor to light.

shot. The shot noise variance is

$$\mathrm{Var}(P) = \frac{\Phi \Delta t}{g}. \tag{2.43}$$

So, the shot noise variance grows with the light intensity. But interestingly, the signal-to-noise ratio (SNR), defined as $\mathbb{E}(P)^2/\mathrm{Var}(P)$ grows with light intensity Φ. Therefore, somewhat counterintuitively, lower illumination levels make the shot noise more apparent [42].

Read noise. After the exposure, the collected photo-electrons are converted to a voltage, which is then amplified, sequentially read, digitized, and possibly re-amplified for processing by the ISP [42]. In an ideal scenario, the digital number read for each pixel is directly proportional to the photon counts, according to the sensor gain, which serves as the conversion factor. However, in reality, all these steps introduce noise into the measurements. A common assumption is that the readout noise, represented by the random variable R, is independent of the signal intensity [44]. It can be modeled as a zero-centered Gaussian distribution with a variance σ_R^2 that depends on the ISO gain and characteristics of the sensor [44].

Relationship with ISO. Modeling the dependence of read noise on ISO is often helpful, especially when optimizing ISO, exposure time, and aperture settings during a burst [42]. The read noise is partly ISO-dependent because some readout noise is amplified alongside the amplification stage. A simple yet effective model assumes that read noise consists of two independent noise sources [41, 23]: pre-amplification and post-amplification noises. The variance of the sum of these two independent random variables is then given by:

$$\mathrm{Var}(R) = \left(\frac{\sigma_{\text{read}}}{g}\right)^2 + \sigma_{\text{ADC}}^2. \tag{2.44}$$

The first term represents noise from the sensor readout, while the last term combines quantization noise and amplifier noise. To determine these two linear coefficients, we can utilize the EXIF data from the camera, as read noise at a specific ISO setting is often provided in the raw file's metadata. However, it is worth noting that most modern cameras use multi-stage amplification schemes, rendering a linear fitting of the read noise/ISO curve inaccurate. In practice, more than two coefficients must be used for an accurate fit [41]. For further details on this topic, refer to [41].

Quantization error. After converting the photon flux to a voltage and amplifying the analog signal, the signal is converted to a digital signal by the ADC converter. During this process, a portion of the information is lost due to quantization errors, also known as *posterization*. A common noise model assumes that quantization errors are not significantly correlated and follow a uniform distribution. Under the assumption that the quantization step of the digital signal is Δ and is small relative to the variation in the signal being quantized [45], the variance of quantization error will be approximately $\frac{\Delta^2}{12}$. For modern sensors with resolutions ranging from 10 to 14 bits, this average quantization error of approximately 0.3 is often negligible in comparison to the read noise of the camera [41].

Other sources of noise. Below, we briefly discuss additional noise sources, even though they are often neglected when modeling sensor noise.

- **Pattern noise.** In real-world scenarios, read noise is often not spatially independent and may not be zero mean. As a result, patterns can be observed in the noise fluctuations. Although pattern noise might be a minor contribution to the overall noise, the human eye is sensitive to detecting structured patterns. To mitigate pattern noise, several black frames can be averaged [41].
- **Thermal noise.** Thermal agitation of electrons on pixels can liberate a few that are not distinguishable from electrons liberated from photons' arrivals. Hence, it is another source of noise. The variance of thermal noise is often modeled as a constant function through exposure time and temperature for a given camera. As exposure times are typically short for non-scientific applications, this noise source is often overlooked [41].
- **Pixel response non-uniformity (PRNU).** Due to the design of CMOS sensors (discussed in Section 2.2.1), not all pixels have the same efficiency in converting photons to data numbers. Consequently, there is non-uniformity in pixel responses, resulting in varying read noise for each pixel. Examples of such non-uniformities include hot pixels, which exhibit higher sensitivity than neighboring pixels, leading to quicker saturation, and dead pixels, which do not output any signal at all.

Noise model. Having introduced the various sources of noise, we can summarize them with a single noise model that considers only shot noise and read noise, neglecting other sources for digital sensors. The noise model is often simplified by assuming that both shot noise and read noise can be represented using a single zero-centered heteroscedastic Gaussian distribution

$$N \sim \mathcal{N}(0, \sigma_N^2). \tag{2.45}$$

These two noise sources, shot noise and read noise, are entirely independent, and their variances add up. For pixels below the saturation point, we have

$$\sigma_N^2 = \mathrm{Var}(P) + \mathrm{Var}(R) \tag{2.46}$$

$$= \frac{\Phi \Delta t}{g^2} + \frac{\sigma_{\text{read}}^2}{g^2} + \sigma_{\text{ADC}}^2. \tag{2.47}$$

The first term represents the shot noise and depends on the light intensity. The second and the last terms account for the read noise variance. The two terms, respectively, correspond to the post-amplifier and pre-amplifier noise. In this way, we can model the total noise variance for digital sensors in terms of shot noise and read noise contributions.

Signal-to-noise-ratio. Using the simplified noise model, we can derive the squared SNR for a pixel, denoted as $\mathrm{SNR}(\Phi)^2$, where Φ is the photon flux received by the pixel during exposure time Δt. The SNR is given by

$$\mathrm{SNR}(\Phi)^2 = \frac{(\Phi \Delta t / g)^2}{\mathrm{Var}(N)} = \frac{(\Phi \Delta t)^2 \cdot \mathbb{1}_{s(\Phi) < s_{\max}}}{\Phi \Delta t + \sigma_{\text{read}}^2 + \sigma_{\text{ADC}}^2 \cdot g^2}, \tag{2.48}$$

Where $\mathbb{1}_{s(\Phi\Delta t)<s_{\max}}$ models the case when the pixel is saturated, resulting in a null SNR.

The SNR increases monotonically with the number of photons collected, proportional to $\Phi\Delta t$ until it reaches saturation, where the SNR is reduced to zero. When graphing the SNR using the noise parameters of real cameras, we observe two phases. In the first phase, additive noise dominates, so the SNR increases linearly with Φ. In the second phase, shot noise becomes dominant, and the SNR increases with the square root of Φ [42, 44].

Choosing the exposure time involves a tradeoff between SNR and pixel saturation, disregarding motion blur. Longer exposures result in higher SNR, but they also risk saturating highlights in the image. Therefore, photographers must carefully balance exposure time to achieve the desired noise level and prevent saturation in important image regions.

Now that we have reviewed the noise model and discussed various noise sources, we have the necessary tools to delve into another key performance metric of imaging devices: dynamic range.

2.2.3 Dynamic Range

An essential aspect of imaging sensors is dynamic range, which accounts for the capacity of the sensor to capture details in a broad range of illumination. The human eye can distinguish details in a formidable range of brightness [6, 15, 46]. We give some examples of real-world luminance [11] values in the table below (from [31, 15]):

Scene	Luminance [cd.sr/m2]
Moonless Sky	$2 \cdot 10^{-3}$
Full Moon	$2.5 \cdot 10^{-1}$
Sunrise/Sunset	$4 \cdot 10^{2}$
Blue Sky	$2 \cdot 10^{4}$
Sunlight (zenith)	$1.2 \cdot 10^{5}$

The dynamic range may have different definitions depending on the context. From an engineering perspective, this is the ratio d between the highest value $s_{\max}$ that can be recorded by the sensor and the lowest discernable value that the sensor can read $s_{\min}$ [46]:

$$d = \frac{s_{\max}}{s_{\min}}. \tag{2.49}$$

This ratio is generally defined in stops, orders, or decibels [15]. We summarize commonly used formulas to express dynamic range in the table below:

Name	Formula	Unit
Log exposure range	$d = \log_2(s_{\max}) - \log_2(s_{\min})$	stops
Peak signal to noise ratio	$d = 20 \cdot \log_1(s_{\max}/s_{\min})$	dB

It is important to note that because sensors are noisy, the minimal discernable value is never zero. The minimum discernable signal is often chosen as the noise floor of the sensor. For a camera, it makes sense to take the sensor's read noise standard deviation as the minimal discernable value:

$$d = \frac{s_{\max}}{\sigma_{\mathrm{R}}}. \tag{2.50}$$

[11] Luminance, akin to radiance, accounts for the human eye's perception of light by weighting spectral radiance based on the eye's sensitivity to different wavelengths.

Practical dynamic range. The last definition of dynamic range may not always provide practical insights because the lowest discernable value is subjective and varies depending on the specific use context. For some applications, it can be more beneficial to consider a minimum discernable with a S/N ratio greater than one [41].

Sensor's dynamic range. As discussed in Section 2.2.1, the dynamic range of a camera's sensor is influenced by various factors, including the Full Well capacity and sensor noise. Increasing the Full Well capacity, achieved through larger pixel area, optimized doping, and improved detector properties, helps expand the dynamic range by reducing the risk of saturation in high-intensity scenes. On the other hand, readout noise affects the lower limit of the dynamic range. Employing denoising algorithms can also be effective in minimizing noise and thus extending the useful range of the sensor. Exposure bracketing is also often used to increase the dynamic range by capturing multiple images at different exposure settings and then combining them (as discussed in Section 2.6).

Human eye. The human eye excels in dynamic range, although comparing it to cameras is problematic [47]. The zone of visual acuity is narrow for the human eye [31], with a significant decrease in visual ability beyond the center. At the periphery, we only perceive large-scale contrast and minimal color. To construct a detailed mental image of a scene, our eyes rapidly focus on various regions of interest [31]. In scenes with a wide dynamic range, our eyes adapt quickly as our pupils adjust to different brightness levels. Some estimates suggest that the human eye can distinguish up to 24 f-stops of dynamic range [47], whereas most DSLRs offer a usable range of 5-9 f-stops. Regarding instantaneous dynamic range (with a fixed pupil opening), it is estimated that our eyes can perceive approximately only 10-14 f-stops [47].

2.2.4 Image Sampling and Aliasing

In the context of digital imaging, the irradiance image formed on the sensor plane is inherently *continuous* in space. However, camera sensors produce digital images by capturing discrete pixel values. This discretization process can lead to discrepancies between the continuous and digital representations. In this Section, we briefly explore the considerations associated with accurately representing continuous images through discrete digital versions through sampling. We refer the reader to the two thesis [30, 48] as excellent introductions to the topic.

Ideal sampling. We focus first on the case of ideal sampling of a continuous 1-D signal $f : \mathbb{R} \to \mathbb{R}$ for simplicity. We can model the sampling in equidistant steps by the product of the sampled continuous signal $f(t)$ and the Dirac comb [49]

$$y(t) = \mathcal{S}_T\{f(t)\} = \sum_{m=-\infty}^{\infty} f(t) \cdot \delta(t - mT), \tag{2.51}$$

where $m \in \mathbb{Z}$ and δ is the Dirac delta impulse

$$\delta(t) = \begin{cases} 1, & \text{if } t = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.52}$$

The resulting signal $y(t)$ is continuous and represents the discrete values $y[n]$ for $t = nT$. Let $F(\nu) = \mathcal{F}\{f(t)\}$ be the continuous Fourier transform of the signal f. Analysis of $y(t)$ in the Fourier domain shows that y is sufficient to obtain a periodic summation of $F(\nu)$ in the Fourier domain. Given appropriate conditions, on the sampling frequency that we will review in the next paragraph, it is then possible for the copies to remain distinct for a band-limited signal f.

Nyquist sampling. The quality of the reconstruction depends on the sampling frequency, which is defined as

$$\nu_s = \frac{1}{T}. \tag{2.53}$$

Assuming that the signal is *band-limited*, i.e., if there exists a cut-off frequency ν_0 such that the Fourier spectrum is null above that cut-off frequency, then according to the Nyquist-Shannon theorem, if the sampling frequency ν_s is chosen with $\nu_s > 2\nu_0$, it is feasible to fully recover f from its discrete samples $y[n]$. This means that in that setting, the sampling does not result in any loss of information [30]. And according to the Shannon-Whitaker theorem, the continuous signal can be reconstructed *exaclty* via sinc convolution [50]

$$f(t) = \sum_{n \in \mathbb{Z}^2} y[n] \cdot \mathrm{sinc}(t - n). \tag{2.54}$$

Undersampling with aliasing. In undersampling, if ν_s is chosen as $\nu_s < 2\nu_0$, aliasing occurs. A reconstruction of the clean signal from the discrete samples is ambiguous, and this can result in various artifacts (such as jagged edges, moiré patterns, or distortion for 2-D images) that do not accurately reflect the original signal [51, 30]. One solution to overcome this problem is to filter the measured signal using a low-pass filter [30] to erase high frequencies and alleviate aliasing issues. In the case of images, one other solution is to use several frames to super-resolve these aliased frequencies [52, 53, 54, 30, 29] we address this approach in Sections 2.6.

Discretization of 2-D images. The sampling theorem can be extended to functions of two variables such as images [30]. We can model the bi-dimensional sampling process of the continuous signal $f : \mathbb{R}^2 \to \mathbb{R}$ performed by the sensor array as

$$y(\mathbf{u}) = \mathcal{S}_\Delta\{f(\mathbf{u})\} = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(u, v) \cdot \delta(u - m\Delta, v - n\Delta), \tag{2.55}$$

where $m, n \in \mathbb{Z}$. Here, Δ is the pixel pitch which is assumed to be the same in both directions (u, v), and δ is the 2-D Dirac delta impulse such as

$$\delta(\mathbf{u}) = \begin{cases} 1, & \text{if } (u, v) = 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.56}$$

We denote by $\mathbf{y} \in \mathbb{R}^{M_u \cdot M_v}$ the vector representing the digitized version of the image f over the image domain $\Omega_f \subset \mathbb{R}^2$. Note that for convenience, the values $y(\mathbf{u})$ at pixel positions $\mathbf{u} \in \Omega_f$ are usually reorganized into a vector using a line-by-line scanning

$$\mathbf{y} = \mathbf{S}_\Delta\{f\} = [y(\Delta, \Delta), y(\Delta, 2\Delta), \cdots, y(M_u\Delta, M_v\Delta)]^T. \tag{2.57}$$

Here, Δ is the pixel pitch, which is assumed to be the same in both directions (u, v). In this manuscript, we will denote by $\mathbf{S}$ the bi-dimensional ideal sampling operator producing an image vector from a continuous signal f.

Real sampling. The sampling process was considered ideal such that a Dirac delta could be used to model the sampling operation. However, this assumption is not met in practice. In the case of an imaging sensor, the image is integrated over a finite pixel area, and this averaging process needs to be accounted for. Mathematically this is modeled by introducing a blurring kernel $h(t)$ with shape depending on the shape of the integration surface[12].

$$y(t) = \mathcal{S}_T\{f(t) * h(t)\}. \tag{2.58}$$

To model imperfect sampling and integration of the detector over a finite area, first, a filtered version of the signal is obtained based on the blurring kernel h, and finally, a sampled signal is obtained with an ideal sampler $\mathcal{S}$.

Sensor's resolution. According to the Shannon-Nyquist sampling theorem, the sensor must provide sufficiently high pixel density to avoid aliasing due to undersampling. Two scenarios can occur: optic-limited cameras and detector-limited cameras. In the first case, the resolution is limited by the sharpness of the image formed with the optic. In practice, this happens when the lens produces a relatively large lens spot size compared to the pixel size. In the second case, the sensor's resolution is not high enough to properly reconstruct the sharp image formed on the image plane. This occurs when the lens is sharp enough to produce a relatively small lens spot size compared to the pixel size. When enhancing the resolution, a distinction can be made between two methods: diffractive and geometrical approaches. Geometrical has the role of circumventing limitations of the sensor and reconstructing images at a finer sampling grid. In contrast, diffractive aims to alleviate the low pass effect of diffraction and other sources of blurs.

To conclude this Section, note that increasing pixel density or decreasing fill factor -the fraction of the pixel surface that converts received light- is not always the solution, as it also reduces the sensor's size. As discussed in Section 2.2.2, small pixels generally come with worse dynamic range and SNRs due to the reduced photon detection efficiency of the pixels.

2.2.5 Sensing Colors and Human Perception

In the upcoming Section, we will explore color perception and color sensing. While our earlier discussion mainly centered around gray images formed with monochromatic light, we will now delve into representing colors and faithfully reproducing the colors captured by a camera on display.

Light spectrum. Color is the sensation produced by the way objects reflect or emit light and is linked to its spectral repartition [17]. A light source can be characterized by its radiated power spectrum $\Phi_\lambda(\lambda)$, which indicates the power emitted at each wavelength. When such a light source illuminates objects, the spectrum of the reflected light is determined by the combination of the light source spectrum and the reflection spectrum of the illuminated object [15, 7]. Thus, the spectrum of the reflected light conveys information on both the light source and the illuminated object [15]. Analyzing the reflected spectrum provides valuable insights into the physical composition of illuminated objects.

[12] It's worth noting that in the case of imaging sensors, pixel's shape may not necessarily be square due to the circuitry on the sensor's surface (cf Section 2.2).

Color perception. Contrary to other stimuli such as soundwaves[13], the human eye is not able to decompose with great precision the power spectrum of a light source and distinguish the different monochromatic radiations composing it. Instead, the sensation of color comes from the stimulation of specialized cells called *cones* [6] [14]. Humans typically have three different kinds of cones, which are more excited for distinct parts of the light spectrum, but other species can have a different number of cones: the Mantis shrimp has 16 different cones [6]. Human cones are usually labeled as *short*, *long*, and *medium* cones, and their peak sensibility is blue, red, and green, respectively. Figure 2.28 shows the sensitivity of cones across the visible spectrum. The combination of the stimulation of the three different cones gives our brain color sensation. Different activations of the cone triplet will lead to different perceived colors. Hence, the generation of pink stimuli through monochromatic radiation is unfeasible, as it necessitates the concurrent stimulation of S and L cones while minimizing M cone excitation. Achieving this balance with a single-wavelength light source is impracticable due to the inherent spectral sensitivities of these cones, as depicted in Figure 2.28.

Metamerism. Another consequence of the last paragraph is that two light sources of different combinations of wavelengths can produce the same color perception. Despite differences in spectral power distributions, these two light sources can cause similar activation of LMS cones. This phenomenon is known as *metamerism*. And as we will explain, this phenomenon is in fact crucial in the color imaging pipeline to reproduce faithfully colors captured by a camera on display [15, 55].

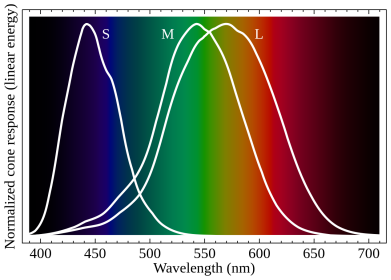


Figure 2.28: The graph depicts the sensitivities of human cones, with each curve representing the sensitivity of a specific cone to different wavelengths. The peak wavelength, found at the highest point of each curve, signifies the wavelength to which the cone exhibits the highest sensitivity. Image credit: Wikipedia.

Quantifying color. In colorimetry, accurately quantifying colors is vital to ensure color consistency in how objects appear to the human eye or when captured by a camera and rendered on display or in print. That task is challenging due to the subjective nature of color perception in our brains. Color spaces offer a valuable solution by providing a standardized representation of colors. Since the LMS cones'

[13]Humans can distinguish harmonics in a musical chord whereas, but cannot discern whether perceived color stimuli result from pure monochromatic radiation or a combination of multiple pure radiations. Differentiating whether perceived green arises from a combination of blue and yellow pigments or single radiation in the 500-578 nm range remains impossible!

[14]The Human eye is made of rods and cones; rods are predominately important in low light and don't play an important role in color vision [15], so we focus on cones for our analysis.

activation level entirely defines color, a color can be defined as a (L, M, S) triplet of the cones' activation levels. But the LMS color's representation has very unpractical properties: not all triplet values are physically possible, as stimulation of only one cone at a time is impossible [15]. And it would be problematic to build display hardware as one would need a way to stimulate each cone independently. Furthermore, cones' spectral sensitivities were discovered only recently. So, different representation systems are used.

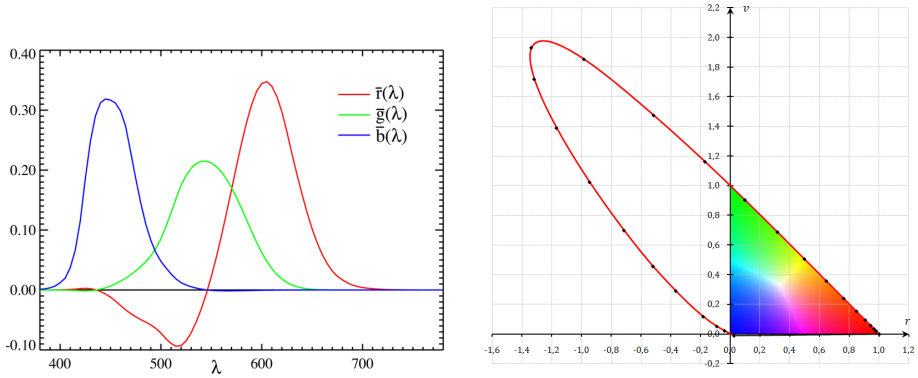


Figure 2.29: RGB colorspace. **Left:** Color matching functions. The CIE RGB color matching functions illustrate the proportions of primary colors needed to replicate a monochromatic color, with its wavelength indicated on the horizontal axis. **Right:** Rg chromaticity diagram. The red curve represents the different wavelengths in terms of chromaticity. Some wavelengths cannot be represented with the tree primaries of the RGB colorspaces. Therefore, they map to a region of space with negative components. Image credit: Wikipedia.

Color Matching Functions. One of the main results of colorimetry is that almost all colors may be reproduced by mixing light from three -suitably pure[16]- sources called *primaries* [15]. In the 20s, researchers proposed to define colors precisely from the mixing of three predefined monochromatic light sources with colors in the red, green, and blue range. In their experiments, a subject should match a target wavelength by mixing different amounts of the primaries. By recording the intensity to match each wavelength, three functions $\bar{r}(\lambda)$,$\bar{g}(\lambda)$ and $\bar{b}(\lambda)$ can be constructed. They are called the *color matching function* (CMF). They are plotted in Figure 2.29. Note that they can take negative values; this will be explained in the next paragraph. Thus, a light source of spectral distribution Φ_λ can be associated with a linear combination of three spectral sources as they produce the same stimuli:

$$\Phi_\lambda \equiv \bar{r}(\lambda)R + \bar{g}(\lambda)G + \bar{b}(\lambda)B, \tag{2.59}$$

where R, G and B are scalar values. The relation is a visual equivalence relation, denoted by $\equiv$. It means that the produced colors are equivalent but does not imply the equality of the two spectra (metamerism). Here, we will call the (R, G, B) triplet the *tristimulus value* of Φ. Note that for a given light source of spectral distribution Φ_λ

[15] this would be impossible to achieve the coordinate $(0, 1, 0)$

[16] Primaries with a pure power spectrum Φ_λ close to monochromatic lights can approximate a wider range of color. This can be seen on the CIE XYZ diagram that we will discuss in the next paragraphs

the tristimulus values may be obtained by integration along the visible spectrum Λ:

$$R = \int_{\lambda \in \Lambda} \Phi_\lambda \bar{r}(\lambda) \mathrm{d}\lambda, \tag{2.60}$$

$$G = \int_{\lambda \in \Lambda} \Phi_\lambda \bar{g}(\lambda) \mathrm{d}\lambda, \tag{2.61}$$

$$B = \int_{\lambda \in \Lambda} \Phi_\lambda \bar{b}(\lambda) \mathrm{d}\lambda. \tag{2.62}$$

The CMF leads to the CIE RGB colorspace proposed in 1931 by the CIE[17].

Color substractions. It is worth noting that color-matching functions (CMFs) may exhibit negative values, as not all colors, including monochromatic lights, can be fully represented using a combination of these three primary colors. To address this limitation, an additional color contribution was introduced to the target wavelength being reproduced. This addition can be thought of as a color subtraction, resulting in negative components in the CMFs [15]. This approach ensures that all parts of the visible spectrum can be decomposed into three primaries, even if some have negative components. Alternative color spaces are often used to mitigate this issue, as we will see below.

Finally, it is important to acknowledge that the color-matching functions (CMFs) are relative to each individual, and there is no guarantee that cone sensitivity is exactly the same for everyone. To address this variability, the authors proposed conducting experiments with several subjects and then averaging the results. However, in the study, only a few individuals from the same age and ethnicity group were included.

XYZ colorspace. To make color calculations more practical, the Commission Internationale de l'Éclairage (CIE) introduced the XYZ colorspace, which ensures that all visible color values are positive within the range [0, 1] on all the axes. It is derived from the RGB colorspace through a carefully chosen linear transformation [55]. However, to accomplish this, the XYZ colorspace introduces virtual colors that do not exist in the real world, often referred to as *imaginary* colors. These additional colors facilitate the mapping of all chromaticities within the desired range.

Chromaticity diagram (r,g) The XYZ colorspace is associated with tristimulus values (X, Y, Z). From these values, chromaticity coordinates (x, y, z) can be calculated, where:

$$x = \frac{X}{X + Y + Z}, \tag{2.63}$$

$$y = \frac{Y}{X + Y + Z}, \tag{2.64}$$

$$z = \frac{Z}{X + Y + Z} = 1 - x - y. \tag{2.65}$$

Since z is known if x and y, only (x, y) coordinates need to be kept. Chromaticity coordinates (x, y) characterize a color's chromaticity independently of its luminance. These 2-dimensional coordinates can be plotted on a chromaticity diagram. In this

[17]Comission Internationale de l'éclairage: the international authority on light, illumination, color, and color spaces.

diagram shown in Figure 2.35, all monochromatic light maps to a position along the curved boundary which is called the *spectral locus*. Three primaries will map to three points on the diagram. The triangle formed by the three primaries contains a range of colors that can be reconstructed with the primaries and is called the *gamut*.

Gamut. The gamut of a color space is the range of colors that can be represented within that color space. This is the boundary of all possible values that can be captured or reproduced by the device. Chromaticities out of the gamut cannot be measured or reproduced by the device. For instance, a laptop screen can only reproduce a fraction of all existing colors.

Color spaces. Various devices such as cameras, monitors, scanners, TVs, and projectors use their unique sets of primaries to represent colors. Consequently, each device employs its specific color space, determined by factors like RGB color filter array spectral responsivity for cameras, LED color spectra for displays, or the three phosphors of a CRT display [15, 55]. Conversion between these color spaces is essential to ensure the faithful reproduction of colors. If the spectral responsitivity of the camera filters are known, as well as the emission spectra of the photo elements of the display, we can specify a transformation between a tristimulus value of the camera and a tristimulus value of the display and thus reproduce the same colors. This is an important consequence of *metamerism*.

Color spaces conversion. We can convert from one tristimulus colorspace to another tristimulus colorspace using 3×3 matrix color transformation. For example, converting a color captured with specific tristimulus values to the XYZ colorspace is performed using a matrix transform of the following form

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{2.66}$$

where $[X, Y, Z]$ are the desired CIE XYZ tri-stimulus values, $[R, G, B]$ are the tristimulus values obtained from the device. Here, the 3x3 matrix is the measured tri-stimulus values for the device, where $[Xr, Yr, Zr]$ $[Xg, Yg, Zg]$, $[Xb, Yb, Zb]$ are the measured CIE XYZ tri-stimulus values for the three channels, respectively, at maximum emission. Conversely, to convert from XYZ to the RGB colorspace, we can use the inverse form of the matrix

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{2.67}$$

However, note that in many applications, measuring the tri-stimulus values for the device is impossible as it requires difficult and expensive calibrations [55]. Usually, the primaries of the device are only known by the xy chromaticity coordinates, a white point $[x_w, y_w]$, and the maximum luminance Y_w of the device. In this case, it is possible to obtain the color transformation matrix by solving a 3×3 linear system [55, 15].

Sensing colors. When it comes to imaging sensors, photosites are sensitive to light across a broad spectrum, including nonvisible infrared, leading to the emission of electrons. A solution to capture colors is to emulate the human eye by placing

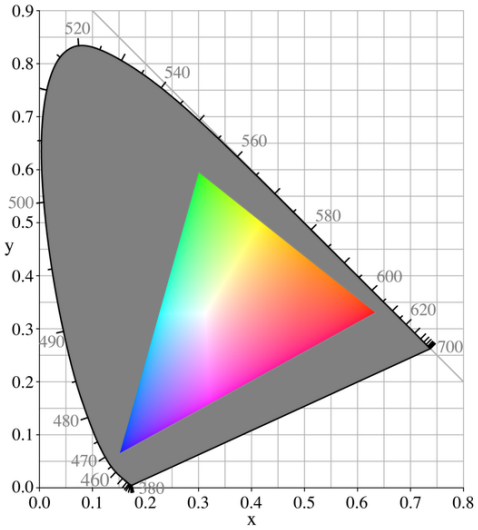


Figure 2.30: xy chromaticity diagram. Inside are represented the three primaries used to define the RGB colorspace. The triangle encompasses colors that can be reproduced with the three RGB primaries. Image credit: Wikipedia.

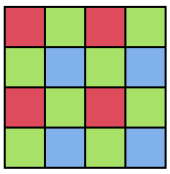


Figure 2.31: Color Filter Array (CFA): Each pixel is covered by a colored filter based on a specific pattern. Demosaicking algorithms are necessary to reconstruct the complete image (including the two missing colors for each pixel). Consequently, the sensor does not capture 2/3 of RGB images.

a bandpass color filter in front of each photosite, selecting either the light's red, green, or blue component. The frequencies of light reaching the sensor depend on the pixel's location, following a specific pattern. One commonly used pattern is the Bayer pattern, which is prevalent in most cameras. A visual representation of this system is illustrated in Figure 2.31. However, this method means that each pixel in the final color photograph needs to be reconstructed through a *demosaicking* step to recover the missing color information. Consequently, approximately 2/3 of the pixels in a colored image are not directly captured by the sensor but instead reconstructed using interpolation algorithms.

Gamma encoding and gamma display. Colorspaces often use gamma correction, a non-linear transform. Our eyes perceive light differently from cameras. Unlike cameras, where twice the photons yield twice the signal (linear relationship), our eyes perceive twice the light as only slightly brighter, especially at higher light intensities (non-linear relationship). Compared to cameras, we are more sensitive to changes in dark tones than in bright tones. In general form, gamma encoding has the form

$$I_{\text{out}} = A I_{\text{in}}^{\gamma}. \tag{2.68}$$

Gamma-encoded images efficiently store tones by aligning tonal levels with how our eyes perceive them. Gamma encoding reduces the need for many bits to describe a given tonal range. Without this encoding, too many bits would be used for brighter tones (where cameras are more sensitive) and too few for darker tones (where cameras are less sensitive). Finally, it is important to note that to view a gamma-encoded image accurately. Gamma correction is needed. This process converts the image back into light from the original scene by applying inverse operation, aligning it with human vision. This correction is sometimes called gamma correction. For further understanding, we recommend exploring additional resources [15].

2.3 Image Formation Model

In this section, we present a comprehensive formation model that takes into account optics and imaging sensors, serving as a summary of the preceding sections. The model utilized in this thesis is based on the seminal work by Elad et al. [56], which has been widely employed in numerous super-resolution algorithms. We present here a refined model, drawing inspiration from Delbracio's work [32].

Analytical formation model. Let f represent the 2-D irradiance image of the 3D world projected onto the image plane of an ideal pinhole camera. For each color channel, the entire image formation process can be succinctly described by the following equation

$$\mathbf{y} = \mathbf{S}\Big\{ g\big((f * h_{\text{extrinsic}}) \circ F\big) * h_{\text{intrinsic}} \Big\} + \mathbf{n},$$

where F is geometric distortion map on the focal plane into itself, $h_{\text{intrinsic}}$ is the blurring kernel varying smoothly across the image plane that accounts for intrinsic blur (such as the pixel integration, diffraction, lens aberrations), and $h_{\text{extrinsic}}$ is blurring operator accounting for extrinsic blurring effects happening outside the camera (atmospheric turbulence). g is a monotone increasing function describing each pixel's non-linear response, the operator $\mathbf{S}$ is an ideal 2-D image sampler modeling the sensor array, and last, $\mathbf{n}$ models sensor's noise. Please note that we consider each color channel independently. For instance, aberrations and diffraction are assumed to vary for each band [8].

Such a model will prove useful in the subsequent chapters as it accounts for degradations induced by non-ideal digital cameras, enabling the recovery of high-quality images by solving inverse problems.

2.4 Camera Imaging Pipeline

After describing the hardware aspects of digital cameras, we continue our tour by focusing on the software and the classical processing techniques commonly used to generate high-quality and aesthetically pleasing images.

In photography, the ISP (Image Signal Processor) is the specialized software part responsible for processing the image data captured by a camera sensor. The ISP performs various tasks to optimize the quality of the final image output. It varies for each device depending on the task the device is specialized for, and each manufacturer has its own pipeline. Making an exhaustive presentation impossible to achieve. We review here the main stages.

Automatic camera settings. Before the image is taken, some parameters are usually automatically set in real-time [57]. *Auto Exposure* adjusts the exposure settings (shutter speed, aperture, and ISO) to achieve proper exposure for the scene and determines the amount of light that the sensor will collect. *Auto Focus* adjusts the lens focus settings to ensure that the subject is in sharp focus. *Auto white balance* adjusts the color of the image to ensure an accurate representation of colors by making white objects appear neutral regardless of the lighting conditions. More on white balancing later.

Frame acquisition. After the camera has been set, images can be taken. During exposure, photons hitting the sensor accumulate electrons, which are converted to a numerical signal. The acquired signal is generally a 10-12-bit signal of raw sensor data, which needs to be processed by the camera processor to produce an image interpretable by the human eye. The processing steps of a simplistic ISP are depicted in Figure 2.32. We describe each of these steps below.

Figure 2.32: Simplistic ISP for raw image processing.

Linearization. In general, the raw measurements are not in the $[0, 1]$ range and have an offset, called the *black level*) and a scaling, sometimes called *white level*. The first step of the ISP generally consists of normalizing the measurements in the range $[0, 1]$ by applying affine transformations to each raw measurement as follows [58]

$$\text{linear} = \text{clip}\left(\frac{\text{raw} - \text{black}}{\text{saturation} - \text{black}}, 0, 1\right). \tag{2.69}$$

Pixels below the black level and above the saturation level can be clipped but are sometimes kept as it generally helps for denoising in the dark regions (clipping complicates noise modeling in dark regions). Note that each Bayer channel may have different black and saturation levels.

Lens shading correction. A lens shading correction is commonly employed to address the radial decrease in light reaching the sensor caused by vignetting effects. This non-uniformity is typically rectified using a calibrated shading mask [57].

White balancing. White balancing involves correcting for color differences caused by the lighting conditions in which the image was taken. White balancing adjusts the colors in an image to ensure that white objects appear white without any unwanted color casts. The purpose is to imitate the chromatic adaptation ability[18] of the visual system to adjust to the dominant illumination of a scene [59].

White balance requires an estimate of the sensor's response to the chromaticity dominant light source $[l_r, l_g, l_b]$. This response can be pre-calibrated for diverse illumination conditions (sunlight, neon light, etc.) or estimated by an auto-white

[18]Human eye can adapt to the chromaticity of the dominant light source, in particular to white illuminant of various temperatures. If the light source is changed gradually, the human eye will adapt and perceive the same color [15]. This is sometimes called *color consistency* in the literature.

balance algorithm [57]. Note that since only the chromaticity of the color matters, one of the channels, usually the green one, can be set to 1 [58]. Once scene dominant illumination has been estimated, the image is white-balanced by dividing each channel by the corresponding channel of the illuminant to emulate a neutral illuminant as follows

$$\begin{bmatrix} r_{\text{wb}} \\ g_{\text{wb}} \\ b_{\text{wb}} \end{bmatrix} = \begin{bmatrix} 1/l_r & 0 & 0 \\ 0 & 1/l_g & 0 \\ 0 & 0 & 1/l_b \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix}. \tag{2.70}$$

See Figure 2.33 for an illustration of white balance with different color temperatures.

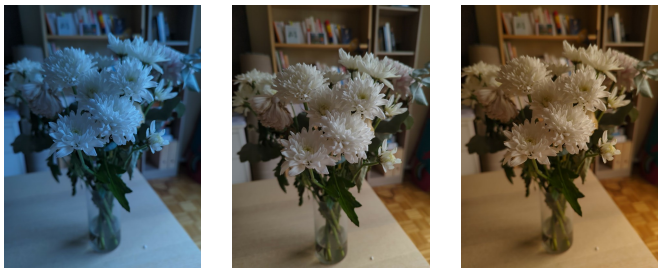


Figure 2.33: White roses shot white balanced with various settings. The correct setting is in the middle. Pictures were taken with a Pixel3a camera.

Demosaicking. Demosaicking is the process of reconstructing a full-color image from the incomplete color information captured by a camera sensor equipped with a filter array (introduced in Section 2.2.5). The demosaicking algorithm uses interpolation techniques to estimate the missing color values for each pixel, resulting in a full-color image. It is somewhat counter-intuitive to realize that 2/3 of images are generated by interpolation and do not correspond to true measurements.

Color correction. As briefly explained in Section 2.2.5, the spectral sensitivities of the camera primaries (the red, green, and blue color filters) are specific to a particular sensor. Because of this, it is necessary to convert these sensor-specific RGB values to a device-independent color space, such as CIE XYZ colorspace or the sRGB colorspace (one of the most commonly used colorspace). We can convert each color pixel triplet $\mathbf{lin}_{\text{rgb}} = [r, g, b]^\top$ to the right colorspace by using the proper (and normalized) color transformation matrix $A_{\text{cam}\to\text{sRGB}}$ as follows

$$\mathbf{lin}_{\text{srgb}} = \text{clip}\left(A_{\text{cam}\to\text{sRGB}} \cdot \mathbf{lin}_{\text{rgb}}, 0, 1\right). \tag{2.71}$$

Tonemaping. Finally, tone mapping (TM) is an essential step in the image processing pipeline. It helps ensure that the final image accurately represents the original scene while maintaining a natural and visually appealing appearance. The TM algorithm applies various techniques to adjust the brightness, contrast, and color saturation to produce an image that looks visually pleasing and natural on the output device. For instance, a simplistic tone mapper can apply a global gamma correction to brighten dark regions. In contrast, more elaborate tone mappers mimic the human visual system by locally adapting brightness and contrast to the content of the image [15, 46].

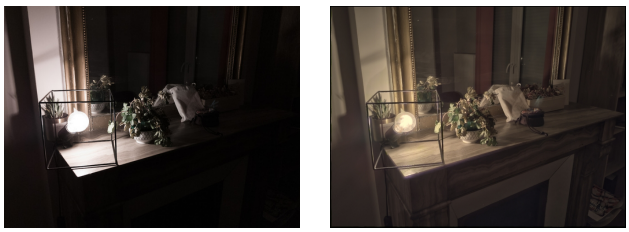


Figure 2.34: **Left:** Unprocessed image saved as jpg without tone-mapping, resulting in lost details due to saturations or obscured areas in shadows. **Right:** Enhanced image following the application of content-aware tone-mapping, locally adjusting brightness and contrast, revealing previously hidden details in the raw image. Picture taken with a Pixel3a camera.

2.5 Algorithms for Image Restoration

In the previous sections, we explored a minimalist ISP pipeline. Now, we delve into the heart of this thesis, focusing on algorithms designed to enhance image quality. While methods focusing on aesthetic improvements in photographs are a fascinating topic, we focus solely on "quantitive" improvements in terms of resolution, SNR, and dynamic range, which we denote as image restoration algorithms. Such improvements -with no hallucinations[19]- hold value in scientific and medical applications as they enable capturing images with more information. Our exploration begins with an introduction to inverse problems in computational imaging. We then delve into data-driven approaches, often based on deep learning techniques. Lastly, we show how to combine these two families of approaches to get the best from both worlds. The core works of this thesis are based on this class of methods.

2.5.1 Inverse Problems

We consider the task of estimating an unknown high-quality[20] image $\mathbf{x} \in \mathbb{R}^n$ from its noisy measurement(s) $\mathbf{y} \in \mathbb{R}^m$ with a camera. We assume that we can represent the camera's behavior with a forward model, which is the operator characterizing the response of the imaging sensor. See 2.3 for an example of a camera forward model. It is common to address this problem with optimization. We refer to this as an inverse problem.

Forward model. In computational imaging, a common assumption is that the system can be represented by a linear operator A. The observed data can then be modeled using the following linear degradation process:

$$\underbrace{\mathbf{y}}_{\text{observation}} = \underbrace{A}_{\substack{\text{forward} \\ \text{operator}}} \underbrace{\mathbf{x}}_{\text{image}} + \underbrace{\mathbf{n}.}_{\text{noise}} \tag{2.72}$$

where $\mathbf{y} \in \mathbb{R}^m$ represents the measured signal resulting from the matrix-vector multiplication with the system matrix $A^{m \times n}$ and the original image $\mathbf{x}$, along with the

[19] In the literature, images enhanced with plausible yet incorrect details, not corresponding to the true scene, are sometimes referred to as "hallucinations." Whether to seek or avoid such hallucinations depends on the specific context and purpose.

[20] As it is not feasible to reconstruct a continuous representation of the ideal image $x(\mathbf{u})$, we generally limit ourselves in this thesis to the reconstruction of digital images with a finer spatial sampling. See [30] for related discussions on this topic and for continuous modeling of the forward operator.

noise term $\mathbf{n}$. In practice, the matrix A is often very large, so it is more efficient to compute the matrix-vector multiplication using operators such as local convolution (in the case of a circulant matrix). This formulation is versatile and applicable to various problems, including denoising, deconvolution, super-resolution, or inpainting [60].

Maximum a posteriori estimation. Even without any noise present, it's common for the number of observations n to be smaller than the number of unknowns m ($m < n$), leading to an under-determined linear system. Consequently, a multitude of solutions could yield accurate measurements. The task of determining which solution to select can be addressed through Bayesian modeling. Let us consider $p(\mathbf{x})$, a prior distribution on the high-resolution image $\mathbf{x} \in \mathbb{R}^n$. Then using Bayes'rule the posterior distribution $p(y|\mathbf{x})$ is given by

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}). \tag{2.73}$$

One solution to estimate $\mathbf{x}$ is to use maximum a posteriori (MAP) estimation [30] to maximize $p(x|y)$ according to $\mathbf{x}_{\text{MAP}} = \arg\max_{\mathbf{x}\in\mathbb{R}^n} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$. Taking the negative log-likelihood, this maximization problem is equivalent to

$$\mathbf{x}_{\text{MAP}} = \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min} \underbrace{L(\mathbf{x})}_{\text{data-fitting term}} + \underbrace{\lambda R(\mathbf{x})}_{\text{image prior}}, \tag{2.74}$$

where $R(\mathbf{x}) \propto -\log p(\mathbf{x})$ and $L(\mathbf{x}) \propto -\log p(\mathbf{y}|\mathbf{x})$. For additive white Gaussian noise, the data-fitting term boils down to the least square $L(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2$ [61]. We discuss in the next section how to choose the prior function R.

2.5.2 Image Priors

Image priors in image processing are statistical assumptions or constraints used to regularize the solution of ill-posed inverse problems [61, 60]. In this section, we present classical regularizing functions classically used for some inverse problems in imaging.

Quadratic regularizer. In cases where the problem is over-constrained yet ill-posed, a common and straightforward choice for regularization is the ℓ_2-norm on the image, denoted as $\|\mathbf{x}\|_2^2$. Another fundamental image prior encourages high-resolution images $\mathbf{x}$ as spatially smooth signals [30]. This prior can be parameterized as follows

$$R(\mathbf{x}) = \|Q\mathbf{x}\|_2^2, \tag{2.75}$$

where $Q \in \mathbb{R}^{n\times n}$ is a circulant matrix that implements a high-pass filter. A typical selection for Q includes the spatial gradient of the image or the Laplacian operator [30]. Note that these priors offer the advantage of being relatively easy to optimize.

Sparsity promoting regularizer. In the context of under-determined problems, commonly used regularizers are the sparsity-promoting regularizers of the form:

$$R(\mathbf{x}) = \|S\mathbf{x}\|_1, \tag{2.76}$$

where S represents a suitable transform [61], and $\|\cdot\|_1$ is the ℓ_1-norm, which encourages sparsity, see [60]. These regularizers aim to enforce the sparsity of the image in specific domains, meaning they encourage the image to have many zero or close-to-zero values in certain representations. This approach is valuable for promoting simplicity or compressibility in the image, which is particularly useful in cases where the underlying solution is expected to have sparse representations in certain domains. Note that a direct sparsity measure such as ℓ_0 may also be used. But it produces combinatorially hard problem. While 2.76 is convex.

Total variation. A popular regularizer often used for image restoration problems is the Total variation (TV). See [62]. The TV of an image is defined as the sum of the absolute differences between neighboring pixel values. By minimizing the TV of an image, the algorithm promotes sparse image gradients and hence enforces the image to have sharp edges and flat regions. The isotropic version of the prior, which is convex but not differentiable, is

$$R_{\mathrm{TV}}(\mathbf{x}) = \sum_{i=1}^{N} \sqrt{\nabla_u \mathbf{x}[i]^2 + \nabla_v \mathbf{x}[i]^2}, \tag{2.77}$$

where $\nabla_u \mathbf{x}$ and $\nabla_u \mathbf{x}$ are the spatial gradient in the u and v directions

2.5.2.1 Successful Image Priors for Denoising

While not directly applicable to regularizing inverse problems, we highlight two crucial image priors that form the foundation of many modern denoising techniques today, such as [63].

Sparse patch decomposition. Another effective way to leverage sparsity is sparse coding. It consists in decomposing corrupted signals on the basis of elementary signals [60]. This leverages the observation that natural images can be well reconstructed with few atoms on a well-chosen basis. The key idea is that corrupted signals are highly entropic and can't be recovered with a sparse linear combination of atoms. To perform image restoration of a noisy image, overlapping patches are first extracted from, and each noisy patch is reconstructed by solving the Lasso with $p = 0$ or $p = 1$

$$\min_{\mathbf{a}_i \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{y}_i - D\mathbf{a}_i\|_2^2 + \lambda \|\mathbf{a}_i\|_p, \tag{2.78}$$

$D = [\mathbf{d}_1, \ldots, \mathbf{d}_p] \in \mathbb{R}^{m \times p}$ is a basis of elementary signals called atoms, and $\|.\|_p$ induces sparsity. Atoms of the dictionary can be chosen, for example, from a DCT basis or learned on a set of natural images with alternate optimization. See [60] for more details on this topic.

Self-similar structures. The non-local means technique involves averaging akin patches affected by independent and identically distributed (i.i.d.) noise with zero mean. This averaging diminishes the noise variance while preserving the signal integrity. The underlying concept is built on the observation that natural images exhibit numerous instances of local self-similar patterns.

Procedure 1 ISTA

$f, g : \mathbb{R}^n \to \mathbb{R}$ are convex, ∇f L-Lipschitz and g non-smooth
Input: $\mathbf{x}^0 \in \mathbb{R}^n$,
repeat
$\mathbf{z}^{k+1} \leftarrow \mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}^k)$
$\mathbf{x}^{k+1} \leftarrow \text{Prox}_{g/L}(\mathbf{z}^{k+1})$
until convergence

2.5.3 Optimization

Having introduced inverse problems in imaging, we are interested in minimizing functions of the form

$$F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}), \tag{2.79}$$

where f is a convex data-fitting term, and g is a regularizing function that is convex and non-smooth. Proximal algorithms are typically used to minimize equation 2.79 when g is not smooth [64]. One of the most simple algorithms is the proximal gradient descent (PGD or ISTA) which is summarized in Algorithm 1. ISTA alternates between gradient steps with respect to f and evaluation of the proximal operator of the non-smooth term. The proximal operator of g is defined as the unique solution of

$$\text{Prox}_{\gamma g}(\mathbf{z}) = \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|^2 + \gamma g(\mathbf{x}) \right\}, \tag{2.80}$$

for any convex function $g : \mathbb{R}^n \to \mathbb{R}$. Proximal operators play a key role in optimization and admit a closed form for many regularizers [64]. For example, for the ℓ_1 norm, the proximal operator is the soft-thresholding operator $\mathcal{S}_\lambda(u) = \text{sign}(u)\max(|u| - \lambda, 0)$ which is easy and fast to compute.

However, the algorithm has a slow convergence [65] of $\mathcal{O}(1/k)$. An accelerated version called FISTA was proposed in [65]. FISTA uses a slightly different gradient step to obtain a better convergence rate $\mathcal{O}(1/k^2)$. In practice, more elaborate solvers are also used. We can cite the alternating direction method of multipliers (ADMM) or Half Quadratic Splitting (HQS); see [64] for more details on this topic.

2.5.4 Deep Learning

Of course, deep learning has successfully addressed various inverse problems in imaging. In its most straightforward form, this involves feeding a neural network $h_{\boldsymbol{\Theta}}$ with corrupted measurements $\mathbf{y}$ and training it to generate clean images $\mathbf{x}$ using a supervised approach. More formally, the objective is to minimize the discrepancy between pairs of corrupted/clean data $(\mathbf{y}, \mathbf{x}) \sim \mathcal{P}$ by minimizing

$$\min_{\boldsymbol{\Theta}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{P}}\, \mathcal{L}\big(\mathbf{x}, h_{\boldsymbol{\Theta}}(\mathbf{y})\big). \tag{2.81}$$

Here, $\mathcal{L}$ represents the loss function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ used to quantify the distortion between the reconstructed image and its corresponding ground truth. The mean square error is frequently adopted for image restoration tasks.

Minimizing equation 3.7 is achieved in practice by minimizing the empirical risk on a large set of training data $\{(\mathbf{y}_1, \mathbf{x}_1), \cdots, (\mathbf{y}_N, \mathbf{x}_N)\}$. The neural network is trained with backpropagation, fine-tuning its parameters Θ to minimize

$$\min_{\boldsymbol{\Theta}} \frac{1}{n} \sum_{i=1}^{N} \mathcal{L}\big(\mathbf{x}_i, h_{\boldsymbol{\Theta}}(\mathbf{y}_i)\big). \tag{2.82}$$

Once training is finished, the model can be employed to infer clean images from noisy measurements it hasn't encountered during training.

Training pairs generation. Note that training pairs can be synthetically generated using the forward model. Achieving accurate simulations of real defects in practical systems can be challenging, acting as a primary bottleneck that limits performance on real images. We provide a concise overview of the advantages and limitations of deep learning for image restoration over classical methods presented earlier.

Pros: Deep learning methods offer faster processing speeds with a single forward pass, unlike iterative optimization-based methods that may require a prohibitive number of iterations to converge. Second, deep learning somewhat simplifies the design phase thanks to its data-driven approaches, sometimes requiring less tuning and expert knowledge. With available training data, a working solution can be quickly designed. Additionally, deep learning methods have the potential to produce images of superior quality compared to concurrent methods on some specific problems.

Cons: A first significant limitation is the heavy reliance on data simulation quality, which significantly impacts the generated results. Additionally, deep neural networks can produce undesirable artifacts, and unfortunately, the black-box nature of deep learning makes it challenging, if not impossible, to control or detect failure cases producing artefacts [57]. This also can lead to the model reconstructing fake details (hallucinations) that may be harmful (in scientific or medical imaging). Lastly, the computational cost in terms of FLOPS (floating-point operations per second) and/or memory requirements often exceeds the capacity of modern embedded devices.

Pros (+)	Cons (-)
Fast inference	Computational cost (flops/memory)
Reconstruction quality	Robustness
Data-driven (easy to design)	Interpretability (artefacts detection/hallucinations)
	Need for accurate data simulation

In the next Sections, we discuss how learned models can be enhanced by integrating physical models in computational imaging. The main advantages and drawbacks will also be presented.

2.5.5 Plug and Play

Classical inverse methods discussed above require handcrafted priors and hyperparameters tuning (such as the regularization penalty λ). An essential property of PGD or ADMM is the computation of the proximal operator, which can be viewed as solving a denoising problem [66, 61]. This perspective led to the development of Plug-and-Play methods, where the prox step is replaced by an image denoiser, such as BM3D [63], or a trained deep neural network like DnCNN [66]. Consequently, a denoiser $\mathcal{D}_{\boldsymbol{\theta}}$ can be employed to tackle various inverse problems. See Algorithm 2 for a simplistic instance of this approach PnP-ISTA. Originally introduced for ADMM, the PnP framework has seen numerous variations and adaptations to many algorithms. Some variations have convergence guarantees, established using monotone operator theory [61]. Even when using black box CNN-based denoisers, convergence guarantees can be obtained by training the CNN to satisfy contractive conditions, for example through spectral norm techniques.

Procedure 2 PnP-ISTA

Input: $\mathbf{x}^0 \in \mathbb{R}^n$, a black-box denoiser $\mathcal{D}_{\boldsymbol{\Theta}}$
  repeat
    $\mathbf{z}^{k+1} \leftarrow \mathbf{x}_k - \gamma \nabla_f(\mathbf{x}^k)$
    $\mathbf{x}^{k+1} \leftarrow \mathcal{D}_{\boldsymbol{\Theta}}(z^k)$
  until convergence

Optimization problem. However, it is crucial to emphasize that PnP algorithms with black-box denoisers in their simplest form do not necessarily solve an optimization problem like ADMM and FISTA algorithms unless specific constraints are imposed on the denoiser, as the deep neural network may not have symmetric Jacobian (we refer the reader to [67] for more details).

2.5.6 Deep Unfoldings

It is possible to go one step further and directly finetune the denoiser parameters $\boldsymbol{\Theta}$ to maximize performance for a specific restoration task by unfolding the iterative algorithm. This is generally called "deep unfolding" in the literature. By truncating a PnP algorithm like PnP-ISTA to a certain number of iterations ($K \geq 1$), it forms a differentiable computational graph, which becomes a trainable model. Denoting $\mathbf{x}^K_{\Theta,\gamma}(\mathbf{y})$ as the K-th iterate, the weights Θ of the denoiser $\mathcal{D}_\Theta$, and the gradient step γ, can be adjusted to minimize reconstruction errors on a training set of pairs comprising corrupted/ground-truth images. Therefore, the model's training involves solving the learning problem

$$\min_{\boldsymbol{\Theta},\gamma} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{P}} \mathcal{L}\big(\mathbf{x}, \mathbf{x}^K_{\boldsymbol{\Theta},\gamma}(\mathbf{y})\big). \tag{2.83}$$

Of course, numerous variations based on this formulation exist, and both PnP and deep unfoldings have been adapted to various algorithms. In this discussion, however, we concentrate solely on the core concept.

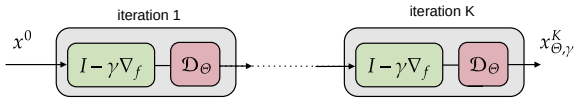


Figure 2.35: Deep unfolding of PnP-ISTA. Figure inspired from [61].

Deep unfoldings have been empirically shown to achieve better performance compared to pure Plug-and-Play (PnP) models. This enhancement is generally attributed to the fine-tuning of the denoiser, allowing it to correct artifacts induced by the inverse solver for specific problems.

Furthermore, unlike purely deep learning approaches, hybrid methods often necessitate smaller models, making them particularly advantageous for integration on embedded devices. Additionally, these methods demonstrate higher stability when applied to real-world unseen images during inference, and they require fewer training examples to attain satisfactory performance. These advantages may arise from the reduced workload on the neural network, which solves a somewhat simpler problem. Below, we outline some advantages and drawbacks of such approaches:

Pros (+)	Cons (-)
Compact DNNs (better portability)	Harder to design
Reconstruction qual. matching larger DNNs	Need for accurate data simulation
Robustness / fewer hallucinations	
Faster training/ requiring less data	

2.5.7 Bilevel Optimization

We presented PnP models as a convenient way for mixing data-driven learning with model-based optimization, but we saw that PnP models generally do not have guarantees to solve an optimization problem. Another strongly connected approach is the bi-level formulation, which mitigates that issue and therefore has better interpretability. Bilevel optimization is a class of problems where one optimization problem (the upper-level problem) depends on the solution of another optimization problem (the lower-level problem). For instance, assuming that one is given corrupted/clean pairs $(\mathbf{x}_i, \mathbf{y}_i)_{i=1\dots n}$, one may consider the following bi-level objective

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \quad & \frac{1}{n}\sum_{i=0}^{N} \mathcal{L}\Big(\mathbf{x}_i, g_{\boldsymbol{\Theta}}\big(\mathbf{z}^\star\big)\Big) \\ \text{s.t.} \quad & \mathbf{z}^\star \in \underset{\mathbf{z}\in\mathbb{R}^p}{\arg\min}\, h_{\boldsymbol{\Theta}}(\mathbf{y}_i, \mathbf{z}). \end{aligned} \tag{2.84}$$

where $\mathbf{z}^\star$ is the result of some model-based optimization obtained by minimizing some function $h_{\boldsymbol{\Theta}}$. The reconstructed image $g_{\boldsymbol{\theta}}(\mathbf{z}^\star)$ is compared to the ground-truth $\mathbf{y}_i$ through a loss function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$. That approach was, for example, used in [60] in the context of sparse coding. In that case, $h_{\boldsymbol{\Theta}}$ is a Lasso optimization problem, and $\mathbf{z}^\star$ is the optimal sparse code, encoding corrupted image $\mathbf{y}$, while g is chosen here as a linear reconstruction operator W.

Optimization

To solve such optimization problems, one needs to compute the gradient of the trainable parameters with respect to the lower-level problem solution $\nabla_{\boldsymbol{\Theta}}\mathbf{z}^\star$, sometimes called the hyper-gradient in the literature. Different methods have emerged in the literature; we summarize two techniques.

Implicit differentiation. The first approach consists in deriving the exact gradient $\nabla_{\boldsymbol{\Theta}}\mathbf{z}^\star$ by leveraging the implicit function theorem. Assuming that h is twice differentiable and that its optimal solution $\mathbf{z}^\star(\boldsymbol{\Theta})$ uniquely exists. By the implicit function theorem, the derivative of g with respect to $\boldsymbol{\Theta}$ can be written as

$$\nabla_{\Theta} z^\star = -\nabla_{\boldsymbol{\Theta},z} h \nabla_{z,z}^{-1} h \nabla_{\mathbf{z}^\star} f, \tag{2.85}$$

where $f = \frac{1}{n}\sum_{i=0}^{N} \mathcal{L}\Big(\mathbf{x}_i, g_{\boldsymbol{\Theta}}\big(\mathbf{z}^\star\big)\Big)$. Note that a linear problem must be solved to compute the gradient. A solution can be computed with a conjugate gradient algorithm and only requires computing the Hessian vector product. This can be done efficiently with double auto diff. Refer to [68] for more details on this topic.

Unrolled optimization. A commonly used approach is choosing an iterative method for minimizing the lower problem $h_{\boldsymbol{\Theta}}$ and then computing an approximated solution to the optimization problem after a truncated number of K iterations $\mathbf{z}^K$. The unrolled iterations can be seen as a differentiable function of $\boldsymbol{\Theta}$ and $\nabla_{\boldsymbol{\Theta}}\mathbf{z}^K$

can be computed with auto-diff. The concept of unrolled optimization concept was initially introduced to accelerate the solution of the Lasso with an unrolled ISTA algorithm called LISTA [69]. More formally, given an initial condition z_0, the iterative optimization algorithm solving the lower problem can be written as $z_{t+1} = U_{\boldsymbol{\Theta}}(z_t)$. We can view $\mathbf{z}^K$ as a function of $\boldsymbol{\Theta}$ by unrolling the iterative scheme for K iterations; it can be shown with the chain rule

$$\nabla_{\boldsymbol{\Theta}} z^K = \sum_{k=0}^{K} \nabla_{\boldsymbol{\Theta}} U \nabla_{\mathbf{z}^k} U \cdots \nabla_{\mathbf{z}^{K-1}} U. \tag{2.86}$$

See Figure 2.36 for an illustration and refer to [70] for more details on this topic.

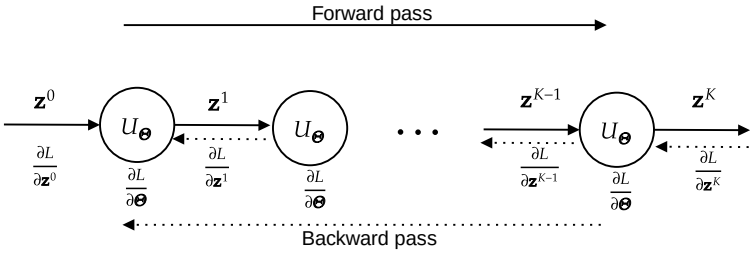


Figure 2.36: Unrolled optimization. Intermediate values need to be accumulated in memory for the backward pass. Therefore, memory consumption grows linearly with the number of unrolled iterations.

Discussions. Note that in the first case, there is a possible mismatch between the approximated solution $\mathbf{z}^K$ and the computed hypergradient $\nabla_{\boldsymbol{\Theta}} \mathbf{z}^\star$ if the solver of the lower problem has not fully converged. Due to inexact gradient information, this mismatch may lead to instabilities during the training phase [68]. For this reason, unrolled optimization may exhibit better stability during training if the solver has not fully converged. But on the other hand, unrolled optimization has a large memory footprint, as the intermediate results computed during the forward pass must be stored in memory to accumulate gradients during the backward pass to compute the full gradient $\nabla_{\boldsymbol{\Theta}} \mathbf{z}^K$. This may result in memory issues if the number of iterations is too large. Note that solutions like truncated back propagation [71] or checkpointing can be used for memory savings. See [70, 68] for related discussions.

Learned Inverse Problems (LIP). Note that with the bilevel framework, $\mathbf{z}^\star$ may also be chosen as the solution of an inverse problem, and g is set to the identity. In this fashion, the hyperparameters of the inverse problem, such as the regularization strength -a key parameter to tune- can be adjusted end-to-end on a set of training data. Furthermore, using unrolled optimization, it is also possible to tune the optimizer's parameters (such as the gradient steps or a preconditioner's parameters). In this manuscript, we refer to this setting as *trainable prior* or *learned inverse problems*. LIP allows very compact models with optimized performance as it alleviates the cumbersome task of hyperparameters tuning, often crucial, when designing an optimization-based inverse solver.

2.6 Burst Photography

To conclude our background section, we focus on a specific class of image restoration algorithms using bursts of images rather than individual frames. Burst photography involves capturing multiple shots of the same scene in rapid succession, usually within a few seconds, and with varying settings such as ISO, exposure, or aperture. By increasing and diversifying the number of observations of the underlying scene, these algorithms can achieve significant improvements compared to single-frame approaches.

Classical methods for processing bursts typically adopt a two-step approach, encompassing (1) frame alignment and (2) merging of the frames. Nevertheless, recent advances in deep learning have enabled simultaneous handling of both tasks, occasionally removing the explicit need for frame alignment.

2.6.1 Registration

Image registration is the process of aligning images to ensure they are in the same coordinate system. This is generally the first step of multi-frame methods.

2.6.1.1 Motion Parameterization

Various parameterizations can be chosen to represent motion induced by hand tremors when taking the burst. Motion representation is a crucial aspect when registering images. There is generally a tradeoff between the expressivity of the motion model and its robustness.

Parametric motion. In the case of a parametric motion, the pixel $\mathbf{u}$ is assumed to be transformed by a global transformation that is characterized by a small number of parameters. We define the projective homography in homogeneous coordinates, which has 8 degrees of freedom according to

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \cong \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} \mathbf{u} \\ 1 \end{bmatrix}, \tag{2.87}$$

where $\cong$ indicates equality up to a scale factor. Homographies can represent motions in case of a pure rotation of the camera with a rigid scene or a general motion of the camera with translations but only for a plane.

Homographies given the plane. Suppose a plane π in 3d-space specified by its coordinate in the world frame. Let us assume we have two cameras, C_1 and C_2, and $[R, \mathbf{t}]$ is the relative motion between them. The two cameras C_1, C_2 look at a point $\mathbf{x}$ on the plane π of normal vector $\mathbf{n}$ and of the distance d from the origin to the plane, so that $\mathbf{n}^\top \mathbf{x} + d = 0$, see Figure 2.37 for an illustration. Let $\mathbf{u}$ and $\mathbf{u}'$ be the projection of $\mathbf{x}$ in the two image planes. The authors of [72] showed that the coordinates of the two projections are related by a homography H, which can be expressed as

$$H = K' \left(R - \frac{\mathbf{t}\mathbf{n}^\top}{d} \right) K^{-1}, \tag{2.88}$$

where K and K' are the camera's intrinsic parameter matrices. Note that when the image region in which the homography is computed is small, or the image has been

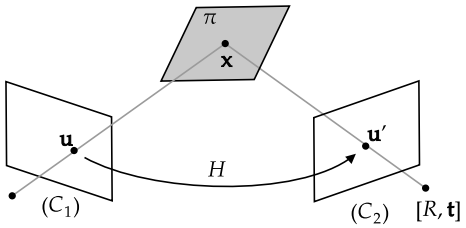


Figure 2.37: Homography induced by a plane: the ray corresponding to point $\mathbf{u}$ in camera C_1 is extended to meet the plane π in a point $\mathbf{x}$ in 3d-space. This point is projected to a point $\mathbf{x}'$ in the other image. The from $\mathbf{u}$ to $\mathbf{u}'$ is the homography induced by the plane π. Figure inspired from [72]

acquired with a large focal length, an affine transformation where the last row is fixed to $p_{31} = p_{3,2} = 0$ is a valid model of image displacements.

Rigid transformation. In several cases, it is practical to use a transformation with fewer degrees of freedom, such as an affine or rigid transformation. A rigid transformation has 3 degrees of freedom and can be parametrized with three parameters $\mathbf{p} = [t_u, t_v, \theta]^\top$ according to

$$M = \begin{bmatrix} \cos\theta & -\sin\theta & t_u \\ \sin\theta & \cos\theta & t_v \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.89}$$

A rigid transformation describes a motion composed of a rotation and translation. This transform preserves the ratio of distance in images as well as angles between lines.

A rigid transformation or even a homography may not be sufficient to represent faithfully the motion of the camera on a rigid scene if the scene is not planar and the motions induce parallax effects. A common way to solve this is to consider local parametric transformation. Block parametric transformations involve dividing the image sequence into small blocks and then estimating the motion of each block between consecutive frames based on a set of parameters.

Optical flow. A more flexible approach is the use of a dense vector flow field to describe motion according to

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} u + m_u(\mathbf{u}) \\ v + m_v(\mathbf{u}) \end{bmatrix}. \tag{2.90}$$

The displacements $m(\mathbf{u})$ with $m : \mathbb{R}^2 \to \mathbb{R}^2$ describe the motion from the image to the reference image at the pixel position $\mathbf{u}$. This has the advantage of modeling the non-rigid motion of the scene or the motion of a projective surface on a non-planar surface.

2.6.1.2 Registration Algorithms

Optimization methods. Optimization-based methods analyze images using the first or higher-order derivatives of the image intensity function to estimate the trans-

formation between them. One of the earliest image registration algorithms is the Lucas-Kanade method [73, 74] that we present in the next paragraph.

The goal of this algorithm is to minimize a photometric similarity measure between a source image $\mathbf{I}_2$ and target image $\mathbf{I}_1$ with respect to parameters $\mathbf{p} \in \mathbb{R}^p$ of a transformation $\mathbf{u}'(\mathbf{u}, \mathbf{p})$, where $\mathbf{u} = [u, v]^\top$ are the pixels coordinates. The algorithm starts with an initial guess $\mathbf{p}$ and iteratively seeks the optimal increment on the motion parameters by solving the non-linear least-square optimization problem:

$$\min_{\Delta \mathbf{p}} \sum_{\mathbf{u} \in \Omega} \Big| I_1(\mathbf{u}) - I_2\big(\mathbf{u}'(\mathbf{u}, \mathbf{p} + \Delta \mathbf{p})\big) \Big|^2 \tag{2.91}$$

This a non-linear least-square optimization problem because even if the transformation is linear in $\mathbf{p}$ the pixel intensities are not linearly related to the pixel coordinates vector $\mathbf{u}$. This minimization problem is generally solved with a Gauss-Newton algorithm, and the parameters are updated as $\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$ until convergence, where

$$\Delta \mathbf{p} = H^{-1} \sum_{\mathbf{u} \in \Omega} \nabla I_2\big(\mathbf{u}'(\mathbf{u}, \mathbf{p})\big) J(\mathbf{u}, \mathbf{p})^\top \Big(I_1(\mathbf{u}) - I_2\big(\mathbf{u}'(\mathbf{u}, \mathbf{p})\big) \Big), \tag{2.92}$$

where, $\nabla I = [\nabla_u I(\mathbf{u}), \nabla_v I(\mathbf{u})]^\top$ denotes the spatial gradient of the image and H is the approximated hessian computed as

$$H = \sum_{\mathbf{u} \in \Omega} \Big(\nabla I_2\big(\mathbf{u}'(\mathbf{u}, \mathbf{p})\big) J(\mathbf{u}, \mathbf{p}) \Big)^\top \Big(\nabla I_2\big(\mathbf{u}'(\mathbf{u}, \mathbf{p})\big) J(\mathbf{u}, \mathbf{p}) \Big). \tag{2.93}$$

Multiscale algorithm. The Lucas-Kanade method assumes that the displacement of image contents between two frames is small. To handle large motions, a coarse-to-fine approach is typically employed. This involves creating a pyramid of downsampled images. At the coarsest scale, the motion is initialized as zero, and the motion is estimated. This initial solution is then iteratively refined as we move to finer scales in the pyramid.

2.6.2 High Dynamic Range

In scenes with high dynamic range, a single exposure may fail to capture details in both the highlights due to the sensor's saturation and shadows due to low SNRs. Taking multiple exposures is an effective way to extend the dynamic range and increase SNR in photographs.

To estimate the radiance map of a scene with large dynamic, K images $(\mathbf{I}_1, \cdots \mathbf{I}_K)$ are shot with different gains $(g_1, \cdots g_K)$ and exposures $(\Delta t_1, \cdots \Delta t_K)$. Choosing this set of exposure parameters optimally is a challenging problem called metering. See [42] for a discussion on this problem. The frame k with normalized exposure can be expressed as

$$\tilde{\mathbf{I}}_k = \frac{\mathbf{I}_k g_k}{\Delta t_k}. \tag{2.94}$$

Assuming perfect alignment of the images (this is considered true for statics scenes and for a burst of images captured on a tripod), Hasinoff et al. [42] derived the minimum variance estimator of the radiant power Φ for each pixel. This blends the K measurements with normalized exposures $\tilde{\mathbf{I}}_k$. The estimate of the radiant power $\hat{\Phi}$incident to the pixel at location j is then

$$\hat{\Phi}[j] = \frac{\sum_k w_k \tilde{\mathbf{I}}_k[j]}{\sum_k w_k}. \tag{2.95}$$

The fusion weights depend on the exposure parameters (i.e., gain g_k and exposure time Δt_k) and are set to 0 in case of saturated pixels. See [42, 75] for a detailed expression of these weights.

While the fusion process is relatively straightforward, the main challenge lies in accurately aligning the frames. This difficulty arises due to the high heterogeneity of frame content, including variations in saturation areas and significant differences in SNR within dark zones. Consequently, aligning frames in this context becomes highly challenging [57]. To tackle this challenge, several methods have been proposed, including those suggested by Sen et al. [76], Hu et al. [77], and Gallo et al. [78].

2.6.3 Super-Resolution

Burst super-resolution improves the resolution of images by combining multiple images captured in quick succession. By taking a burst of photos of the same scene, each with a slightly different viewpoint, the images can be aligned and merged to create a final image with greater detail and higher resolution than any of the individual images. Multi-frame super-resolution is commonly formulated as an inverse problem as in [79, 29] and can be represented as follows

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min} \sum_{k=1}^{K} \|DBW_{\mathbf{p}_i}\mathbf{x} - \mathbf{y}_i\|^2 + \lambda R(\mathbf{x}). \tag{2.96}$$

Here, $\mathbf{x}$ represents the high-resolution image, while $\mathbf{y}_i$ corresponds to one of the observed low-resolution frames from the burst. The operators B, D, and W_i are responsible for blurring, decimation (possibly considering spectral decimation like the Bayer pattern), and image deformations caused by slight camera displacements, respectively. The function $R : \mathbb{R}^n \to \mathbb{R}$ acts as a regularization term, ensuring desirable properties in the final high-resolution image.

One critical aspect of super-resolution lies in the quality of image registration, as accurate alignment is essential for restoring high-frequency details. Particularly, dealing with non-rigid motions poses challenges during frame alignment. To address this, a common approach involves using a method to detect misalignments, allowing the rejection of frames that haven't been properly aligned. These algorithms, often referred to as deghosting algorithms in the literature, play a significant role in ensuring the overall success of the super-resolution technique.

There are alternate formulations of the problem. For instance, some methods involve reconstructing high-resolution images in the Fourier domain or performing non-uniform interpolation, as discussed in [80, 48]. Additionally, some approaches incorporate motion estimation as an initial step, while others jointly estimate motion parameters along with the high-resolution image. For an extensive overview of super-resolution algorithms, a recommended resource is the detailed taxonomy provided in [30].

2.6.4 Low-Light Imaging

In low-light conditions, achieving sharp images is difficult. Night scenes require long exposure times to capture enough photons to produce images with a sufficient SNR. However, this extended exposure time can introduce motion blur when images are captured by hand due to hand tremors.

One solution to combat this problem is to capture a burst of rapid sequences of sharp photographs with short exposures. These photos are aligned and merged, resulting in a final image with improved SNR and reduced motion blur.

Once the frames are accurately aligned, a common approach is to employ the empirical mean of the pixel values from the different frames as the clean image estimator, considering the noise of each measurement to be independent. Additionally, techniques can be applied to detect blurry frames and reject them from the merging process. Utilizing deghosting algorithms can also aid in the rejection of poorly aligned frames, enhancing the quality of the produced image.

2.6.5 Focus Stacking

Burst photography can generate images with an extended depth of field by merging multiple shots captured at different focus points. By capturing a burst of photos, each focused at a slightly different distance from the camera. The individual images can be combined to produce a final image with a greater depth of field. This technique is particularly advantageous in macro photography, where a shallow depth of field can result in difficulties capturing the entire subject in focus.

However, aligning frames with different focus planes and integrating them into the same frame coordinate poses important challenges similar to HDR registration. Furthermore, due to the change of the lens-to-sensor distance, the varying focus settings introduce scaling and other subtle geometric effects [81] that must be considered when registering the frames.

Chapter 3

Differentiable Non-Local Sparse Model

Chapter abstract: Non-local self-similarity and sparsity principles have proven to be powerful priors for natural image modeling. We propose a novel differentiable relaxation of joint sparsity that exploits both principles and leads to a general framework for image restoration which is (1) trainable end to end, (2) fully interpretable, and (3) much more compact than competing deep learning architectures. We apply this approach to denoising, blind denoising, jpeg deblocking, and demosaicking, and show that, with as few as 100K parameters, its performance on several standard benchmarks is on par or better than state-of-the-art methods that may have an order of magnitude or more parameters.

B. Lecouat, J. Ponce, J. Mairal. Fully Trainable and Interpretable Non-Local Sparse Models for Image Restoration. *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Contents

3.1 Introduction

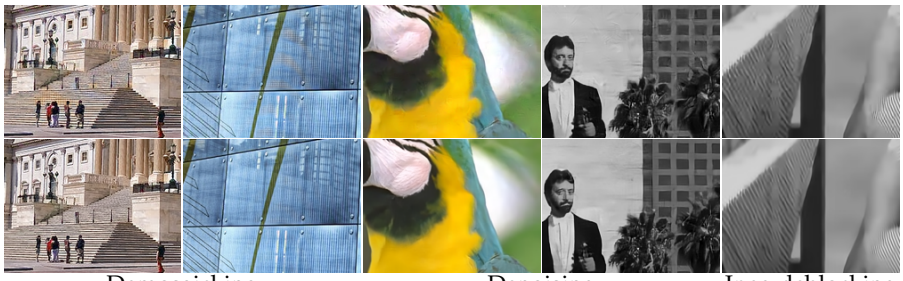


Figure 3.1: Effect of combining sparse and non-local priors for different reconstruction tasks. Top: reconstructions with sparse prior only, exhibiting artefacts. Bottom: reconstruction with both priors, artefact-free. Best seen in color by zooming on a computer screen.

The image processing community has long focused on designing handcrafted models of natural images to address inverse problems, leading, for instance, to differential operators [82], total variation [83], or wavelet sparsity [49] approaches. More recently, image restoration paradigms have shifted towards data-driven approaches. For instance, non-local means [84] exploits self similarities, and many successful approaches have relied on unsupervised methods such as learned sparse models [85, 60], Gaussian scale mixtures [86], or fields of experts [87]. More powerful models such as BM3D [63] have also been obtained by combining several priors, in particular self similarities and sparse representations [88, 63, 89, 90, 91].

These methods are now often outperformed by deep learning models, which are able to leverage pairs of corrupted/clean images for supervised learning, in tasks such as denoising [92, 93, 94, 95], demoisaicking [96, 97, 98], upsampling [99, 100], or artefact removal [98]. Yet, they also suffer from lack of interpretability and the need to learn a huge number of parameters. Improving these two aspects is one of the key motivation of this paper. Our goal is to design algorithms that bridge the gap in performance between earlier approaches that are parameter-efficient and interpretable, and current deep models.

Specifically, we propose a differentiable relaxation of the non-local sparse model LSSC [91]. The relaxation allows us to obtain models that may be trained end-to-end, and which admit a simple interpretation in terms of joint sparse coding of similar patches. The principle of end-to-end training for sparse coding was introduced in [101], and later combined in [102] for super-resolution with variants of the LISTA algorithm [103, 69, 104]. A variant based on convolutional sparse coding was then proposed in [105] for image denoising, and another one based on the K-SVD algorithm [106] was introduced in [107]. Note that these works are part of a vast litterature on model-inspired methods, where the model architecture is related to an optimization strategy for minimizing an objective, see [92, 108, 109].

In contrast, our main contribution is to extend the idea of differentiable algorithms to *structured* sparse models [110], which is a key concept behind the LSSC, CSR, and BM3D approaches. To the best of our knowledge, this is the first time that non-local sparse models are shown to be effective in a supervised learning setting. As [107], we argue that bridging classical successful image priors within deep

learning frameworks is a key to overcome the limitations of current state-of-the-art models. A striking fact is notably the performance of the resulting models given their low number of parameters.

For example, our method for image denoising performs on par with the deep learning baseline DnCNN [95] with 8x less parameters, significantly outperforms the color variant CDnCNN with 6x less parameters, and achieves state-of-the-art results for blind denoising and jpeg deblocking. For these two last tasks, relying on an interpretable model is important; most parameters are devoted to image reconstruction and can be shared by models dedicated to different noise levels. Only a small subset of parameters can be seen as regularization parameters, and may be made noise-dependent, thus removing the burden of training several large independent models for each noise level. For image demosaicking, we obtain similar results as the state-of-the-art approach RNAN [98], while reducing the number of parameters by 76x. Perhaps more important than improving the PSNR, the principle of non local sparsity also reduces visual artefacts when compared to using sparsity alone, which is illustrated in Figure 3.1.

3.2 Preliminaries and Related Work

In this section, we introduce non-local sparse coding models for image denoising and present a differentiable algorithm for sparse coding [69].

Sparse coding models on learned dictionaries. A simple approach for image denoising introduced in [106] consists of assuming that natural image patches can be well approximated by linear combinations of few dictionary elements. Thus, a clean estimate of a noisy patch is obtained by computing a sparse approximation. Given a noisy image, we denote by $\mathbf{y}_1, \ldots, \mathbf{y}_n$ the set of n overlapping patches of size $\sqrt{m} \times \sqrt{m}$, which we represent by vectors in $\mathbb{R}^m$ for grayscale images. Each patch is then processed by solving the sparse decomposition problem

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \tag{3.1}$$

where $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ in $\mathbb{R}^{m \times p}$ is the dictionary, which we assume given at the moment, and $\|.\|_1$ is the ℓ_1-norm, which is known to encourage sparsity, see [60]. Note that a direct sparsity measure such as ℓ_0-penalty may also be used, at the cost of producing a combinatorially hard problem, whereas (3.1) is convex.

Then, $\mathbf{D}\boldsymbol{\alpha}_i$ is a clean estimate of $\mathbf{y}_i$. Since the patches overlap, we obtain m estimates for each pixel and the denoised image is obtained by averaging:

$$\hat{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^{n} \mathbf{R}_i \mathbf{D}\boldsymbol{\alpha}_i, \tag{3.2}$$

where $\mathbf{R}_i$ is a linear operator that places the patch $\mathbf{D}\boldsymbol{\alpha}_i$ at the position centered on pixel i on the image. Note that for simplicity, we neglect the fact that pixels close to the image border admit less estimates, unless zero-padding is used.

Whereas we have previously assumed that a good dictionary $\mathbf{D}$ for natural images is available, the authors of [106] have proposed to learn $\mathbf{D}$ by solving a matrix factorization problem called *dictionary learning* [111].

Differentiable algorithms for sparse coding. ISTA [112] is a popular algorithm to solve problem (3.1), which alternates between gradient descent steps with respect to the smooth term of (3.1) and the soft-thresholding operator $S_\eta(x) = \text{sign}(x)\max(0, |x| - \eta)$.

Note that such a step performs an affine transformation followed by the pointwise non-linear function S_η, which makes it tempting to consider K steps of the algorithm, see it as a neural network with K layers, and learn the corresponding weights. Following such an insight, the authors of [69] have proposed the LISTA algorithm, which is trained such that the resulting neural network learns to approximate the solution of (3.1). Other variants were then proposed, see [103, 104]; as [105], the one we have adopted may be written as

$$\boldsymbol{\alpha}_i^{(k+1)} = S_{\boldsymbol{\Lambda}_k}\left[\boldsymbol{\alpha}_i^{(k)} + \mathbf{C}^\top\left(\mathbf{y}_i - \mathbf{D}\boldsymbol{\alpha}_i^{(k)}\right)\right], \tag{3.3}$$

where $\mathbf{C}$ has the same size as $\mathbf{D}$ and $\boldsymbol{\Lambda}_k$ in $\mathbb{R}^p$ is such that $S_{\boldsymbol{\Lambda}_k}$ performs a soft-thresholding operation with a different threshold for each vector entry. Then, the variables $\mathbf{C}, \mathbf{D}$ and $\boldsymbol{\Lambda}_k$ are learned for a supervised image reconstruction task.

Note that when $\mathbf{C} = \eta\mathbf{D}$ and $\boldsymbol{\Lambda}_k = \eta\lambda\mathbf{1}$, where η is a step size, the recursion recovers exactly the ISTA algorithm. Empirically, it has been observed that allowing $\mathbf{C} \neq \mathbf{D}$ accelerates convergence and could be interpreted as learning a preconditioner for ISTA [104], whereas allowing $\boldsymbol{\Lambda}_k$ to have entries different than $\lambda\eta$ corresponds to using a weighted ℓ_1-norm and learning the weights.

There have been already a few attempts to leverage the LISTA algorithm for specific image restoration tasks such as super-resolution [102] or denoising [105], which we extend in our paper with non-local priors and structured sparsity.

Exploiting self-similarities. The non-local means approach [84] consists of averaging similar patches that are corrupted by i.i.d. zero-mean noise, such that averaging reduces the noise variance without corrupting the signal. The intuition relies on the fact that natural images admit many local self-similarities. This is a non-parametric approach (technically a Nadaraya-Watson estimator), which can be used to reduce the number of parameters of deep learning models.

Non local sparse models. The LSSC approach [91] relies on the principle of joint sparsity. Denoting by S_i a set of patches similar to $\mathbf{y}_i$ according to some criterion,

we consider the matrix $\mathbf{A}_i = [\boldsymbol{\alpha}_l]_{l \in S_i}$ in $\mathbb{R}^{p \times |S_i|}$ of corresponding coefficients. LSSC encourages the codes $\{\boldsymbol{\alpha}_l\}_{l \in S_i}$ to share the same sparsity pattern—that is, the set of non-zero entries. This can be achieved by using a group-sparsity regularizer

$$\|\mathbf{A}_i\|_{1,2} = \sum_{j=1}^{p} \|\mathbf{A}_i^j\|_2, \tag{3.4}$$

where $\mathbf{A}_i^j$ is the j-th row in $\mathbf{A}_i$. The effect of this norm is to encourage sparsity patterns to be shared across similar patches, as illustrated in Figure 3.2. It may be seen as a convex relaxation of the number of non-zero rows in $\mathbf{A}_i$, see [91].

Building a differentiable algorithm relying on both sparsity and non-local self-similarities is challenging, as the clustering approach used by LSSC (or CSR) is typically not a continuous operation of the dictionary parameters.

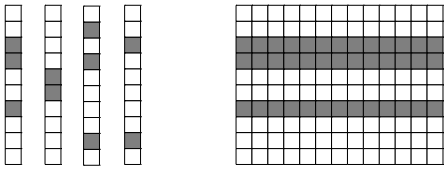


Figure 3.2: (Left) sparsity pattern of codes with grey values representing non-zero entries; (right) group sparsity of codes for similar patches. Figure from [91].

Deep learning models. In the context of image restoration, successful principles for deep learning models include very deep networks, batch norm, and residual learning [113, 95, 114, 98]. Recent models also use attention mechanisms to model self similarities, which are pooling operations akin to non-local means. More precisely, a non local module has been proposed in [93], which performs weighed average of similar features, and in [94], a relaxation of the k-nearest selection rule is introduced for similar purposes.

Model-based methods. Unfolding an optimization algorithm to design an inference architecture is not limited to sparse coding. For instance [108, 115] propose trainable architectures based on unrolled ADMM. The authors of [92, 113] propose a deep learning architecture inspired from proximal gradient descent in order to solve a constrained optimization problem for denoising; [116] optimize hyperparameters of non linear reaction diffusion models; [117] unroll an interior point algorithm. Finally, Plug-and-Play [109] is a framework for image restoration exploiting a denoising prior as a modular part of model-based optimization methods to solve various inverse problems. Several works leverage the plug-in principle with half quadratic spliting [118], deep denoisers [97], message passing algorithms [119], or augmented Lagrangian [120].

3.3 Proposed Approach

We now present trainable sparse coding models for image denoising, following [105], with a few minor improvements, before introducing differentiable relaxations for the LSSC method [91] . A different approach to take into account self similarities in sparse models is the CSR approach [89]. We have empirically observed that it does not perform as well as LSSC. Nevertheless, we believe it to be conceptually interesting, and provide a brief description in the appendix.

3.3.1 Trainable Sparse Coding (without Self-Similarities)

In [105], the sparse coding approach (SC) is combined with the LISTA algorithm to perform denoising tasks.[1] The only modification we introduce here is a centering step for the patches, which empirically yields better results.

SC Model - inference with fixed parameters. Following the approach and notation from Section 3.2, the first step consists of extracting all overlapping patches

[1]Specifically, [105] proposes a model based on convolutional sparse coding (CSC). CSC is a variant of SC, where a full image is approximated by a linear combination of small dictionary elements. Unfortunately, CSC leads to ill-conditioned optimization problems and has shown to perform poorly for image denoising. For this reason, [105] introduces a hybrid approach between SC and CSC. In our paper, we have decided to use the SC baseline and leave the investigation of CSC models for future work.

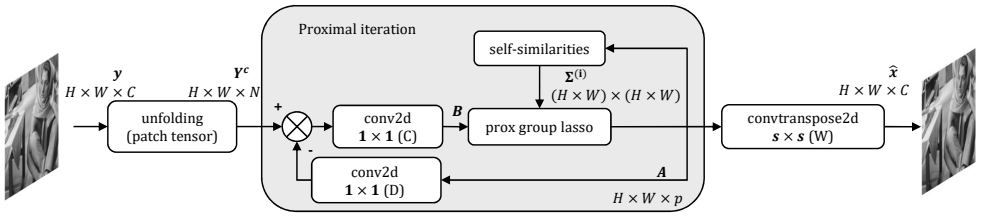


Figure 3.3: An illustration of the main inference algorithm for GroupSC. See Figure 3.4 for an illustration of the self-similarity module.

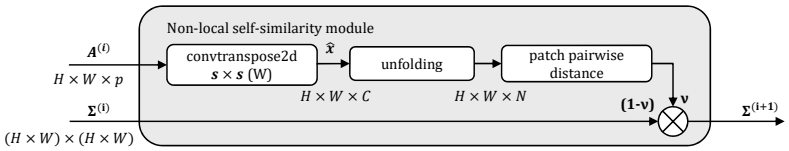


Figure 3.4: An illustration of the self-similarity module used in our GroupSC algorithm.

$\mathbf{y}_1, \dots, \mathbf{y}_n$. Then, we perform the centering operation for every patch

$$\mathbf{y}_i^c \triangleq \mathbf{y}_i - \mu_i \mathbf{1}_m \quad \text{with} \quad \mu_i \triangleq \frac{1}{m} \mathbf{1}_m^\top \mathbf{y}_i. \tag{3.5}$$

The mean value μ_i is recorded and added back after denoising $\mathbf{y}_i^c$. Hence, low-frequency components do not flow through the model. The centering step is not used in [105], but we have found it to be useful.

The next step consists of sparsely encoding each centered patch $\mathbf{y}_i^c$ with K steps of the LISTA variant presented in (3.3), replacing $\mathbf{y}_i$ by $\mathbf{y}_i^c$ there, assuming the parameters $\mathbf{D}, \mathbf{C}$ and $\mathbf{\Lambda}_k$ are given. Here, a minor change compared to [105] is the use of varying parameters $\mathbf{\Lambda}_k$ at each LISTA step. Finally, the final image is obtained by averaging the patch estimates as in (4.3), after adding back μ_i:

$$\hat{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{N} \mathbf{R}_i (\mathbf{W} \boldsymbol{\alpha}_i^{(K)} + \mu_i \mathbf{1}_m), \tag{3.6}$$

but the dictionary $\mathbf{D}$ is replaced by another matrix $\mathbf{W}$. The reason for decoupling $\mathbf{D}$ from $\mathbf{W}$ is that the ℓ_1 penalty used by the LISTA method is known to shrink the coefficients $\boldsymbol{\alpha}_i$ too much. For this reason, classical denoising approaches such as [106, 91] use instead the ℓ_0-penalty, but we have found it ineffective for end-to-end training. Therefore, as in [105], we have chosen to decouple $\mathbf{W}$ from $\mathbf{D}$.

Training the parameters. We now assume that we are given a training set of pairs of clean/noisy images $(\mathbf{x}, \mathbf{y}) \sim \mathcal{P}$, and we minimize in a supervised fashion

$$\min_{\mathbf{\Theta}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{P}} \, \|\hat{\mathbf{x}}(\mathbf{y}) - \mathbf{x}\|_2^2, \tag{3.7}$$

where $\mathbf{\Theta} = \{\mathbf{C}, \mathbf{D}, \mathbf{W}, (\mathbf{\Lambda}_k)_{k=0,1\dots K-1}, \boldsymbol{\kappa}, \nu\}$ is the set of parameters to learn and $\hat{\mathbf{x}}$ is the denoised image defined in (3.6).

3.3.2 Differentiable Relaxation for Non-Local Sparse Priors

Self-similarities are modeled by replacing the ℓ_1-norm by structured sparsity-inducing regularization functions. In Algorithm 4, we present a generic approach

Procedure 3 Pseudo code for the inference model of GroupSC.

Extract patches $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ and center them with (3.5);
Initialize the codes $\boldsymbol{\alpha}_i$ to 0;
Initialize image estimate $\hat{\mathbf{x}}$ to the noisy input $\mathbf{y}$;
Initialize pairwise similarities $\boldsymbol{\Sigma}$ between patches of $\hat{\mathbf{x}}$;
for $k = 1, 2, \ldots K$ **do**
    Compute pairwise patch similarities $\hat{\boldsymbol{\Sigma}}$ on $\hat{\mathbf{x}}$;
    Update $\boldsymbol{\Sigma} \leftarrow (1-\nu)\boldsymbol{\Sigma} + \nu\hat{\boldsymbol{\Sigma}}$;
    for $i = 1, 2, \ldots, N$ in parallel **do**
        $\boldsymbol{\alpha}_i \leftarrow \text{Prox}_{\boldsymbol{\Sigma}, \boldsymbol{\Lambda}_k}\left[\boldsymbol{\alpha}_i + \mathbf{C}^\top(\mathbf{y}_i^c - \mathbf{D}\boldsymbol{\alpha}_i)\right]$;
    end for
    Update the denoised image $\hat{\mathbf{x}}$ by averaging (3.6);
end for

to use this principle within a supervised learning approach, based on a similarity matrix $\boldsymbol{\Sigma}$, overcoming the difficulty of hard clustering/grouping patches together. In Figure 4.1, we also provide a diagram of one step of the inference algorithm. At each step, the method computes pairwise patch similarities $\boldsymbol{\Sigma}$ between patches of a current estimate $\hat{\mathbf{x}}$, using various possible metrics that we discuss in Section 3.3.3. The codes $\boldsymbol{\alpha}_i$ are updated by computing a so-called proximal operator, defined below, for a particular penalty that depends on $\boldsymbol{\Sigma}$ and some parameters $\boldsymbol{\Lambda}_k$. Practical variants where the pairwise similarities are only updated once in a while, are discussed in Section 3.3.6.

Definition 1 (Proximal operator)**.** *Given a convex function $\Psi : \mathbb{R}^p \to \mathbb{R}$, the proximal operator of Ψ is defined as the unique solution of*

$$Prox_\Psi[\mathbf{z}] = \underset{\mathbf{u} \in \mathbb{R}^p}{\arg\min} \frac{1}{2}\|\mathbf{z} - \mathbf{u}\|^2 + \Psi(\mathbf{u}). \tag{3.8}$$

The proximal operator plays a key role in optimization and admits a closed form for many penalties, see [60]. Indeed, given Ψ, it may be shown that the iterations $\boldsymbol{\alpha}_i \leftarrow \text{Prox}_{\eta\Psi}\left[\boldsymbol{\alpha}_i + \eta\mathbf{D}^\top(\mathbf{y}_i^c - \mathbf{D}\boldsymbol{\alpha}_i)\right]$ are instances of the ISTA algorithm [65] for minimizing

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^p} \frac{1}{2}\|\mathbf{y}_i^c - \mathbf{D}\boldsymbol{\alpha}_i\|^2 + \Psi(\boldsymbol{\alpha}_i),$$

and the update of $\boldsymbol{\alpha}_i$ in Algorithm 4 simply extend LISTA to deal with Ψ. Note that for the weighted ℓ_1-norm $\Psi(\mathbf{u}) = \sum_{j=1}^p \lambda_j |\mathbf{u}[j]|$, the proximal operator is the soft-thresholding operator S_Λ introduced in Section 3.2 for $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_p)$ in $\mathbb{R}^p$, and we simply recover the SC algorithm from Section 3.3.1 since Ψ does not depend on the pairwise similarities $\boldsymbol{\Sigma}$. Next, we present different structured sparsity-inducing penalties that yield more effective algorithms.

3.3.2.1 Group-SC.

For each location i, the LSSC approach [91] defines groups of similar patches $S_i \triangleq \left\{j = 1, \ldots, n \text{ s.t. } \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \leq \xi\right\}$ for some threshold ξ. For computational reasons, LSSC relaxes this definition in practice, and implements a clustering method such that $S_i = S_j$ if i and j belong to the same group. Then, under this clustering

assumption and given a dictionary $\mathbf{D}$, LSSC minimizes

$$\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{Y}^c - \mathbf{D}\mathbf{A}\|_{\mathrm{F}}^2 + \sum_{i=1}^{N} \Psi_i(\mathbf{A}) \quad \text{with} \quad \Psi_i(\mathbf{A}) = \lambda_i \|\mathbf{A}_i\|_{1,2}, \tag{3.9}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$ in $\mathbb{R}^{m \times N}$ represents all codes, $\mathbf{A}_i = [\boldsymbol{\alpha}_l]_{l \in S_i}$, $\|.\|_{1,2}$ is the group sparsity regularizer defined in (3.4), $\|.\|_{\mathrm{F}}$ is the Frobenius norm, $\mathbf{Y}^c = [\mathbf{y}_1^c, \ldots, \mathbf{y}_N^c]$, and λ_i depends on the group size. As explained in Section 3.2, the role of the Group Lasso penalty is to encourage the codes $\boldsymbol{\alpha}_j$ belonging to the same cluster to share the same sparsity pattern, see Figure 3.2. For homogeneity reasons, we also consider the normalization factor $\lambda_i = \lambda / \sqrt{|S_i|}$, as in [91]. Minimizing (3.9) is easy with the ISTA method since we know how to compute the proximal operator of Ψ, which is described below:

Lemma 1 (Proximal operator for the Group Lasso)**.** *Consider a matrix* $\mathbf{U}$ *and call* $\mathbf{Z} = \mathit{Prox}_{\lambda\|.\|_{1,2}}[\mathbf{U}]$*. Then, for all row* $\mathbf{Z}^j$ *of* $\mathbf{Z}$*,*

$$\mathbf{Z}^j = \max\left(1 - \frac{\lambda}{\|\mathbf{U}^j\|_2}, 0\right) \mathbf{U}^j. \tag{3.10}$$

Unfortunately, the procedure used to design the groups S_i does not yield a differentiable relation between the denoised image $\hat{\mathbf{x}}$ and the parameters to learn. Therefore, we relax the hard clustering assumption into a soft one, which is able to exploit a similarity matrix $\boldsymbol{\Sigma}$ representing pairwise relations between patches. Details about $\boldsymbol{\Sigma}$ are given in Section 3.3.3. Yet, such a relaxation does not provide distinct groups of patches, preventing us from using the Group Lasso penalty (3.9).

This difficulty may be solved by introducing a joint relaxation of the Group Lasso penalty and its proximal operator. First, we consider a similarity matrix $\boldsymbol{\Sigma}$ that encodes the hard clustering assignment used by LSSC—that is, $\boldsymbol{\Sigma}_{ij} = 1$ if j is in S_i and 0 otherwise. Second, we note that $\|\mathbf{A}_i\|_{1,2} = \|\mathbf{A}\,\mathrm{diag}(\boldsymbol{\Sigma}_i)\|_{1,2}$ where $\boldsymbol{\Sigma}_i$ is the i-th column of $\boldsymbol{\Sigma}$ that encodes the i-th cluster membership. Then, we adapt LISTA to problem (3.9), with a different shrinkage parameter $\boldsymbol{\Lambda}_j^{(k)}$ per coordinate j and per iteration k as in Section 3.3.1, which yields

$$\begin{aligned} \mathbf{B} &\leftarrow \mathbf{A}^{(k)} + \mathbf{C}^\top(\mathbf{Y}^c - \mathbf{D}\mathbf{A}^{(k)}) \\ \mathbf{A}_{ij}^{(k+1)} &\leftarrow \max\left(1 - \frac{\Lambda_j^{(k)}\sqrt{\|\boldsymbol{\Sigma}_i\|_1}}{\|(\mathbf{B}\,\mathrm{diag}(\boldsymbol{\Sigma}_i)^{\frac{1}{2}})^j\|_2}, 0\right) \mathbf{B}_{ij}, \end{aligned} \tag{3.11}$$

where the second update is performed for all i, j, the superscript ^{j} denotes the j-th row of a matrix, as above, and $\mathbf{A}_{ij}$ is simply the j-th entry of $\boldsymbol{\alpha}_i$.

We are now in shape to relax the hard clustering assumption by allowing any similarity matrix $\boldsymbol{\Sigma}$ in (3.11), leading to a relaxation of the Group Lasso penalty in Algorithm 4. The resulting model is able to encourage similar patches to share similar sparsity patterns, while being trainable by minimization of the cost (3.7).

3.3.3 Similarity Metrics

We have computed similarities $\boldsymbol{\Sigma}$ in various manners, and implemented the following practical heuristics, which improve the computional complexity.

Online averaging of similarity matrices. As shown in Algorithm 4, we use a convex combination of similarity matrices (using ν_k in $[0,1]$, also learned by backpropagation), which provides better results than computing the similarity on the current estimate only. This is expected since the current estimate $\hat{\mathbf{x}}$ may have lost too much signal information to compute accurately similarities, whereas online averaging allows retaining information from the original signal. We run an ablation study of our model reported in appendix to illustrate the need of similarity refinements during the iterations. When they are no updates the model perfoms on average 0.15 dB lower than with 4 updates.

Semi-local grouping. As in all methods that exploit non-local self similarities in images, we restrict the search for similar patches to $\mathbf{y}_i$ to a window of size $w \times w$ centered around the patch. This approach is commonly used to reduce the size of the similarity matrix and the global memory cost of the method. This means that we will always have $\mathbf{\Sigma}_{ij} = 0$ if pixels i and j are too far apart.

Learned distance. We always use a similarity function of the form $\mathbf{\Sigma}_{ij} = e^{-d_{ij}}$, where d_{ij} is a distance between patches i and j. As in classical deep learning models using non-local approaches [93], we do not directly use the ℓ_2 distance between patches. Specifically, we consider

$$d_{ij} = \| \operatorname{diag}(\boldsymbol{\kappa})(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)\|^2, \tag{3.12}$$

where $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ are the i and j-th patches from the current denoised image, and $\boldsymbol{\kappa}$ in $\mathbb{R}^m$ is a set of weights, which are learned by backpropagation.

3.3.4 Extension to Blind Denoising and Parameter Sharing

The regularization parameter λ of Eq. (3.1) depends on the noise level. In a blind denoising setting, it is possible to learn a shared set of dictionnaries $\{\mathbf{D}, \mathbf{C}, \mathbf{W}\}$ and a set of different regularization parameters $\{\Lambda_{\sigma_0}, \ldots, \Lambda_{\sigma_n}\}$ for various noise intensities. At inference time, we use first a noise estimation algorithm from [121] and then select the best regularization parameter to restore the image.

3.3.5 Extension to Demosaicking

Most modern digital cameras acquire color images by measuring only one color channel per pixel, red, green, or blue, according to a specific pattern called the Bayer pattern. Demosaicking is the processing step that reconstruct a full color image given these incomplete measurements.

Originally addressed by using interpolation techniques [122], demosaicking has been successfully tackled by sparse coding [91] and deep learning models. Most of them such as [97, 98] rely on generic architectures and black box models that do not encode a priori knowledge about the problem, whereas the authors of [96] propose an iterative algorithm that relies on the physics of the acquisition process. Extending our model to demosaicking (and in fact to other inpainting tasks with small holes) can be achieved by introducing a mask $\mathbf{M}_i$ in the formulation for unobserved pixel values. Formally we define $\mathbf{M}_i$ for patch i as a vector in $\{0,1\}^m$, and $\mathbf{M} = [\mathbf{M}_0, \ldots, \mathbf{M}_N]$ in $\{0,1\}^{n \times N}$ represents all masks. Then, the sparse coding formulation becomes

$$\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{M} \odot (\mathbf{Y}^c - \mathbf{D}\mathbf{A})\|_{\mathrm{F}}^2 + \sum_{i=1}^{N} \Psi_i(\mathbf{A}), \tag{3.13}$$

where $\odot$ denotes the elementwise product between two matrices. The first updating rule of equation (3.11) is modified accordingly. This lead to a different update which has the effect of discarding reconstruction error of masked pixels,

$$\mathbf{B} \leftarrow \mathbf{A}^{(k)} + \mathbf{C}^\top(\mathbf{M} \odot (\mathbf{Y}^c - \mathbf{D}\mathbf{A}^{(k)})). \tag{3.14}$$

3.3.6 Practical variants and implementation

Finally, we discuss other practical variants and implementation details.

Dictionary initialization. A benefit of designing an architecture with a sparse coding interpretation, is that the parameters $\mathbf{D}, \mathbf{C}, \mathbf{W}$ can be initialized with a classical dictionary learning approach, instead of using random weights, which makes the initialization robust. To do so, we use SPAMS toolbox [123].

Block processing and dealing with border effects. The size of the tensor $\mathbf{\Sigma}$ grows quadratically with the image size, which requires processing sequentially image blocks. Here, the block size is chosen to match the size w of the non local window, which requires taking into account two important details:

(i) Pixels close to the image border belong to fewer patches than those from the center, and thus receive less estimates in the averaging procedure. When processing images per block, it is thus important to have a small overlap between blocks, such that the number of estimates per pixel is consistent across the image.

(ii) We also process image blocks for training. It then is important to take border effects into account, by rescaling the loss by the number of pixel estimates.

3.4 Experiments

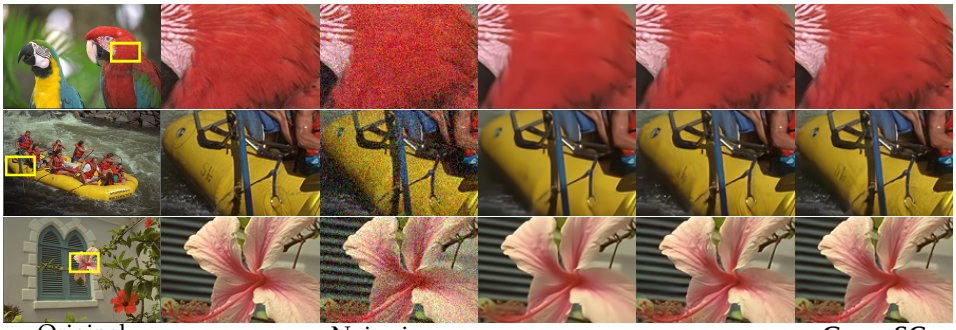


Figure 3.5: Color denoising results for 3 images from the Kodak24 dataset. Best seen in color by zooming on a computer screen. More qualitative results for other tasks are in appendix.

Training details and datasets. In our experiments, we adopt the setting of [95], which is the most standard one used by recent deep learning methods, allowing a simple and fair comparison. In particular, we use as a training set a subset of

[1]We run here the model with the code provided by the authors online on the smaller training set BSD400.

Table 3.1: **Blind denoising** on CBSD68, training on CBSD400. Performance is measured in terms of average PSNR. SSIMs are in the appendix. Best is in bold, second is underlined.

Noise level	CBM3D[63] -	CDnCNN-B [95] 666k	CUNet[113] 93k	CUNLnet[113] 93k	**SC** 115k	**GroupSC** 115k
5	40.24	40.11	40.31	<u>40.39</u>	40.30	**40.43**
10	35.88	36.11	36.08	<u>36.20</u>	36.07	**36.29**
15	33.49	33.88	33.78	<u>33.90</u>	33.72	**34.01**
20	31.88	<u>32.36</u>	32.21	32.34	32.11	**32.41**
25	30.68	<u>31.22</u>	31.03	31.17	30.91	**31.25**

Table 3.2: **Color denoising** on CBSD68, training on CBSD400 for all methods except CSCnet (Waterloo+CBSD400). Performance is measured in terms of average PSNR. SSIMs are reported in the appendix.

Method	Trainable	Params	Noise level (σ) 5	10	15	25	30	50
CBM3D [88]	✗	-	40.24	-	33.49	30.68	-	27.36
CSCnet [105]		186k	-	-	33.83	31.18	-	28.00
CNLNet[92]		-	-	-	33.69	30.96	-	27.64
FFDNET [114]		486k	-	-	33.87	31.21	-	27.96
CDnCNN [95]		668k	40.50	36.31	33.99	31.31	-	28.01
RNAN [98]		8.96M	-	**36.60**	-	-	**30.73**	**28.35**
SC (baseline)		119k	40.44	-	33.75	30.94	-	27.39
GroupSC (ours)		119k	<u>40.58</u>	<u>36.40</u>	<u>34.11</u>	<u>31.44</u>	<u>30.58</u>	<u>28.05</u>

the Berkeley Segmentation Dataset (BSD) [128], called BSD400. We evaluate our models on 3 popular benchmarks: BSD68 (with no overlap with BSD400), Kodak24, and Urban100 [129] and on Classic5 for Jpeg deblocking, following [124, 130]. For gray denoising and Jpeg deblocking we choose a patch size of 9×9 and dictionary with 256 atoms for our models, whereas we choose a patch size of 7×7 for color denoising and demosaicking. For all our experiments, we randomly extract patches of size 56×56 whose size equals the neighborhood for non-local operations and optimize the parameters of our models using ADAM [131]. Similar to [105], we normalize the initial dictionnary $\mathbf{D}_0$ by its largest singular value, which helps the LISTA algorithm to converge. We also implemented a backtracking strategy that automatically decreases the learning rate by a factor 0.5 when the training loss diverges. Additional training details can be found in the appendix for reproductibility purposes.

Performance measure. We use the PSNR as a quality measure, but SSIM scores for our experiments are provided in the appendix, leading to similar conclusions.

Grayscale Denoising. We train our models under the same setting as [95, 92, 93]. We corrupt images with synthetic additive gaussian noise with a variance $\sigma = \{5, 15, 25, 50\}$ and train a different model for each σ and report the performance in terms of PSNR. Our method appears to perform on par with DnCNN for $\sigma \geq 10$ and performs significantly better for low-noise settings. Finaly we provide results on other datasets in the appendix. On BSD68 the light version of our method runs 10 times faster than NLRN [93] (2.17s for groupSC and 21.02s for NLRN), see the

Table 3.3: **Grayscale Denoising** on BSD68, training on BSD400 for all methods except CSCnet (Waterloo+BSD400). Performance is measured in terms of average PSNR. SSIMs are reported in the appendix.

Method	Trainable	Params	Noise Level (σ)			
			5	15	25	50
BM3D [88]	✗	-	37.57	31.07	28.57	25.62
LSSC [91]	✗	-	37.70	31.28	28.71	25.72
BM3D PCA [63]	✗	-	37.77	31.38	28.82	25.80
TNRD [116]		-	-	31.42	28.92	25.97
CSCnet [105]		62k	37.84	31.57	29.11	26.24
CSCnet(BSD400) [105][2]		62k	37.69	31.40	28.93	26.04
LKSVD [107]		45K	-	31.54	29.07	26.13
NLNet [92]		-	-	31.52	29.03	26.07
FFDNet [114]		486k	-	31.63	29.19	26.29
DnCNN [95]		556k	37.68	<u>31.73</u>	29.22	26.23
N3 [94]		706k	-	-	<u>29.30</u>	<u>26.39</u>
NLRN [93]		330k	<u>37.92</u>	**31.88**	**29.41**	**26.47**
SC (baseline)		68k	37.84	31.46	28.90	25.84
GroupSC (ours)		68k	**37.95**	31.71	29.20	26.17

Table 3.4: **Jpeg artefact reduction** on Classic5 with training on CBSD400. Performance is measured in terms of average PSNR. SSIMs are reported in the appendix.

Quality factor	jpeg	SA-DCT [124]	TNRD[116]	DnCNN-3 [95]	SC	GroupSC
10	27.82	28.88	29.28	<u>29.40</u>	29.39	**29.61**
20	30.12	30.92	30.12	<u>31.63</u>	31.58	**31.78**
30	31.48	32.14	31.47	<u>32.91</u>	32.80	**33.06**
40	32.43	33.00	-	<u>33.75</u>	33.75	**33.91**

Table 3.5: **Demosaicking.** Training on CBSD400 unless a larger dataset is specified between parenthesis. Performance is measured in terms of average PSNR. SSIMs are reported in the appendix.

Method	Trainable	Params	Kodak24	BSD68	Urban100
LSSC	✗	-	41.39	40.44	36.63
IRCNN [97] (BSD400+Waterloo [125])		-	40.54	39.9	36.64
Kokinos [126] (MIT dataset [127])		380k	41.5	-	-
MMNet [96] (MIT dataset [127])		380k	42.0	-	-
RNAN [98]		8.96M	**42.86**	<u>42.61</u>	-
SC (ours)		119k	42.34	41.88	37.50
GroupSC (ours)		119k	<u>42.71</u>	**42.91**	**38.21**

appendix for detailed experiments concerning the running time our our method ans its variants.

Color Image Denoising We train our models under the same setting as [92, 95]; we corrupt images with synthetic additive gaussian noise with a variance $\sigma = \{5, 10, 15, 25, 30, 50\}$ and we train a different model for each variance of noise. For reporting both qualitative and quantitative results of BM3D-PCA [63] and DnCNN [95] we used the implementation realeased by the authors. For the other methods we provide the numbers reported in the corresponding papers. We report the performance of our model in Table 3.2 and report qualitative results in Fig-

ure 3.5, along with those of competitive approaches, and provide results on other datasets in the appendix. Overall, it seems that RNAN performs slightly better than GroupSC, at a cost of using 76 times more parameters.

Blind Color Image Denoising. We compare our model with [113, 95, 63] and report our results in Table 3.1. [113] trains two different models in the range [0,25] and [25,50]. We compare with their model trained in the range [0,25] for a fair comparaison. We use the same hyperparameters than the one used for color denoising experiments. Our model performs consistently better than other methods.

Demosaicking. We follow the same experimental setting as IRCNN [97], but we do not crop the output images similarly to [97, 91] since [98] does not seem to perform such an operation according to their code online. We compare our model with sate-of-the-art deep learning methods [126, 96, 98] and also report the performance of LSSC. For the concurrent methods we provide the numbers reported in the corresponding papers. On BSD68, the light version of our method(groupsc) runs at about the same speed than RNAN for demosaicking (2.39s for groupsc and 2.31s for RNAN). We observe that our baseline provides already very good results, which is surprising given its simplicity, but suffers from more visual artefacts than GroupSC (see Fig. 3.1). Compared to RNAN, our model is much smaller and shallower (120 layers for RNAN and 24 iterations for ours). We also note that CSR performs poorly in comparison with groupSC.

Compression artefacts reduction. For jpeg deblocking, we compare our approach with state-of-the-art methods using the same experimental setting: we only restore images in the Y channel (YCbCr space) and train our models on the CBSD400 dataset. Our model performs consistently better than other approaches.

3.5 Centralised Sparse Representation

A different approach to take into account self similarities in sparse models is the CSR approach of [89]. This approach is easier to turn into a differentiable algorithm than the LSSC method, but we have empirically observed that it does not perform as well. Nevertheless, we believe it to be conceptually interesting, and we provide a brief description below. The idea consists of regularizing each code $\boldsymbol{\alpha}_i$ with the function

$$\Psi_i(\boldsymbol{\alpha}_i) = \|\boldsymbol{\alpha}_i\|_1 + \gamma\|\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\|_1, \tag{3.15}$$

where $\boldsymbol{\beta}_i$ is obtained by a weighted average of prevous codes. Specifically, given some codes $\boldsymbol{\alpha}_i^{(k)}$ obtained at iteration k and a similarity matrix $\boldsymbol{\Sigma}$, we compute

$$\boldsymbol{\beta}_i^{(k)} = \sum_j \frac{\boldsymbol{\Sigma}_{ij}}{\sum_l \boldsymbol{\Sigma}_{il}} \boldsymbol{\alpha}_j^{(k)}, \tag{3.16}$$

and the weights $\boldsymbol{\beta}_i^{(k)}$ are used in (3.15) in order to compute the codes $\boldsymbol{\alpha}_i^{(k+1)}$. Note that the original CSR method of [89] uses similarities of the form $\boldsymbol{\Sigma}_{ij} = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{W}\boldsymbol{\alpha}_i - \mathbf{W}\boldsymbol{\alpha}_j\|_2^2\right)$, but other similarities functions may be used.

Even though [89] does not use a proximal gradient descent method to solve the problem regularized with (3.15), the next proposition shows that it admits a closed form, which is a key to turn CSR into a differentiable algorithm. To the best of our knowledge, this expression is new; its proof is given in the appendix.

Proposition 1 (Proximal operator of the CSR penalty)**.** *Consider* Ψ_i *defined in (3.15). Then, for all* $\mathbf{u}$ *in* $\mathbb{R}^p$*,*

$$Prox_{\lambda\Psi_i}[\mathbf{u}] = S_\lambda\big(S_{\lambda\gamma}\left(\mathbf{u} - \boldsymbol{\beta}_i - \lambda\,\mathrm{sign}(\boldsymbol{\beta}_i)\right) + \boldsymbol{\beta}_i + \lambda\,\mathrm{sign}(\boldsymbol{\beta}_i)\big),$$

where S_λ *is the soft-thresholding operator, see Figure 3.6.*

Despite the apparent complexity of the formula, it remains a continuous function of the input and is differentiable almost everywhere, hence compatible with end-to-end training. Qualitatively, the shape of the proximal mapping has a simple interpretation. It pulls codes either to zero, or to the code weighted average $\boldsymbol{\beta}_i$.

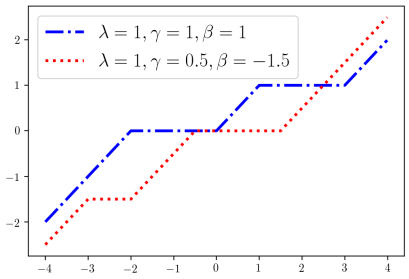


Figure 3.6: $\mathrm{Prox}_{\lambda\Psi_i}$ for various λ, γ, β

At each iteration, the similarity matrix is updated along with the codes $\boldsymbol{\beta}_i$. The proximal operator can then easily be plugged into our framework. We reported performance of the CSR approach in the main paper for grayscale denoising, color denoising and demosaicking. Performance of the CSR approach are reported in Tables 3.6, 3.7, 3.8. We observe that it performs significantly better than the baseline SC but is not as effective as GroupSC overall.

Table 3.6: **Color denoising** on CBSD68, training on CBSD400 for all methods except CSCnet (Waterloo+CBSD400). Performance is measured in terms of average PSNR. SSIMs are reported in the appendix.

Method	Trainable	Params	Noise level (σ) 5	10	15	25	30	50
CBM3D [88]	✗	-	40.24	-	33.49	30.68	-	27.36
CSCnet [105]		186k	-	-	33.83	31.18	-	28.00
CNLNet[92]		-	-	-	33.69	30.96	-	27.64
FFDNET [114]		486k	-	-	33.87	31.21	-	27.96
CDnCNN [95]		668k	40.50	36.31	33.99	31.31	-	28.01
RNAN [98]		8.96M	-	**36.60**	-	-	**30.73**	**28.35**
SC (baseline)		119k	40.44	-	33.75	30.94	-	27.39
CSR (ours)		119k	40.53	-	34.05	31.33	-	28.01
GroupSC (ours)		119k	<u>40.58</u>	<u>36.40</u>	<u>34.11</u>	<u>31.44</u>	<u>30.58</u>	<u>28.05</u>

3.6 Conclusion

We have presented a differentiable algorithm based on non-local sparse image models, which performs on par or better than recent deep learning models, while using significantly less parameters. We believe that the performance of such approaches—including the simple SC baseline—is surprising given the small model size, and given the fact that the algorithm can be interpreted as a single sparse coding layer operating on fixed-size patches. This observation paves the way for future work for sparse coding models that should be able to model the local stationarity of natural images at multiple scales, which we expect should perform even better. We believe

Table 3.7: **Grayscale Denoising** on BSD68, training on BSD400 for all methods except CSCnet (Waterloo+BSD400). Performance is measured in terms of average PSNR. SSIMs are reported in the appendix.

Method	Trainable	Params	Noise Level (σ) 5	15	25	50
BM3D [88]	✗	-	37.57	31.07	28.57	25.62
LSSC [91]	✗	-	37.70	31.28	28.71	25.72
BM3D PCA [63]	✗	-	37.77	31.38	28.82	25.80
TNRD [116]		-	-	31.42	28.92	25.97
CSCnet [105]		62k	37.84	31.57	29.11	26.24
CSCnet(BSD400) [105][2]		62k	37.69	31.40	28.93	26.04
LKSVD [107]		45K	-	31.54	29.07	26.13
NLNet [92]		-	-	31.52	29.03	26.07
FFDNet [114]		486k	-	31.63	29.19	26.29
DnCNN [95]		556k	37.68	<u>31.73</u>	29.22	26.23
N3 [94]		706k	-	-	<u>29.30</u>	<u>26.39</u>
NLRN [93]		330k	<u>37.92</u>	**31.88**	**29.41**	**26.47**
SC (baseline)		68k	37.84	31.46	28.90	25.84
CSR (ours)		68k	37.88	31.64	29.16	26.08
GroupSC (ours)		68k	**37.95**	31.71	29.20	26.17

that our work also confirms that model-based image restoration principles developed about a decade ago are still useful to improve current deep learning models and are a key to push their current limits.

3.a Appendix

This supplementary material is organized as follows: In Section 3.a.1, we provide implementation details that are useful to reproduce the results of our paper (note that the code is also provided). In Section 3.a.2, we present additional quantitative results that were not included in the main paper for space limitation reasons; we notably provide the SSIM quality metric [132] for grayscale, color, and demosaicking experiments; the SSIM score is sometimes more meaningful than PSNR (note that the conclusions presented in the main paper remain unchanged, except for grey image denoising, where our method becomes either closer or better than NLRN, whereas it was slightly behind in PSNR); we also present ablation studies and provide additional baselines for demosaicking and denoising. Section 3.a.3 is devoted to the proof of Proposition 1, and finally in Section 3.a.4, we present additional qualitative results (which require zooming on a computer screen). Finally, in section 3.a.5 we included Visualizations of parameters learned by our model to provide better intuition regarding our approach.

3.a.1 Implementation Details and Reproducibility

Training details. During training, we randomly extract patches 56×56 whose size equals the window size used for computing non-local self similiarities. We apply a mild data augmentation (random rotation by 90° and horizontal flips). We optimize the parameters of our models using ADAM [131] with a minibatch size of 32. All the models are trained for 300 epochs for denoising and demosaicking. The learning rate is set to 6×10^{-4} at initialization and is sequentially lowered during training

Table 3.8: **Demosaicking.** Training on CBSD400 unless a larger dataset is specified between parenthesis. Performance is measured in terms of average PSNR. SSIMs are reported in the appendix.

Method	Trainable	Params	Kodak24	BSD68	Urban100
LSSC	✗	-	41.39	40.44	36.63
IRCNN [97] (BSD400+Waterloo [125])		-	40.54	39.9	36.64
Kokinos [126] (MIT dataset [127])		380k	41.5	-	-
MMNet [96] (MIT dataset [127])		380k	42.0	-	-
RNAN [98]		8.96M	**42.86**	<u>42.61</u>	-
SC (ours)		119k	42.34	41.88	37.50
CSR (ours)		119k	42.25	-	-
GroupSC (ours)		119k	<u>42.71</u>	**42.91**	**38.21**

by a factor of 0.35 every 80 training steps, in the same way for all experiments. Similar to [105], we normalize the initial dictionary $\mathbf{D}_0$ by its largest singular value, which helps the LISTA algorithm to converge faster. We initialize the matrices $\mathbf{C}$,$\mathbf{D}$ and $\mathbf{W}$ with the same value, similarly to the implementation of [105] released by the authors. [2] Since too large learning rates can make the model diverge (as for any neural network), we have implemented a backtracking strategy that automatically decreases the learning rate by a factor 0.8 when the loss function increases too much on the training set, and restore a previous snapshot of the model. Divergence is monitored by computing the loss on the training set every 20 epochs. Training the GroupSC model for color denoising takes about 2 days on a Titan RTX GPU.

Accelerating inference. In order to make the inference time of the non-local models faster, we do not update similarity maps at every step: we update patch similarities every $1/f$ steps, where f is the frequency of the correlation updates. We summarize in Table 4.9 the set of hyperparameters that we selected for the experiments reported in the main tables.

Table 3.9: Hyper-parameters chosen for every task.

Experiment	Color denoising	Gray denoising	Demosaicking	Jpeg Deblocking
Patch size	7	9	7	9
Dictionary size	256	256	256	256
Nr epochs	300	300	300	300
Batch size	32	32	32	32
K iterations	24	24	24	24
Middle averaging	✓	✓	✓	✓
Correlation upd frequency f	1/6	1/6	1/8	1/6

3.a.2 Additional Quantitative Results and Ablation Studies

3.a.2.1 Results on Other Datasets and SSIM Scores

We provide additional grayscale denoising results of our model on the datasets BSD68, Set12, and Urban100 in terms of PSNR and SSIM in Table 3.10. Then, we present additional results for color denoising in Table 3.11, for demosaicking in

[2] The implementation of CSCnet [105] is available here `https://github.com/drorsimon/CSCNet/`.

Table 3.10, and for jpeg artefact reduction in Table 3.12. Note that we report SSIM scores for baseline methods, either because they report SSIM in the corresponding papers, or by running the code released by the authors.

Table 3.10: **Grayscale denoising** results on different datasets. Training is performed on BSD400. Performance is measured in terms of average PSNR (left number) and SSIM (right number).

Dataset	Noise	BM3D	DnCNN 556k	NLRN 330k	GroupSC 68k
Set12	15	32.37/0.8952	32.86/0.9031	**33.16/0.9070**	32.85/0.9063
	25	29.97/0.8504	30.44/0.8622	**30.80/0.8689**	30.44/0.8642
	50	26.72/0.7676	27.18/0.7829	**27.64/0.7980**	27.14/0.7797
BSD68	15	31.07/0.8717	31.73/0.8907	**31.88**/0.8932	31.70/**0.8963**
	25	28.57/0.8013	29.23/0.8278	**29.41**/0.8331	29.20/**0.8336**
	50	25.62/0.6864	26.23/0.7189	**26.47/0.7298**	26.18/0.7183
Urban100	15	32.35/0.9220	32.68/0.9255	**33.45/0.9354**	32.72/0.9308
	25	29.70/0.8777	29.91/0.8797	**30.94/0.9018**	30.05/0.8912
	50	25.95/0.7791	26.28/0.7874	**27.49/0.8279**	26.43/0.8002

Table 3.11: **Color denoising** results on different datasets. Training is performed on CBSD400. Performance is measured in terms of average PSNR (left number) or SSIM (right number).

Dataset	Noise	CDnCNN 668k	GroupSC 119k
Kodak24	15	34.84/0.9233	**35.00/0.9275**
	25	32.34/0.8812	**32.51/0.8867**
	50	29.15/0.7985	**29.19/0.7993**
CBSD68	15	33.98/0.9303	**34.11/0.9353**
	25	31.31/0.8848	**31.44/0.8917**
	50	28.01/0.7925	**28.05/0.7974**
Urban100	15	34.11/0.9436	**34.14/0.9461**
	25	31.66/0.9145	**31.69/0.9178**
	50	28.16/0.8410	**28.23/0.8513**

Table 3.12: **Jpeg artefact reduction** on Classic5 with training on CBSD400. Performance is measured in terms of average PSNR.

Quality factor	AR-CNN [130]	TNRD[116]	DnCNN-3 [95]	GroupSC
10	29.04/0.7929	29.28/0.7992	29.40/0.8026	**29.61/ 0.8166**
20	31.16/0.8517	31.47/0.8576	31.63/0.8610	**31.78/ 0.8718**
30	32.52/0.8806	32.78/0.8837	32.91/0.8861	**33.06/ 0.8959**
40	33.34/0.8953	-	33.75/0.9003	**33.91/ 0.9093**

3.a.2.2 Inference Speed and Similarity Refinements

In table 4.10, we provide a comparison of our model in terms of speed. We compare our model for demosaicking and color denoising with the methods NLRN. This

Table 3.13: **Demosaicking** results. Training on CBSD400 unless a larger dataset is specified between parenthesis. Performance is measured in terms of average PSNR (left) and SSIM (right).

Method	Params	**Kodak24**	**BSD68**	**Urban100**
IRCNN (BSD400+Waterloo)	107k	40.54/0.9807	39.96/0.9850	36.64/0.9743
GroupSC (CBSD400) (ours)	118k	**42.71/0.9901**	**42.91/0.9938**	**38.21/0.9804**

study shows how to balance the trade-off between speed and accuracy. Whereas the best model in accuracy achieves 31.71dB in PSNR with about 30s per image, a "light" version can achieve 31.67dB in only 2.35s per image. This ablation study also illustrates the need of similarity refinements during the iterations. When they are no updates the model perfoms on average 0.15 dB lower than with 4 updates.

Table 3.14: **Inference time (s)** per image / PSNR (in dB) for gray denoising task with $\sigma = 15$, computed on BSD68. Inference time is measured using a Titan RTX gpu.

Middle averaging (6)	$f_{\hat{\Sigma}}$	Stride between image blocks			
		$s = 56$	$s = 48$	$s = 24$	$s = 12$
✗	∞	1.30 / 31.29	1.75 / 31.57	6.00 / 31.58	22.57 / 31.59
	12	1.41 / 31.36	1.85 / 31.64	6.57 / 31.66	24.44 / 31.66
	8	1.51 / 31.37	2.90 / 31.65	7.06 / 31.68	26.05 / 31.68
	6	1.59 / 31.38	2.15 / 31.65	7.48 / 31.68	27.60 / 31.69
✓	∞	1.30 / 31.29	1.75 / 31.57	6.00 / 31.58	22.57 / 31.59
	12	1.45 / 31.36	1.95 / 31.65	6.82 / 31.66	25.40 / 31.67
	8	1.63 / 31.38	2.17 / 31.66	7.61 / 31.68	27.92 / 31.70
	6	1.77 / 31.39	2.35 / 31.67	8.25 / 31.69	30.05 / 31.71
NLRN	330k	23.02 / 31.88			

3.a.2.3 Influence of Patch and Dictionary Sizes

We measure in Table 3.15 the influence of the patch size and the dictionary size for grayscale image denoising. For this experiment, we run a lighter version of the model groupSC in order to accelerate the training. The batch size was decreased from 25 to 16, the frequency of the correlation updates was decreased from 1/6 to 1/8 and the intermediate patches are not approximated with averaging. These changes accelerate the training but lead to slightly lower performances when compared with the model trained in the standard setting. As can be seen in the table, better performance can be obtained by using larger dictionaries, at the cost of more computation. Note that all other experiments conducted in the paper use a dictionary size of 256. Here as well, a trade-off between speed/number of parameters and accuracy can be chosen by changing this default value.

3.a.2.4 Number of Unrolled Iterations

We also investigated the impact of the depth of the model on the performance. To do so, we conducted a denoising experiment using the light version of our model with a model with various number of unrolled steps. When changing the depth from K=12, to 36, we only measure a difference of 0.02dB.

Table 3.15: **Influence of the dictionary size and the patch size** on the denoising performance. Grayscale denoising on BSD68. Models are trained on BSD400. Models are trained in a light setting to accelerate training.

Noise (σ)	Patch size	n=128	n=256	512
5	k=7	37.91	37.92	-
	k=9	37.90	37.92	37.96
	k=11	37.89	37.89	-
15	k=7	31.60	31.63	-
	k=9	31.62	31.67	31.71
	k=11	31.63	31.67	-
25	k=7	29.10	29.11	-
	k=9	29.12	29.17	29.20
	k=11	29.13	29.18	-

Table 3.16: **Influence of the number of unrolled iterations**.Grayscale denoising on BSD68. Models are trained on BSD400. Models are trained in a light setting to accelerate training.

Model	Unrolled iterations		
SC	28.90	28.91	28.90
GroupSC (light)	29.10	29.12	29.12

3.a.3 Proof of Proposition

The proximal operator of the function $\Psi_i(\mathbf{u}) = \|\mathbf{u}\|_1 + \gamma\|\mathbf{u} - \boldsymbol{\beta}_i\|_1$ for $\mathbf{u}$ in $\mathbb{R}^p$ is defined as

$$\text{Prox}_{\lambda\Psi_i}[\mathbf{z}] = \underset{\mathbf{u}\in\mathbb{R}^p}{\arg\min}\ \frac{1}{2}\|\mathbf{z} - \mathbf{u}\|^2 + \lambda\|\mathbf{u}\|_1 + \lambda\gamma\|\mathbf{u} - \boldsymbol{\beta}_i\|_1$$

The optimality condition for the previous problem is

$$0 \in \nabla(\frac{1}{2}||\mathbf{z} - \mathbf{u}||_2^2) + \partial(\lambda||\mathbf{u}||_1) + \partial(\lambda\gamma||\mathbf{u} - \boldsymbol{\beta}_i||_1)$$

$$\Leftrightarrow 0 \in \mathbf{u} - \mathbf{z} + \lambda\partial||\mathbf{u}||_1 + \lambda\gamma\partial||\mathbf{u} - \boldsymbol{\beta}_i||_1$$

We consider each component separately. We suppose that $\boldsymbol{\beta}_i[j] \neq 0$, otherwise $\Psi_i(\mathbf{u})[j]$ boils down to the ℓ_1 norm. And we also suppose $\lambda, \gamma > 0$.

Let us examine the first case where $u[j] = 0$. The subdifferential of the ℓ_1 norm is the interval $[-1, 1]$ and the optimality condition is

$$\begin{aligned} 0 \in \mathbf{u}[j] - \mathbf{z}[j] + [-\lambda, \lambda] + \lambda\gamma\,\text{sign}(\mathbf{u}[j] - \boldsymbol{\beta}_i[j]) \\ \Leftrightarrow \mathbf{z}[j] \in [-\lambda, \lambda] - \lambda\gamma\,\text{sign}(\boldsymbol{\beta}_i[j]) \end{aligned}$$

Similarly if $\mathbf{u}[j] = \boldsymbol{\beta}_i[j]$

$$\mathbf{z}[j] \in \boldsymbol{\beta}_i[j] + \lambda\,\text{sign}(\boldsymbol{\beta}_i[j]) + [-\lambda\gamma, \lambda\gamma]$$

Finally let us examine the case where $u[j] \neq 0$ and $u[j] \neq \boldsymbol{\beta}_i[j]$: then, $\partial||\mathbf{u}||_1 = \text{sign}(\mathbf{u}[j])$ and $\partial||\mathbf{u} - \boldsymbol{\beta}_i||_1 = \text{sign}(\mathbf{u}[j] - \boldsymbol{\beta}_i[j])$. The minimum $u[j]^*$ is obtained as

$$\begin{aligned} 0 = \mathbf{u}[j] - \mathbf{z}[j] + \lambda\,\text{sign}(\mathbf{u}[j]) + \lambda\gamma\,\text{sign}(\mathbf{u}[j] - \boldsymbol{\beta}_i[j]) \\ \Leftrightarrow \mathbf{u}[j]^* = \mathbf{z}[j] - \lambda\,\text{sign}(\mathbf{u}[j]^*) - \lambda\gamma\,\text{sign}(\mathbf{u}[j]^* - \boldsymbol{\beta}_i[j]) \end{aligned}$$

We study separately the cases where $\mathbf{u}[j] > \boldsymbol{\beta}[j]$, $0 < \mathbf{u}[j] < \boldsymbol{\beta}[j]$ and $\mathbf{u}[j] < 0$ when $\boldsymbol{\beta}_i[j] > 0$ and proceed similarly when $\boldsymbol{\beta}_i < 0$. With elementary operations we can derive the expression of $\mathbf{z}[j]$ for each case. Putting the cases all together we obtain the formula.

3.a.4 Additional Qualitative Results

We show qualitative results for jpeg artefact reduction, color denoising, grayscale denoising, and demosaicking in Figures 3.8, 3.9, 3.10, respectively.

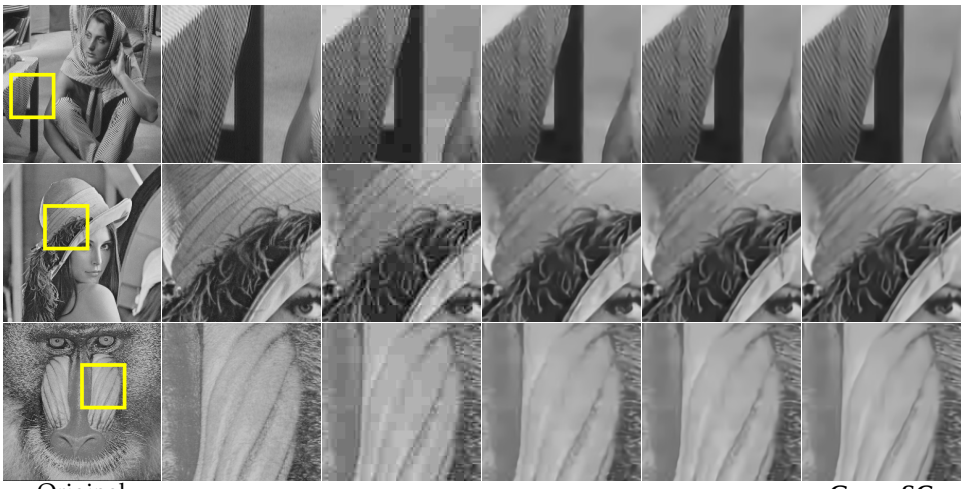


Figure 3.7: Jpeg artefact reduction results for 2 images from the Classic5 dataset. Best seen in color by zooming on a computer screen.

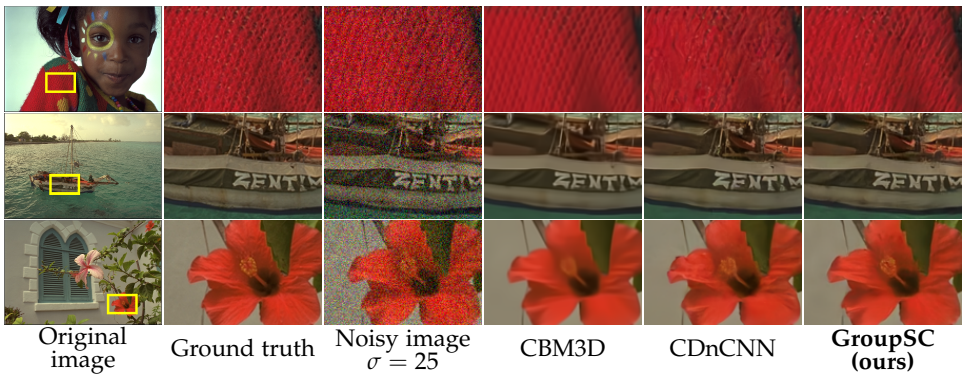


Figure 3.8: Color denoising results for 3 images from the Kodak24 dataset. Best seen in color by zooming on a computer screen. Artefact reduction compared to CDnCNN can be seen in the top and bottom pictures (see in particular the flower's pistil).

Figure 3.9: Grey denoising results for 3 images from the BSD68 dataset. Best seen by zooming on a computer screen. GroupSC's images are slightly more detailed than DnCNN on the top and middle image, whereas DnCNN does subjectively slightly better on the bottom one. Overall, these two approaches perform similarly on this dataset.

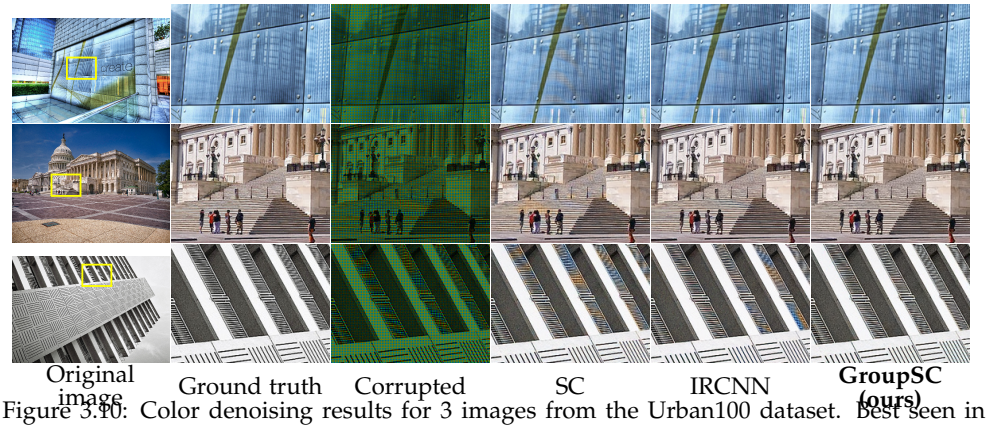


Figure 3.10: Color denoising results for 3 images from the Urban100 dataset. Best seen in color by zooming on a computer screen. On the three images, our approach groupSC exhibits significantly less artefacts than IRCNN and our baseline SC.

3.a.5 Parameters Visualization

We present in this section some visualizations of the learned parameters of our introduced model groupsc for a dnoising task. We reported in Figure 3.12 learned dictionaries $\mathbf{D}$ and $\mathbf{W}$ (model trained with $\mathbf{C} = \mathbf{D}$). We observe that dictionaries are coupled. We reported in Figure 3.13 the sequence of regularization parameters $(\mathbf{\Lambda}_k)_{k=0,1\ldots K-1}$ for a denoising task, and $(\Lambda_{\sigma_0}, \ldots, \Lambda_{\sigma_n})$. for blind denoising. Finally, we reported in Figure 3.11 the learned weights $\boldsymbol{\kappa}$ of the gaussian kernel for comparing patches.

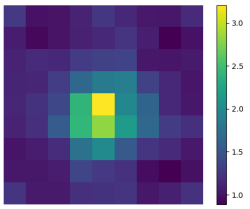


Figure 3.11: Weights $\boldsymbol{\kappa}$ for comparing patches.

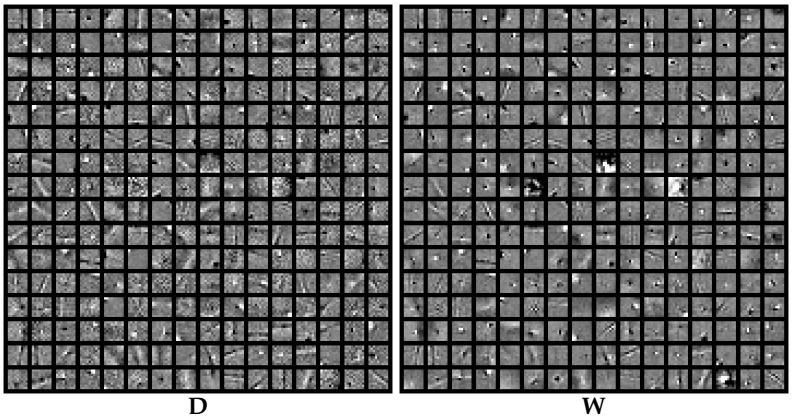
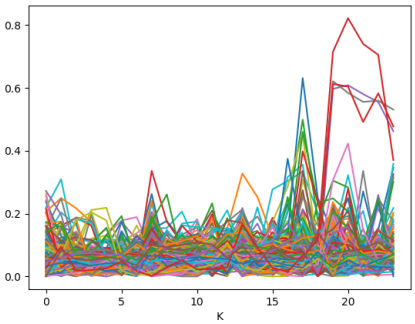
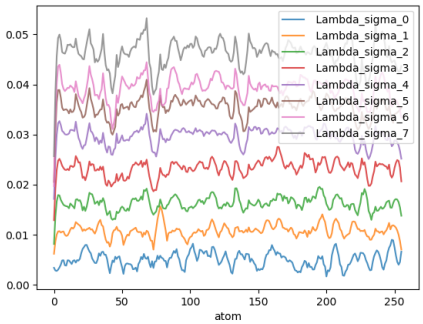


Figure 3.12: Learned dictionnaries of groupSC for denoising.



Sequence of regularization parameters $\mathbf{\Lambda}_i$ of a non-blind models.



Set of regularization parameters $(\Lambda_{\sigma_0}, \ldots, \Lambda_{\sigma_n})$ of a blind model.

Figure 3.13: Learned regularization parameters of groupSC for denoising and blind denoising. Models are trained on BSD400.

Chapter 4

A Framework for Designing Trainable Priors

Chapter abstract:
We introduce a general framework for designing and training neural network layers whose forward passes can be interpreted as solving non-smooth convex optimization problems, and whose architectures are derived from an optimization algorithm. We focus on convex games, solved by local agents represented by the nodes of a graph and interacting through regularization functions. This approach is appealing for solving imaging problems, as it allows the use of classical image priors within deep models that are trainable end to end. The priors used in this presentation include variants of total variation, Laplacian regularization, bilateral filtering, sparse coding on learned dictionaries, and non-local self similarities. Our models are fully interpretable as well as parameter and data efficient. Our experiments demonstrate their effectiveness on a large diversity of tasks ranging from image denoising and compressed sensing for fMRI to dense stereo matching.

B. Lecouat, J. Ponce, J. Mairal. A Flexible Framework for Designing Trainable Priors with Adaptive Smoothing and Game Encoding. *In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Contents

4.1 Introduction

Despite the undeniable successes of deep learning in domains as varied as image processing [95] and recognition [133], natural language processing [134], speech [135] or bioinformatics [136], feed-forward neural networks are often maligned as being "black boxes" that, except perhaps for their top classification or regression layers, are difficult or even impossible to interpret. In imaging applications, for example, the elementary operations typically consist of convolutions and pointwise nonlinearities, with many parameters adjusted by backpropagation, and no obvious functional interpretation.

In this paper, we consider instead network architectures explicitly derived from an optimization algorithm, and thus interpretable from a functional point of view. The first instance of this approach we are aware of is LISTA [69], which provides a fast approximation of sparse coding. Yet, we are not content to design an architecture that provides a fast approximation to a given optimization problem, but we also want to learn a data representation pertinent for the corresponding task. This yields an unusual machine learning paradigm, where one learns the parameters of a parametric objective function used to represent data, while designing an optimization algorithm to minimize it efficiently.

Even though interpretability is not always necessary to achieve good prediction, this point of view, sometimes called algorithm unrolling [119, 137], has proven successful for solving inverse imaging problems, providing effective and parameter-efficient models. This approach allows the use of domain-specific priors within trainable deep models, leading to a large number of applications such as compressive imaging [108, 115], demosaicking [1], denoising [1, 107, 105], and super-resolution [102] .

However, existing approaches are often limited to simple image priors such as sparsity induced by the ℓ_1-norm [105], or differentiable regularization functions [113], and a general algorithmic framework for combining complex, possibly non-smooth, regularization functions is still missing. Our paper addresses this issue and is able to leverage a large class of image priors such as total variation [83], the ℓ_1-norm, structured sparse coding [91], or Laplacian regularization, where local optimization problems interact with each others. The interaction can be local among direct neighbors on an image grid, or non-local, capturing for instance similarities between spatially distant image patches [84, 63].

In this context, we adopt a more general and flexible point of view than the standard convex optimization paradigm, and consider formulations to represent data based on non-cooperative games [138] potentially involving non-smooth terms, which are tackled by using the Moreau-Yosida regularization technique [139, 140]. Unrolling the resulting optimization algorithm results in a network architecture that can be trained end-to-end and capture any combination of the domain-specific priors mentioned above. This approach includes and improves upon specific trainable sparse coding models based on the ℓ_1-norm for example [105, 102]. More importantly perhaps, it can be used to construct several interesting new image priors: In particular, we show that a trainable variant of total variation and its non-local variant based on self similarities is competitive with the state of the art in imaging tasks, despite using up to 50 times fewer parameters, with corresponding gains in

speed. We demonstrate the effectivness and the flexibility of our approach on several imaging tasks, namely denoising, compressed fMRI reconstruction, and stereo matching.

Summary of our contributions. First, we provide a new framework for building trainable variants of a large class of domain-specific image priors. Second, we show that several of these priors match or even outperform existing techniques that use a much larger number of parameters and training data. Finally, we present a set of practical tricks to make optimization-driven layers easy to train.

4.2 Background and Related Work

Classical image priors. Inverse imaging problems are often solved by minimizing a data fitting term with respect to model parameters, regularized with a penalty that encourages solutions with a particular structure. In image processing, the community long focused on designing handcrafted priors such as sparse coding on learned dictionaries [106, 60], diffusion operators [82], total variation [83], and non-local self similarities [84], which is a key ingredient of successful restoration algorithms such as BM3D [63]. However these methods are now often outperformed by deep learning models [93, 95, 114], which leverage pairs of corrupted/clean training images in a supervised fashion.

Bilevel optimization. A simple method for mixing data representation learning with optimization is to use a bi-level formulation [101]. For instance, assuming that one is given pairs $(\mathbf{x}_i, \mathbf{y}_i)_{i=1\ldots n}$ of corrupted/clean signals with $\mathbf{x}_i$ and $\mathbf{y}_i$ in $\mathbb{R}^m$, one may consider the following bi-level objective

$$\min_{\boldsymbol{\theta}\in\Theta, \mathbf{W}\in\mathbb{R}^{m\times p}} \frac{1}{n}\sum_{i=1}^{n} L(\mathbf{y}_i, \mathbf{W}\boldsymbol{\alpha}^\star_{\boldsymbol{\theta}}(\mathbf{x}_i)) \quad \text{where} \quad \boldsymbol{\alpha}^\star_{\boldsymbol{\theta}}(\mathbf{x}_i) \in \operatorname*{arg\,min}_{\boldsymbol{\alpha}\in\mathbb{R}^p} h_{\boldsymbol{\theta}}(\mathbf{x}_i, \boldsymbol{\alpha}), \tag{4.1}$$

where $\boldsymbol{\theta}$ is a set of model parameters, $\mathbf{W}\boldsymbol{\alpha}^\star_{\boldsymbol{\theta}}(\mathbf{x}_i)$ is a prediction which is compared to $\mathbf{y}_i$ through a loss function $L : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$, and the data representation $\boldsymbol{\alpha}^\star_{\boldsymbol{\theta}}(\mathbf{x}_i)$ in $\mathbb{R}^p$ is obtained by minimizing some function $h_{\boldsymbol{\theta}}$. Note that for simplicity, we have considered here a multivariate regression problem, where given a signal $\mathbf{x}$ in $\mathbb{R}^m$, we want to predict another signal $\mathbf{y}$ in $\mathbb{R}^m$, but this formulation also applies to classification problems. It was first introduced for sparse coding in [101, 141] and it has recently been extended to the case when $\boldsymbol{\alpha}^\star_{\boldsymbol{\theta}}(\mathbf{x}_i)$ is replaced by an approximate minimizer of $h_{\boldsymbol{\theta}}$.

Unrolled algorithms. A common approach to solving (4.1) consists in choosing an iterative method for minimizing $h_{\boldsymbol{\theta}}$ and then define $\boldsymbol{\alpha}^\star_{\boldsymbol{\theta}}(\mathbf{x}_i)$ as the output of the optimization method after K iterations. The sequence of operations performed by the optimization method can be seen as a computational graph and $\nabla_{\boldsymbol{\theta}}\mathbf{z}^\star_{\boldsymbol{\theta}}$ can be computed by automatic differentiation. This often yields neural-network-like computational graphs, which we call *optimization-driven layers*. Such architectures have found multiple applications such as training of conditional random fields [142], stabilization of generative adversarial networks [143], structured prediction [144], or hyper-parameters tuning [145]. For image restoration, various optimization problems have been explored including for example sparse coding [1, 105, 115], non linear diffusion [116] and differential operator regularization [113]. Many inference algorithms have been investigated including proximal gradient descent [113, 105], ADMM [119], half quadratic spliting [118], or augmented Lagrangian [120].

4.3 A General Framework for Learning Optimization-Driven Layers

4.3.1 Proposed Approach

We adopt a more general point of view than (4.1), where we assume that input signals admit a local "patch" structure (*e.g.*, rectangular image regions) and the data representation encodes individual patches. Assuming that there are m patches in $\mathbf{x}$, we denote by $\mathbf{Z}^\star(\mathbf{x}) = [\mathbf{z}_1^\star(\mathbf{x}), \ldots, \mathbf{z}_m^\star(\mathbf{x})]$ in $\mathbb{R}^{p \times m}$ the representation of $\mathbf{x}$ and by $\mathbf{z}_j^\star(\mathbf{x})$ the representation of patch j (we omit the dependency on the model parameters $\boldsymbol{\theta}$ for simplicity). In imaging applications and as in previous models [105], $\mathbf{Z}^\star(\mathbf{x})$ can be seen as a feature map akin to that of a convolutional neural network with p channels.

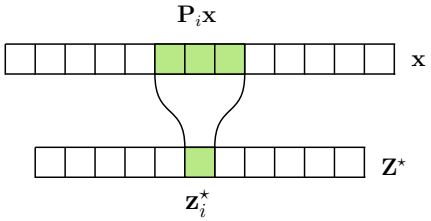


Figure 4.1: Our models encode locally an input feature vector. The local optimal solutions $\mathbf{z}_j^\star$ interact trough the regularization function $\psi_\theta^j(\mathbf{Z})$.

Encoding with non-cooperative convex games. Concretely, given a signal $\mathbf{x}$, we denote by $\mathbf{P}_j\mathbf{x}$ the patch of $\mathbf{x}$ centered at position j, where $\mathbf{P}_j$ is a linear patch extraction operator, and we define the optimal encoding $\mathbf{Z}^\star(\mathbf{x})$ of $\mathbf{x}$ as a Nash equilibrium of the set of problems

$$\min_{\boldsymbol{\alpha}_j \in \mathcal{Z}} h_{\boldsymbol{\theta}}(\mathbf{P}_j\mathbf{x}, \boldsymbol{\alpha}_j) + \psi_{\boldsymbol{\theta}}^j(\mathbf{Z}) \quad \text{for} \quad j = 1, \ldots, m, \tag{4.2}$$

where $h_{\boldsymbol{\theta}}$ is a a convex reconstruction objective for each patch, parametrized by $\boldsymbol{\theta}$, $\psi_{\boldsymbol{\theta}}^j$ is a convex regularization function encoding interactions between the variable $\mathbf{z}_j$ and the remaining ones $\mathbf{z}_l$ for $l \neq j$, and $\mathcal{Z}$ is a convex subset of $\mathbb{R}^p$. When $\mathcal{Z}$ is compact, the problem is a specific instance of a non-cooperative convex game [138], which is known to admit at least one Nash equilibrium—that is, a solution such that one of the objectives in (4.2) is optimal with respect to its variable $\mathbf{z}_j$ when the other variables $\mathbf{z}_l$ for $l \neq j$ are fixed. The conditions under which an optimization algorithm is guaranteed to return such an equilibrium point are well studied, see Section 4.3.3, and in many situations the compactness of $\mathcal{Z}$ is not required, as also observed in our experiments where we choose $\mathcal{Z} = \mathbb{R}^p$. For instance, in several practical cases, (4.2) can be solved by minimizing the sum of m convex terms, a setting called a *potential game*, which boils down to a classical convex optimization problem.

4.3.2 Application of our Framework to Inverse Problems

In this section, we show how to leverage our optimization-driven layers for imaging. For the sake of clarity we choose to narrow down the scope of this presentation to imaging, even though our method is not limited to this single application: different modalities including for example genomic/graph data could benefit from our methodology.

Examples of models $h_{\boldsymbol{\theta}}$. We consider two cases in the rest of this presentation:

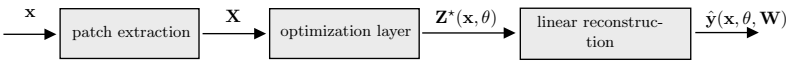


Figure 4.2: Architecture of our trainable models for image restoration.

- *Pixel reconstruction*: $h_{\boldsymbol{\theta}}(\mathbf{P}_j\mathbf{x},\mathbf{z}_j) = (\mathbf{x}_j - \mathbf{z}_j)^2$, where $\mathbf{x}_j$ is the pixel j of $\mathbf{x}$ and $\mathbf{z}_j$ is a scalar, corresponding to patches of size $q = 1 \times 1$ and $p = 1$.
- *Patch encoding on a dictionary*: $h_{\boldsymbol{\theta}}(\mathbf{P}_j\mathbf{x},\mathbf{z}_j) = \|\mathbf{P}_j\mathbf{x}_j - \mathbf{D}\mathbf{z}_j\|^2$, where $\mathbf{D}$ in $\mathbb{R}^{q\times p}$ is a dictionary, q is the patch size, and p is the number of dictionary elements. This is a classical model where patch j is approximated by a linear, often sparse, combination of dictionary elements [106].

Only the second choice involves model parameters $\mathbf{D}$ (represented by $\boldsymbol{\theta}$). These two loss functions are common in image processing [106], but other losses may be used for other modalities.

Linear reconstruction with a dictionary. Assuming that $\mathbf{y}$ and $\mathbf{x}$ have the same size m for simplicity, predicting $\mathbf{y}$ from a feature map $\mathbf{Z}^\star(\mathbf{x})$ is typically achieved by using a learned dictionary matrix $\mathbf{W}$ in $\mathbb{R}^{q\times p}$ where q is the patch size. Then, $\mathbf{W}\mathbf{z}_j^\star(\mathbf{x})$[1] can be interpreted as a reconstruction of the j-patch of $\mathbf{y}$. Since the patches overlap, we obtain q estimators for every pixel, which can be combined by averaging (neglecting border effects below for simplicity), yielding the prediction

$$\hat{\mathbf{y}}(\mathbf{x},\theta,\mathbf{W}) = \frac{1}{q}\sum_{j=1}^{m}\mathbf{P}_j^\top\mathbf{W}\mathbf{z}_j^\star(\mathbf{x}), \tag{4.3}$$

where $\mathbf{P}_j^\top$ is the linear operator that places a patch of size q at position j in a signal of dimension m. Patch averaging is a classical operation in patch-based image restoration algorithms, see [106], which can be interpreted in terms of transposed convolution[2] and admits fast implementations on GPUs.

Learning problem. For image restoration, given training pairs of corrupted / clean images $\{\mathbf{x}_i,\mathbf{y}_i\}_{i=1,\dots,n}$, we consider the regression problem

$$\min_{\boldsymbol{\theta}\in\Theta,\mathbf{W}\in\mathbb{R}^{q\times p}} \|\mathbf{y}_i - \hat{\mathbf{y}}(\mathbf{x}_i,\theta,\mathbf{W})\|^2 \quad \text{where } \hat{\mathbf{y}}(\mathbf{x}_i) \text{ is defined in (4.3).} \tag{4.4}$$

Examples of regularization functions $\psi_{\boldsymbol{\theta}}^j$. Our framework allows the use of several regularization functions, which are presented in the table 4.1 below. We assume that the patches are nodes in a graph, and denote by $\mathcal{N}_j$ the set of neighbors of the patch j. For natural images, the graph may be a two-dimensional grid with edge weights $a_{j,k}$ that depend on the relative position of the patches j and k, which we denote by a_{j-k}, but it may also be a non-local graph based on some similarity function as in [1, 93]. Concretely, we can consider:

- the distance $d_{\mathrm{NL}}^{j,k} = \|\operatorname{diag}(\boldsymbol{\kappa})(\mathbf{P}_j\mathbf{x} - \mathbf{P}_k\mathbf{x})\|^2$ between patches j and k of the image $\mathbf{x}$, where $\boldsymbol{\kappa}$ in $\mathbb{R}^q$ is a set of parameters to learn, and q is the patch size, and we define normalized weights $a_{\mathrm{NL}}^{j,k} = e^{-d_{\mathrm{NL}}^{j,k}}/\sum_{l\in\mathcal{N}_j} e^{-d_{\mathrm{NL}}^{j,k}}$.

[1] We employ a debiasing dictionary $\mathbf{W} \neq \mathbf{D}$ to improve the quality of the reconstructions. Debiaising is commonly used when dealing with ℓ_1 penalty which is known to shrink the coefficients $\mathbf{Z}$ too much.

[2] `torch.nn.functionnal.conv2D_transpose` on PyTorch [146].

- or a distance inspired from the bilateral filter [147]. In that case we define the distance $d_{\text{BL}}^{j,k} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_d^2} + \frac{\|i-j\|^2}{2\sigma_r^2}$ between pixels on a local window $\mathcal{N}_j$ centered around pixel j, and we define again normalized weights $a_{\text{BL}}^{j,k} = e^{-d_{\text{BL}}^{j,k}} / \sum_{l \in \mathcal{N}_j} e^{-d_{\text{BL}}^{j,k}}$.

Table 4.1: A non-exhaustive list of regularization functions $\psi_{\boldsymbol{\theta}}$ covered by our framework.

	$\psi_{\boldsymbol{\theta}}^j(\mathbf{Z})$	Model parameters
Laplacian	$\sum_{k \in \mathcal{N}_j} a_{j-k} \|\mathbf{z}_j - \mathbf{z}_k\|^2$	weights in $\mathbb{R}^{\lvert\mathcal{N}\rvert}$
Non-local Laplacian	$\sum_{k \in \mathcal{N}_j} a_{\text{NL}}^{j,k} \|\mathbf{z}_j - \mathbf{z}_k\|^2$	$\boldsymbol{\kappa}$ in $\mathbb{R}^q$
Bilateral filter (BF)	$\sum_{k \in \mathcal{N}_j} a_{\text{BL}}^{j,k} \|\mathbf{z}_j - \mathbf{z}_k\|^2$	$\sigma_d \in \mathbb{R}$ and $\sigma_r \in \mathbb{R}$
Total variation (TV)	$\sum_{k \in \mathcal{N}_j} a_{j-k} \|\mathbf{z}_j - \mathbf{z}_k\|_1$	weights in $\mathbb{R}^{\lvert\mathcal{N}\rvert}$
Non-local total variation (NLTV)	$\sum_{k \in \mathcal{N}_j} a_{\text{NL}}^{j,k} \|\mathbf{z}_j - \mathbf{z}_k\|_1$	$\boldsymbol{\kappa}$ in $\mathbb{R}^q$
Bilateral TV (BLTV)	$\sum_{k \in \mathcal{N}_j} a_{\text{BL}}^{j,k} \|\mathbf{z}_j - \mathbf{z}_k\|_1$	$\sigma_d \in \mathbb{R}$ and $\sigma_r \in \mathbb{R}$
Weighted ℓ_1-norm (sparse coding)	$\sum_{l=1}^p \lambda_l \lvert\mathbf{z}_j[l]\rvert$	$\boldsymbol{\lambda}$ in $\mathbb{R}^p$
Non-local group regularization	$\sum_{l=1}^p \lambda_l \sqrt{\sum_{k \in \mathcal{N}_j} a_{j,k} \mathbf{z}_k[l]^2}$	$\boldsymbol{\lambda}$ in $\mathbb{R}^p$ and $\boldsymbol{\kappa}$ in $\mathbb{R}^q$
Variance reduction	$\|\mathbf{W}\mathbf{z}_j - \mathbf{P}_j\hat{\mathbf{y}}\|^2$ with $\hat{\mathbf{y}}$ from (4.3)	$\mathbf{W}$ from (4.3)

Novelty of the proposed formulation and relation to previous work.

- *Total variation*: to the best of our knowledge, the basic anisotropic TV penalty [62] does not seem to appear in the literature on unrolled algorithms with end-to-end training. Note also that our TV variant allows learning non-symmetric weights $a_{j,k} \neq a_{k,j}$, leading to a non-cooperative game that goes beyond the classical convex optimization framework typically used with the TV penalty.

- *Non-local TV*: the non-local TV penalty presented above is based on a classical formulation [148], but can be incorporated within a trainable deep network with non-symmetric weights.

- *Bilateral filtering*: the bilateral filter and its TV variant implemented in this paper are based on classical formulations [147, 149]. But they have not, to the best of our knowledge, been implemented as trainable priors.

- *Sparse coding and variance reduction*: the weighted ℓ_1-norm combined with the patch encoding loss $h_{\boldsymbol{\theta}}$ yields a sparse coding formulation (SC) that has been well studied within optimization-driven layers [107, 105]. Yet, the codes $\mathbf{z}_j$ in the SC setup are obtained by solving independent optimization problems, which has motivated by Simon and Elad [105] to propose instead a Convolutional Sparse Coding model (CSC), where the full image is approximated by a linear combination of small dictionary elements. Unfortunately, as noted in [105], CSC leads to ill-conditioned optimization problems, making a hybrid approach between SC and CSC more effective. Our paper proposes an alternative solution combining the weighted ℓ_1-norm regularization with a variance reduction penalty, which forces the codes $\mathbf{z}_j$ to reach a consensus when reconstructing the image $\hat{\mathbf{y}}$. Our experiments show that this approach outperforms [105] for image denoising.

Procedure 4 Pseudocode of the general training procedure for image restoration

Sample a minibatch of pairs of corrupted/clean images $\{(\mathbf{x}_0, \mathbf{y}_0), \cdots, (\mathbf{x}_K, \mathbf{y}_K)\}$;
Extract overlapping patches of corrupted images to form tensors $\mathbf{X}_i = [\mathbf{P}_1\mathbf{x}_i, \cdots, \mathbf{P}_n\mathbf{x}_i]$;
for $t = 1, 2, \ldots, K$ **do** ▷ Compute an approximate Nash equilibrium $\mathbf{Z}^\star$ of the convex games
    $\mathbf{Z}_{t+1} \leftarrow \mathbf{Z}_t - \eta_t H_\theta(\mathbf{Z}_t, \mathbf{X})$;
end for
Approximate clean images by linear reconstruction $\hat{\mathbf{y}} = \frac{1}{q}\sum_{j=1}^m \mathbf{P}_j^\top \mathbf{W}\mathbf{z}_j^\star(\mathbf{X}, \theta)$;
Compute the ℓ_2 reconstruction loss $\|\mathbf{y} - \hat{\mathbf{y}}(\mathbf{x}, \theta, \mathbf{W})\|_2^2$ on the minibatch;
Compute an estimate of the gradients wrt. $(\theta, \mathbf{W})$ with auto-diff;
Update trainable parameters $(\theta, \mathbf{W})$ with Adam;

- *Non-local group regularization*: This regularization function corresponds to a soft variant of the Group Lasso penalty [150], which encourages similar patches to share similar sparsity patterns (set of non-zero elements of the codes $\mathbf{z}_j$). It was originally used in [91] and was recently revisited within optimization-driven layers with an heuristic algorithm [1]. Our paper provides a better justified algorithmic framework as well as the ability to combine this penalty with other ones.

4.3.3 Differentiability and End-to-end Training

In this section we adress end-to-end training of the optimization-driven layers. Given pairs of training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1,\ldots,n}$, we consider the learning problem

$$\min_{\boldsymbol{\theta}\in\Theta} \frac{1}{n}\sum_{i=1}^n L\left(\mathbf{y}_i, g_\theta\left(\boldsymbol{\alpha}_{\boldsymbol{\theta}}^\star(\mathbf{x}_i)\right)\right), \tag{4.5}$$

where g_θ is a differentiable function. We consider the approximation where the codes $\mathbf{z}_j^\star(\mathbf{x})$ are obtained as the K-th step of an optimization algorithm for solving the problem (4.2). To obtain these codes, we leverage (i) iterative gradient and extra-gradient methods, which are classical for solving game problems [151, 152], and (ii) a smoothing technique for dealing with the regularization functions $\psi_{\boldsymbol{\theta}}^j$ above when they are non-smooth. Refer to Algorithm 4 for an overview of the training procedure. We start with the first point when dealing with smooth objectives.

Unrolled optimization for convex games. Consider a set of m objective functions of the form

$$\min_{\mathbf{z}_j\in\mathbb{R}^p} h_j(\mathbf{Z}) \quad \text{with} \quad \mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_m], \tag{4.6}$$

where the functions h_j are convex and differentiable and may depend on other parameters than $\mathbf{z}_j$. Our objective is to find a zero of the simultaneous gradient

$$H(\mathbf{Z}) = [\nabla_{\mathbf{z}_1} h_1(\mathbf{Z}), \cdots, \nabla_{\mathbf{z}_m} h_m(\mathbf{Z})], \tag{4.7}$$

which corresponds to a Nash equilibrium of the game (4.6). In the rest of this presentation, we consider both the general setting and the simpler case of so-called potential games, for which the equilibrium can be found as the optimum of a single convex objective. This is the case for several of our regularizers, for example the TV penalty with symmetric weights. More details are provided in Appendix 4.a.1 on the nature of the non-cooperative games corresponding to our penalties.

Table 4.2: Gradient descent (GD) vs. Extra-gradient. Denoising results in avg. PSNR with $\sigma = 25$ on BSD68 [128].

Method	GD (24 iters)	GD (48 iters)	Extra-gradient (24 iters)
Trainable TV *symmetric*	27.58	27.50	27.82
Trainable TV *assymetric*	27.99	27.89	28.24

Two standard methods studied in the variational inequality litterature [151, 152, 153, 154] are the *gradient* and the *extra-gradient* [155] methods. The iterates of the basic gradient method are given by

$$\mathbf{Z}_{t+1} = \mathbf{Z}_t - \eta_t H(\mathbf{Z}_t), \tag{4.8}$$

where $\eta_t > 0$ is a step-size. These iterates are known to converge under a condition called strong monotonicity of the operator H, which is related to the concept of strong convexity in optimization, see [151]. Because this condition is relatively stringent, the extra-gradient method is often prefered [155], as it is known to converge under weaker conditions, see [153, 154]. The intuition of the method is to compute a look ahead step in order to compute more stable directions of descent:

$$\begin{aligned} \text{Extrapolation step} \quad & \mathbf{Z}_{t+1/2} = \mathbf{Z}_t - \eta_t H(\mathbf{Z}_t) \\ \text{Update step} \quad & \mathbf{Z}_{t+1} = \mathbf{Z}_t - \eta_t H(\mathbf{Z}_{t+1/2}). \end{aligned} \tag{4.9}$$

In this paper, our strategy is to unroll iterates of one of these two algorithms, and then to use auto differentiation for learning the model parameters $\boldsymbol{\theta}$. Furthermore, parameters that control the optimization process (*e.g.*, step size η_t) can also be learnt with this approach. It should be noted that optimization-driven layers have never been used before in the context of non-cooperative games, to the best of our knowledge, and therefore an empirical study is needed to choose between the strategies (4.8) or (4.9). In our experiments, extra-gradient descent has always performed at least as well, and sometimes significantly better, than plain gradient descent for comparable computational budgets. See for example Table 4.2 for a smoothed variant of the TV penalty.

Moreau-Yosida smoothing. The non-smooth regularization functions we consider can be written as a sum of simple terms. Omitting the dependency on $\boldsymbol{\theta}$ for simplicity, we may indeed write

$$\psi^j(\mathbf{Z}) = \textstyle\sum_{k=1}^{r} \phi_k(L_{k,j}(\mathbf{Z})) \quad \text{for some } r \geq 1,$$

where $L_{k,j}$ is a linear mapping and ϕ_k is either the ℓ_1- or ℓ_2-norm. For instance, ϕ_k is the ℓ_1-norm with $L_{k,j}(\mathbf{Z}) = a_{k,j}(\mathbf{z}_j - \mathbf{z}_k)$ in $\mathbb{R}^p$ for the TV penalty, and $L_{k,j}(\mathbf{Z}) = [\sqrt{a_{1,j}}\mathbf{z}_1(k), \ldots, \sqrt{a_{1,j}}\mathbf{z}_q(k)]^\top$ in $\mathbb{R}^q$ with ϕ_k being the ℓ_2-norm for the non-local group regularization. Handling such non-smooth convex terms may be achieved by leveraging the so-called Moreau-Yosida regularization [156, 157, 158]

$$\Phi_k(\mathbf{u}) = \min_{\mathbf{v}} \left\{ \phi_k(\mathbf{v}) + \frac{\alpha}{2}\|\mathbf{v} - \mathbf{u}\|^2 \right\},$$

which defines an optimization problem whose solution is called the proximal operator $\text{Prox}_{\phi_k/\alpha}[\mathbf{u}]$. As shown in [156], Φ_k is always differentiable and $\nabla\Phi_k(\mathbf{u}) = \alpha(\mathbf{u} - \text{Prox}_{\phi_k/\alpha}[\mathbf{u}])$, which can be computed in closed form when $\phi_k = \ell_1$ or ℓ_2.

The positive parameter α controls the trade-off between smoothness (the gradient of Φ_k is α-Lipschitz) and the quality of approximation. It is thus natural to define a smoothed approximation Ψ^j of ψ^j as $\Psi^j(\mathbf{Z}) = \sum_{k=1}^r \Phi_k(L_{k,j}(\mathbf{Z}))$.

Note that when the proximal operator of ψ^j can be computed efficiently, as is the case for the ℓ_1-norm, gradient descent algorithms can typically be adapted to handle the non-smooth penalty without extra computational cost [60], and there is no need for Moreau-Yosida smoothing. However, the proximal operator of the TV penalty and the non-local group regularization do not admit fast implementations. For the first one, computing the proximal operator requires solving a network flow problem [159], whereas the second one is essentially easy to solve when the weights $a_{j,k}$ form non-overlapping groups of variables, leading to a penalty called group Lasso [150].

We are now ready to present our unrolled algorithm as we have previously discussed gradient-based algorithms for solving convex smooth games and a smoothing technique for handling non-smooth terms. Generally, at iteration t, the gradient algorithm (4.8) performs the following simultaneous updates for all problems j

$$\begin{aligned}
\mathbf{u}_{k,j}^{(t)} &\leftarrow \mathrm{Prox}_{\phi_k/\alpha_{k,t}}[L_{k,j}(\mathbf{Z}^{(t)})] \qquad \text{for } \; k = 1, \ldots, r \\
\mathbf{z}_j^{(t+1)} &\leftarrow \mathbf{z}_j^{(t)} - \eta_t \left(\nabla_{\mathbf{z}_j} h_\theta \left(\mathbf{P}_j \mathbf{x}, \mathbf{z}_j^{(t)} \right) + \sum_{k=1}^r \alpha_{k,t} \left[L_{k,j}^* \left(L_{k,j}(\mathbf{Z}^{(t)}) - \mathbf{u}_{k,j}^{(t)} \right) \right]_j \right),
\end{aligned}$$

where $L_{k,j}^*$ is the adjoint of the linear mapping $L_{k,j}$. The computation of the gradients can be implemented with simple operations allowing auto-differentiation in deep learning frameworks. Interestingly, the smoothing parameter α can be made iteration-dependent, and learned along with other model parameters such that the amount of smoothing is chosen automatically.

4.3.4 Tricks of the Trade for Unrolled Optimization

Our strategy is to unroll iterates of our algorithms, and then compute $\nabla_{\boldsymbol{\theta}} \mathbf{z}_{\boldsymbol{\theta}}^\star$ by automatic differentiation. We present here a set of practical rules, some old and some new, facilitating training when $h_{\boldsymbol{\theta}}$ is a patch encoding function on a dictionary $\mathbf{D}$.

Initialization. To help the algorithm converge, we choose an initial stepsize $\eta_t \leq \frac{1}{L}$, where L is the Lipschitz constant of $\nabla_{\mathbf{z}} h_{\boldsymbol{\theta}}$, which is the classical step-size used by ISTA [65]. To do so, inspired by [105] we normalize the initial dictionary by its largest singular value and take $\eta_0 = 1$. Note that we can go one step further and normalize the dictionary throughout the training phase. This is in fact equivalent to the spectral normalization that has received some attention recently, notably for generative adversarial networks [160].

Untied parameters. In our framework, $\nabla_{\mathbf{z}_j} h_{\boldsymbol{\theta}}(\mathbf{P}_j \mathbf{x}, \mathbf{z}_j) = \mathbf{D}^\top \left(\mathbf{D}\mathbf{z}_j - \mathbf{P}_j \mathbf{x} \right)$. It has been suggested in previous work [69, 1, 105] to introduce an additional parameter $\mathbf{C}$ of the same size as $\mathbf{D}$, and consider instead the parametrization $\mathbf{C}^\top \left(\mathbf{D}\mathbf{z}_j - \mathbf{P}_j \mathbf{x} \right)$, $\mathbf{C}$ acting as a learned preconditioner. Even though the theoretical effect of this modification is not fully understood, it has been observed to accelerate convergence and boost performance for denoising tasks [1]. In our experiments, we will indicate in which cases we use this heuristic.

Backtracking. A simple way for handling the potential instability of the unrolled algorithm is to use a backtracking scheme which automatically decreases the

stepsize when the training loss diverges. This heuristic was used for instance in [1]. More details are provided in Appendix 4.a.2.

Barzilai-Borwein method for choosing the stepsize. A different, perhaps more principled, approach to improved stability consists in adaptively choosing an adaptive stepsize η_t. The literature on convex optimization proposes a set of effective rules, known as Barzilai-Borwein (BB) step size rules [161]. Even though these rules were not designed for convex games, they appear to be very effective in practice in the context of our optimization-driven layers. Concretely, they lead to step sizes $\eta_{t,j} = \|\mathbf{D}^\top \mathbf{D}\mathbf{s}_j\|_2 / \|\mathbf{D}\mathbf{s}_j\|^2$ with $\mathbf{s}_j = \mathbf{z}_j^{(t)} - \mathbf{z}_j^{(t-1)}$ for problem j at iteration t.

Table 4.3: Study of stabilization techniques for learnt sparse coding. Denoising results in average PSNR with $\sigma = 25$ on BSD68.

Method	Psnr (dB)	
	D	C,D
BM3D [88]	28.57	
Sparse Coding (SC)	✗	✗
SC + Backtracking	28.71	28.83
SC + Spectral norm	28.69	28.82
SC + Barzilai-Borwein	28.82	**28.86**

In our experiments, we observed that spectral normalization, backtracking, and Barzilai-Borwein step size were all effective to stabilize training. We have noticed that the spectral normalization impacts negatively the reconstruction accuracy, while the BB method tend to improve it by using larger stepsizes, at the expense of a larger computational cost. This is illustrated in Table 4.3 for a smoothed variant of sparse coding (we indicate with a crossmark when the algorithm diverges). In addition, we observe that the untied models brings a small boost in reconstruction accuracy.

4.4 Experiments

We consider three different tasks, illustrated with various combinations of regularization functions in order to demonstrate the wide applicability of our approach and its flexibility. A software package and additional details are provided in the supplementary material for reproducibility purposes.

Image denoising. For image denoising experiments, we use the standard setting of [95] with BSD400 [128] as a training set and on BSD68 as a test set. We optimize the parameters of our models using Adam [131] and also use the backtracking strategy described in Section 4.3.4 that automatically decreases the learning rate by a factor 0.5 when the training loss diverges. For the non-local models, we follow [1] and update the similarity matrices three times during the inference step. We use the parametrization with the $\mathbf{C}$ matrix for our patch-based experiments. We also combine our variance regularization with [1]. Additional training details and hyperparameters choices can be found in Appendix 4.a.2. We report performance in terms of averaged PSNR in Table 4.4, and more detailed tables with additional results are available in Appendix 4.a.3 for pixel-level models, and for the patch-based models involving a dictionary $\mathbf{D}$. Our models based on non-local sparse approximations perform better than the competing deep learning models with the exception of [93] for $\sigma \geq 15$ with much fewer parameters. In addition, we also observed that our assymetric TV models are almost on par with BM3D while being significantly faster (see Appendix 4.a.3 for more details) with only a very small amount of parameters.

Table 4.4: **Grayscale denoising** on BSD68, training on BSD400 for all methods. Performance in terms of average PSNR. Tiny CNN is a CNN baseline with few parameters. See Appendix 4.a.3 for qualitative results.

Method	Params	Noise Level (σ)			
		5	15	25	50
Tiny CNN *(ours)*	326	35.17	29.42	26.90	24.06
Tiny CNN *(ours)*	1200	36.47	30.36	27.70	24.60
BM3D [88]	-	37.57	31.07	28.57	25.62
LSCC [91]	-	37.70	31.28	28.71	25.72
CSCnet [105]	62k	37.69	31.40	28.93	26.04
GroupSC [1]	68k	37.95	31.71	29.20	26.17
FFDNet [114]	486k	N/A	31.63	29.19	26.29
DnCNN [95]	556k	37.68	31.73	29.22	26.23
NLRN [93]	330k	37.92	**31.88**	**29.41**	**26.47**
Pixel-reconstruction					
TV *symmetric*	288	36.91	30.27	27.66	24.51
TV *assymetric - extra-grad*	480	37.30	30.76	28.24	25.32
Laplacian *symmetric*	288	35.17	28.42	26.14	23.70
Laplacian *assymetric - extra-grad*	480	35.20	28.46	26.39	23.77
Bilateral - *extra-grad*	146	36.75	29.89	27.20	23.72
Bilateral TV - *extra-grad*	146	36.94	30.46	27.78	24.52
Non-local TV - *extra-grad*	307	37.53	31.03	28.50	25.26
Non-local Laplacian - *extra-grad*	307	37.54	31.00	28.47	25.46
Patch-reconstruction					
Sparse Coding (SC)	68k	37.84	31.46	28.90	25.84
Sparse Coding + Variance	68k	37.83	31.49	29.00	26.08
Sparse Coding + TV	68k	37.84	31.50	29.02	26.10
Sparse Coding + TV + Variance	68k	37.84	31.51	29.03	26.09
Non-local group	68k	37.95	31.69	29.19	26.19
Non-local group + Variance	68k	**37.96**	31.70	29.22	26.28
GroupSC + Variance	68k	**37.96**	<u>31.75</u>	<u>29.24</u>	<u>26.34</u>

Compressed Sensing for fMRI. Compressed Sensing for functional magnetic resonance imaging (fMRI) aims at reconstructing functional MR images from a small number of samples in the Fourier space. The corresponding inverse problem is

$$\min_{\mathbf{y}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\Psi(\mathbf{y}), \tag{4.10}$$

where the degradation matrix is $\mathbf{A} = \mathbf{P}\mathcal{F}$, $\mathbf{P}$ is a diagonal binary sampling matrix for a given sub-sampling pattern, $\mathcal{F}$ is the discrete Fourier transform such that the observed corrupted signal $\mathbf{x}$ is in the Fourier domain, and Ψ is a regularization function. This problem highlights the ability of our framework to handle both localized and non localized constraints. In our paper, we implemented two models revisiting some well studied priors for compressed sensing in an end-to-end fashion:

- *Pixel reconstruction* with total variation: we aim at solving the optimization for each node $\min_{\mathbf{y}_i\in\mathbb{R}} \|\mathbf{A}\mathbf{y} - \mathbf{x}\|_2^2 + \mathrm{TV}_i(\mathbf{y})$. In the past, total variation has been widely used for MRI [162], often in combination with sparse regularization in the wavelet domain.
- *Patch encoding on a dictionary with sparse coding*: we solve a collection of optimization problems of the form $\min_{\mathbf{z}_i\in\mathbb{R}^n} \|\mathbf{A}\hat{\mathbf{y}}(\mathbf{z}) - \mathbf{x}\|_2^2 + \lambda\|\mathbf{z}_i\|_1$, with

$\mathbf{y} = \frac{1}{n}\sum_j \mathbf{R}_j^\top \mathbf{D}\mathbf{z}_j$ the average of the overlapping patches. Some previous methods have explored dictionary-based reconstruction [163], but they were not investigated from a task-driven manner with end-to-end training.

In our experiments, we use the same setting as [108] for fair comparison: we train and test our models on the brain MRI dataset studied in that paper. Our models are trained separately for each sampling rate. We used the pseudo radial sampling for the matrix $\mathbf{P}$ similarly to the other methods. The reconstruction accuracy are reported in term of PSNR over the test set in Table 4.5. Our trainable model relying on a trainable TV prior performs surprisingly well given the conceptual simplicity of the prior. Also importantly, it runs significantly faster than all competing methods with a very small number of parameters. Furthermore, our trainable sparse coding method for fMRI gives strong performance and exceeds the state of the art for sampling rates larger than 30%. Note that architecture choices (patch and dictionary size) of our models are the same as for the denoising task, and we did not try to optimize them for the considered task, thus demonstrating the robustness of our approach.

Table 4.5: **Compressed sensing for fMRI** on the MR brain dataset using a pseudo radial sampling pattern. Performance comparisons in terms of PSNR (dB).

Method	Params	20 %	30 %	40%	50%	Test time
TV [162]	-	35.20	37.99	40.00	41.69	0.731s (cpu)
RecPF [164]	-	35.32	38.06	40.03	41.71	0.315s (cpu)
SIDWT	-	35.66	38.72	40.88	42.67	7.867s (cpu)
PANO [165]	-	36.52	39.13	40.31	41.81	35.33s (cpu)
BM3D-MRI [166]	-	<u>37.98</u>	40.33	41.99	43.47	40.91s (cpu)
ADMM-net [108]	-	37.17	39.84	41.56	43.00	0.791s (cpu)
ISTA-net [115]	337k	**38.73**	**40.89**	<u>42.52</u>	<u>44.09</u>	0.143s (gpu)
CS-TV (ours)	140	36.80	39.63	41.58	43.46	0.015s (gpu)
CS-Sp. cd. (ours)	68k	37.80	40.50	42.46	44.16	0.213s (gpu)
CS-Sp. cd. + Var (ours)	68k	37.79	<u>40.67</u>	**42.54**	**44.17**	0.213s (gpu)

Dense Stereo Matching. Our approach can be used to provide a generic regularization module that can easily be integrated into various neural architectures. We showcase its versatility by using it for deep stereo matching [168]. Given aligned image pairs, the goal is to compute disparity $\mathbf{d}$ for each pixel. Traditionally stereo matching is formulated as minimization of an energy function $E_{\text{data}}(\mathbf{d}) + \lambda E_{\text{smooth}}(\mathbf{d})$ where the data term, E_{data} measures how well $\mathbf{d}$ agrees

Table 4.6: **Denoising with less data**. Results in terms of average PSNR(dB) on BSD68 with $\sigma = 15$. All the models are trained on a similar subset of BSD400 for fair comparaison.

Method	Params	Training images			
		400	200	100	50
DnCNN [95]	556k	<u>31.73</u>	<u>31.65</u>	<u>31.47</u>	31.23
TV *extra-grad*	480	30.75	30.72	30.67	30.66
SC+Var	68k	31.49	31.49	<u>31.47</u>	<u>31.40</u>
GroupSC+Var	68k	**31.75**	**31.66**	**31.62**	**31.54**

Table 4.7: **Dense stereo matching** fine-tuning on kitti2015 train set, performance reported on the kitti2015 validation set.

Model	3-px error (%)
PSMNet [167]	2.14 ± 0.04
PSMNet+TV 12	2.11 ± 0.03
PSMNet+TV 24	2.11 ± 0.04
PSMNet+TV *extra*	$\mathbf{2.10 \pm 0.03}$

with the input image pairs, E_{smooth} enforces consistency among neighboring pixels' disparities: TV is a commonly chosen regularizer. Recent deep learning methods tackle the problem as a supervised regression to estimate continuous disparity map given pairs of stereo views and ground truth disparity maps [167]. We propose to combine our smoothing TV block with a state-of-the-art deep learning model [167]. In practice, we combine our block with a pretrained model on the SceneFlow [169] dataset, and fine-tune the pretrained model on the kitti2015 [170] train set, following the training procedure described in [167]. We used the original implementation of [167] available online and did not change any hyperparameters. We report in Table 4.7 the performance on the validation set in term of 3 pixels error which counts predicted pixel as correct if the disparity deviates from the ground truth from 3 pixels or less. We ran the experiment 10 times for each model (with and without the TV regularization). We observed that our TV block introduces very few additional parameters and consistently boosts performances.

Training with few examples. We conducted denoising experiments with less training data and report corresponding results in Table 4.6. We use the code released by the authors for training DnCNN with less data. Very interestingly the gap between our best model and CNN-based models increases when decreasing the size of the training set. We believe that this is an appealing feature, particularly relevant for applications in medical imaging or microscopy where the amount of training data can be very limited.

4.5 Discussion

We have presented a general framework based on non-cooperative games to train end-to-end imaging priors. Our experiments demonstrate the flexibility and the effectiveness of our approach on diverse tasks ranging from image denoising to fMRI reconstruction and dense stereo matching. Beyond image processing, we believe that the issue of interpretability is important. We consider models with a clear mathematical description of the decision function they produce. As a by-product, our models are also more parameter efficient than classical deep learning models. We believe that these are important steps to build systems that should not be seen as black boxes anymore, that produce explanable decisions, and that do not require training a system for days on a huge corpus of annotated data. These are important questions, which we are planning to address explicitly in the future.

4.a Appendix

This supplementary material is organized as follows: In Section 4.a.1, we discuss additional priors that were not presented in the main paper, but which are in prin-

ciple compatible with our framework, and we provide more details about potential games. In Section 4.a.2, we provide implementation details that are useful to reproduce the results of our paper (note that the code is also provided). In Section 4.a.3, we present additional quantitative results and additional results regarding inference speed of our models that were not included in the main paper for space limitation reasons. Finally, in Section 4.a.4, we present additional qualitative results (which require zooming on a computer screen).

4.a.1 Discussion on Models and Priors

4.a.1.1 Additional Priors

Our framework makes it possible to handle models of the form:

$$h_j(\mathbf{Z}) = h_{\boldsymbol{\theta}}(\mathbf{P}_j\mathbf{x}_j, \mathbf{z}_j) + \lambda \sum_{k=1}^{r} \phi_k(L_{k,j}(\mathbf{Z})), \tag{4.11}$$

where ϕ_k is a simple convex function that admits a proximal operator in closed form, and $L_{k,j}$ is a linear operator. In the main paper, several regularization functions have been considered, including the total variation, variance reduction, or non-local group regularization penalties. Here, we would like to mention a few additional ones, which are in principle compatible with our framework, but which we did not investigate experimentally. In particular, two of them may be of particular interest, and may be the topic of future work:

- the regularization $\lambda\|\mathbf{H}^\top\mathbf{z}_j\|_1$, where $\mathbf{H}$ is a matrix, may correspond to several settings. The matrix $\mathbf{H}$ may be for instance a wavelet basis, or may by learned, corresponding then to the penalty used in the analysis dictionary learning model from the paper "The cosparse analysis model and algorithms" of Nam et al., 2013.
- the regularization $\lambda\phi(\mathbf{H}^\top\mathbf{z}_j)$ where ϕ is a smooth function is closely related to the model introduced in [113], and to the Field of experts model of Roth and Black from the 2005 paper "Fields of Experts: A Framework for Learning Image Priors", even though the functions used in these other works are not convex.

4.a.1.2 Potential Games

A potential game is a non-cooperative convex game whose Nash equilibria correspond to the solutions of a convex optimization problem. We will now consider problems of the form (4.11), and show that all penalties that admit some symmetry are in fact potential games. Assuming the functions ϕ_k to be smooth for simplicity, optimality conditions for the convex problems (4.11) are, for all $j = 1, \ldots, m$:

$$\nabla_{\mathbf{z}_j} h_{\boldsymbol{\theta}}(\mathbf{P}_j\mathbf{x}_j, \mathbf{z}_j) + \lambda \sum_{k=1}^{r} \nabla_{\mathbf{z}_j} \tilde{\phi}_{k,j}(\mathbf{Z}) = 0, \quad \text{with} \quad \tilde{\phi}_{k,j}(\mathbf{Z}) = \phi_k(L_{k,j}(\mathbf{Z})). \tag{4.12}$$

Let us now assume the following symmetry condition such that if problem l involves a variable $\mathbf{z}_j$ through a function $\tilde{\phi}_{k,l}(\mathbf{Z})$, then problem j also involves the same term. Based on this assumption, we may define the potential function

$$V(\mathbf{Z}) := \sum_{j=1}^{m} \left(h_{\boldsymbol{\theta}}(\mathbf{P}_j\mathbf{x}_j, \mathbf{z}_j) + \frac{\lambda}{2} \sum_{k=1}^{r} \tilde{\phi}_{k,j}(\mathbf{Z}) \right).$$

The partial derivative of this potential function with respect to $\mathbf{z}_j$ is then

$$\nabla_{\mathbf{z}_j} h_{\boldsymbol{\theta}}(\mathbf{P}_j \mathbf{x}_j, \mathbf{z}_j) + \frac{\lambda}{2} \sum_{l=1}^{m} \sum_{k=1}^{r} \nabla_{\mathbf{z}_j} \tilde{\phi}_{k,l}(\mathbf{Z}) = \nabla_{\mathbf{z}_j} h_{\boldsymbol{\theta}}(\mathbf{P}_j \mathbf{x}_j, \mathbf{z}_j) + \frac{\lambda}{2} \sum_{l=1}^{m} \sum_{k \in \mathcal{N}_{j,l}} \nabla_{\mathbf{z}_j} \tilde{\phi}_{k,l}(\mathbf{Z}),$$

where $\mathcal{N}_{j,l}$ is the set of functions $\tilde{\phi}_{k,l}$ involving variable $\mathbf{z}_j$. The previous gradient can then be simplified into

$$\nabla_{\mathbf{z}_j} h_{\boldsymbol{\theta}}(\mathbf{P}_j \mathbf{x}_j, \mathbf{z}_j) + \frac{\lambda}{2} \sum_{j=1}^{r} \nabla_{\mathbf{z}_j} \tilde{\phi}_{k,l}(\mathbf{Z}) + \frac{\lambda}{2} \sum_{l \neq j} \sum_{k \in \mathcal{N}_{j,l}} \nabla_{\mathbf{z}_j} \tilde{\phi}_{k,l}(\mathbf{Z}).$$

Since the symmetry condition can be expressed as $\sum_{j=1}^{r} \tilde{\phi}_{k,l}(\mathbf{Z}) = \sum_{l \neq j} \sum_{k \in \mathcal{N}_{j,l}} \tilde{\phi}_{k,l}(\mathbf{Z})$, the condition $\nabla V(\mathbf{Z}) = 0$ is then equivalent to (4.12). Note that we have assumed the functions ϕ_k to be smooth for simplicity, but a similar reasoning can be conducted for non-smooth functions, by using the concept of subgradients.

Examples of potential games.

- the ℓ_1-norm: with $r = 1$ and $\tilde{\phi}_{1,j} = \|\mathbf{z}_j\|_1$, since problem j does not involve any variable $\mathbf{z}_l$ for $l \neq j$;
- Symmetric TV / Laplacian: problem j may involve a variable $\mathbf{z}_l$ through a term $a_{j,l}\|\mathbf{z}_j - \mathbf{z}_l\|_1$. Then, problem l involves the same term $a_{l,j}\|\mathbf{z}_j - \mathbf{z}_l\|_1$ under the condition $a_{j,l} = a_{l,j}$.
- Symmetric non local group with $r = p$ and $\tilde{\phi}_{k,j} = \lambda_k \|[\sqrt{a_{j,1}}\mathbf{z}_1(k), \ldots, \sqrt{a_{j,m}}\mathbf{z}_m(k)]^\top\|_2$. Under the condition of symmetric weights $a_{j,l} = a_{l,j}$, we obtain again a potential game.

Potential games are appealing as they provide guarantees about the existence of Nash equilibria without requiring optimizing over a compact set. Yet, we have found that allowing non-symmetric weights often performs better. This is illustrated in Table 4.8 for a simple denoising experiment.

Table 4.8: **Symmetric vs assymmetric** grayscale denoising on BSD68, training on BSD400 for all methods. Performance is measured in terms of average PSNR.

Method	Params	Noise Level (σ)			
		5	15	25	50
TV *symmetric*	72	36.08	30.21	27.58	24.74
TV *assymetric - extra-grad*	480	37.30	30.76	28.24	25.32
Laplacian *symmetric*	72	34.88	28.14	25.90	23.45
Laplacian *assymetric - extra-grad*	480	35.20	28.46	26.39	23.77
Non-local group - *symmetric*	68k	37.94	31.67	29.17	26.16
Non-local group - *assymetric*	68k	37.95	31.69	29.20	26.19

4.a.2 Implementation Details and Reproducibility

4.a.2.1 Training Details

For the training of patch-based models for denoising, we randomly extract patches of size 56×56 whose size equals the window size used for computing non-local

self-similarities; whereas we train pixel level models on the full size images. For fMRI experiments we also trained the models on the full sized images. We apply a mild data augmentation (random rotation by 90° and horizontal flips). We optimize the parameters of our models using ADAM [131].
The learning rate is set to 6×10^{-4} at initialization and is sequentially lowered during training by a factor of 0.35 every 80 training steps, in the same way for all experiments. Similar to [105], we normalize the initial dictionary $\mathbf{D}_0$ by its largest singular value as explained in the main paper in Section 4.3.4. We initialize the dictionary $\mathbf{C}$,$\mathbf{D}$ and $\mathbf{W}$ with the same dictionary obtained with an unsupervised dictionary learning algorithm (using SPAMS library).
We have implemented the backtracking strategy described in Section 4.3.4 of the main paper for all our algorithms, which automatically decreases the learning rate by a factor 0.8 when the loss function increases too much on the training set, and restore a previous snapshot of the model. Divergence is monitored by computing the loss on the training set every 10 epochs. Training the non-local models for denoising are the longer models to train and takes about 2 days on a Titan RTX GPU. We summarize the chosen hyperparameters for the experiments in Table 4.9.

Table 4.9: Hyper-parameters chosen for every task.

Experiment	Gray denoising (patch)	Gray denoising (pixel)	fMRI
Patch size	9	-	9
Dictionary size	256	-	256
Nr epochs	300	300	150
Batch size	32	32	1
K iterations	24	24	24
Middle averaging	✓	✓	-
Correlation update frequency f	1/6	1/12	-

4.a.3 Additional Quantitative Results

4.a.3.1 Inference Speed

In Table 4.10 we provide a comparison of our TV models in terms of speed with BM3D for grayscale denoising on the BSD68 dataset. For fair comparison, we reported computation time both on gpu and cpu.

Table 4.10: Inference speed for image denoising.

	Params	Psnr	Speed
BM3D [88]	-	25.62	7.28s (cpu)
TV assymetric	240	24.93	0.014s (gpu) / 0.18s (cpu)
TV assymetric (extra)	480	25.32	0.021s (gpu) / 0.28s (cpu)

4.a.3.2 Image Denoising

We provide additional results for grayscale denoising with different variations of the prior introduced in the main paper, as well as combination of different priors. We reported performances for gray denoising in Table 4.11 for the pixel based

models, and in Table 4.12 for the patch based models. In Table 4.11 *untied* κ denotes when we used a different set of learned parameters κ at each stage of the refinement step of the similarity matrix for the non-local models.

Table 4.11: **Pixel level** grayscale denoising on BSD68, training on BSD400 for all models. Performance is measured in terms of average PSNR.

Method	Params	Noise Level (σ)			
		5	15	25	50
BM3D [88]	-	<u>37.57</u>	**31.07**	**28.57**	**25.62**
Tiny CNN	326	35.17	29.42	26.90	24.06
Tiny CNN	1200	36.47	30.36	27.70	24.60
TV *symmetric*	288	36.08	30.21	27.58	24.74
TV *symmetric - extra-grad*	144	37.02	30.33	27.82	24.81
TV *assymetric-*	240	36.83	30.49	27.99	24.93
TV *assymetric - extra-grad*	480	37.30	30.76	28.24	25.32
Laplacian *symmetric*	288	34.88	28.14	25.90	23.45
Laplacian *symmetric - extra-grad*	144	33.87	28.14	25.91	23.45
Laplacian *assymetric*	240	35.20	28.48	26.17	23.78
Laplacian *assymetric - extra-grad*	480	35.20	28.46	26.39	23.77
Non-local TV *assymmetric*	154	37.25	30.86	28.28	25.42
Non-local TV *assymmetric* (untied κ)	235	37.12	31.01	28.37	25.24
Non-local TV *assymmetric - extra-grad*	226	**37.83**	30.98	28.34	25.31
Non-local TV *assymmetric - extra-grad* (untied κ)	307	37.53	<u>31.03</u>	28.50	25.26
Non-local Laplacian *assymmetric*	154	37.31	30.75	28.33	25.15
Non-local Laplacian *assymmetric* - (untied κ)	235	37.53	31.01	28.37	<u>25.47</u>
Non-local Laplacian *assymmetric - extra-grad*	226	37.51	30.99	28.34	25.13
Non-local Laplacian *assymmetric - extra-grad* (untied κ)	307	37.54	31.00	<u>28.47</u>	25.46
Bilateral	74	36.76	29.89	27.16	23.97
Bilateral TV	74	36.60	29.82	27.23	24.00
Bilateral - *extra-grad*	146	36.75	29.89	27.20	23.72
Bilateral TV - *extra-grad*	146	36.94	30.46	27.78	24.52

4.a.4 Additional Qualitative Results

Finaly, we show qualitative results for grayscale denoising in Figures 4.3, 4.4.

Table 4.12: **Patch level** grayscale denoising on BSD68, training on BSD400 for all methods. Performance is measured in terms of average PSNR.

Method	Params	Noise Level (σ)			
		5	15	25	50
BM3D [88]	-	37.57	31.07	28.57	25.62
LSCC [91]	-	37.70	31.28	28.71	25.72
CSCnet [105]	62k	37.69	31.40	28.93	26.04
FFDNet [114]	486k	N/A	31.63	29.19	26.29
DnCNN [95]	556k	37.68	31.73	29.22	26.23
NLRN [93]	330k	37.92	**31.88**	**29.41**	**26.47**
GroupSC [1]	68k	37.95	31.71	29.20	26.17
Sparse Coding + Barzilai-Borwein	68k	37.85	31.46	28.91	25.84
Sparse Coding + Variance	68k	37.83	31.49	29.00	26.08
Sparse Coding + TV	68k	37.84	31.50	29.02	26.10
Sparse Coding + TV + Var	68k	37.84	31.51	29.03	26.09
Sparse Coding + TV + Var + BB	68k	37.86	31.52	29.04	26.04
Non-local group - *symmetric*	68k	37.94	31.67	29.17	26.16
Non-local group - *assymetric*	68k	37.95	31.69	29.20	26.19
Non-local group - *assymetric* + TV	68k	37.96	31.71	29.22	26.26
Non-local group - *assymetric* + Var	68k	37.96	31.70	29.23	26.28
Non-local group - *assymetric* + Var + TV	68k	37.95	31.71	<u>29.24</u>	26.30
GroupSC + Variance	68k	**37.96**	<u>31.75</u>	<u>29.24</u>	<u>26.34</u>

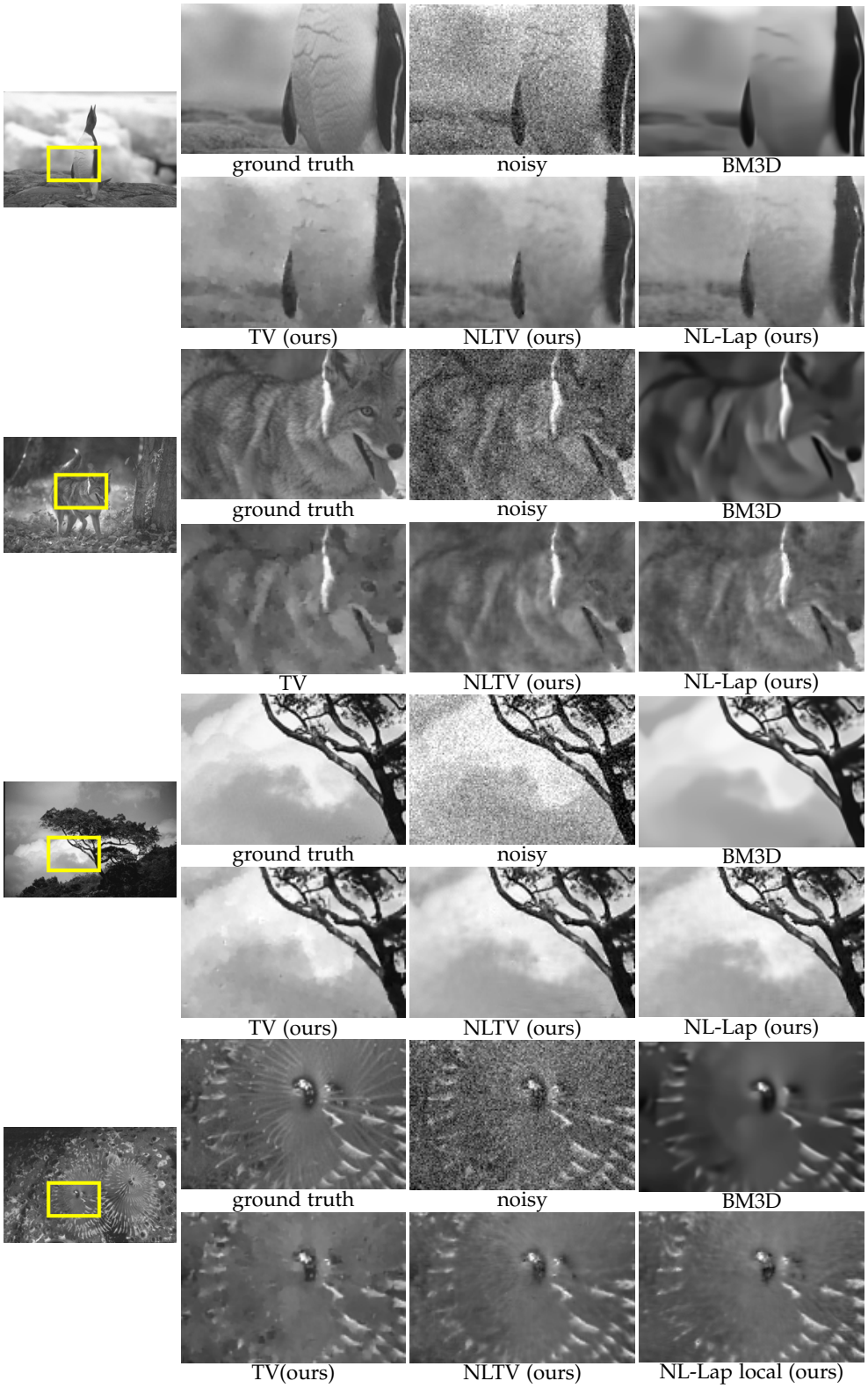


Figure 4.3: Grayscale denoising for 4 images from the BSD68 dataset. Best seen by zooming on a computer screen.

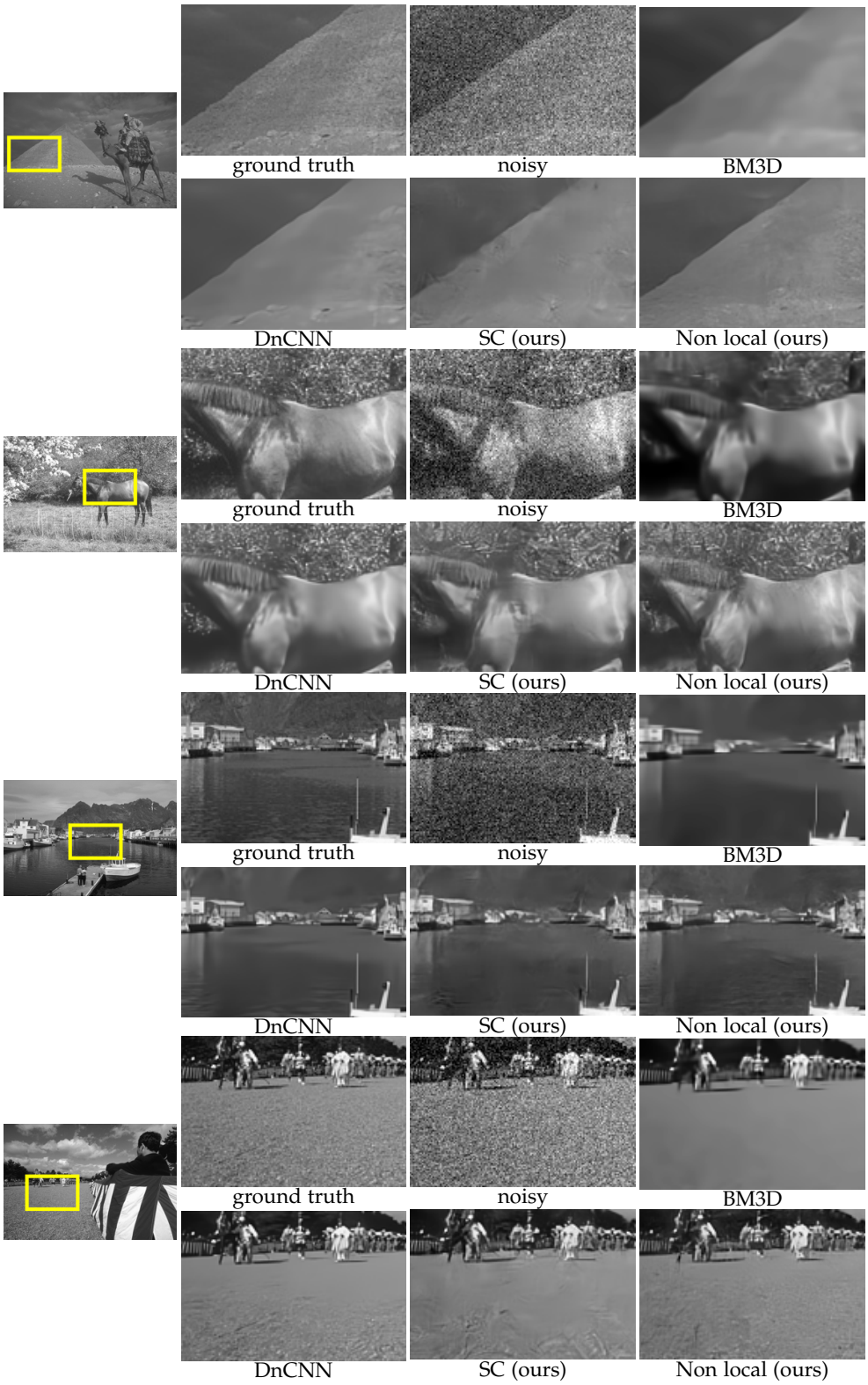


Figure 4.4: Results of our patch level models for grayscale denoising for 4 images from the BSD68 dataset. Best seen by zooming on a computer screen.

Chapter 5

Super-Resolution from Raw Image Bursts

Chapter abstract:
This presentation addresses the problem of reconstructing a high-resolution image from multiple lower-resolution snapshots captured from slightly different viewpoints in space and time. Key challenges for solving this *super-resolution* problem include (i) aligning the input pictures with sub-pixel accuracy, (ii) handling raw (noisy) images for maximal faithfulness to native camera data, and (iii) designing/learning an image prior (regularizer) well suited to the task. We address these three challenges with a hybrid algorithm building on the insight from [80] that aliasing is an ally in this setting, with parameters that can be learned end to end, while retaining the interpretability of classical approaches to inverse problems. The effectiveness of our approach is demonstrated on synthetic and real image bursts, setting a new state of the art on several benchmarks and delivering excellent qualitative results on real raw bursts captured by smartphones and prosumer cameras.

B. Lecouat, J. Ponce, J. Mairal. Lucas Kanade Reloaded : End-to-End Super-Resolution from Raw Image Bursts. *International Conference on Computer Vision (ICCV)*, 2021.

Contents

5.1 Introduction

The problem of reconstructing high-resolution (HR) images from lower resolution (LR) ones comes in multiple flavors, that may significantly differ from each other in both technical detail and overall objectives. When a single LR image is available, the corresponding inverse problem is severely ill-posed, requiring very strong priors about the type of picture under consideration [171, 172]. For natural images, data-driven methods based on convolutional neural networks (CNNs) have proven to be very effective [173, 174]. Generative adversarial networks (GANs) have also been used to synthesize impressive HR images that may, however, contain "hallucinated" high-frequency details [99, 175].

In the true *super-resolution* setting [176, 177, 172],[1] where multiple LR frames are available, HR details *are* present in the data, but they are spread among multiple misaligned images, with technical challenges such as recovering sub-pixel registration, but also the promise of recovering veridical information in applications ranging from amateur photography to astronomy, biological and medical imaging, microscopy imaging, and remote sensing.

Videos are of course a rich source of multiple, closely-related pictures of the same scene, with several recent approaches to super-resolution in this domain, often combining data-driven priors from CNNs with self-similarities between frames [178, 179, 180]. However, most digital videos are produced by a complex pipeline mapping raw sensor data to possibly compressed, lower-resolution frames, resulting in a loss of high-frequency details and spatially-correlated noise that may be very difficult to invert [149]. With the ability of modern smartphone and prosumer cameras to record raw image bursts, on the other hand, there is a new opportunity to restore the corresponding frames *before* the image signal processor (ISP) of the camera produces irremediable damage [181, 80]. This is the problem addressed in this presentation, and it is challenging for several reasons: (i) images typically contain unknown motions due to hand tremor,[2] making subpixel alignment difficult; (ii) converting noisy raw sensor data to full-color images is in itself a difficult problem known as *demosaicking* [182, 1]; and (iii) effective image priors are often data driven, thus requiring a differentiable estimation procedure for end-to-end learning.

In this paper, we jointly address these issues and propose a new approach that retains the interpretability of classical inverse problem formulations while allowing end-to-end learning of models parameters. This may be seen as a bridge between the "old world" of signal processing and the "brave new one" of data-driven black boxes, without sacrificing interpretability: On the one hand, we address an inverse problem with a model-based optimization procedure alternating motion and HR image estimation steps, directly building on classical work from the 1980s [73, 183] and 1990s [184]. On the other hand, we also fully exploit modern technology in the form of a *plug-and-play* prior [185, 109] that gracefully mixes deep neural networks with variational approaches. In turn, unrolling the optimization procedure [103, 1,

[1] "Single-image super-resolution" has become a popular nickname for single-image upsampling under strong priors; here, we use the classical definition of super-resolution from multiple LR snapshots [177, 172].

[2] Image bursts acquired on a tripod may also present subpixel misalignments in practice due to floor vibrations, as observed in our experiments.

186] allows us to learn the model parameters end to end by using training data with synthetic motions [181].
Since aliasing produces low-frequency artefacts associated with undersampled high-frequency components of the original signal, it is typically considered a nuisance, motivating camera manufacturers to add anti-aliasing (optical) filters in front of the sensor.[3] Yet, aliased images carry high-frequency information, which may be recovered from multiple shifted measurements. Perhaps surprisingly, aliasing is thus an ally in the context of super-resolution, a fact already noted in earlier references, see [187]. As shown in the rest of this presentation, our approach to raw burst super-resolution also exploits this insight, and it achieves a new state of the art on several standard benchmarks that use synthetic motion for ground truth. It also gives excellent qualitative results on real data obtained with smartphone and prosumer cameras. Interestingly, as illustrated by Figure 6.1, our method has turned out to be surprisingly robust to noise given the particularly challenging setting of raw image super-resolution, which involves simultaneous blind denoising, demosaicking, registration, and upsampling.

Summary of contributions.

- To the best of our knowledge, we propose the first model-based architecture learnable end to end for joint image alignment and super-resolution from raw image bursts.
- We introduce a new differentiable image registration module that can be applied to images of different resolutions, is readily integrable in neural architectures, and may find other uses beyond super-resolution.
- We show that our approach gives excellent results on both real image bursts (with up to $\times 4$ upsampling for raw images) and synthetic ones (up to $\times 16$ for RGB images).

5.2 Related Work

Classical multiframe super-resolution. Tsai and Huang wrote the seminal paper in this setting [177], with a restoration model in the frequency domain assuming known translations between frames. Most latter approaches have focused on the spatial domain, and they generally fall into two main categories [30]: In interpolation-based methods, LR snapshots aligned with sub-pixel precision are jointly interpolated into an HR image [188, 189]. Impressive results have recently been obtained for hand-held cameras using the variant of this method proposed by Wronski *et al.* [80], whose insight of exploiting aliasing effects has been one of the inspirations of our work. However, due to the sequential nature of their algorithm, errors may propagate from one stage to the next, leading to sub-optimal reconstructions [190]. In contrast, iterative spatial domain techniques iteratively refine an estimate for the super-resolved image so as to best explain the observed LR frames under some image formation model. Variants of this approach include the early iterated backprojection algorithm of Irani *et al.* [191], the maximum likelihood technique of Elad and Feuer [56], and the model regularized by bilateral total variation of Farsiu *et al.* [79]. The image formation parameters are either be

[3] There is, however, a trend today toward removing these filters, as in the prosumer camera used in some of our experiments with real images.

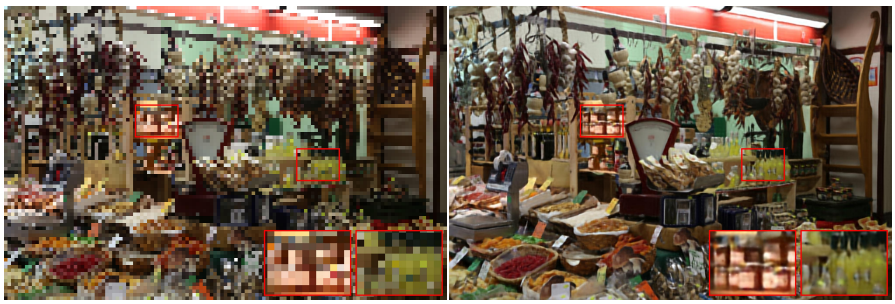


Figure 5.1: Proof of concept for extreme $\times 16$ upsampling. The right image is obtained by processing a burst of 20 LR images presented on the left obtained with synthetic random affine movements and bilinear downsampling.

assumed to be known a priori through calibration, or estimated jointly with the HR image. In general, inter-frame motion can either be estimated separately, or be treated as an integral part of the super-resolution problem [192, 184], thus avoiding motion estimation between LR frames, whose accuracy may be affected by undersampling [193]. The method proposed in the rest of this paper combines the best of both worlds since it performs joint estimation while aligning the LR frames with the reconstructed HR image.

Learning-based approaches. In this context, the multiframe case has received less attention than its single-image counterpart, for which several loss functions and architectures have been proposed [99, 175, 186]. Most multi-frame algorithms focus on video super-resolution. Model-based techniques learn non-uniform interpolation or motion compensation using convolutional neural networks [194] but the most successful approaches so far are model free, leveraging instead diversity with 3D convolutions or attention mechanisms [178, 180]. Learning-based methods have also been used in remote sensing applications, using 3D convolutions [195] or joint registration/fusion architectures [196] for example. Finally, and closer to our work, Bhat et al. [181] have recently proposed a network architecture for raw burst super-resolution, together with a very interesting dataset featuring both synthetic and real images for training and testing. It is important to note that learning-based approaches to super-resolution are typically trained on synthetically generated LR images [197], a strategy that may not generalize well to real photographs unless great care is taken in modeling the image corruption process [198]. Learning super-resolution models from real LR/HR image pairs is quite challenging since it requires in general using separate cameras with different lenses and spatial resolution, with inevitable spatial and spectral misalignments. As shown by our experiments, our method, although trained from synthetic LR images, gives excellent results with real bursts taken from different smartphones and cameras. Leveraging real images at training time is, for now, left for future work.

5.3 Proposed Approach

This section presents the three main components of our approach: its image formation model, an optimization procedure for solving the corresponding inverse problem, and its unrolled implementation in a feedforward architecture whose parameters can be learned end to end.

5.3.1 Image Formation Model

Image acquisition in a digital camera starts from an instantaneous irradiance function $f_{\gamma,t} : [0,1]^2 \to \mathbb{R}^+$ defined on a continuous retinal domain with nonnegative values, such that $f_{\gamma,t}(\mathbf{u})$ is the spectral irradiance value at point $\mathbf{u}$, time t, and wavelength γ, accounting for blur due to optics, atmospheric effects, etc. The camera sensor integrates $f_{\gamma,t}$ in the spatial, time, and spectral domains to construct a *raw* digital image $\mathbf{y} : [1,\dots,n]^2 \to \mathbb{R}^+$, where each pixel's spectral response is typically dictated by the 2×2 RGGB Bayer pattern, with twice as many measurements for the green channel than for the red and blue ones [182]. Modern cameras turn the raw image $\mathbf{y}$ into a full blown, three-channel *RGB* image $\mathbf{x}$ with the same spatial resolution through an interpolation process called *demosaicking*.
In practice, we do not have access to $f_{\gamma,t}$ to use as ground truth for learning an image restoration process, even when an accurate model of the $f_{\gamma,t} \mapsto \mathbf{x}$ map is available. Thus, we model instead the process $\mathbf{x} \mapsto \mathbf{y}_k$, where $\mathbf{x}$ is a latent high-resolution (HR) image we wish to recover, and the low-resolution (LR) images $\mathbf{y}_k$ ($k = 1,\dots,K$) have been observed in a burst of length K. We assume that $\mathbf{x}$ is sharp, without any blur, and noiseless. The burst images are obtained through the following forward model (Figure 5.2):

$$\mathbf{y}_k = DBW_{\mathbf{p}_k}\,\mathbf{x} + \boldsymbol{\varepsilon}_k \text{ for } k = 1,\dots,K, \tag{5.1}$$

where $\boldsymbol{\varepsilon}_k$ is some additive noise. Here, both the HR image $\mathbf{x}$ and the frames $\mathbf{y}_k$ of the burst are flattened into vector form. The operator $W_{\mathbf{p}_k}$ parameterized by $\mathbf{p}_k$ warps $\mathbf{x}$ to compensate for misalignments between $\mathbf{x}$ and $\mathbf{y}_k$ caused by camera or scene motion between frames, assumed here to be a 6-parameter affine transformation of the image plane, then resamples the warped image to align its pixel grid with that of $\mathbf{y}_k$. Finally, the corresponding HR image is blurred to account for integration over both space (the LR pixel area, using either simple averaging or, as in the figure, a Gaussian filter) and time (accounting for camera and/or scene motion during exposure), and it is finally downsampled in both the spatial and spectral domains by the operator D, with an (a priori) arbitrary choice of *where* to pick the sample from (pixel corner or center for example), the spectral part correponding to selecting one of the three RGB values to assemble the raw image. It will prove convenient in the sequel to rewrite (5.1) as $\mathbf{y} = U_{\mathbf{p}}\mathbf{x} + \boldsymbol{\varepsilon}$, where

$$U_{\mathbf{p}} = \begin{bmatrix} DBW_{\mathbf{p}_1} \\ \vdots \\ DBW_{\mathbf{p}_K} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_K \end{bmatrix}, \mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_K \end{bmatrix}, \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_K \end{bmatrix}. \tag{5.2}$$

Before closing this section, let us note that simpler motion models with two (translation) or three (rigid motion) parameters, or (much) more complex piecewise-affine or elastic models could be considered depending on the application. We focus here on the scenario where a user wishes to zoom in on a relatively small crop (say, between 100×100 to 800×800 pixels) of a multi-megapixel image, and the affine model has proven effective with real handheld cameras in this setting. This implicitly corresponds to a globally piecewise-affine motion model.

5.3.2 Inverse Problem and Optimization

Given the image formation model of Eqs. (5.1)–(8.2), recovering the HR image $\mathbf{x}$ from the K LR frames $\mathbf{y}_k$ in the burst can be formulated as finding the values of $\mathbf{x}$

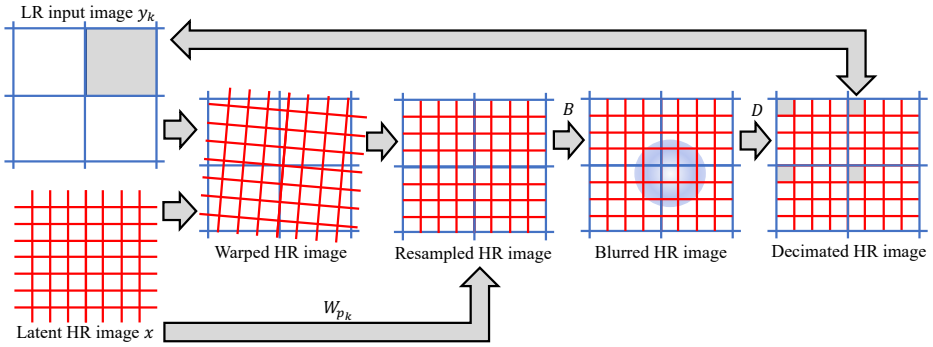


Figure 5.2: Image formation: The HR image $\mathbf{x}$ is warped then resampled to align it with the LR image $\mathbf{y}$ using the operateur $W_{\mathbf{p}_k}$. It is then blurred by the operator B to account for integration over LR pixels and finally downsampled in the spatial and spectral domains by the operator D (the spectral downsampling from RGB to R, G or B is not illustrated here for simplicity).

and $\mathbf{p}$ that minimize

$$\frac{1}{2}\|\mathbf{y} - U_{\mathbf{p}}\,\mathbf{x}\|^2 + \lambda\phi_\theta(\mathbf{x}), \tag{5.3}$$

where ϕ_θ is a parameterized regularizer, to be detailed later, and λ is a parameter balancing the data-fidelity and regularization terms. Many methods are of course available for minimizing this function. Like others (e.g., [199]), and mainly for simplicity, we choose here a quadratic penalty method [200, Sec. 17.1] often called half-quadratic splitting (or HQS) [201]: the original objective is replaced by

$$E_\mu(\mathbf{x}, \mathbf{z}, \mathbf{p}) = \frac{1}{2}\|\mathbf{y} - U_{\mathbf{p}}\,\mathbf{z}\|^2 + \frac{\mu}{2}\|\mathbf{z} - \mathbf{x}\|^2 + \lambda\phi_\theta(\mathbf{x}), \tag{5.4}$$

where $\mathbf{z}$ is an auxiliary variable, and μ is a parameter increasing at each iteration, such that, as $\mu \to +\infty$, the minimization of (5.4) with respect to $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{p}$ becomes equivalent to that of (5.3) with respect to $\mathbf{x}$ and $\mathbf{p}$ alone. Each iteration of HQS can be viewed as one step of a block-coordinate descent procedure for minimizing E, changing one of variables $\mathbf{z}$, $\mathbf{x}$ and $\mathbf{p}$ at a time while keeping the others fixed, with the value of μ increasing after each iteration. Convergence guarantees for quadratic penalty methods require an approximate minimization of Eq. (5.4) with increasing precision over time [200]. Following common practice in computer vision (e.g. [199]), we use HQS without formally checking that its precision indeed increases with iterations. This very simple procedure turns out to work well in practice. Its steps are detailed in the next three paragraphs, the exponent t being used to designate the value of the variables at iteration t. The sequence of weights $(\mu^t)_{t\geq 0}$ is learned end-to-end as explained in Section 5.3.3.

Updating z. Several strategies are possible for minimizing Eq. (5.4) with respect to $\mathbf{z}$. Given the dimension of the problem, one may choose for instance a fast iterative minimization procedure such as conjugate gradient descent. Since an approximate minimization is sufficient for our needs, we have chosen to use instead a single step of plain gradient descent, which converges more slowly in theory, but is also simpler and more easily amenable to the unrolled optimization strategy for end-to-end learning that will be presented next. The update at iteration t is given by

$$\mathbf{z}^t \leftarrow \mathbf{z}^{t-1} - \eta_t\left[U_{\mathbf{p}^{t-1}}^\top(U_{\mathbf{p}^{t-1}}\mathbf{z}^{t-1} - \mathbf{y}) + \mu(\mathbf{z}^{t-1} - \mathbf{x}^{t-1})\right], \tag{5.5}$$

where $\eta_t > 0$ is some step size, also learned end to end.

Updating the motion parameters p. Let $\mathbf{p}_k$ denote the part of the parameter vector $\mathbf{p}$ responsible for the alignment of $\mathbf{z}^t$ and $\mathbf{y}_k$ in (5.4). The corresponding optimization problem can be rewritten as

$$\min_{\mathbf{p}_k} \frac{1}{2} \|\mathbf{y}_k - DBW_{\mathbf{p}_k}\mathbf{z}^t\|^2. \tag{5.6}$$

This is a non linear least-squares problem, which can once again be solved using many different techniques. Here, we pick a Gauss-Newton approach, which corresponds to a variant of the Lucas-Kanade algorithm [73, 183], showing again that a 40-year old technique can still be relevant today. Specifically, we perform one Gauss-Newton step at each iteration t for each $\mathbf{p}_k$ in parallel:

$$\mathbf{p}_k^t \leftarrow \mathbf{p}_k^{t-1} - \left(\mathbf{J}_k^{t\top}\mathbf{J}_k^t\right)^{-1} \mathbf{J}_k^{t\top}\mathbf{r}_k^t, \tag{5.7}$$

where $\mathbf{r}_k^t = U_{\mathbf{p}_k^{t-1}}\mathbf{z}^t - \mathbf{y}_k$ is the residual of the non-linear least-squares problem (5.6), and $\mathbf{J}_k^t = (\partial U_{\mathbf{p}_k^{t-1}}/\partial \mathbf{p}_k)\mathbf{z}^t$ is the Jacobian of the $DBW_{\mathbf{p}_k}$ operator. The only difference with a Lucas-Kanade iteration is the presence of a high-resolution frame $\mathbf{z}^t$ and the downsampling operator DB. This is similar to [184], or more recently [192, 202], which align high-resolution images with low-resolution ones.

Estimating the HR image x. The $\mathbf{x}$ update is obtained as

$$\mathbf{x}^t \leftarrow \arg\min_{\mathbf{x}} \frac{\mu_{t-1}}{2}\|\mathbf{z}^t - \mathbf{x}\|^2 + \lambda\phi_\theta(\mathbf{x}),$$

which amounts to computing the proximal operator of the prior ϕ_θ. In practice, we follow a "plug-and-play" approach [185, 203, 109], and replace the proximal operator by a parametric function $f_\theta(\mathbf{z}_t)$ (here, a CNN, see implementation details). Using such an implicit prior has proven very effective in our setting. More traditional image priors such as total variation could of course have been used as well.

5.3.3 Unrolled Optimization and Backpropagation

The optimization procedure described so far requires choosing hyper-parameters such as the sequence $(\mu_t)_{t\geq 0}$, and its implicit prior also involves model parameters θ. By using a training set of n LR burst/HR image pairs, we propose to learn all these parameters in a supervised fashion. We denote the training set by $(\mathbf{Y}_i, \mathbf{x}_i)_{i=1}^n$, where $\mathbf{Y}_i = \{\mathbf{y}_j^i\}_{j=1}^K$ is the i-th burst of LR images associated to the HR image $\mathbf{x}_i$. We then unroll the optimization procedure for T steps and, denoting by $\hat{\mathbf{x}}_T(\mathbf{Y}_i)$ the HR image estimated from burst $\mathbf{Y}_i$, we consider the objective function

$$\frac{1}{n}\sum_{i=1}^n L(\hat{\mathbf{x}}_T(\mathbf{Y}_i), \mathbf{x}_i), \tag{5.8}$$

where L is the ℓ_2 or ℓ_1 loss (in practive we have observed that the ℓ_1 loss performs slightly better). Because every step of our estimation procedure is differentiable, we minimize (5.8) by stochastic gradient descent.

Learned data prior. Good image priors are essential for solving ill-posed inverse problems. As noted earlier, instead of using a classical one, such as total variation (TV) or bilateral total variation (BTV) [79], we learn an implicit prior parameterized by a convolutional neural network f_θ in a data-driven manner. We use the ResUNet architecture introduced in [186] in practice. It involves four scales, each of which has an identity skip connection between downscaling and upscaling operations.

5.3.4 Implementation Details and Variants

Downsampling and blurring operators $\mathbf{D}, \mathbf{B}$. We have tried different variants of downsampling/blurring strategies such as Gaussian smoothing. In practice, we have observed that simple averaging, which is differentiable and parameter-free, gives good results in all our experiments. As a consequence, we do not assume any knowledge about the blur used to generate data, corresponding to an operator B that only captures blur due to photon integration on the sensor without addressing optical blur. We argue that this limited model is relevant because modern cameras and smartphone are aliased [80], which may explain the generalization to real images, as soon as the scene is static.

Initialization by coarse alignement. To initialize the motion parameters $\mathbf{p}$, we cannot minimize (5.6) as in the previous section, because no good estimate of the HR image is available. Therefore, we align each LR frame to an arbitrary one from the burst (*e.g.*, the first one) by using the Lucas-Kanade forward additive algorithm [73, 74] which is known to be robust to noise. Note that another difficulty lies in the raw format of images. To overcome this issue, we simply convert raw images into grayscale images by using bilinear interpolation. This is of course sub-optimal, but sufficient for obtaining coarse motion parameters.

Initialization via coarse-to-fine strategy. For extreme upsampling factors ($\times 16$), we found a coarse-to-fine initialization strategy to be useful: We initialize the motion parameters $\mathbf{p}_j^0$ and high-resolution image $\mathbf{z}^0$ by using the output of the algorithm trained at a lower upsampling factor. For instance, $\times 16$ can be obtained by applying twice a $\times 4$ algorithm, or four times $\times 2$ algorithm.

5.4 Experiments

Experiments were conducted on synthetic and real raw image bursts. We also provide experiments on RGB bursts in the appendix, allowing easier comparison with earlier approaches that cannot handle raw data.

Training procedure and data. For synthetizing realistic *raw bursts* from groundtruth RGB images, we follow the approach described in [181], using the author's publicly available code[4] on the training split of the Zurich raw to RGB dataset [204]. The approach consists of applying the inverse RGB to raw pipeline introduced in [198]. Displacements are randomly generated with Euclidean motions and frames are downscaled with bilinear interpolation in order to simulate LR frames containing aliasing. Synthetic, yet realistic, noise is added to the frames, and color values are discarded according to the Bayer pattern. Then, we train our models for minimizing the loss (5.8). We perform 100 000 iterations of the ADAM

[4] `https://github.com/goutamgmb/NTIRE21_BURSTSR`.

optimizer with a batch size of 10, a burst size of 14 and with a learning rate of 3×10^{-5} decaying by a factor 2 after 50 000 iterations. Our approach is implemented in Pytorch and takes approximately 1.5 days to train on an Nvidia Titan RTX GPU. We evaluate our models in all our experiments with a burst size of 14 unless specified.

Extreme $\times 16$ upsampling on RGB images. As a proof-of-concept, we also perform experiments for an unusual $\times 16$ super-resolution task, using the coarse-to-fine strategy of Sec. 5.3.4. A result is presented in Fig. 5.1, showing impressive reconstruction and additional ones can be found in the appendix. Even though not realistic, we believe the experiment to be of interest, as it demonstrates the effectiveness of our approach in an idealistic, yet extreme, setting.

Demosaic+SISR ETH [181] Ours

Figure 5.3: Visual comparison on **synthetic raw image bursts** used in [181]. Demosaic+SISR is our single-image baseline based on the ResUNet architecture [186] (see main text). The two right columns are produced by methods dedicated to raw burst processing, respectively [181] and ours.

Evaluation on synthetic RAW images. The evaluation protocol of [181] allows us to perform quantitative comparison with their state-of-the-art method for processing raw image bursts. An additional comparison with [80] would have been interesting but this method is part of a commercial product that could not be shared with us.

We provide a quantitative comparison in Table 5.1 with the model introduced in [181], as well as a single-image upsampling baseline based on the ResUNet architecture [186], which we use as a plug-and-play prior in our model.

To that effect, we first use the validation set of [181] available online (with no overlap with the training set), for which motions are unknown, allowing us to compare with their method, which we outperform by more than 2dBs. In order to perform further comparison and conduct the ablation study, we also build an additional validation set by randomly extracting 266 images from the Zurich raw to RGB dataset, allowing us to generate validation data with known motion. We evaluate variations of our model in the same table, notably comparing the registration accuracy

Method	PSNR (db)	Geom (pix)	SSIM
Scores on public validation set			
ETH [181]	39.09	-	-
Ours (refine)	**41.45**	-	0.95
Scores on our own validation set to conduct the ablation study			
Bicubic Single Image	33.45	-	-
Multiframe L2 only	34.21	-	-
Multiframe L2 + TV prior	34.48	-	-
Single Image	36.80	-	-
Ours (no refinements)	40.38	0.55	0.958
Ours (refinements)	**41.30**	0.32	0.963
Ours (known motion)	42.41	0.00	0.971

Table 5.1: **Results with synthetic raw image bursts** of 14 images generated from the Zurich raw to RGB dataset [204] with synthetic affine motions. Reconstruction error in average PSNR and geometrical registration error in pixels for our models. "known $\mathbf{p}$" is the oracle performance our model could achieve, if motion estimation was perfect.

achieved by these variants by using the geometrical error presented in [74]. More precisely, we perform a small ablation study by introducing a simpler baseline that does not perform joint alignement and only exploits the coarse registration module (no refine baseline). Performing motion refinement significantly improves the registration accuracy and subsequently the image reconstruction quality. Last, we also report the oracle performance of our model with known motions.
We provide a visual comparison in Figure 5.3 with single-image SR baselines and the state-of-the-art method [181] for processing raw image bursts. Only the two approaches processing bursts are able to recover high-frequency details, demonstrating their ability to leverage and remove aliasing artefacts, which are very present in the top image. Significantly better quality results are obtained with our approach.

Impact of burst length and cropping size. The dataset Zurich rgb-to-raw [204] was very useful for training our models, but it unfortunately features relatively small image crops of size 96×96 without giving access to the original megapixel images. By experimenting with real raw data, it became apparent to us to our method was performing better with larger crops (*e.g.*, more than 200×200 pixels), achieving better registration and visually better results. To study the impact of the crop size and burst length, we have thus synthesized additional raw bursts from the DIV2K dataset, and report our experimental results in Figure 5.4, confirming our findings. Note that this does not appear to be a strong limitation of our approach, since in real-life scenarios, we can always assume that the original megapixel image is available. As expected, the performance of our approach is also increasing with the burst size, even though our models were trained with bursts of size 14.

Results on real raw image bursts, dataset of [181]. In Figure 5.5, we show a comparison with [181] using their dataset featuring small crops of size 96×96. As discussed previously, this setup is suboptimal for our approach, but still produces visually pleasant results. Choosing which method performs best here is however very subjective and we found conclusions hard to draw on this dataset. Whereas the images produced by [181] may sometimes look slightly sharper, one may argue that our approach seems to recover more reliable details, *e.g.*, the text is perhaps

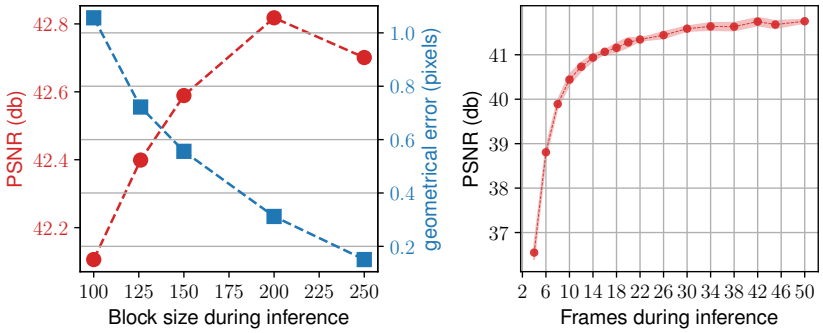


Figure 5.4: Left: Impact of the crop size on the registration and reconstruction performance. Right: Impact of the burst length, see main text for details.

easier to read. Note that our models were trained on synthetic data only and we leave fine-tuning with real data on this dataset for future work. There is an attempt in [181] to address the open problem of quantitative evaluation with real data using a custom metric, but, like any other attempt so far, it is flawed since (i) it is based on the alignment method of [181], with an unavoidable slight bias in its favor, and (ii) it assumes ground truth from a particular Canon camera. Interestingly, this score improvement does not always correlate with visual quality, as shown by Figure 6. This is by no means a criticism of [3]: we believe instead that quantitative evaluation on real images is an extremely challenging problem, far from being solved. Since the submission of our paper, the results of the NTIRE 2021 burst super-resolution challenge have been published [205]. Our method ranked third quantitatively in the "synthetic data" part of the challenge that we entered.

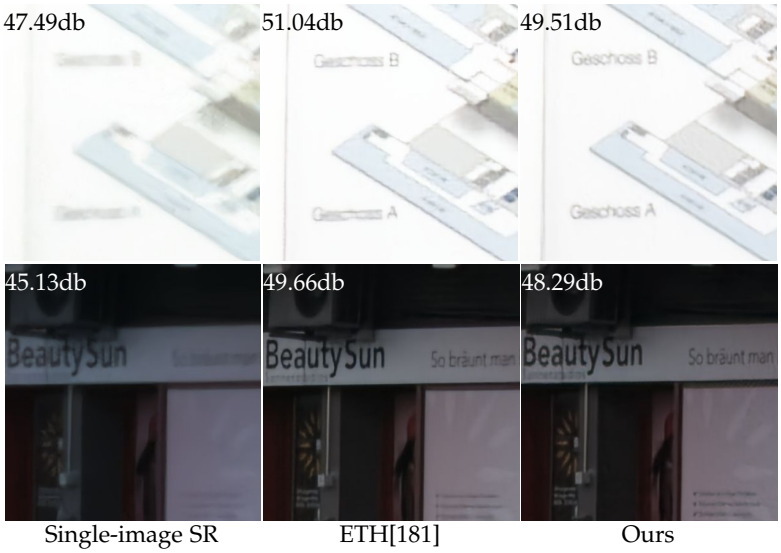


Figure 5.5: **Results from real raw bursts** from dataset of [181] including Aligned PSNR score (see main text).

Results on real raw image bursts from various devices. Finally, we demonstrate the effectiveness of our approach on real raw bursts acquired by different devices. We consider a Panasonic Lumix GX9 camera, which is interesting for SR as it does not feature an optical anti-aliasing filter, a Canon Powershot G7X camera, a Sam-

sung S7 and a Pixel 4a smartphones. Results obtained in high noise regimes have already been presented in Figure 6.1, showing that our approach is surprisingly robust to noise. We believe that the result is of interest since it may allow photographers to use high ISO settings in low-light conditions, without sacrificing image quality. Other results are presented in Figure 5.5 on low-noise outdoor conditions with bursts of 20 to 30 raw images. In all cases, the method succeeds at recovering high-frequency details. Many more examples and comparisons with other multiframe methods are provided in the supplementary material. We also present failure cases, corresponding in large parts to scene motion. Last, we remark that our method is relatively fast at inference time. Processing a burst of 20 raw 300×300 images takes for instance about $1s$ on an Nvidia Titan RTX GPU, producing an upsampled image of size 1200×1200.

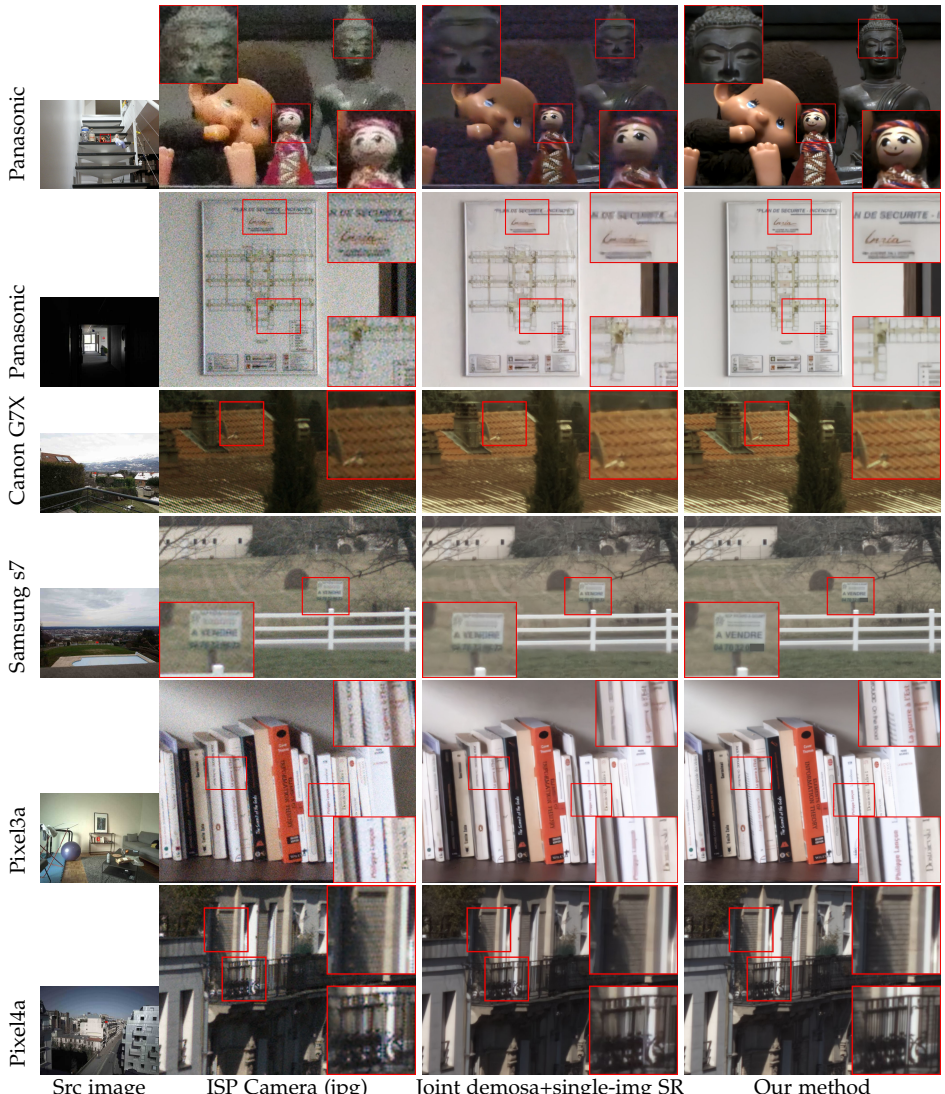


Figure 5.6: Results from real raw image bursts obtained with various cameras. We provide comparisons with single image and multiframe baselines. Finest restored details can be seen by zooming on a computer screen. The last three digits of the phone number, only legible in our reconstruction, are masked in the figure for privacy concerns.

5.5 Conclusion

We have presented a simple but effective method for superresolution that combines the interpretability of model-based approaches to inverse problems with the flexibility of data-driven architectures and can be learned from pairs of synthetic LR and real HR images. We plan several extensions, including using multiple cameras to add real LR-burst/HR-image pairs to the training mix, and at test time to take advantage of the multiplicity of imaging devices now available on high-end smartphones. This will open the door to wide-baseline super-resolution applications, such as the construction of high quality panoramas and finely detailed texture maps in multi-view stereo reconstructions. Finally, we plan to explore several other extensions of our approach, including tackling blurry bursts, extending super-resolution to reconstruct HDR images, and pursuing applications in the astronomy and microscopy domains

5.a Appendix

This supplementary material presents additional qualitative and quantiative results. In Figure 5.7 we present additional visual comparison with two burst denoising methods on real images. In Table 5.2 we present additional experiments on RGB images. Figures 5.8 and 5.9 are devoted to super-resolution experiments from real raw data from different smartphones (Google Pixel 3a and 4a, Samsung S7 and S10) and cameras (Panasonic Lumix GX9 and Canon Powershot G7X) and comparison with additional baselines. In Figures 5.10, we present extreme upsampling results by using synthetic RGB image bursts. In Figure 5.11, we present restoration results obtained from real images with very low SNR to illustrate the efficiency of our method to perform blind denoising. In Figures 5.12 and 5.13, we study the effect of the number of frames in the burst on the reconstruction, both in the low SNR and high SNR settings. Finally, we present failure cases in Figure 5.14, where fast moving objects are present in the scene.

5.a.1 Comparison with burst denoising methods

We perform additional qualitative comparison on a real image with two burst denoising methods. We compare our method with [206] which performs joint denoising and demosaicking on a burst of raw images. We use the code and the pretrained model made available online. We also use the code and pretrained model of [207]. However the model is only designed to perform grayscale burst denoising, so we perform denoising independently on each RGB channel and then perform demosaicking to get an RGB image. Despite our best efforts for tuning the parameters of these methods to maximize visual quality, the results obtained are not as good as our method (see Figure 5.7 below). We believe this is not surprising since each one of these methods only addresses a subset of our problem. Adapting them successfully to our general setting is not trivial.

Figure 5.7: Comparison with joint denoising and demosaicking methods.

5.a.2 Evaluation on RGB Images

We compare our approach on the BSD68 dataset against state-of-the-art single-image and video super-resolution algorithms (considering a burst as a video sequence) and report the HR image reconstruction accuracy in terms of average PSNR in Table 5.2. For the training with RGB data, we perform 80 000 iterations of the ADAM optimizer with a batch size of 10, a burst size of 14 and with a learning rate of 3×10^{-5} decaying by a factor 2 after 40 000 iterations. For evaluating the model VSR-DUF [178], we use the code and the pretrained models made available online by the authors. Other single-image reconstruction results are from [186].

In the present setting, we consistently outperform other baselines, notably demonstrating that burst SR cannot simply be addressed effectively by current video SR approaches. We also note that our models perfom better with less blurring (and more aliasing). Finally, we evaluate variations of our model in the same table, notably comparing the registration accuracy achieved by these variants by using the geometrical error presented in [74]. More precisely, we perform a small ablation study by introducing a simpler baseline that does not perform joint alignement and only exploits the coarse registration module (no refine baseline). Performing joint alignment and image estimation systematically improves motion estimation. Last, we also report the oracle performance of our model with known motions.

Method	Scaling factor / blurring kernel std		
	$\times 2/\sigma = 0.7$	$\times 3/\sigma = 1.2$	$\times 4/\sigma = 1.6$
Single Image SR			
RCAN [208]	29.48	27.30	25.59
IRCNN [97]	29.60	26.89	25.32
USRNet [186]	30.55	27.76	26.18
Video SR			
VSR-DUF[178]	-	31.03	29.24
Ours (no refine)	42.36/0.10	32.63/0.14	**30.00**/0.19
Ours	**43.73**/0.07	**33.10**/0.10	29.87/0.14
Ours (known $\mathbf{p}$)	45.72/0.00	34.47/0.00	31.32/0.00

Table 5.2: **Results for RGB with synthetic affine motions**, of different methods for different combinations of scale factors and blur kernels. Results are given in term of average PSNR in dBs and geometrical registration error in pixels for our models. "known $\mathbf{p}$" is the oracle performance our model could achieve, if motion estimation was perfect.

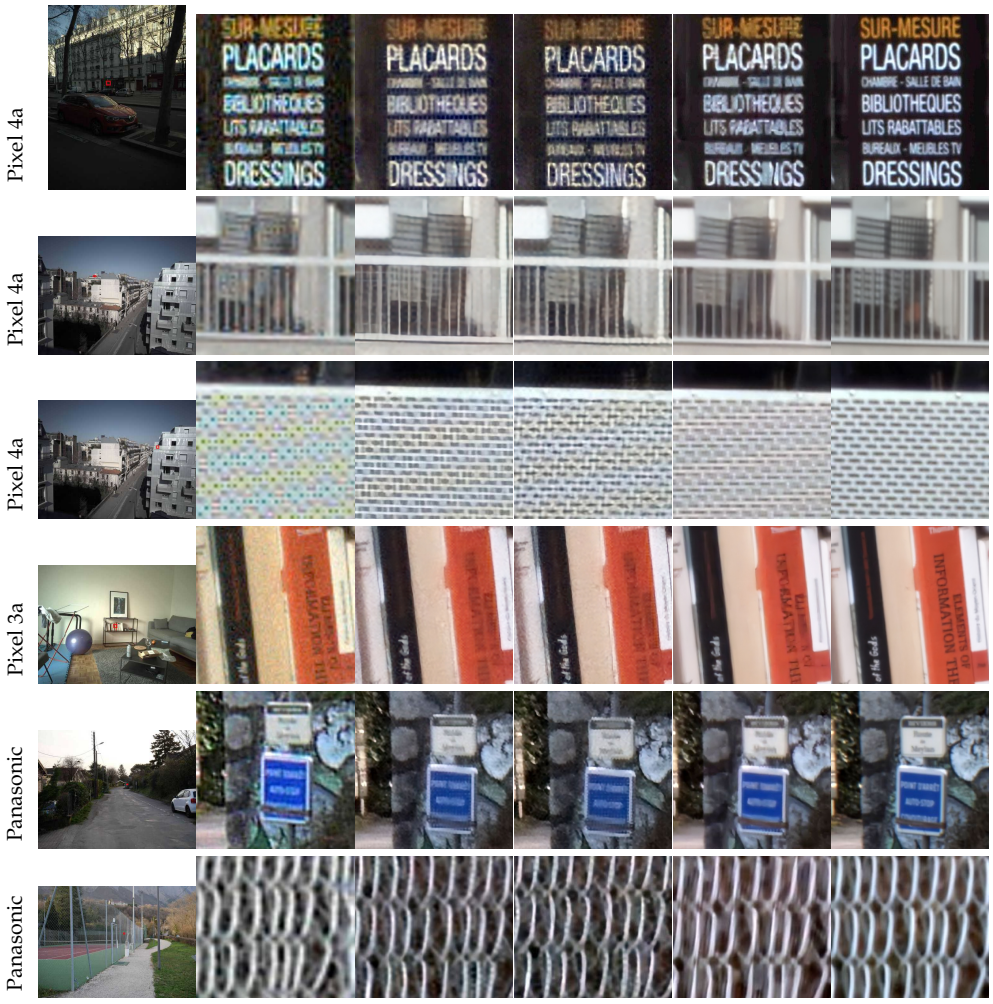


Figure 5.8: Results from real raw image bursts obtained with various cameras. We provide comparisons with single image and multiframe baselines. Finest restored details can be seen by zooming on computer screen.

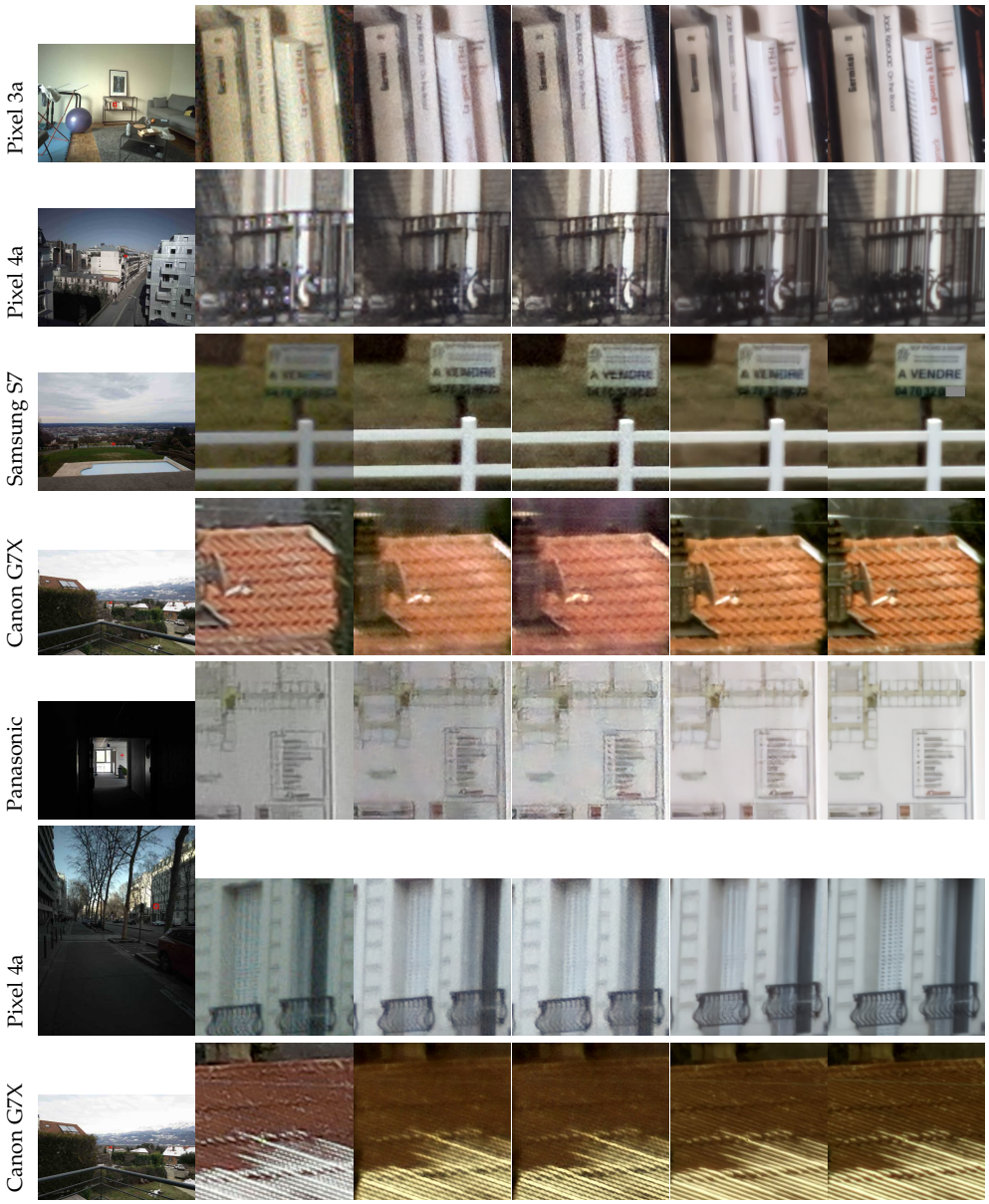


Figure 5.9: Results from real raw image bursts obtained with various cameras. We provide comparisons with single image and multiframe baselines. Finest restored details can be seen by zooming on computer screen.

Figure 5.10: Extreme $\times 16$ upsampling experiment. The right image is obtained by processing a burst of 20 LR images presented on the left obtained with synthetic random affine movements and average pooling downsampling

Figure 5.11: Image restoration of images taken at night with very low signal to noise ratio by using a Panasonic GX9 camera.

Figure 5.12: Visual differences caused by merging a different number of frames in the case of low SNR scenes. With a larger number of frames we can observe a quality increase and better denoising.

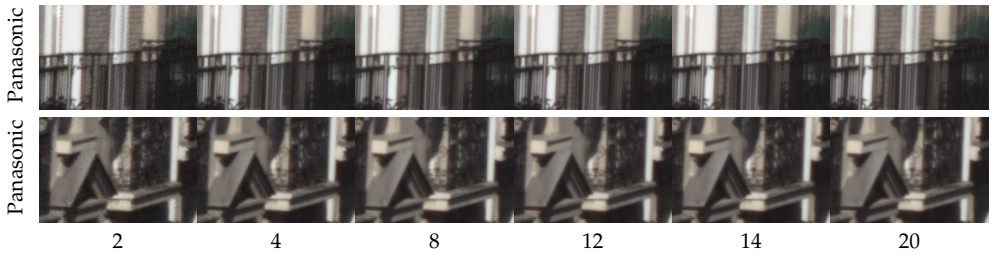


Figure 5.13: Visual differences caused by merging a different number of frames in the case of high SNR scenes. With a larger number of frames we can observe a quality increase.

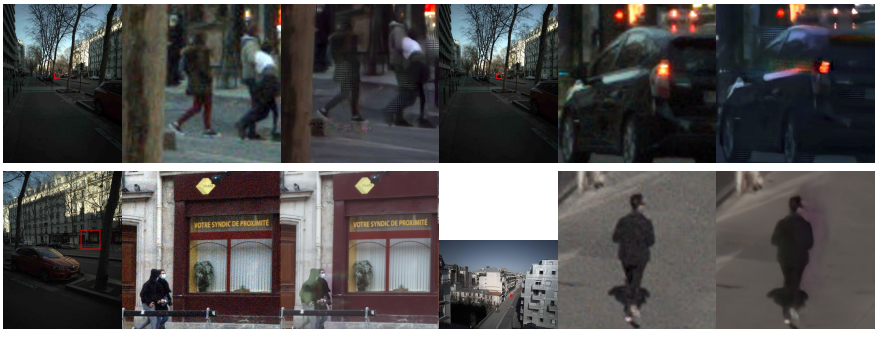


Figure 5.14: Misalignements artefacts due to moving objects in the scene. Our current implementation does not handle fast moving objects and then generates visual artefacts. Dealing with fast dynamic scenes will be the focus of future work.

Chapter 6

Joint HDR and Super-Resolution from Bracketed Raw Bursts

Chapter abstract: Photographs captured by smartphones and mid-range cameras have limited spatial resolution and dynamic range, with noisy response in underexposed regions and color artefacts in saturated areas. This paper introduces the first approach (to the best of our knowledge) to the reconstruction of high-resolution, high-dynamic range color images from raw photographic bursts captured by a handheld camera with exposure bracketing. This method uses a physically-accurate model of image formation to combine an iterative optimization algorithm for solving the corresponding inverse problem with a learned image representation for robust alignment and a learned natural image prior. The proposed algorithm is fast, with low memory requirements compared to state-of-the-art learning-based approaches to image restoration, and features that are learned end to end from synthetic yet realistic data. Extensive experiments demonstrate its excellent performance with super-resolution factors of up to $\times 4$ on real photographs taken in the wild with hand-held cameras, and high robustness to low-light conditions, noise, camera shake, and moderate object motion.

B. Lecouat, T. Eboli, J. Ponce, J. Mairal. High Dynamic Range and Super-Resolution From Raw Image Bursts. *ACM Transactions on Graphics (SIGGRAPH)*, 2022.

Contents

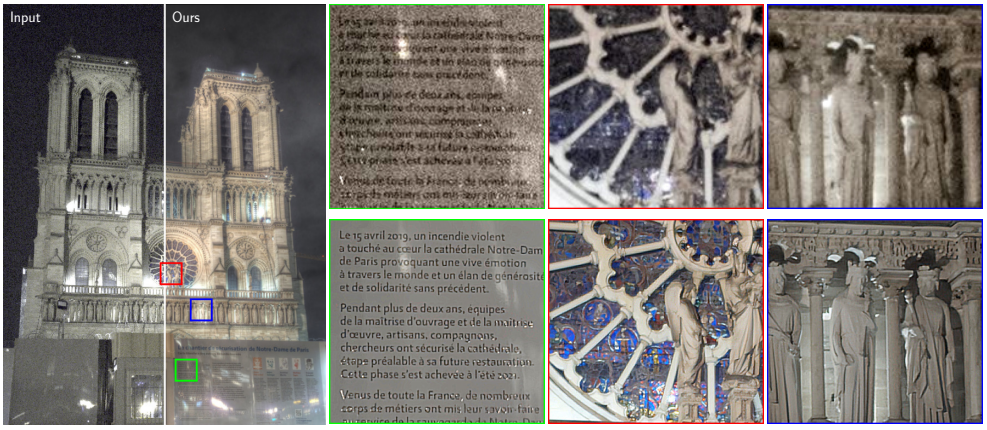


Figure 6.1: An example of joint super-resolution (SR) and high-dynamic range (HDR) imaging. **Left:** An 18-photo burst was shot at night from a hand-held Pixel 4a smartphone at 12MP resolution with an exposure time varying from 1/340s to 1/4s. The left half of the central image from the burst is shown along with the right half of the 192MP HDR image reconstructed by our algorithm with a super-resolution factor of $\times 4$ (after tone mapping). **Right:** Three small crops of the two images corresponding to the colored square regions on the left. Crops from the central image of the burst are rendered using Adobe Camera Raw to convert raw files into jpg with highest quality setting. The HDR/SR results are rendered using the PhotoMatix tone mapper`https://www.hdrsoft.com/`. Note that the 192MP HDR image on the left is not reproduced at full resolution because of the corresponding file's size.

6.1 Introduction

Key factors limiting the level of detail of photographs captured by digital cameras are their spatial resolution and dynamic range: High resolution is necessary to zoom on small image regions, and high dynamic range is needed to reveal details hidden in dark areas (*e.g.*shadows) and avoid color artefacts due to saturation in bright ones (*e.g.*highlights). For a given sensor size, higher resolution also means smaller pixel size, with less light reaching each photoreceptor, resulting in lower dynamic range and increased noise in dark regions, an effect exacerbated in smartphones by their small sensor size. It is natural, and by now rather common, to

use multiple photographs to reconstruct an image with higher spatial resolution, a process known as *super-resolution* (or *SR* for short in this presentation, see, for example [53]), or dynamic range, a process known as *high dynamic range* (or *HDR*) imaging (see, for example [209]).
We propose in this paper a novel method for *joint* SR and HDR imaging from the *raw* image bursts featuring a range of different exposures that can now be captured by most smartphones and mid-range cameras (Figure 6.1). A major challenge tackled by our algorithm is the automated alignment with sub-pixel accuracy of the burst elements required to compensate for camera shake and possibly (moderate) object motion, despite the variations in saturation and signal-to-noise ratio due to the different exposures used across the burst. Other notable difficulties include the high contrasts and noise levels encountered in night scenes for example, where a photo might feature both very dark and noisy regions and saturated ones near light sources, as well as the fact that a digital camera only captures one color channel at each pixel according to the corresponding *color filter array* (or *CFA*, often a Bayer pattern). Despite the latter challenge, it now seems clear that it is better to work directly with the raw image data than with the sRGB pictures produced by the *image signal processor* (or *ISP*) of the camera since their construction involves several steps, including white balance, denoising, demosaicking, gamma correction, compression of each color channel content to 8 bits, etc., that result in an unavoidable loss of information in high spatial frequencies and dynamic range.
The approach proposed in the rest of this presentation extends the algorithm for multi-frame super-resolution of [52] to jointly perform blind denoising, demosaicking, super-resolution and HDR image reconstruction from raw bursts. Its key features can be summarized as follows:

- Our method uses a physically-accurate model of image formation that accounts for the successive transformations applied to the original analog irradiance image, including quantization of the signal, noise, exposure and spatial quantization.

- We combine an iterative optimization algorithm for solving the corresponding inverse problem with a learned image representation for robust alignment and a learned natural image prior. This is the first main technical novelty of our paper, enabling us to address the joint reconstruction of high-resolution, high-dynamic range color images from raw photographics bursts captured by a handheld camera with exposure bracketing.

- The proposed algorithm is fast, with low memory requirements compared to state-of-the-art learning-based approaches to image restoration, and features that are learned end to end from synthetic yet realistic data, generated using again our image formation model.

- We introduce an image alignment method to compensate for camera shake which is robust to (moderate) object motions and an image fusion technique which is itself tolerant to alignment errors. Together, these form the second main technical novelty of our paper, and they are key factors in the robustness of our algorithm in both the SR and HDR imaging tasks with, notably, significant improvement over [52] in super-resolution.

- Extensive experiments demonstrate the excellent performance of the proposed approach with super-resolution factors of up to $\times 4$ on real photographs taken in the wild with hand-held cameras, and high robustness to low-light conditions, noise, camera shake, and moderate object motion. These results are confirmed by quantitative and qualitative comparisons with the state of the art in super-resolution and HDR imaging tasks on synthetic and real image bursts.

6.2 Background

6.2.1 High Dynamic Range Imaging

Bracketing techniques. [210, 209, 211, 212] construct an HDR image by combining multiple photographs of the same scene with different exposures. The darkest pictures are used to reconstruct areas prone to saturation and the brightest ones are needed for restoring dark regions that are likely to be noisy (we will come back to that point later). They typically work on *linRGB* images, that is, demosaicked images *before* they are transformed by the camera's ISP into *sRGB* images ready for display. A sequence of sRGB input photographs must therefore in general be "linearized" by inverting this mapping, also known as the camera response function (or *CRF*). The HDR image is then reconstructed as a weighted sum of the linearized bracket images, normalized by the corresponding shutter speed. Its pixel values are typically represent as single-precision floating-point numbers, with min and max those of the image bracket. Bracketing-based approaches to HDR imaging face a number of classical issues, including choosing the optimal fusion weights, estimating the CRF [209], leveraging accurate raw image noise models [211, 213, 43], selecting the best exposure parameters for a fixed number of frames in the bracket [214, 212], registering images with different exposures [215, 216], which is significantly more challenging than aligning same-exposure images [217], and removing ghosting artefacts [218, 219] due to misalignment.

Using raw bursts with constant shutter speed. Unlike classical exposure bracketing techniques, HDR+ [220] takes as input a burst of raw underexposed images captured with the same exposure time. These are mostly free of saturation but noisy in dark regions. A 12-bit, denoised raw image is obtained by aggregating the 10-bit photos of the burst. It is then demosaicked and tone mapped. Recent updates of HDR+ use a couple of well-exposed frames to achieve better denoising and deghosting [221], or leverage the metering technique of [212] to adapt the original algorithm to low-light situations [222].

Using pixelwise ISO sensitivities. Instead of relying on classical imaging devices, [223] reconstruct a single HDR image from a sensor with spatially-varying pixel exposures. This approach can be further combined with learning-based methods [224, 225]. Even though our work focuses on standard sensors, we believe it to be flexible enough to be adapted to pixelwise ISO sensitive sensors under simple modification of the image formation model. This is an interesting research direction for future work, but beyond the scope of our paper.

Learning-based methods for HDR imaging have also been proposed. [226] introduce a convolutional neural network (CNN) to predict the irradiance from three

LR frame 1 LR frame 6 LR frame 12 LR frame 18 Ours (HDR+SR $\times 4$)

Figure 6.2: Exposure bracketing: **Left:** Three high dynamic range, high-resolution images obtained by our method from 18-image bursts taken by a handheld Pixel 4a smartphone with a $\times 4$ super-resolution factor. We show post-processed sRGB pictures for the sake of presentation. **Right:** Small crops from sample photos in the burst and our reconstruction. Note the high level of noise in the short-exposure images, in particular in the second row, and the saturated regions in the long-exposure ones. As shown by the last column of the figure, our algorithm recovers details in saturated areas and remove noise in the darkest regions. The reader is invited to zoom in on a computer screen.

low-dynamic range (LDR) images, with different exposures, camera poses and possibly moving subjects, pre-aligned with an optical flow algorithm. Most recent CNN-based multi-image methods [227, 228, 229, 230, 231] learn to align and fuse demosaicked images in an end-to-end manner, and they typically operate on image triplets such as those in the dataset of [226]. CNN-based approaches to single-image HDR include [232, 233, 234, 235]. They rely on machine learning to recover missing details in the darkest and saturated areas of tone-mapped images.

6.2.2 Super-Resolution

We limit here our discussion to multi-frame super-resolution algorithms. Although single-image learning-based techniques have been used to generate very impressive and highly-detailed images [236, 237], their objective is not the same as ours: they aim at generating a high-resolution picture *compatible* with one input photograph, whereas we want to reconstruct the details that are *actually available* in the input burst.

Energy-based methods. High-frequency information present in low resolution (LR) photos with aliasing artefacts is useful for reconstructing a high-resolution (HR) image from multiple LR frames [238]. Unfortunately, this information is typically lost during the denoising and demosaicking steps performed by the camera ISP pipeline to produce sRGB images. [238] estimate an HR demosaicked image from a sequence of raw photographs by minimizing a penalized energy—that is, they solve an inverse problem via optimization. [53] adapt the kernel method of [239] and exploit natural hand tremor to jointly demosaick and super-resolve a raw image burst with magnification factors up to $\times 3$ in a fraction of a second on a handheld smartphone.

Learning-based techniques. [54] learn a CNN with attention module to align, demosaick and super-resolve a burst of raw images. In a follow-up work, [240] minimize a penalized energy including a data term comparing the sum of parameterized features residuals. [52], learn instead a hybrid neural network alternating between aligning the images with the Lucas-Kanade algorithm [241], predicting an HR image by solving a model-based least-squares problem and evaluating a learned prior function. [242] propose a neural network architecture that aligns an input burst of images while performing super-resolution with a non-local fusion module.

6.2.3 Joint HDR Imaging and Super-Resolution

The algorithms proposed by [243, 244] address joint SR and HDR imaging with an existing SR energy-based solver. To tackle the multi-exposure setting, they introduce weights inspired by bracketing techniques in the least-squares term. More generally, this joint image restoration problem has been addressed in a two-stage fashion: (i) image registration with an algorithm robust to varying exposures and (ii) solving a least-squares problem including operators modelling both SR and HDR. For instance, [245] propose an exposure-invariant transform before applying the FFT-based registration technique of [246]. The image is then obtained by solving a penalized least-squares problem. [215] use an optical flow approach with normalized gradients for robustness to changes of exposure, and the HR/HDR image is found by solving again a penalized least-squares problem. [247] adapt a backprojection algorithm to the multi-exposure setting and simply solve a weighted least-squares problem without prior, with comparable performance but lower computational cost. [248] explore the case where the LDR SR images are also blurred with camera shake or motion blur. Similar to the HDR case, CNNs have also been proposed for single-image joint SR and HDR, *e.g.*[249], while [250] address instead joint SR, HDR and tone mapping by merging a pair of previously aligned over- and under-exposed images with a two-stream CNN. In contrast with these techniques, we use trainable image features to adapt the raw image registration module of [52] to the varying-exposure setting in a robust manner, and jointly learn these features and a parametric image prior in an end-to-end manner.
Figure 6.2 shows examples of the input data these methods use and samples of the the predicted high-resolution HDR images we predict with the proposed approach.

6.3 Image formation model

We now describe the process generating a burst of low-dynamic low-resolution raw images from a high-resolution HDR image. This process yields a natural inverse problem formulation, which we will leverage later to build a trainable architecture.

6.3.1 Dynamic Range

After analog-to-digital conversion, a camera sensor outputs a black-and-white mosaicked image whose pixel values are integers obtained by quantizing the number of photons collected by each photosite on a linear q-bit scale [251], where q is called the *bit depth* of the sensor. We denote by P_q the set of the discrete values a pixel may take, as measured in *data numbers* (or *DNs* [44, 251]), from 0 to $2^q - 1$.
The dynamic range $R(u)$ for a pixel u is defined as the ratio of the largest to the smallest values this pixel may take: the larger the bit depth of the sensor, the greater is its maximal value in P_q. The ratio is usually given in photographic *stops*, where

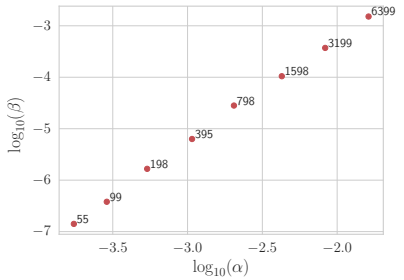


Figure 6.3: Empirical measurements of the shot and read noise levels α and β from the metadata of raw images taken with the Google Pixel3a smartphone. The numbers next to the markers are the corresponding empirical ISO levels. As observed by [255], there exists a linear relationship between $\log_{10}(\alpha)$ and $\log_{10}(\beta)$ that we leverage to train our models.

each stop corresponds to a multiple of 2. In practice, the largest value u can take is limited either by the bit depth q or the white level c set by the camera, to prevent color artefacts in highlights [252], whereas the lowest value is actually limited by the noise $\varepsilon(u)$ and by the camera black level b [253]. Note that even in the absence of light, $\varepsilon(u)$ is never 0 since any digital camera suffers from various sources of electronic noise [212]. This also shows that increasing dynamic range is strongly related to denoising, as discussed later in this section.

6.3.2 Exposure

As mentioned above, raw pixel values depend linearly on the number of photons captured by each photosite (ignoring quantization effects) and thus on exposure time. In photography, this effect is quantified by the *exposure value* (or *EV*): Increasing it by +1EV (resp. decreasing by -1EV) corresponds to doubling (resp. halving) the raw pixel values. The EV depends on the *ISO gain*, aperture size and exposure time. In this work, we will only control the exposure time Δt, keeping it small enough to (mostly) avoid motion blur, and keep the other two quantities constant since modifying the ISO gain may change the noise distribution [212] and adjusting the aperture size changes the blur of out-of-focus regions [254].
The raw value $y(u)$ in P_q recorded at some pixel u is thus related to the irradiance $x(u)$ in $\mathbb{R}^+$ at the same location by

$$y(u) = S(\Delta t x(u)), \tag{6.1}$$

where S is the function mapping pixel values from $\mathbb{R}^+$ to P_q. This equation is only valid when $S(\Delta t x(u)) < 2^q - 1$, with saturation occurring for higher values. Using short exposure times limits saturation, but, as shown in the next section, leads to a poor *signal-to-noise ratio* (or *SNR*).

6.3.3 Noise and SNR

The raw image noise $\varepsilon(u)$ at each pixel comes from the physics of light and the electronics of the camera. The former is called *shot* noise, and it can be modelled with a Poisson distribution [253]. The latter is often referred to as *read* noise and corresponds to random signal fluctuations caused by the electronics and quantization effects. It is usually modelled with a zero-mean Gaussian distribution [253]. The combination of shot and read noise can be modelled by a single random variable

$\varepsilon(u)$ following a zero-mean Gaussian distribution with pixel-dependent standard deviation, defined for any pixel value $y(u)$ as [253, 255, 256]:

$$s(u) = \sqrt{\alpha y(u) + \beta}, \tag{6.2}$$

where α and β are respectively the variances of the shot and read noise. Figure 6.3 shows the distribution of α (shot noise level) and β (read noise level) for the Google Pixel3a camera. We have obtained theses values from the EXIF metadata of raw images taken with the smartphone. Each marker corresponds to a couple $(\log_{10}(\alpha), \log_{10}(\beta))$ for an ISO level. In dark regions, read noise dominates shot noise, and limits the total dynamic range.
For the Poissonian-Gaussian noise model of Eq. (6.2), the SNR is:

$$\mathrm{SNR}(u) = \frac{m(u)y(u)}{s(u)} = \frac{m(u)y(u)}{\sqrt{\alpha y(u) + \beta}}, \tag{6.3}$$

where m is a binary mask excluding the saturated pixels. It is a monotonically increasing function of the pixel value $y(u)$, essentially linear in dark regions (*e.g.*shadows) where read noise dominates shot noise, and essentially proportional to $\sqrt{y(u)}$ in bright regions (*e.g.*highlights) where the opposite occurs [211]. As already discussed in the previous section, noise removal is essential for generating images with high dynamic range, and Equation (6.3) shows that high raw pixel values lead to better SNR and thus better dynamic range in both dark and bright image regions. But high pixels values everywhere in an image can typically only be achieved at the cost of saturating the brightest areas. Exposure bracketing avoids this problem by using the longest exposures to eliminate read noise from dark regions and the shortest ones to avoid saturation in bright spots.

6.3.4 Overall Image Formation Model

The original analog image cannot be recovered on a computer and we instead focus on estimating a discrete HR/HDR $sh \times sw \times 3$ photograph x with pixel values in $\mathbb{R}_+$ from a burst of K raw LR and LDR images y_k ($k = 1, \dots, K$) of size $h \times w$ with entries in P_q. The integer s is the super-resolution factor. Following [52], let us introduce the warp operator W_k associated with the kth photo in the burst and accounting for camera shake, the blur operator B taking into account the integration of the signal over the pixel area is modeled by a convolution, the decimation operator D_s associated with the super-resolution factor s, and the C operator is a binary mask modeling the sensor CFA. Putting them together and taking into account the exposure time Δt_k, the analog low-resolution image associated with the irradiance image x is $a_k = CD_sBW_k(\Delta t_k x)$, which can be rewritten as $a_k = A_k x$, where $A_k = \Delta t_k CD_sBW_k$ (the factor Δt_k commutes with the operators since it only scales the image values).
Combining this model with Eq. (6.2) and (6.1) yields, for all $k = 1, \dots, K$:

$$y_k = S\left(A_k x + \varepsilon_k\right), \tag{6.4}$$

where we abuse the notation so S operates on a whole image instead of a scalar, and ε_k is a zero-mean Gaussian noise with pixel-dependent variance $\alpha A_k x + \beta$ according to Eq. (6.2). The operator CD_sB impacts the spatial resolution, while S and the noise variance limit the dynamic range of each image y_k.
Note that our model assumes that the scene is static during burst acquisition, which may result in ghosting artefacts in the presence of scene motion, when using this

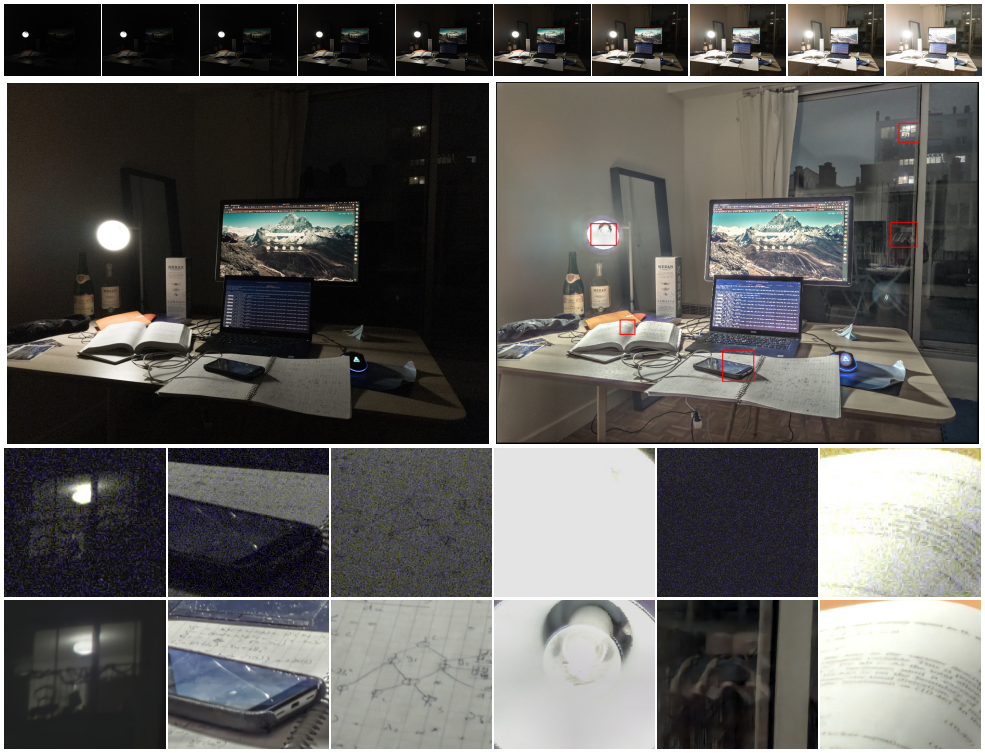


Figure 6.4: Joint HDR imaging and super-resolution $\times 4$ with a burst taken with a hand-held Pixel4a at night, facing a spotlight. **Top:** The original burst. **Middle:** The central image in the burst (left) and the reconstructed HDR/SR image after tone mapping (right). **Bottom:** Six crops showing details of the original and HDR/SR images, presented respectively in the first and second rows.

model within an inverse problem formulation. We will, however, introduce in the next section simple weighting strategies to make our approach robust to moderate scene motion.

6.4 Proposed Approach

The goal of this work is to design a function F_θ with learnable parameter θ which, given K raw images $Y = \{y_1, \dots, y_K\}$ and corresponding exposure times $\Delta = \{\Delta t_1, \dots, \Delta t_K\}$, predicts a single-precision floating-point estimate $\hat{x}$ of the the HR $sh \times sw \times 3$ irradiance map:

$$\hat{x} = F_\theta(Y, \Delta). \tag{6.5}$$

As explained later in this section, all images of the burst are automatically aligned on a reference frame y_{k_0} (typically the central one that has in general a reasonable exposure).

6.4.1 Formulation of the Problem

Inverse problem. Our image formation model (6.4) suggests using an inverse problem formulation to the design of F_θ and the recovery of $\hat{x}$. We first convert the discrete raw pixel values from y_k in P_q into 32-bits real values in $[0, 1]$,

and construct the binary mask $m(y_k)$ representing saturated pixels containing non-informative values. With an abuse of notation, we keep the notation y_k for the floating-point burst images in the rest of this presentation, and formulate the solution of our inverse problem as the joint recovery of the warp operators $W_1, \dots, W_K$ (parameterized with a piecewise-affine model, as detailed later), and the irradiance image x:

$$\min_{x, W_1, \dots, W_K} \frac{1}{2} \sum_{k=1}^{K} \|w_k \odot (y_k - A_k x)\|_F^2 + \lambda \Omega(x), \tag{6.6}$$

where A_k is the image formation operator defined in the previous section, $\odot$ denotes pointwise multiplication, and the function Ω is a regularizer, and it will be discussed in details later. The $h \times w$ maps w_k store pixel-wise weights used to control the relative contribution of each frame to the reconstruction of each pixel, a key factor for robustness in bracketing methods [211, 213].

A robust weighting strategy. We write

$$w_k = \frac{\Delta t_k m(y_k)}{\sum_{j=1}^{K} \Delta t_j m(y_j)} \odot g\left(y_k, W_k y_1\right), \tag{6.7}$$

where $m(y_k)$ is the binary with zero values at saturated pixels (this formulation assumes the existence of non-saturated pixels at corresponding locations in the burst; when all pixels are saturated, we use uniform weights instead). Here, the function g is a confidence factor, often used in HDR imaging to weight down images incorrectly aligned [219] and avoid ghosting effects. It can be handcrafted from classical image features and/or priors, but we will instead follow a plug-and-play strategy (detailed in the next section) to directly learn a parametric function g from supervisory data. Our overall weighting strategy is useful for HDR since it provides larger weights to frames obtained with longer exposure time that are less noisy, but it also accounts for registration errors through the learned function g, which turns out to be critical for robustness to moderate scene motion.

Warp parameterization. We align images with piecewise-affine warps $W_k = W_{p_k}$, where W_{k_0} is the identity and $p = \{p_1, \dots, p_K\}$ is the set of warp parameters. This is implemented by tiling the images into small (*e.g.*200×200) crops, that are aligned independently with affine transformations with 6 parameters.

Regularizer. Many classical regularizers can be used in the formulation of inverse problems in image processing applications, for example sparse total-variation priors [243] or combinations of penalty functions computed from pixel or histogram values [257, 245, 209]. We instead follow the same plug-and-play strategy as for the confidence function g, and learn a CNN in place of the proximal operator [64] of the penalty function Ω. We detail its implementation in Sec. 6.4.3.

6.4.2 Optimization Strategy

We solve our optimization problem with half-quadratic splitting (or HQS) [258] by introducing an auxiliary variable z and minimize

$$\min_{x, z, p} \frac{1}{2} \sum_{k=1}^{K} \|w_k \odot (y_k - A_k z)\|_F^2 + \frac{\eta}{2} \|x - z\|_F^2 + \lambda \Omega(x). \tag{6.8}$$

The parameter η is usually increased at each iteration according to some preset schedule, which guarantees that, as η grows, the solution of this relaxed problem converges to that of the original one (6.6) [258]. As detailed in Sec. 6.4.3, we choose instead to learn this parameter from training data, which improves performance in practice. Note that we now find the warp operators by minimizing the energy with respect to the warp parameters p, and that all operators involved are implemented efficiently by exploiting the image structure (*e.g.*convolutions instead of large sparse operators, etc.). The optimization is carried out by first initializing z and p, then, in an alternate fashion, repeating T times ($T = 3$ in our implementation) an HQS stage consisting of the three steps detailed below. The motivation for this strategy is that it allows us to gracefully convert our optimization method into a trainable architecture, as discussed in Sec. 6.4.3, thanks to automatic differentiation tools [259] implemented in modern deep learning frameworks.

Updating z. The auxiliary image z is updated by a few steps of a simple gradient descent (GD) algorithm:

$$z \leftarrow z - \delta \left(\eta(z - x) + \sum_{k=1}^{K} A_k^\top \left(w_k^2 \odot (A_k z - y_k) \right) \right), \tag{6.9}$$

where δ is a step size (which will be learned automatically by the procedure presented in the next section), and of course A_k depends on the current warping parameters p.

Updating x. Minimizing (6.8) with respect to the image x while keeping the other variables fixed amounts to compute the so-called proximal operator G of Ω [64]:

$$x = G(z, \lambda/\eta) = \arg\min_x \frac{1}{2}\|x - z\|_{\mathrm{F}}^2 + \frac{\lambda}{\eta}\Omega(x). \tag{6.10}$$

We will detail in the next section how we implement G.

Updating p. [52] estimate the warp parameters p_k ($k \neq k_0$) on 200×200 tiles in a 4-scale Gaussian image pyramid, running three stages of the Lucas-Kanade algorithm [241] at each stage. We will show in Sec. 6.4.3 how to do significantly better, both quantitatively and qualitatively, by using a similar approach to align *learned* features instead.

Initialization of p and z. A fast and coarse initialization of the warp parameters p is obtained using a sub-pixel variant of the FFT-based algorithm of [260] with the features of [261]. After having estimated p for the first time with the Lucas-Kanade algorithm and before the first z-update stage, we initialize z as follows: we demosaick each frame y_k with bilinear interpolation, align them with the warping operators W_k, average them with the normalized weights $\Delta_k / \sum_{j=1}^{K} \Delta_j$, and finally upscale the resulting image by a factor s with bilinear interpolation. This procedure yields a fast and coarse estimate of the HR and HDR image to start the GD algorithm in Eq. (6.9).

6.4.3 Learnable Architecture

The optimization procedure described in the previous section is implemented as a function F_θ that produces an estimate $\hat{x}$ from a burst Y and exposure times Δ,

according to Eq. (6.5). By writing this function as a finite sequence of operations that are differentiable with respect to the model parameters θ, it is then possible to leverage training data—that is, pairs of HR/HDR images x associated to LR/LDR bursts—to *learn* these parameters for the reconstruction task. This of course raises questions about data collection and generation, which are discussed later, but it also opens up many possibilities for further improvements. In particular, as described in the rest of this section, this allows us to learn implicitly the regularization function Ω by taking advantage of deep learning principles, as well as learning appropriate weighting strategies, and robust features to improve image alignment.

Learnable proximal operator G. Following the *plug-and-play* strategy [66] which has proven powerful in the signal processing literature, we replace the proximal operator G above by a function G_ω represented by a CNN and parameterized by ω, such that the update (6.10) becomes

$$x = G_\omega(z, \gamma), \tag{6.11}$$

where γ is also a trainable parameter. The CNN has a residual U-net architecture, which is a smaller variant of the network of [186] for single-image super resolution. This network has four scales with respectively 32,64,128,128 channels per scale. We also run experiments with an even smaller version of the network with 32 features per channel (dubbed *small*) and 16 features per channel (dubbed *tiny*). Note that for our problem, the first layer has 4 input channels: three for the predicted RGB auxiliary variable z and one for the scalar γ.

Learnable confidence function g. Similarly, since designing the function g by hand is difficult, we choose to learn instead a CNN g_ρ, and the fusion weights w_k become for all $k \neq k_0$:

$$w_k = \frac{\Delta t_k m(y_k, c)}{\sum_{j=1}^{K} \Delta t_j m(y_j, c)} \odot g_\rho(y_k, W_k y_{k_0}), \tag{6.12}$$

The function g_ρ is implemented with the tiny variant of the U-Net architecture used above. The network takes as input the concatenation along the channel dimension of RGB versions of the images y_k and $W_k y_{k_0}$ obtained by bilinear interpolation.

Learnable features for alignment. A classical approach to the registration of frame captured with different exposure times is to use MTB features [261]. Here, we construct instead a single-channel feature map for each raw image using again the tiny CNN with U-net architecture, then perform the multi-scale Lucas Kanade algorithm for a fixed number of iterations (3 iteration per scale of the pyramid) *directly on the feature map*. Our implementation of the forward additive version of the Lucas Kanade algorithm is fully differentiable. Therefore we can learn the parameters of the feature map jointly with all the trainable parameters of our model, following a strategy similar to [262]. As shown in the experimental section this significantly improves registration performance.

6.4.4 Learning the Model Parameters θ

We denote here by θ all the learnable parameters of our methods, including those of the CNNs and the scalar parameters involved in the HQS optimization procedure introduced above (*e.g.*, δ, η, ...). We use triplets of the form $(x^{(i)}, Y^{(i)}, \Delta^{(i)})$

Figure 6.5: Day-time comparisons of joint HDR imaging and super-resolution algorithms with bursts acquired by a Pixel4a. **Left:** The central image in the burst (top) and our reconstruction (bottom). **Right:** Comparison of close-ups of the reconstructions obtained by the CNN-based Adobe Camera Raw single-image algorithm for $\times 2$ super-resolution and demosaicking, the CNN-based $\times 2$ super-resolution method of [249], and our method. (Note: part of the phone number legible in our case is masked for privacy reasons.)

$(i = 1, \dots, n)$ of training data to supervise the learning procedure. In our setting where ground-truth HDR/HR images are normally not available for real image bursts, the training data is necessarily semi-synthetic, that is, obtained by applying various transformations to real images. Obtaining robust inference with real raw bursts is thus challenging. The hybrid nature of our algorithm, which exploits both a learning-free inverse problem formulation and data-driven priors, appears to be a key to achieving good generalization on real raw data acquired in various conditions that do not necessarily occur in the training dataset.

Dataset generation. Given a collection of *sRGB* images, we construct bursts of LDR/LR raw images and HDR/HR RGB targets using the ISP inversion method of [255] and our image formation pipeline, adjusting the gain to simulate different exposure times. The noise levels are sampled following the empirical model of Figure 6.3.
We generate n bursts $Y^{(i)}$ of synthetic raw SR images from both `.jpg` and `.hdr` images with various simulated exposures $\Delta^{(i)}$. The latter images are important to make our network robust to artefacts occurring near saturated areas. We use the `.hdr` images from [235].

Training loss. With this training data in hand, we supervise our model using the ℓ_1 distance between the target irradiance images $x^{(i)}$ and the predicted ones $F_\theta(Y^{(i)}, \Delta^{(i)})$, and minimize the cost function:

$$\min_\theta \sum_{i=1}^{n} \left\| x^{(i)} - F_\theta(Y^{(i)}, \Delta^{(i)}) \right\|_1 . \tag{6.13}$$

By using a normalized scheme, we avoid the sigmoid activation at the top layer of recent CNNs for HDR imaging [226, 227, 228] forcing the output to be between 0

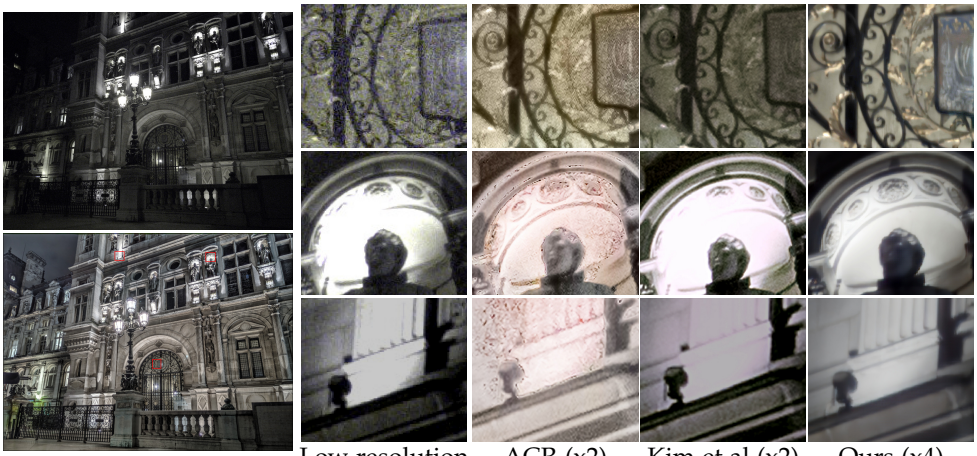


Figure 6.6: Night-time comparisons of joint HDR imaging and super-resolution algorithms with bursts acquired by a Pixel4a. **Left:** The central image in the burst (top) and our reconstruction (bottom). **Right:** Comparison of close-ups of the reconstructions obtained by the CNN-based Adobe Camera Raw single-image algorithm for $\times 2$ super-resolution and demosaicking, the CNN-based $\times 2$ super-resolution method of [249], and our method.

and 1. We have also tried to use the so-called μ-law [226] to include some kind of tone mapping in the supervision but it only marginally improved the visual quality of the images predicted by our model.

Optimizer. We minimize Eq. (6.13) using Adam optimizer with learning rate set to 10^{-4} for 400k iterations. We decrease the learning rate by 0.5 every 100k iterations. The weights of the CNNs are randomly initialized with the default setting of the PyTorch library.

6.5 Results

We first show in Section 6.5.1 several qualitative results illustrating the performance of our method for joint HDR imaging, super-resolution, demosaicking and denoising from real raw image bursts. Qualitative and quantitative comparisons with existing methods for super-resolution, HDR imaging, and registrations are presented in Section 6.5.2, Section 6.5.3 and Section 6.5.4 respectively. The effect of the choice of prior and the robustness of our method for real images are discussed in Section 6.5.5. Additional results, ablations studies, and discussions of its limitations can be found in the appendix.
Note that all the HDR images are rendered using *Photomatix*[1] for tone mapping, which is itself a challenging task [59] beyond the scope of this paper. For baselines operating on RGB images instead of raw photographs, we first process raw files with Adobe Camera Raw to generate RGB images with the highest quality possible.

[1] `https://www.hdrsoft.com/`

6.5.1 Joint SR and HDR on Raw Image Bursts

To the best of our knowledge, we are the first to address jointly HDR, super resolution, demosaicking, and denoising on bursts of raw images. Therefore, we will mostly present here qualitative results, and will defer quantitative comparisons to the following sections that evaluate the performance of our algorithm on separate HDR or SR tasks.
We consider bursts acquired in different settings by a Pixel 3a or 4a camera, by using an Android application to shoot bursts of 11 to 18 raw images. We choose an EV step of 1/3 to 2/3 between each shot. This is particularly important for night scenes to avoid motion blur in the longest-exposure frames. Our method successfully restores finer details and extends the dynamic range of the original shot by denoising dark areas and restoring clipped signals. More precisely:

- Figures 6.4 and 6.6 show night-time photos with large dynamics, similar to Figure 6.1, with both under- and over-exposed areas in the low-resolution central frame. Both the dynamics and the resolution are significantly improved by our algorithm.
- An outdoor day-time photograph is shown in Figure 6.5, with a particularly large dynamic range. The scene contains both under-exposed, noisy areas in the shadows and large bright saturated areas. Note also that the scene contains patterns which are smaller than the resolution of the native image which is a particularly hard setting for demosaicking. Our approach handles such situations well.
- A night scene with both very dark building parts and light bulbs, resulting in a very large dynamic range (Figure 6). Our approach, unlike our competitors, can recover details in both the dark and saturated areas.

Figure 6.7: Visual comparison for super-resolution only on real same-exposure raw bursts, of respectively $K = 20$ and $K = 30$ frames, with state-of-the-art competitors. We *do not* present HDR results in this figure. Our approach limits Moiré artefacts in the first row and reveals in general more high frequency image details in both rows. The last row shows an example requiring deghosting. The ghosted LR image on the left is obtained by averaging the whole burst to show the pedestrian's motion. [54] and our method effectively handle small object motions. The reader is invited to zoom in.

Low resolution [54] [52] [242] Ours

6.5.2 Pure Super-Resolution

We now move to pure super-resolution from raw image bursts, and compare our approach with [52], using examples from their paper. The bursts in this section all have the same exposure, making the alignment simpler compared to the previous section. We first perform a quantitative evaluation on the semi-synthetic benchmark of [54], following their experimental setup and using their dataset. Table 6.1 presents a comparison of our approach for SR, which can be seen as an improved variant of [52]. All methods in the comparison are designed to process raw image bursts. We first note that our improvements in the image registration module yields +1dB over [52] for similar network capacities. The geometric error, measuring alignment discrepancies, is also four times smaller than that of [52], which further suggests the usefulness of our modified Lucas-Kanade module. Since the other methods of the panel do not explicitly predict any motion vector, we cannot compute the corresponding geometric errors. We also have a PSNR gain of about 0.5 to 1dB over three of the recent competitors and fall only behind [242] by less than 1dB but with 13 times fewer learnable parameters. Therefore, the proposed approach is also a compact and competitive algorithm for SR alone. A speed comparison, presented later in Section 6.a.1.4 also shows that our method is faster at inference time.

The previous comparison is conducted on semi-synthetic data, both for training the models and for testing, which makes its conclusions difficult to generalize to the real world of raw bursts from handheld cameras. Nevertheless, it remains the best existing quantitative experimental setup, to our knowledge, since it is not possible to acquire reliable HR ground-truth data along with LR raw bursts. Figure 6.7 shows two challenging real-world examples on which we compare qualitatively the approaches of [54], [52] and [242] to ours, for $\times 4$ super-resolution factor. We display in the first row the results for a burst of $K = 20$ raw frames of a textured surface. Moiré artefacts and aliasing can respectively be noticed in the results from [54] and [242]. Such artefacts are not visible in our reconstruction and that of of [52]. The second row shows the results for a burst of $K = 30$ raw images from [52]. Amongst the four methods in the panel, ours returns the sharpest image, with for instance easier-to-read characters than competitors. We point out that we *have not* used any sharpening algorithm on any of these images.

As remarked by [53], there is a physical limit to the maximum frequency one can reconstruct with aliasing, due to the sensor pitch or the lens point-spread function. We verify this property in Figure 6.8 where we show two crops from the same image, with $\times 2$ and $\times 4$ resolution factors. The first row shows details of a balcony clearly benefiting from a $\times 4$ gain in resolution compared to its $\times 2$ counterpart. The second row shows however that sometimes, as predicted by [53], $\times 4$ upsampling factor may not reveal finer details that its $\times 2$ counterpart.

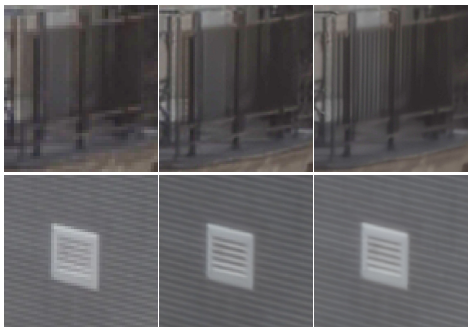
6.5.3 Pure HDR Imaging

We evaluate the ability of our approach to align and merge raw images into HDR image at the same resolution as the input.

We compare our approach with a bracketing technique, implemented with the weights of [212], two state-of-the-art CNNs [227, 228] trained to predict a 32-bit image from only three LDR images with -2, 0 and +2EV or -3, 0 and +3EV, and recent single-image HDR CNNs [234, 235]. We generate 266 raw bursts with 32-bit ground-truth images, each burst containing 11 synthetic raw images with small random shifts and rotations and Poissonian-Gaussian noise with parameters α and

Table 6.1: Super-resolution ($\times 4$) comparison with a selected panel of recent methods with average PSNR and geometric error when it can be computed. We *do not* perform HDR generation in this experiment. Our method falls behind that [242] within a margin of less than 1dB but with 13 times fewer parameters. We gain 1dB compared to [52] with a similar number of parameters by upgrading the registration module.

Model	# parameters	PSNR	Geom (avg)
[54]	13M	40.76	N/A
[52]	3M	41.45	<u>2.56</u>
[240]	-	41.56	N/A
[263]	6.6M	41.93	N/A
[242]	26M	**43.35**	N/A
Ours	3M	<u>42.42</u>	**0.80**



LR HR ($\times 2$) HR ($\times 4$)

Figure 6.8: Visualizing super-resolution limit at resolutions increased by $\times 2$ and $\times 4$ with our model. The first image in the first row benefits from the $\times 4$ improvements whereas the one in the second row (from the same photograph) is not further enhanced after $\times 2$. See the discussion in the text.

β selected according the distribution in Figure 6.3. More details about data generation can be in found in Section 6.a.2 of the appendix. To evaluate the CNNs trained on RGB images, we first pick the three raw frames corresponding to {-2.4,0,+2.4} EV in the burst and demosaick them with the approach of [265]. We also demosaick the frames before merging the HDR images with the bracketing technique. If the raw frames are not aligned, after demosaicking, we align the frames either with the phase correlation algorithm [219] on the MTB features [261] or with our Lucas-Kanade-based registration technique. For fairness with the CNNs, we compare our approach when there are only three frames in the bracket (the same as for the CNNs) and with the whole burst.
We present in Table 6.2 the results of our comparison. We evaluate the PSNR and the SSIM metrics on both the output of each algorithm and after evaluating the irradiance maps with μ-law, playing the role of a tone mapping algorithm [226]. However these typical image processing metrics may not be adapted to HDR imaging [266, 267] we thus also report the HDRVDP2 perceptual quality score of [268] (version 2.2.2). Note that [227] and [228] use RGB images for training, while our method leverages more information by directly processing raw frames. We also compare our method to the single-image methods of [234] and [235] running on the central frame of the burst.

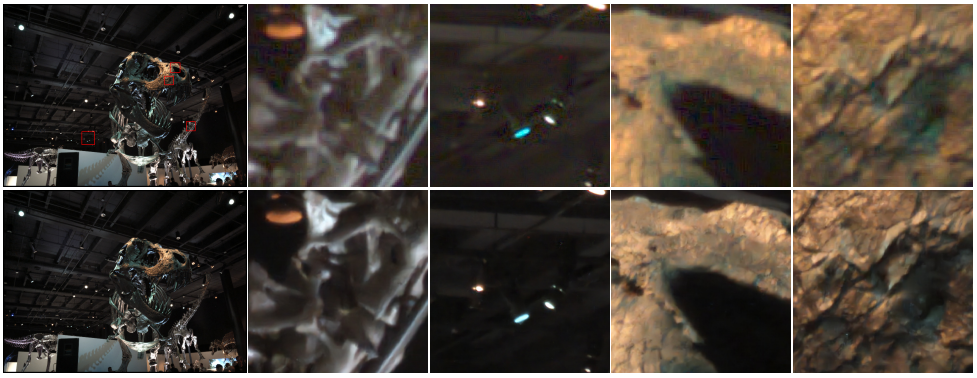


Figure 6.9: Comparison between the public-domain implementation [264] dubbed here HDR+Ipol of Google's HDR+ [220] (top) with our HDR/SR×4 method. **Left:** The images reconstructed by HDR+Ipol (top) and our method (bottom) from a burst of 8 same-exposure images acquired by a Nexus 5. Note that they are barely distinghuisable at this resolution. **Right:** Crops showing that our algorithm reveals finers details while effectively suppressing noise in dark areas.

Our algorithm using 11 frames achieves the best results as expected, with HDRVDP2 margins ranging from +4 to +9 over recent CNN-based methods and of +4 over the bracketing technique of [212] using 11 frames too. The gap with CNNs comes from our ability to restore the darker areas in raw photographs containing large read noise whereas these networks are trained on RGB images only. Figure 6.10 shows qualitative comparisons with the baselines in Table 6.2 for bursts of 21 images taking during day time and night time. Our method achieves the best visual results in both dark and saturated areas. Note that the CNN baselines considered here have been designed to handle 1 or 3 images only, which is not sufficient to achieve effective denoising through image fusion in challenging settings.
We also compare our approach with a public-domain implementation [264] of Google's HDR+ [220] that addresses HDR imaging by fusing images with the same exposure. In this setting, HDR essentially boils down to burst denoising, which is effectively handled by our approach. Figure 6.9 shows a qualitative comparison of HDR+ with our technique. We achieve better denoising, especially in the darkest areas, while also increasing spatial resolution.

6.5.4 Multi-Exposure Registration

We evaluate the performance of our registration module based on learnable features. We measure the geometric alignment error between the ground-truth motion and predicted one [74] computing the Euclidean distance between the aligned image corners with that of the ground-truth ones and is counted in number of pixels in the HR image.
We report the mean and the median over 266 validation bursts (containing 11 images per burst) synthesized with the same protocol as for generating the training data. We compare a typical multi-exposure registration scheme consisting in combining MTB features and phase correlation [219] (used for prealigning the images in our model), with the 3 iterations of the pyramid Lucas-Kanade (PLK) algorithm over plain pixels and deep features learnt in an end-to-end manner. The three methods are run on the mosaicked and possibly noisy images, prior to any ISP processing. We evaluate this panel over three scenarios: (i) HDR generation without SR from

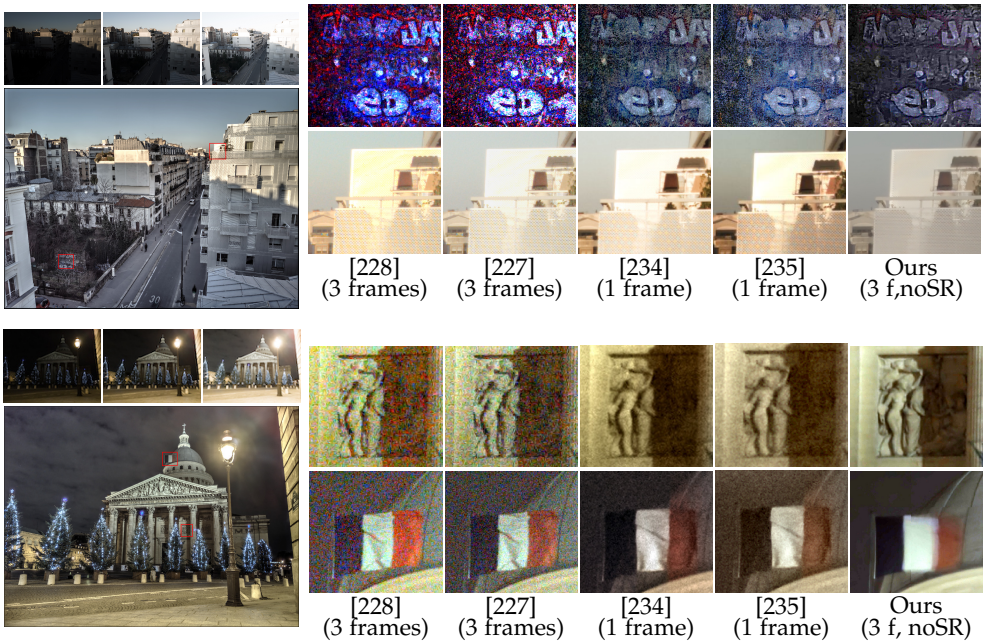


Figure 6.10: Comparison with CNN-based HDR methods processing one to three input frames. **Left:** A sequence of three input frames, followed by our result after tone mapping, for two scenes. **Right:** Small crops from the scenes obtained by various methods. To be fair, we compare them with a version of our model that does not perform super-resolution ($\times 1$ upscaling factor) and only processes a burst of 3 images, in the EV range [-2.4,0,2.4]. We observe that, in well-exposed regions, the reconstruction performances of the three methods are similar. Our method appears to be more robust to noise, but more sensitive to non-rigid motion as shown in the case of the flag.

noise-free raw bursts, (ii) HDR generation without SR from raw bursts with noise and (iii) joint HDR and SR with factor $\times 4$ from raw bursts with noise.
Table 6.3 shows that, in all cases, our approach achieves the best quantitative results, with a margin ranging from 0.5px for the (unrealistic) noise-free benchmark to more than 1px for the more challenging ones featuring noise. Interestingly, using more iterations does not always mean a better alignment. A plausible explanation is that our model is trained for using three iterations of the LK algorithm, and may be sub-optimal for more iterations.
We have also empirically observed that the errors in this table are always greater than that reported by [52] in their work for aligning frames with the same exposure. This gap is caused in practice by the darkest and brightest frames, much harder to align because of the noise in dark regions and large saturated areas.
Figure 6.11 compares the advantage of running the Lucas-Kanade algorithm with deep features and plain pixels in a real situation. Note the purple zipping artefacts caused by faulty alignment before image fusion in the left image obtained with the plain-pixel Lucas-Kanade algorithm. These artefacts vanish in the image on the right using deep features.

6.5.5 Discussion

Choice of the prior function. An important component of our approach is the image proximal operator G_ω. Figure 6.12 shows a qualitative comparison of a prior-

Table 6.2: Quantitative comparison of various algorithms for HDR imaging – we do not perform super-resolution in this experiment – on a synthetic dataset consisting of bracketed raw bursts simulated with our pipeline. Our method directly takes raw frames as an input. The other methods process RGB frames obtained here with VNG demosaicking. Our algorithm quantitatively outperforms the other HDR methods on this dataset, which is not surprising as it is trained leverage the information lost in the raw to rgb conversion.

Method	PSNR (dB)	μ-PSNR (dB)	SSIM (%)	μ-SSIM (%)	HDR-VDP2 (Q)
`K=1 frames`					
[235]	20.11	24.42	0.611	0.690	57.32
[234]	**22.14**	**25.85**	**0.641**	**0.702**	**62.94**
`K=3 frames`					
[212] + MTB	28.08	29.46	0.819	0.847	61.13
[212] + PLK	27.25	28.69	0.814	0.836	60.82
[227]	26.47	27.61	0.771	0.782	61.80
[228]	26.31	27.11	0.761	0.774	61.14
Ours	**33.75**	**34.39**	**0.942**	**0.943**	**63.24**
`K=11 frames`					
[212] + MTB	29.54	30.96	0.862	0.892	62.07
[212] + PLK	28.80	30.21	0.862	0.888	61.95
Ours	**37.83**	**39.22**	**0.964**	**0.971**	**65.44**

LK LK, with deep features

Figure 6.11: Qualitative comparison of the reconstructed image with a pyramid of Lucas-Kanade run on plain pixels or deep features. Note the zipping artefacts along the edges of the large white rectangle. Our learnable variant is faster and leads to more accurate results. The reader is invited to zoom in.

free version, solely aligning and merging the frames, using the image gradients soft-thresholding function derived from the classical TV-ℓ_1 prior, and our approach with a learnable module. The TV-based version is significantly sharper than that the one without prior. The parametric prior returns a better zoomed-in image, *e.g.*next to the head and the dress of the statue.

Robustness on real images. A key advantage of our approach is the accuracy of its registration module, as detailed on Table 6.3 and illustrated in Figure 6.11. We have remarked that this module is particularly efficient for aligning raw frames with the same exposure, as illustrated by Figure 6.7 in the context of SR with factor $\times 4$. Given a burst of raw photographs including moving objects with reasonable motion during exposure, *e.g.*the pedestrian in the figure, we can predict high-quality HR image well-aligned with the reference frame whereas the competitors may introduce ghosting or colored artefacts. Notwithstanding, we have also noted that non-rigid motions in the raw frame burst may lead to blur in the final predicted image. For instance, Figure 6.10 compares the restoration results from [227], [228]

Table 6.3: Quantitative comparison of registration methods on synthetic data with average and median geometric errors [74]. We compare MTB features combined with sub-pixelic phase correlation [219], the pyramid Lucas-Kanade (PLK) algorithm and our variant of PLK using deep features. The three algorithms are run on the mosaicked images. On each benchmark, we outperform both vanilla PLK and the MTB-based approach.

Model	Geom (avg.)	Geom (med.)
$\times 1$ - No noise - 11 raw frames		
MTB + phase correlation	2.93	2.61
3 PLK iterations	1.32	0.97
3 PLK iteration +deep features (ours)	<u>0.91</u>	**0.60**
5 PLK iterations	1.47	1.10
5 PLK iterations + deep features (ours)	**0.88**	<u>0.61</u>
$\times 1$ - Noise - 11 raw frames		
MTB + Phase correlation	3.58	2.99
3 PLK iterations	2.77	2.40
3 PLK iteration +deep features (ours)	**1.25**	**0.95**
5 PLK iterations	2.76	2.10
5 PLK iterations + deep features (ours)	<u>1.40</u>	<u>1.00</u>
$\times 4$ (aliasing) - Noise - 11 raw frames		
MTB + Phase correlation	5.93	4.67
3 PLK iterations	3.82	3.58
3 PLK iteration +deep features (ours)	**2.04**	**2.03**
5 PLK iterations	3.87	3.50
5 PLK iterations + deep features (ours)	<u>2.62</u>	<u>2.17</u>

and our model for a crop featuring a waving flag, *i.e.,* a non-rigid motion. We select $K = 3$ images with EV values of {-2.4,0,2.4} EV for the CNNs and for our model. The CNNs trained to remove ghosting artefacts accurately align the flag with the reference frame whereas our prediction is blurry in the red section of the flag. This may stem from the fact that multi-exposure image registration is a very challenging problem and that we have not such non-rigid motions in our training data. For a better deghosting, the injection of non-rigid motion in the training data, similarly to the dataset introduced in [226], is an interesting future research direction.

6.a Appendix

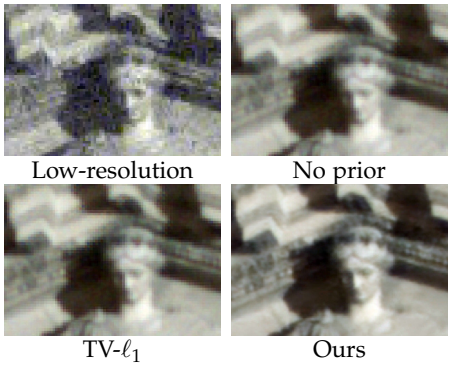


Figure 6.12: Visual comparison of the impact of the prior for joint HDR and $\times 4$ SR. We fuse $K = 20$ images in this example. We note for the three methods effectively suppress the noise present in the original LR frames. However, our learnable prior (here with 300k parameters) yields a higher quality image. The reader is invited to zoom in.

6.a.1 Ablation Studies

In this section, we provide additional experiments to better understand the impact of our method's components.

6.a.1.1 Learning the Image Prior

We quantitatively validate the choice of a parametric proximal function in Eq. (6.11) to address joint HDR and SR by a factor of 4. We compare the proposed CNN-based implementation with the the classical total-variation ℓ_1 (TV-ℓ_1), for instance used by [243], and a simple weighted least-squares problem, *i.e.*, without prior. We generate 266 bursts of $K = 9$ images with exposure values in $[-3, 3]$EV. Table 6.4 shows average PSNRs for methods embedding no prior (simple weighted least-squares), TV-ℓ_1 prior and three variants of the CNN G with parameter ω of sizes 30K, 300K and 3M. The variants with the parametric priors are trained according to the protocol described in the previous section. This table shows, as expected, a clear advantage of learnable penalty functions over handcrafted ones, with a margin of more than 4dB for the shallowest network and more than 7dB for the deepest one over the TV-ℓ_1 variant. Note that the prior-free version is only 0.4dB below its TV-ℓ_1-based counterpart, suggesting that machine learning is important to design an efficient prior function to address joint HDR and SR restoration.

Table 6.4: Quantitative comparison of the choice of the prior over the total performance of the method. Average PSNR on predicted linear HDR images jointly super-resolved by a factor of 4 for typical handcrafted image priors and variants of the proposed parametric one with several parameter sizes. The learnable ones achieve the best scores overall by an important margin. The more parameters yields the best PSNRs.

Prior	PSNR
No prior	26.15
TV-ℓ_1	26.51
Tiny (30k)	30.71
Small Prior (300k)	<u>32.56</u>
Large Prior (3M)	**34.18**

6.a.1.2 Alignement Sub-Components Evaluation

We quantify the impact of each components of the alignment module in Table 6.5 by measuring the mean PSNR of predicted HDR images with resolution enhanced by $\times 4$. We generate 266 raw bursts with 11 frames for each burst with the same protocol than the other experiments. We decompose it into three bricks: using bracketed images, the confidence function g and running the pyramid Lucas-Kanade (PLK) algorithm on deep features. Adding each component one-by-one gradually increases the mean PSNR, the maximum value being naturally reached when the three components are gathered. Note that the PLK algorithm run on deep features brings an improvement of about +2dB, which alone is a better contribution than the total of +1.3dB by combining bracketed images and the confidence function g. We also give an upper-bound to this performance by running a version of our model where we give the ground-truth motion to align the images. Such an oracle model achieves an average PSNR of 34.18dB compared to the 31.42dB of the best setting where the motion is estimated instead. It suggests that there is room to improvement but each

Table 6.5: Ablation study for the alignment module. We report mean PSNR on HDR images with $\times 4$ super-resolution. The first configuration (#1) uses a burst but no bracketing (constant exposure). The fourth configuration (#4) is the setting we use in practice, with bracketed exposures, the confidence function g and the pyramid Lucas-Kanade algorithm run on learnt features. Adding these three components one by one gradually improves the mean PSNR, showcasing the importance of each module. The fifth (#5) configuration is an upper bound where we use the ground-truth motion (and thus do not need LK with deep deatures).

Settings	#1	#2	#3	#4	#5
Bracketing		✓	✓	✓	✓
Confidence function g			✓	✓	✓
LK with deep features				✓	
Oracle motion					✓
PSNR	28.50	29.28	29.77	31.42	34.18

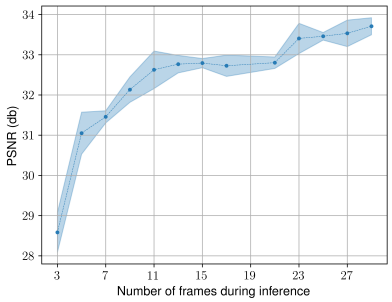


Figure 6.13: Average PSNR on predicted HDR images with spatial resolution increased by ($\times 4$), from a varying number of frames in the burst (from 3 to 30). Our approach benefits from any additional input frame, especially for less than $K = 11$ images. The average PSNR is evaluated from 3 seeds.

sub-components in the alignment actually helps to further narrow the gap with the oracle model.

6.a.1.3 Performance with the Number of Frames

We compare the performance of our approach with respect to the number of frames in the burst for joint SR and HDR on synthetic data. Figure 6.13 shows the mean PSNR taken over 3 seeds for bursts of length ranging from 3 to 30. Our model greatly benefits from additional frames for burst sizes smaller than 11; Starting from 3 images and a PSNR score of 28.6dB, we gain up to 4dB when accumulating 11 frames. Beyond this number, we gain an extra decibel by accumulating more than 20 frames. It is consistent with typical bracketing techniques, *e.g.*[211, 212], for which more images means better noise removal in the dark regions. Thanks to the learnt robust registration algorithm and prior, the performance of our approach hardly falls down when accumulating more and more images, unlike typical multi-image algorithms that may accumulate registration, *e.g.*as noted by [53].

Table 6.6: Comparison of inference speed for different models for burst super-resolution. We have benchmarked the inference speed of different models for processing a burst of 14 12Mpixel raw images (Pixel 4a) on a single Titan RTX GPU. We have used the official implementations released by the authors without any modification. We have not optimized inference speed (yet) using with standard tools such as mixed precision and/or model compression.

Model	# parameters	Runtime	Memory (200x200)	Memory (400x400)
Competitors ($\times$4)				
[54]	13,000k	40.0sec	3.5Gb	11.5Gb
[242]	26,000k	9.5min	3.5Gb	12Gb
Ours ($\times$4)				
Very Small	60k	13.4sec	1.2Gb	2.8Gb
Small	250k	20.0sec	1.2Gb	2.8Gb
Large	3,000k	38.2sec	1.3Gb	3.1Gb
Ours ($\times$2)				
Very Small	60k	4.7sec	800Mb	1.2Gb
Small	250k	6.2sec	820Mb	1.2Gb
Large	3,000k	10.7sec	860Mb	1.3Gb

6.a.1.4 Computational Speed

Our algorithm leverages optimization and machine learning techniques, which leads to a dramatically smaller number of parameters than state-of-the-art CNNs for tasks such as super-resolution. We evaluate the computational speed and memory consumption of our model embedding three variants of the learnable operator G with varying number of parameters. We compare our three versions of the proposed network with that of [54] and [242], the best performers for $\times$4 SR in Table 6.1. We run this five-way comparison within the same python environment, *e.g.*same version of Pytorch, and on the same GPU (Nvidia Titan RTX) for fairness. We show in Table 6.6 that our hybrid method exceeding the SR state of the art in the previous paragraphs, is also the lightest in the panel. The method of [242] has 26 million parameters and that of [54] about 13 million parameters whereas our deepest model has 3 million of them, *i.e.,* four time less than [54]. This gap in size of parameter is due to the building blocks in the competitors' architectures. Indeed, they heavily rely on memory-greedy attention modules, whereas our implementation of G is based on the fully convolutional U-net architecture of [186, 52]. This table also shows that, for resolution factor of $\times$4, our approach is much faster than the state of the art, while coping with them according to Table 6.1. In this table, our "Small" model is less than a decibel below [242]'s model but with an inference time forty times smaller on the same GPU. We are also four times faster than [54]. Likewise, our models require three to four times less GPU memory than our selected competitors, which is an important designing point for deploying such a technology in commercial software running on consumer-grade devices. We also report in Table 6.6 information about the $\times$2 case since in many situations pushing the resolution further brings little improvement, *e.g.*Figure 6.8 and the analysis of [53]. In this configuration, our model requires even less memory to process 400×400 tiles and may run on modest GPUs.

6.a.1.5 Limitations

Albeit our approach favorably addresses HDR, SR and joint HDR and SR against the state of the art, we have noted throughout our experiments a few limitations in certain cases that may degrade the performance of our model.

Lack of robustness to non-rigide motion for joint HDR and SR. We have observed that our model which performs joint HDR and super-resolution are less robust to non-rigid motion than our models performing only burst super-resolution. An example of artefacts that we typically get is shown in Figure 6.10 in the case of the moving flag.

Saturated areas. In the pictures shot with a smartphone, we have sometimes noticed color halos next to saturated areas (Figure 6.14). They may be caused by the fact that the corresponding very high level of contrast is hard to simulate in our synthetic data.

Hot pixels. The method is not trained to correct hot pixels, that may locally alter HDR imaging techniques. We assume that these pixels are corrected upstream in the camera pipeline, which is a classical assumption in the field.

6.a.1.6 Supervision with Various Loss Functions

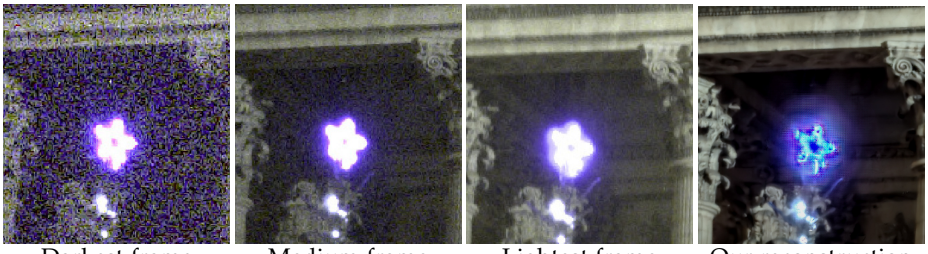
In this subsection, we propose an ablation study to assess the effectivness of different supervision loss instead of the basic L1 loss. We use the same loss function as the one described in [269], which gives a tone curve $\psi(x) = \log(x + \epsilon)$ which more strongly penalizes errors in dark regions. Results of the ablation study are presented in Table 6.7. Our experiment that the log loss gives better result in term of μ-psnr. The selection of the right supervision loss for the training of our model is an interesting direction for future research.

loss	psnr (dB)	μ-psnr (%)	μ-ssim (%)
L_1	36.52	38.08	0.9682
$\log(\epsilon = 10^{-1})$	37.04	37.86	**0.964**
$\log(\epsilon = 10^{-2})$	**37.71**	**38.70**	0.967
$\log\ (\epsilon = 10^{-3})$	13.74	8.07	0.199

Table 6.7: Super resolution factor x1, ablation study with different training loss.

6.a.1.7 Ablation HDR with No Motion

In addition to Table 6.2, we provide in Table 6.8 below a HDR fusion evaluation of the same methods, but on a variant of the synthetic test set where we have not simulated motion between frames. In this new table, our method still achieves the best HDR-VDP scores for brackets of both $K = 3$ and $K = 11$ images, but with a margin of only 1 point over our implementation of [212]. We can conclude that both approaches achieve similar results. However, when we compare these margins that of 3 points in Table 6.2 between our approach over [212], it suggests that our technique is more robust to alignment failures than the pure bracketing technique of [212].



Darkest frame | Medium frame | Lightest frame | Our reconstruction

Figure 6.14: Color halos in very bright areas and saturated regions. Because all input views are saturated in the neighborhood of the star, key info required to correctly reconstruct the image there is missing, leading to severe artefacts there.

Table 6.8: Quantitative comparison of various algorithms for HDR imaging – we do not perform super-resolution in this experiment – on a synthetic dataset consisting of bracketed raw bursts simulated with our pipeline. Our method directly takes raw frames as an input. The other methods process RGB frames obtained here with VNG demosaicking. Our algorithm quantitatively outperforms the other HDR methods on this dataset, which is not surprising as it is trained leverage the information lost in the raw to rgb conversion

Method	psnr (dB)	μ-psnr (dB)	ssim	μ-ssim (dB)	HDR-VDP2(Q)
`K=1 frame`					
Liu *et al.* [235]	19.98	24.25	0.608	0.687	56.51
Santos *et al.* [234]	22.05	25.80	0.635	0.699	61.69
`K=3 frames`					
Wu *et al.* [227]	26.42	27.51	0.765	0.774	61.04
Yan *et al.* [228]	26.22	27.01	0.752	0.768	60.37
Hassinof *et al.* [212]	30.55	31.26	0.874	0.878	67.77
Ours	**34.29**	**34.31**	**0.945**	**0.934**	**68.63**
`K=11 frames`					
Hassinof *et al.* [212]	33.80	33.43	0.917	0.927	68.86
Ours	**38.73**	**38.56**	**0.973**	**0.969**	**70.05**

6.a.2 Implementation Details

We include below details about our datasets and implementation for reproducibility purposes. See also Table 6.6 for the number of parameters used in different variants of our method.

Data Generation. Given a collection of sRGB images, we construct bursts of LDR low-resolution raw images and HDR/high-resolutions RGB targets. For the generation of realistic raw data from sRGB images, we follow the approach described in [54], using the author's publicly available code on the training split of the Zurich raw to RGB dataset [204]. The approach consists of applying the inverse RGB to raw pipeline introduced in [255]. For the training of our model, we generate bursts of 11 frames of size 256x256 with random motions. Displacements are randomly generated, applying random translations of ± 6 pixels and random rotations of $\pm 1^{\circ}$. Frames are downscaled with bilinear interpolation in order to simulate LR frames containing aliasing.

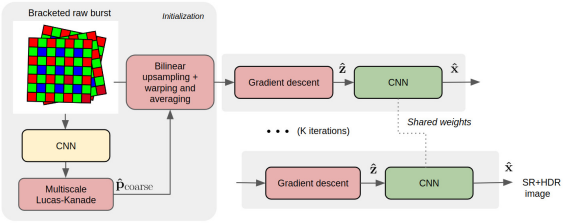


Figure 6.15: A diagramatic view of our model.

We then apply two different random gains in order to simulate frames with varying exposure. First, a random gain in the range [-5ev,5ev] is applied to the ground-truth image, in order to simulate 32-bit ground-trugh images with a large dynamic. We then apply a different gains in the range [-3ev, 3ev] for each image of the burst in order to simulate images with different exposure times. This results in synthetic bursts with different saturated areas and signal to noise ratios. Synthetic noise is added to the frames. The noise levels are sampled following the empirical model of Figure 6.3. Finally, color values are discarded according to the Bayer pattern.

Validation Split In order to perform further comparison and conduct the ablation study, we build a validation set by randomly extracting 266 images from the Zurich raw to RGB dataset [204].

Model. We summarize in Figure 6.15 our proposed pipeline. In all our experiments, we unroll 3 iterations of the HQS algorithm.

Deep Prior. We give more details about the architecture of the deep prior used in our experiments. For all our experiments, we use a smaller variant of the ResUNet architecture introduced in [186] for single-image super resolution. This architecture involves four scales, each of which has an identity skip connection between downscaling and upscaling operations. Downscaling operations are implemented using 2x2 strided convolution while upscaling are implemented with pixel-shuffling. Each residual block is made of two 3x3 convolution layers and ReLU activation combined with an identity skip. For each scale we apply a cascade of 2 residual blocks. The network has respectively 32,64,128,128 channels for each convolution per scale.

Model variants. We also run experiments with an even smaller version of the network with 32 features per channel (dubbed small) and 16 features per channel (dubbed tiny).

Training procedure We minimize Eq. 6.13 using Adam optimizer with learning rate set to $10^{-}5$ for 400k iterations. We decrease the learning rate by a factor 2 every 100k iterations. The weights of the CNNs are randomly initialized with the default setting of the PyTorch library. Our approach is implemented in Pytorch and takes approximately 2 days to train on a Nvidia Titan RTX GPU.

Chapter 7

Dense Image Registration and 3D Reconstruction from Bursts

Chapter abstract: This paper presents a novel approach to the fine alignment of image bursts captured by a handheld camera, with applications to image denoising, super-resolution, and 3D scene capture. Unlike conventional methods, it does not require discrete correspondences, nor does it rely on 2D (e.g., piecewise-affine) transformations, as it directly optimizes the depth and surface orientation at each pixel for a reference image and the extrinsic parameters of all other cameras relative to it. Rough (16×16) and noisy initial depth estimates, as provided nowadays by most high-end smartphone cameras, can be used for additional robustness if necessary. Extensive experiments with synthetic images demonstrate that the proposed method outperforms the state of the art by a significant margin. Preliminary experiments with real image bursts, including denoising, super-resolution, and 3D reconstructions are also presented.

B. Lecouat*, Y. Dubois de Mont-Marin*, T. Bodrito*, J. Mairal, J. Ponce. Dense Image Registration, Camera Pose and Depth Estimation from Bursts [5]. This paper is under review.

7.1 Introduction

We address the problem of *dense* registration, pose estimation, and 3D reconstruction from image bursts captured by a handheld camera, with small motions. Our method estimates optical flows, camera poses, and depth maps which can be used for multiple applications, including denoising, super-resolution, hdr imaging and 3D reconstructions.
In burst photography, a camera captures a short sequence of images (e.g. 10 frames), in rapid succession (e.g. one second), possibly with different camera settings. Exploiting the fact that, for handheld cameras, the images are taken from slightly different viewpoints, bursts can be leveraged for image enhancement as demonstrated by recent approaches to high dynamic range imaging [81] with exposure bracketed, night photography[21], deblurring [270], or super-resolution [80]. Bursts can also be exploited to recover the 3D structure of the captured scenes. [271, 272,

273], shows that it is possible to exploit the parallax in the scene to infer its 3D structure.
A broad family of methods can perform image registration and 3D reconstruction from image bursts, ranging from dense image alignment to structure from motion. However, these methods generally do not fully leverage the specificity of burst photography to their advantage and may therefore be suboptimal for the task. For example, accurate registration is challenging for real bursts captured in the wild, and typical methods [81, 80] align images pairwise and independently with a reference frame. Nevertheless, the quality of the alignment can severely impact the quality of the enhanced image by creating ghosting or zipping artifacts [3] when the registration algorithm does not align frames with sufficient precision. 3D reconstruction from bursts is also challenging due to small baselines [271] (i.e., the displacement of the camera between consecutive frames), and therefore requires very fine alignment. However, small motions also offer some opportunities: it makes matching easier and allows the use of a depth map as a compact and convenient representation of the 3D scene, as explained in [273]. This paper introduces a novel optimization-based algorithm specifically tailored for aligning -and inferring 3D structure- from bursts.
We propose a novel approach to dense burst registration that leverages the multi-image setting by directly modeling the scene structure and the camera poses. To do so, a good choice of parameterization for aligning frames is vital. Homographies are widely used since they are both simple and effective models for planar or distant scenes or when the motion consists mainly of a pure rotation about the optical center, with little parallax. However, they are limited to scenes with little 3D relief [27]. Complex motions can, of course, be approximated by piecewise-simple parametric transformations defined on small tiles [273],[80] or optical flow [274], mitigating the parallax issue. But the price to pay is a large number of parameters to fit, which may impact robustness. Our optimization-based method estimates the camera's pose and 3D scene structure by jointly minimizing the photometric reprojection errors in a reference frame. The pose parameters are fitted individually for each frame, while structure parameters are fitted locally in the images but shared among all the views. The flow between frames can then be computed by reprojecting points in other views. This modeling requires a much smaller number of parameters than block-parametric or optical flow while providing sufficient expressivity level to represent static scenes accurately.
As no existing multi-view stereo dataset we are aware of, exactly covers our typical use cases, we validate our approach with synthetic bursts built with rendering software (the proposed dataset will be made publicly available). Our model turns out to be very versatile in the context of image bursts with small motion; it gives state-of-the-art performance compared with compelling methods specifically designed for optical flow [275], pose estimation [273, 276, 277], and depth estimation [272, 276]. Our results also suggest that, in the small movement case, a dense formulation is very beneficial because many concurrent methods are based -at least partially- on sparse key points [273, 276, 277]. Finally, to validate our approach with real-world data, we demonstrate applications with real bursts captured with a Pixel 6 pro smartphone to night photography denoising, super-resolution, and dense 3D reconstructions.

Contributions. In the context of burst imagery, we propose a versatile multi-frame registration method that excels at various tasks. (1) our algorithm gives state-of-the-art dense alignment metrics on synthetic data: we outperform state-of-the-art

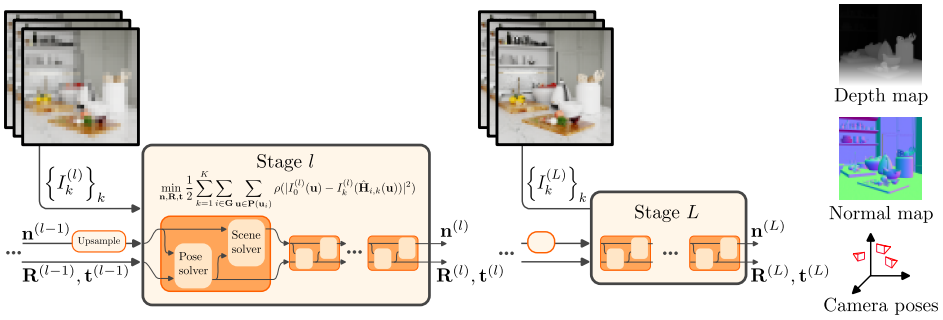


Figure 7.1: Proposed algorithm: Our optimization-based method estimates the camera's pose and 3D scene structure by jointly minimizing the photometric reprojection errors in a reference frame, in a coarse-to-fine fashion.

deep-learning-based methods. The flows estimated on real data can be used for tasks requiring fine alignments, such as burst super-resolution. (2) Our algorithm also gives state-of-the-art metrics for camera pose estimation. (3) Our method is also competitive for depth estimation with small motions. We manage to capture the structure of 3D scenes by only exploiting bursts with small baselines. (4) Finally, we also present a novel fixed-point algorithm to infer depth maps in new camera positions. We use this algorithm to estimate the reverse optical flows and warp reference views onto other views, which is required by downstream tasks such as super-resolution or low-light photography.

7.2 Related work

Burst photography. Burst photography is a technique that involves capturing a sequence of images to improve the overall quality of a photograph by reducing noise, enhancing details, and improving dynamic range. Algorithms are generally built around a registration algorithm to align the frames. Recent research has led to exploring machine-learning techniques for burst photography based solely on deep-learning models that do not necessarily rely on a registration preprocessing step. However, that class of algorithms suffers from several limitations, which make their integration on embedded devices challenging due to computational cost and lack of robustness, as pointed out in [57].

Multi frame image registration. Related work has given relatively little attention to image registration with multiple frames. A straightforward approach often used in practice is simply aligning frames with a reference frame as in [80, 81]. However, it is possible to leverage the multi-view setting to improve the registration quality as in [278, 279, 280], which proposed different optimization-based methods for multi-view image registration, but codes are not publically available. Registration of frames with heterogeneous content is also crucial for burst photography, for example, in focus bracketing or dynamic range imaging, where the saturation zones and signal-to-noise vary among different areas. [78] proposed new algorithms in this setting that can produce visually pleasant fused images.

Depth reconstruction from small motions. Popular 3D reconstruction methods rely on geometric methods such as structure from motion (SfM) [277]. These methods utilize geometric constraints and rely on key point correspondences to recon-

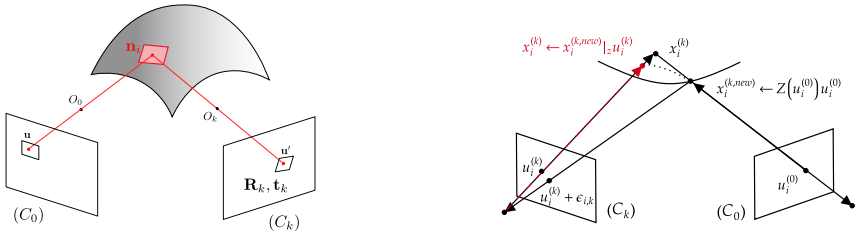


Figure 7.2: **Left**: Image formation model with a patch and its local tomography image. **Right**: fixed-point algorithm to evaluate the reverse flow in a camera (C_k) from the one in (C_0) given by our method.

struct a sparse 3D scene. Dense 3D representations can then be estimated based on the sparse reconstruction, as Colmap software [277] does. Bundle adjustment is a crucial step for refining a scene's estimated 3D structure and camera poses. It operates on 2D image key points, corresponding 3D points, and camera calibration parameters. The goal is to minimize the reprojection error by iteratively optimizing the 3D structure and camera poses, resulting in a more accurate reconstruction of the scene. 3D reconstruction from Methods have specifically been tailored for small motion scenarios. When dealing with small movements, achieving accurate 3D reconstruction relies on employing motion estimation techniques that are highly precise, and depth map has emerged as a popular structure representation in this setting. Im et al. [281] adapted SfM to small motion. [273] efficient method using feature tracking for pairwise key points and bundle adjustment algorithms adapted to small motions. They also estimate the intrinsic parameters of the camera as well as distortion parameters to fit the data better. [271] uses a neural depth model and IMU to initialize the camera poses and lidar measurements to initialize the depth map. [271] makes it non-necessary to initialize with a depth map model even though it may still give better results with the initialization.

7.3 Method

7.3.1 Image Formation Model

We consider a rigid scene described by a piecewise surface and $K+1$ internally calibrated pinhole cameras $(C_i)_{i=0..K}$. A point $\mathbf{u}$ in the (C_0) camera plane is the projection of a point $\mathbf{x}$ of the scene surface. We denote by π, the affine plane tangent to the scene in $\mathbf{x}$ parameterized, by its (non-unit) normal n such that $\pi = \{\mathbf{y} \in \mathbb{R}^3, \mathbf{n}^\top \mathbf{y} = 1\}$. A patch around $\mathbf{u}$ is the projection of a patch around $\mathbf{x}$ in π, and its image in the camera (C_k) is given by a homography uniquely defined by the plane π and the extrinsic parameters $\mathbf{R}_k, \mathbf{t}_k$ of the other camera (Fig. 7.2 left).

7.3.2 Minimization Problem

The parameters of our model are the extrinsic parameters $(\mathbf{R}_k, \mathbf{t}_k)$ for $k = 1..N$ of the camera (C_k) relative to the reference camera (C_0) and the plane normal $\mathbf{n}_i$ associated to the point $\mathbf{u}_i$ for $i \in \mathbf{G}$ a regular grid in the reference camera plane. To estimate the flow induced by the scene and the camera poses, we minimize the photometric error between the reference image on each patch and the other images on patches obtained by the local homography. The previous model is not valid at occluding boundaries. We handle this phenomenon using a robust loss ρ as in [282].

Our minimization problem is:

$$\min_{\mathbf{n},\mathbf{R},\mathbf{t}} \frac{1}{2} \sum_{k=1}^{K} \sum_{i \in \mathbf{G}} \sum_{\mathbf{u} \in \mathbf{P}(\mathbf{u}_i)} \rho(|I_0(\mathbf{u}) - I_k(\hat{H}_{i,k}(\mathbf{u})|^2), \tag{7.1}$$

where $\mathbf{P}(\mathbf{u}_i)$ denotes the set of pixels constituting the patch centered in $\mathbf{u}_i$ and $\hat{H}_{i,k}$ is the homography for the patch i and the view k defined in homogeneous fashion by the 3×3 matrix:

$$H_{i,k} = \mathbf{R}_k + \mathbf{t}_k \mathbf{n}_i^\top, \tag{7.2}$$

as, in [283, 272], which rightly notes that $\mathbf{n}_i$ is not a homogeneous vector defined up to scale and has three full degrees of freedom. So does $\mathbf{t}_k$. On the other hand, and as usual, there is a global scale ambiguity since replacing every $\mathbf{t}_k$ by $\lambda \mathbf{t}_k$ and every $\mathbf{n}_i$ by $\lambda^{-1}\mathbf{n}_i$ does not change the homography and thus the loss. Note that if we take patches of size 1, with mean $\mathbf{P}(\mathbf{u}_i) = \{\mathbf{u}_i\}$, we recover the formula of a pixel-wise loss as used, for example, in [273]. Such a loss only captures the depth corresponding to the points on the $\mathbf{G}$ grid points. On the contrary, estimating the π plane allows a better expressiveness of the model and better optimization stability, as shown in the *ablation study* presented in Appendix 7.a.1.1.

7.3.3 Numerical Procedure

The scale ambiguity suggests that we should fix one of the variables of one of the $\mathbf{t}_k$ or $\mathbf{n}_i$ at an arbitrary non-zero value. However, on the one hand, there is no guarantee that one of the components of one of the $\mathbf{t}_k$ is non-zero, and on the other hand, fixing a value of one component of one of the $\mathbf{n}_i$ will not constrain well the problem since it could be treated as an outlier in the robust loss. This observation leads us to a block coordinate descent with one block constituted of the extrinsic parameters $(\mathbf{R}_k, \mathbf{t}_k)_{k=1..N}$ and one block with the structure of the scene $(\mathbf{n}_i)_{i \in G}$. Indeed, each optimization sub-problem has no more the problem of a scale value which can be arbitrary. The scaling factor then depends on the initialization of the variables. In the case of small movements, it is reasonable to initialize the translation values to 0. Therefore, it is necessary to have a good initialization of the plane parameters, which can be computed from a coarse depth map when initializing the planes as fronto parallel. We use a proximal Gauss-Newton (PGN) for steps on the block $(\mathbf{R}_k, \mathbf{t}_k)$ for $k = 1..N$; and a gradient descent (GD) with Adam momentum and adaptative learning rate for steps on $(\mathbf{n}_i)_{i \in G}$.

7.3.4 Pose Estimation

To estimate the pose, we take advantage of the fact that (7.1) is a robust non-linear least squares problem in $\mathbf{R}_k, \mathbf{t}_k$ to use an algorithm of type Gauss-Newton. We parametrize the poses by their twist in exponential coordinates on the group $SE(3)$: $[\mathbf{R}_k, \mathbf{t}_k] = \mathrm{Exp}(\xi_k)$ where we have the twist $\xi_k = [\mathbf{!}_k, \mathbf{v}_k]$ and Exp is the exponential of the group $SE(3)$. The parameterization in exponential coordinates is relevant in the case of small motion. [273] even uses a linear parametrization $[\mathbf{R}_k, \mathbf{t}_k] = [\mathrm{I}_3 + [\mathbf{!}_k]_\times, \mathbf{v}_k]$ which corresponds to a Taylor series at order 1 of the group exponential. However, as our least squares problem is non-linear, keeping a more expressive non-linear parameterization of the poses for which jacobians in close form exist is relevant [284] (see Appendix 7.a.1.1 for a comparison of the two parameterizations). Noting $\mathbf{r}_k = [I_0(\mathbf{u}) - I_k(\hat{H}_{i,k}(\mathbf{u})]$ the residual vector of the photometric error for every point of every pixel patch for a view in camera (C_k) and abusing

notation we note $r_{k,j}$ its coordinate terms. We take advantage of the fact that $\mathbf{r}_k$ depends only on ξ_k and by following the procedure of the proximal algorithms [200], we use Gauss-Newton algorithm to solve K independent series of minimization problems of the form:

$$\min_{\xi_k} = \frac{1}{2}\sum_j \rho(r_{k,j}^2) + \beta\|\xi_k - \xi_k^*\|_2^2, \tag{7.3}$$

where β is the proximal factor, and ξ_k^* is the last twist of the previous problem in the series. Between each problem, we adjust the proximal factor $\beta \leftarrow \beta/2$; see [200] for more details on the proximal algorithms. For Gauss-Newton steps, as in [282], we use $\mathcal{H}_k = \mathrm{J}_{\mathbf{r}_k}^\top \operatorname{diag}(\rho'(r_{k,j}^2)_j)\,\mathrm{J}_{\mathbf{r}_k} + \beta\,\mathrm{I}_6$ to approximate the loss hessian and $\mathcal{G}_k = -\mathrm{J}_{\mathbf{r}_k}^\top \operatorname{diag}(\rho'(r_{k,j}^2)_j)\mathbf{r}_k + \beta(\xi_k^* - \xi_k)$ as the opposite of the gradient where $\mathrm{J}_{\mathbf{r}_k}$ is the Jacobian of the residuals. We take the step $\xi_k \leftarrow \xi_k + s_k\mathbf{d}_k$ where $\mathbf{d}_k$ is a solution of the linear problem $\mathcal{H}_k\mathbf{x} = \mathcal{G}_k$ and s_k is a step size following a backtracking line search using Armijo's stopping criterion.

7.3.5 Scene Estimation

For the estimation of the scene structure, we perform two reparametrizations. First, we estimate the $\mathbf{n}_i$ on a $\mathbf{G}_v$ grid having a resolution twice as low as $\mathbf{G}$, and we recover the $\mathbf{n}_i$ on the rest of the grid with a bilinear interpolation. This choice acts as an implicit spatial regularization as each variable will intervene in estimating several patch planes. Second, noting $\mathbf{n}_i = [a_i, b_i, c_i]$, we optimize instead the variables $\gamma_i = c_i + [a_i, b_i]^\top \mathbf{u}_i, \alpha_i = a_i/\gamma_i, \beta_i = b_i/\gamma_i$. This change of variable comes from centering and normalizing the parametrizations so that the optimized variables are coherent in any point of the grid and, in turn, that the spatial regularization behaves correctly. We use *autograd* implemented in pytorch [146] to compute the gradient of the loss (7.1) with respect to the variables $[\alpha_j, \beta_j, \gamma_j]_{j\in\mathbf{G}_v}$. We use the optimization method with momentum and adaptive learning rate Adam [131] for the gradient steps. To stabilize the gradient descent on the scene structure, we add penalties. First, a total variation on the variable optimization grid as in [285] and second, an l1 penalization centered in 1 on the determinant of the flow induced by the scene structure. The purpose of this penalization is to favor the rigidity of the flow when the gradient of the loss is not informative enough.

7.3.6 Coarse to Fine Approach

Since we use a photometric loss, the gradients and jacobians obtained depend on the spatial gradient of the I_k images and therefore contain sub-pixel information. When the alignment error is over-pixelated, this can cause a convergence problem. To overcome this problem, we use a *coarse-to-fine* approach as in [286, 282]. For this we solve, a succession of L problems images $I_0^{(l)}$, $I_k^{(l)}$ and grids $\mathbf{G}^{(l)}$, $\mathbf{G}_v^{(l)}$ of a lower resolution of a factor 2^{L-l}. Note that we adopt this coarse-to-fine approach only for the steps of the scene estimation block. We still use high-resolution images and grids for the pose block steps. We use a bilinear sampling to pass from coarse to fine estimates of the structure when we go from one block to the other or from one resolution stage to the next. As we only apply the coarse-to-fine approach to the scene structure block, the optimized losses are different for the two coordinate blocks until the last stage of the coarse-to-fine approach. However, the first stages constitute a strategy to obtain a good initial guess for the last stage optimization

problem, corresponding to the original problem (7.1) and is a well-posed block coordinate descent where the two blocks use the same loss. We illustrate the overall method in Fig. 7.1.

7.3.7 Usage in Downstream Tasks

We need a pixel-wise flow to use our algorithm's output in a downstream task. We choose to take, for each pixel, the local homography evaluated in the center of the pixel patch. The previous flow allows warping the image I_k in the camera (C_0) using a backward warp required by tasks such as denoising. For other tasks like superresolution, we need to be able to do the inverse warp, i.e., warp the image I_0 in each camera (C_k). Doing so with the previous flow and using a forward warp is unstable and may create artifacts in the reconstructed image. Instead, we introduce a fixed point algorithm that estimates the normal map $\mathbf{n}_i$ in the (C_k) cameras from the one in (C_0). With this normal map and noticing that the extrinsic matrix of (C_0) relative to (C_k) is $[\mathbf{R}_k^\top, -\mathbf{R}_k^\top \mathbf{t}_k]$, we can compute the inverse flow and in turn the inverse warp using a backward warp. The algorithm is based on the fact that when we compose the flow and the inverse flow on a regular point grid, we must obtain a regular point grid. We illustrated in Fig. 7.2 right.

7.4 Experiments

7.4.1 Synthetic Burst Simulation.

Method	EPE ↓	RMSE ↓	NPE1 ↑	NPE2 ↑	NPE3 ↑
Blender1 (small motion)					
DfUSMC [273] *	1.4466	2.1723	0.5315	0.7488	0.8477
RCVD [276]*	5.9556	7.678	0.0957	0.2534	0.3763
Saop [272] *	9.7262	12.5891	0.101	0.2457	0.3402
Homography	2.8102	4.7107	0.4998	0.6627	0.7405
Farnebäck [274]	2.6852	4.8478	0.5299	0.6612	0.7278
RAFT [275]	0.9013	1.5396	0.7348	0.9069	0.9443
Ours	**0.6013**	**1.2047**	**0.8392**	**0.9263**	**0.9526**
Blender2 (tiny motion)					
DfUSMC [273] *	4.1356	4.5676	0.2267	0.4278	0.5497
RCVD [276]*	0.4007	0.5316	0.8676	0.9825	0.9959
Saop [272] *	2.0430	2.3563	0.5684	0.7645	0.8424
Homography	0.3008	0.3772	0.9003	0.9921	0.9982
Farnebäck [274]	2.0892	3.8154	0.6480	0.7296	0.7642
RAFT [275]	0.4857	0.5765	0.8664	0.9857	0.9963
Ours	**0.2713**	**0.3287**	**0.9348**	**0.9954**	**0.9999**

Table 7.1: Optical flow errors. The optical flow was predicted from the extrinsic camera parameters and depth maps for the models marked with an asterisk.

We required photorealistic bursts containing ground truth depth and camera poses for evaluating our approach and concurrent methods, but existing public multi-view stereo datasets we are aware of lack the needed characteristics due to non-static scenes or excessively large frame baselines that do not align with our specific use cases. We generate two photorealistic synthetic datasets using CYCLES, the path tracing engine of Blender [287]. We used a set of twelve publicly available

Dataset	Scenes	Frames	Std baselines (m)	Std rotations (deg)	Max depth (m)	Min depth (m)	Mean depth (m)
Blender1	15	20	0.116	0.20	0.316	11.234	3.73
Blender2	10	20	0.010	0.29	1.92	19.453	6.21

Table 7.2: Main characteristics of the two proposed datasets.

indoor scenes made by 3D artists, with detailed and varied scene compositions. Ten scenes come from the Evermotion Archinteriors Vol.43 Collection [288], and two scenes were freely available [289]. Each burst of the dataset consists of 20 frames, with a resolution of 512x512 pixels, a focal length of 50mm, and a sensor size of 35mm. We skipped the post-processing denoising step at the end of the rendering to avoid temporal flickering artifacts. Still, we mitigated the ray tracing noise by using many samples (4096). The camera trajectories and orientations were crafted as follows: a few keyframes were positioned manually to outline the global path, and the other keyframes were obtained with Bezier interpolation. These two datasets are made of small and tiny motions. Their characteristics are summarized in Table 7.2.

7.4.2 Evaluation on Synthetic Data

We follow the standard practice to evaluate pose, depth, and flow as described in [276, 170]. For all the methods, as depth estimation and pose are known up to an unknown scale, we align the predicted depth and the ground truths using median scaling. For pose evaluation, we compute the scale factor as $s = \arg\min_s \|T - s\hat{T}\|^2$, where $T = \left[\mathbf{t}_0, \cdots, \mathbf{t}_N\right]$. In addition, we use the canonic left-invariant distance in $SE(3)$ that combines rotational and translation parts in one quantity; see [290, 291] for details. We report the distance between the ground truth pose and the estimated pose. It reads $d([R,t],[R',t'])^2 = \|t'-t\|_2^2 + \lambda\|\log(R^\top R')\|_2^2$. For λ, we use the median value of the ground truth depth. $\|\log(R^\top R')\|_2$ is the canonic metric on the set of rotation $SO(3)$ and is also reported independantly. Unlike other methods in the literature [276], we chose not to present relative pose error (RPE) as a good RPE may not correlate with good alignment metrics and rely on a time coherent burst. To evaluate the ATE, we did not align the estimated poses with the ground truth poses with rigid transformation, as is common in the SLAM community. Indeed, our loss 7.1 and, more generally, the flow is not invariant by a solid transformation of the poses. As the final goal of our method is alignment, performance evaluation up to a rigid transformation would not be informative.
To assess the effectiveness of our approach, it is necessary, to begin with a rough estimation of the true depth map. In our synthetic experiment, we initialize the depth map with using a very coarse version (16×16) of the ground-truth depth map. We compare our pose and depth estimation method with the method introduced in [276], [272], and [273], using the codes publicly available online. We also initialize the method from [272] with the same low-resolution depth map for a fair comparison. For optical flows, we compare our method on the synthetic datasets with a state-of-the-art deep optical flow method [275] (we register all frames with respect to the reference), and also using a standard homography and the Farneback optical flow [274]. We also compute the optical flow errors for other concurrent methods with estimated pose and depth maps using the pixel-wise projection model. We also compare our method with a monocular depth estimation model [292] for depth estimation, however, monocular methods can only estimate the depth up to an affine

transformation [292]. Thus, these methods are evaluated after an affine registration. In our case, the goal is the alignment of frames, and the optical flow is not invariant by an affine reparametrization, so it does not make full sense to use this registration. The performances in the table 7.4 are obtained after only a rescaling. For the comparison as a pure depth map, we still compared our method and the others to Midas state of the art of monocular depth estimation—appendix 7.a.1.2 sum up the results. By leveraging the multi-image setting and the static scene hypothesis, our method consistently provides better results than [275] in terms of flow accuracy. We also outperform other methods for both pose and depth estimation. We note that [272] gives poorer results than other methods in this setting. We expect it to be because this method is designed to handle very small motions which may be smaller than the ones simulated in our experiments.

Method	Left l2 (m)↓	ATE (m) ↓	Geom (m) ↓	Biinvrot l2 (deg) ↓	Left l2 (m)↓	ATE (m) ↓	Geom (m) ↓	Biinvrot l2 (deg) ↓
Dataset	Blender1 (small motion)				Blender2 (tiny motion)			
Colmap [277]			✗				✗	
DfUSMC[273]	0.0117	0.0108	0.0094	0.1948	0.0046	0.0026	0.0024	0.1918
Saop [272]	0.0274	0.0229	0.0204	0.6369	0.0078	0.0043	0.0040	0.2678
RCVD [276]	0.0168	0.0162	0.0140	0.2158	0.0168	0.0162	0.0140	0.2158
Ours	**0.0032**	**0.0028**	**0.0019**	**0.0727**	**0.0020**	**0.0019**	**0.0018**	**0.0303**

Table 7.3: Pose errors metrics on the two proposed synthetic bursts datasets.

Method	Abs rel ↓	Sqr rel ↓	RMSE↓	Delta 1↑	Delta 2 ↑	Delta 3 ↑
Blender1 (small motion)						
Colmap [277]			✗			
DfUSMC[273]	0.2107	0.4864	**0.9683**	0.7723	0.8877	0.9409
Saop [272]	0.5818	1.8768	1.7900	0.3958	0.6009	0.7198
RCVD [276]	0.3111	0.5382	1.2368	0.5294	0.814	0.9524
Ours	**0.1544**	**0.2229**	0.9258	**0.7881**	**0.9544**	**0.9911**
Blender2 (tiny motion)						
Colmap [277]			✗			
DfUSMC[273]	0.3093	0.9543	2.0499	0.5722	0.7785	0.9187
Saop [272]	0.2936	0.8326	2.002	0.5794	0.7976	0.9263
RCVD [276]	**0.1898**	**0.3492**	**1.3745**	**0.6726**	**0.8816**	0.9693
Ours	0.2496	0.6107	1.8778	0.5834	0.8754	**0.9818**

Table 7.4: Depth errors metrics on the two proposed synthetic bursts datasets.

7.4.3 3D Reconstructions Quality on Synthetic and Real Bursts

We illustrate the high quality of depth reconstruction that our method can achieve on real scenes. To perform 3D reconstruction we feed or solver with RAW image bursts shot with a Pixel 6 pro smartphone. The raw images were demosaicked with simple bilinear filtering and then processed by our model, following the same procedure as described in [272]). We also feed our model an initial low-resolution depth map. We present in comparisons the depth map computed with a monocular method [292], RCVD [276], Saop [272] and DfUSMC [273]. We also illustrate depth map reconstructions on synthetic data from our dataset. Our depth map can have a noisy aspect on a texture-less structure. This is a normal feature as our optimization is not well conditioned on uniform surfaces as small variations in inferred depth will have no effect on the reprojection photometric loss. This noisy effect can be

mitigated by choosing larger spatial regularization coefficients. But this trades with lower performance in terms of flow and pose metrics on synthetic data. We observed that no spatial regularisation plan parameters give the best results for image alignment and pose estimation. Reconstructed depth maps are displayed in Figure 7.3 and Figure 7.4.

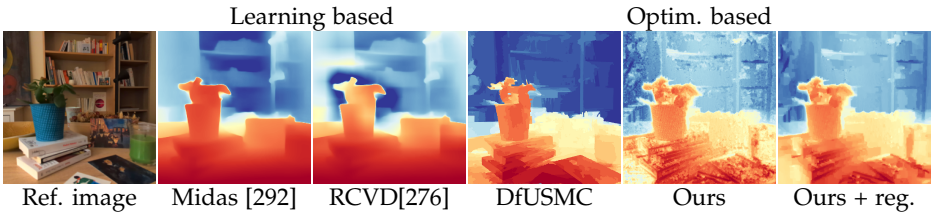


Figure 7.3: Depth estimation from real bursts. We compare the learning-based method and optimization. We present ours result w/o regularisation (*Ours*) and with determinant penalization 7.3 (*Ours + reg*).

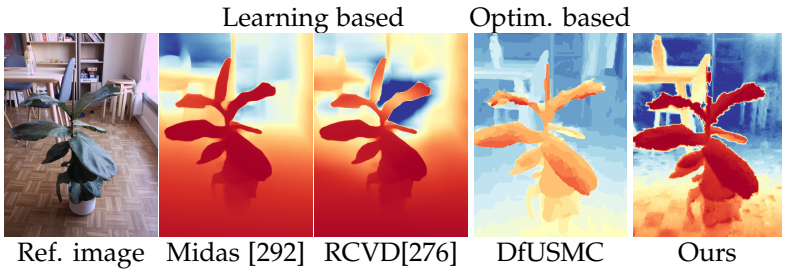


Figure 7.4: Another example of depth estimation from real bursts. Our method recovers more details for the depth than learning methods but also from some artifacts on the floor that look like a chess board.

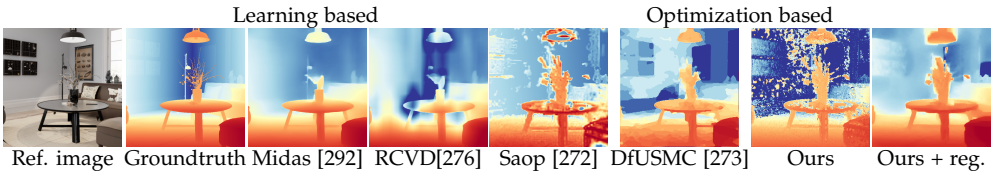


Figure 7.5: Depth estimation from synthetic bursts. It is one of the scenes generated with Blender used in the dataset *Blender 2*.

7.4.4 Low-Light Photography on Real Bursts

We demonstrate the robustness and flexibility of our alignment method in the case of low-light photography. This setup is demanding because low signal-to-noise ratio frames must be aligned. We shot night burst with low light choosing a short exposure time and high ISO to mitigate motion blur, with a Pixel 6 pro smartphone. We then align the frames with our method and with concurrent alignment algorithms. We use a simple homography and a dense optical flow using the Farneback algorithm [274]. We use the implementation from opencv [293]. Then, to perform burst denoising and increase SNR, we average the aligned frame. We do not focus on the fusion of the frames as we aim to highlight the registration quality offered

with our method. A better fusion algorithm may be chosen to alleviate artifacts and improve the overall image's quality; see, for example, the works [21, 81] as a reference. We initialized our method with a low-resolution depth map from the smartphone. We visually compare our results in Figure 7.6. We observe that due to the nonplanar nature of the scene, the homography fails to efficiently align objects in the foreground such as the plant, and objects in the background (e.g. the books) as the denoised image has a blurry aspect. On the other hand, the optical flow model is more flexible and manages to align objects both in the background and foreground. However, it may lack robustness, and some part of the image is not well aligned, such as the white book in the background or the white cup in the foreground.

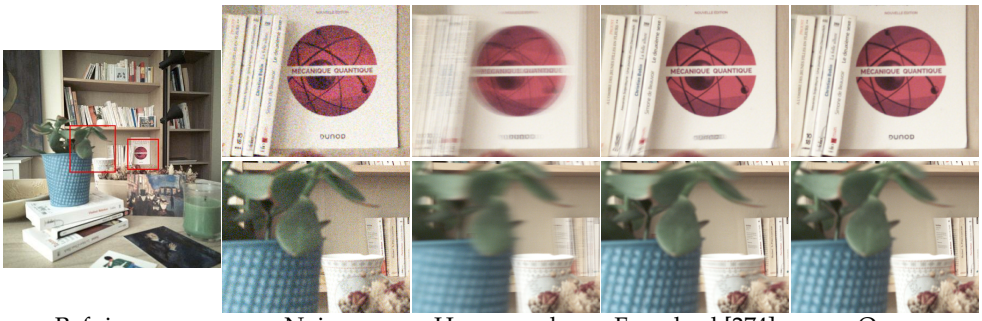


Figure 7.6: Burst denoising for night photography on real bursts exploiting alignment of various algorithms. Left: Full image with bounding boxes highlighting the region of interest. Top line: background region is misaligned for concurrent methods. Bottom line: cup is misaligned for other methods. Homography misaligned the plant as well. Best seen by zooming on a computer screen.

7.4.5 Super-Resolution on Real Bursts

To showcase the ability of our method to produce fine alignments on real images, we perform burst super-resolution (SR) with our alignments. We use the popular inverse problem framework employed in [79, 3] to achieve the task. To recover the high-resolution image $\mathbf{x}$ from a set of K noisy and low-resolution observations $\mathbf{y}_i$ with $i \in [0, K]$ we solve the minimization problem $\min_{\mathbf{x}} \sum_i^K \|DBW_i\mathbf{x} - \mathbf{y}_i\|_2^2$, with a gradient descent algorithm. D is a decimation operator which reduces spatial resolution, B is a blurring operator, and W is a warp parametrized by the optical flow. In our experiments, DB is chosen as the average pooling operator following [3]. The gradient can be derived as $\sum_i^K W_i^\top B^\top D^\top (DBW_i\mathbf{x} - \mathbf{y}_i)$. The adjoint operator W^T of the warp is implemented in our using the backward implementation of `gridsample` operator available in PyTorch. The optical flow to warp the reference high-resolution image $\mathbf{x}$ candidate is estimated in two steps using our method and then the fixed point algorithm presented in Sec. 7.3 to infer the motion field of interest. We perform SR on demosaicked RAW frames with bilinear filtering. We visually compare our results in Figure 7.7. Our algorithm is able to recover fine details, including, for instance, the fine texture on the rum bottle or the hair of the doll, that were not distinguishable in the original frames.

Figure 7.7: burst super-resolution on real raw bursts exploiting our alignment method. Top: low-resolution crops. Bottom: super-resolution exploiting our alignment method. Best seen by zooming aggressively on a computer screen.

7.4.6 Impact of a Good Depth Initialization

We show in Figure 7.8 the impact of the initialization of the depth-map to the performance of our method. We gradually increase the variance of a Gaussian random noise added to the 16×16 initialization depth map and evaluate the performance of our algorithm on our synthetic dataset with various depth, pose, and alignment metrics. This experiment demonstrates that our method is robust to noise on the initialization depth map. Or model only requires a coarse estimate to converge to the right solution.

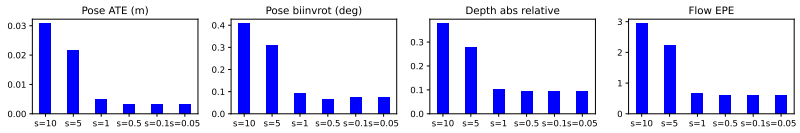


Figure 7.8: Figure: Noise on the initialization depth map. Our method is robust to noise, the performance begins to degrade when the noise on the depth map is greater than 1 meter.

7.a Appendix

7.a.1 Additional Experiments

7.a.1.1 Ablation Study

We make an ablation study to understand the impact of the different choices in our modeling and algorithm. We compare the global algorithm to an identical algorithm using the same hyperparameters but, respectively, without the exponential parametrization of the motion, with regularization (total variation and determinant), without the reparametrization of the plane, with patches of size one, i.e., a pixel-wise loss and without the use of a lower resolution grid for the $\mathbf{n}_i$ structure variables.
We report the performance on the pose estimate in Table. 7.5, depth estimate in Table. 7.7, and flow estimate in Table. 7.6.

7.a.1.2 Comparison with Monocular Method

Monocular depth estimation methods can only estimate depth up to an affine transformation. Therefore, we evaluate them up to an affine correction. It does not make

Method	Left l2 (m)↓	ATE (m) ↓	Geom (m) ↓	Biinvrot l2 (deg) ↓
Dataset	Blender1 (small motion)			
w/o subgrid	0.0033	0.0029	0.0019	0.0784
with $k = 1$	0.0028	0.0024	0.0016	0.0698
w/o plan reparametrization	0.015	0.0133	0.0113	0.3333
with regularization	0.0032	0.0028	0.0019	0.0743
w/o exponential parametrization	0.0032	0.0028	0.0019	0.073
Dataset	Blender2 (tiny motion)			
w/o subgrid	0.0021	0.002	0.0019	0.0322
with $k = 1$	0.0023	0.0021	0.002	0.0367
w/o plan reparametrization	0.0049	0.0048	0.0046	0.0441
with regularization	0.0021	0.002	0.0018	0.0329
w/o exponential parametrization	0.0021	0.002	0.0018	0.0339

Table 7.5: Pose errors in the ablation study.

Method	EPE ↓	RMSE ↓	NPE1 ↑	NPE2 ↑	NPE3 ↑
Blender1 (small motion)					
w/o subgrid	0.6207	1.2487	0.8379	0.924	0.9487
with $k = 1$	0.6327	1.1491	0.8061	0.9192	0.953
w/o plan reparametrization	377.1064	4333.003	0.4215	0.5987	0.6932
with regularization	0.6052	1.2147	0.8386	0.9256	0.9518
w/o exponential parametrization	0.6017	1.2048	0.8392	0.9263	0.9526
Blender2 (tiny)					
w/o subgrid	0.2751	0.3347	0.9334	0.9951	0.9999
with $k = 1$	0.2904	0.3494	0.9299	0.9959	0.9999
w/o plan reparametrization	115.5213	633.4214	0.7204	0.813	0.8333
with regularization	0.2713	0.3287	0.9349	0.9954	0.9999
w/o exponential parametrization	0.2716	0.3292	0.935	0.9954	0.9999

Table 7.6: Optical flow errors ablation study.

Method	Abs rel ↓	Sqr rel ↓	RMSE↓	Delta 1↑	Delta 2 ↑	Delta 3 ↑
Blender1 (small motion)						
w/o subgrid	0.0969	0.1107	0.6958	0.889	0.9642	0.9895
with $k = 1$	0.1058	0.1229	0.7508	0.8726	0.9692	0.9931
w/o plan parametrization	0.3687	0.6707	1.6153	0.3357	0.6333	0.8194
with regularization	0.0967	0.1082	0.6905	0.8872	0.966	0.9915
w/o exponential parametrization	0.0953	0.1058	0.6825	0.8893	0.9669	0.9918
Blender2 (tiny motion)						
w/o subgrid	0.1817	0.3005	1.3918	0.6707	0.9541	0.9969
with $k = 1$	0.1787	0.2902	1.3785	0.6735	0.962	0.998
w/o plan parametrization	0.257	0.5266	1.7724	0.5007	0.8882	0.986
with regularization	0.1773	0.2897	1.3744	0.6834	0.9587	0.9975
w/o exponential parametrization	0.1756	0.2862	1.3685	0.6876	0.9596	0.9976

Table 7.7: Depth errors for the ablation study.

sense to compare them to the binocular method with linear correction as in Table 7.4. On the other hand, to compare them to the latter, we must recalculate the error of each of the methods in Table 7.4 with an affine correction. The results are reported in Table 7.8.

Method	Abs rel ↓	Sqr rel ↓	RMSE↓	Delta 1↑	Delta 2 ↑	Delta 3 ↑
Blender1 (small motion)						
Midas [292]	0.1589	1.0747	1.3148	**0.8019**	0.951	0.9824
RCVD [276]	0.2038	1.0622	1.3888	0.698	0.9191	0.9684
Ours	**0.1544**	**0.2229**	**0.9258**	0.7881	**0.9544**	**0.9911**
Blender2 (tiny motion)						
Midas [292]	**0.0790**	**0.0786**	**0.7166**	**0.9429**	**0.9929**	**0.9986**
RCVD [276]	0.0971	0.1131	0.8244	0.9149	0.988	0.9973
Ours	0.1763	0.2875	1.3711	0.6857	0.9594	0.9976

Table 7.8: Depth errors metrics on the two proposed synthetic bursts datasets.

7.a.1.3 Additional Details on the Experiments

7.a.1.4 Choosing the Best Reference Frame

The performance of our method can be enhanced by more carefully selecting the reference frame. To do so, we run our algorithm only on its first scale with different reference image candidates. We then select the reference image which provided the lowest optimization error and finish the optimization across all the scales for this candidate. This approach increases performance on synthetic datasets, as shown in Tables 7.11, 7.9 and 7.10.

Method	Left l2 (m)↓	ATE (m) ↓	Geom (m) ↓	Biinvrot l2 (deg) ↓
Dataset	Blender1 (small motion)			
Ours	0.0037	0.0032	0.0022	0.0921
Auto ref frame	**0.0032**	**0.0027**	**0.0020**	**0.0795**
Dataset	Blender2 (tiny motion)			
Ours	**0.0020**	0.0020	**0.0018**	**0.0275**
Auto ref frame	**0.0020**	**0.0795**	**0.0020**	**0.0019**

Table 7.9: Pose errors in the ablation study.

Method	EPE ↓	RMSE ↓	NPE1 ↑	NPE2 ↑	NPE3 ↑
Dataset	Blender1 (small motion)				
Ours	0.6966	1.341	0.8166	0.9095	0.9386
Auto ref frame	**0.5449**	**1.0444**	**0.8548**	**0.9324**	**0.9552**
Dataset	Blender2 (tiny motion)				
Ours	0.2586	0.3106	0.9374	**0.9963**	**1.0000**
Auto ref frame	**0.2572**	**0.306**	**0.9331**	0.9959	**1.0000**

Table 7.10: Optical flow errors ablation study.

Method	Abs rel ↓	Sqr rel ↓	RMSE↓	Delta 1↑	Delta 2 ↑	Delta 3 ↑
Blender1 (small motion)						
Ours	**0.1954**	**0.3168**	**1.0253**	**0.7231**	**0.891**	**0.9811**
Auto ref frame	0.2078	0.3500	1.0844	**0.7231**	0.8795	0.9804
Blender2 (tiny motion)						
Ours	**0.2376**	**0.5626**	**1.8198**	**0.5997**	**0.8866**	**0.9853**
Auto ref frame	0.2577	0.652	1.9556	0.5643	0.8535	0.9762

Table 7.11: Depth errors for the ablation study.

7.a.2 Additional Visual Results

7.a.2.1 Estimated Occlusion Mask

We use the previous algorithm on the depth map obtained at the optimization's last step and note the points for which the fixed point algorithm does not converge.
We use a threshold and a maximum number of iterations to construct the non-convergent set. This set constitutes a partial occlusion mask. It can be used in downstream tasks to avoid aggregating erroneous information because it is occluded. Fig. 7.9 shows examples of masks on synthetic data.

7.a.2.2 Depthmaps

We provide additonal examples of depth maps from both synthetic bursts (Fig. 7.10) and real bursts (Fig. 7.11). All disparity maps were aligned to the groundtruth with an affine transform by using the least square criterion of [292].
For a fair comparison, we also show the results of DfUSMC without their additional depth map filtering, which is essential to obtain a visually appealing depth map. However, this step introduces a stratification of the depth map, which is not present with our method.

7.a.2.3 Point Clouds

We provide an example of a point cloud generated with our method and DfUSMC[273] in Fig. 7.12. With the focal length known, the point clouds were obtained by a 3D projection of the depth map. For example, we can see that our method preserves the geometry of the room since the floor is almost perpendicular to the wall. DfUSMC[273] does not perform as well in this regard, in addition to the previously mentioned stratification of the depth map.
In our experiments, whose results are reported in Table 7.1, Table 7.3, Table 7.4, we evaluated the performance of the Saop method [272] by calculating the average results across all scenes where Saop successfully converged. On the *blender 1* dataset, we excluded one scene where Saop did not converge. Excluding this scene for Saop does not change the ranking of the methods and the conclusion of our experiments.

7.a.2.4 Visual Inspection of the Registration of Real Frames

Fig. 7.13 visually demonstrates the alignment quality achieved with our method on a real burst. To assess the alignment quality, we generate images by overlaying the green and blue channels of the warped source images onto the red channel of the target image, following a similar approach as [271] In this example, we observe that the majority of the frames exhibit a good alignment, while a few frames (5 out

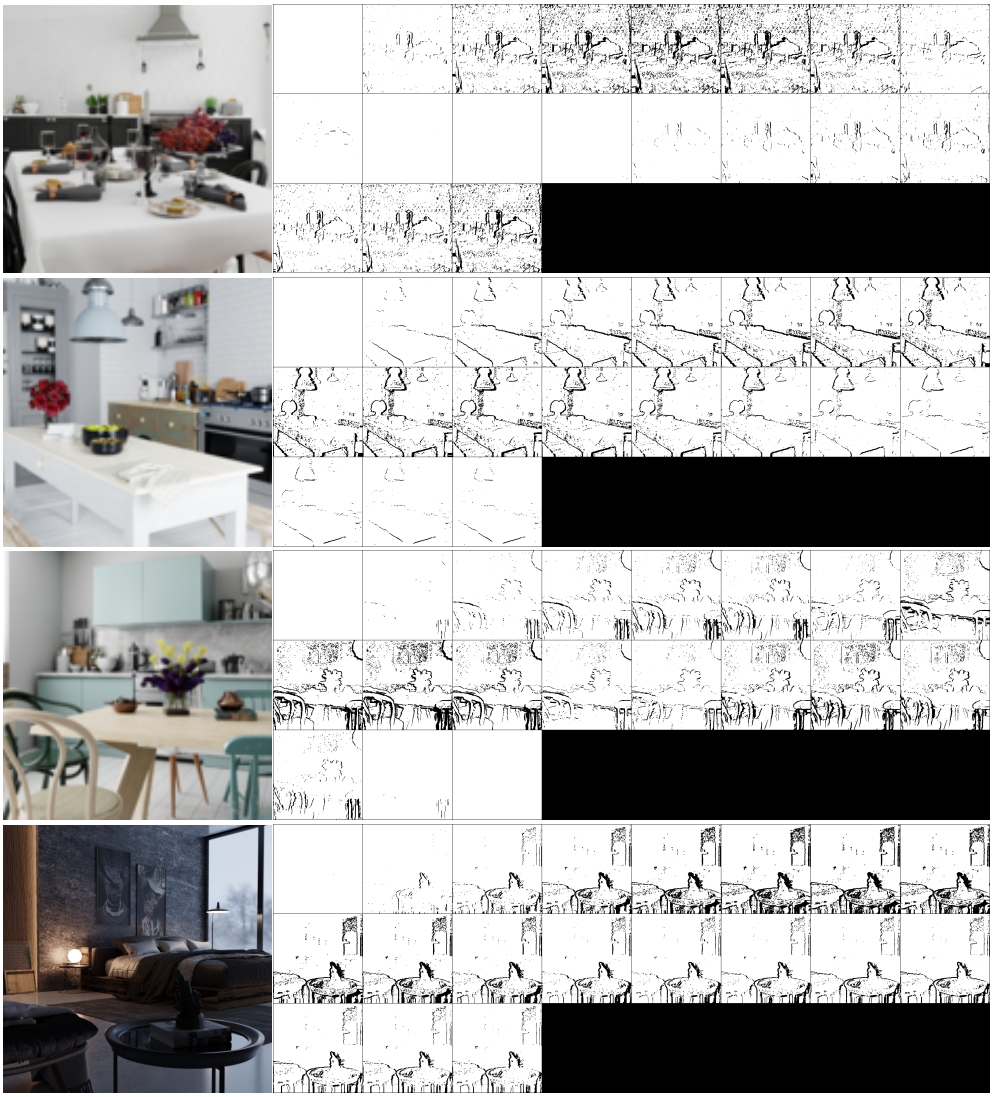


Figure 7.9: Partial occlusion mask obtained using the fixed point algorithm for four example of the blender 2 dataset.

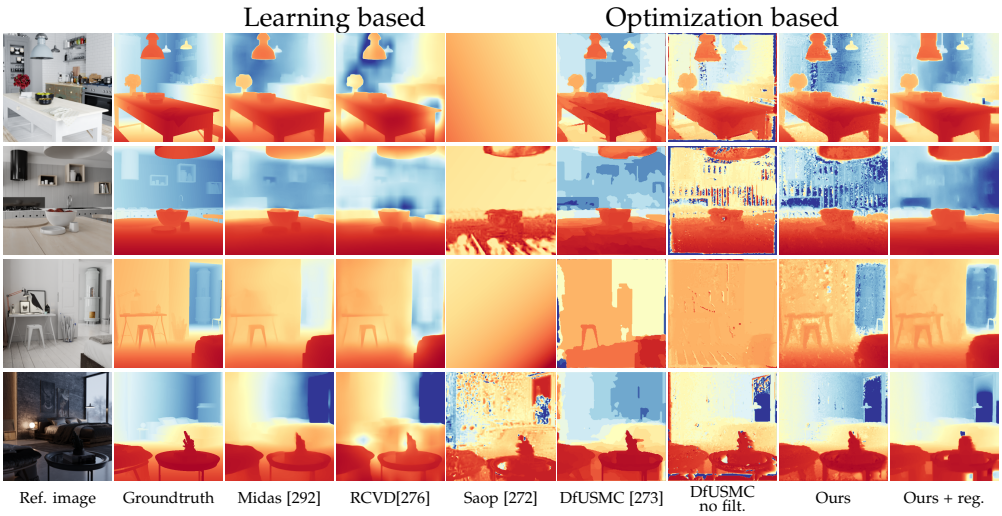


Figure 7.10: Depth estimation from synthetic bursts (*Blender 2* dataset).

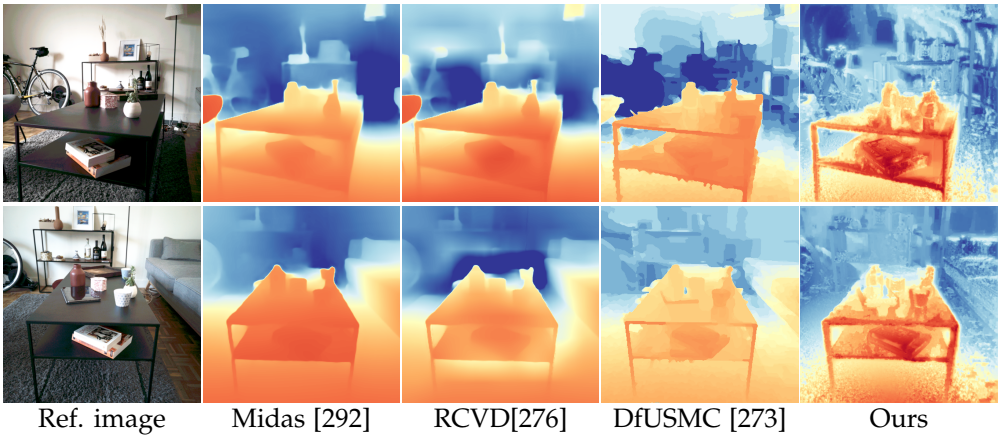


Figure 7.11: Depth estimation from real bursts.

of 15) show inadequate alignment particularly in certain regions of the foreground (see for example the books or the plant).

7.a.2.5 Pose Estimation Visualization

To visualize the positions the algorithm approximates, we can look at the translation part of the positions. Because our images come from a burst, we use the temporal coherence of the series of pictures and can trace the trajectory of the camera center during the burst. After rescaling, we compare the trajectory approximated by the algorithm to the trajectory used to create the burst in Blender. Fig. 7.14 shows examples of trajectories for different images of the Blender 2 dataset during the last three stages.

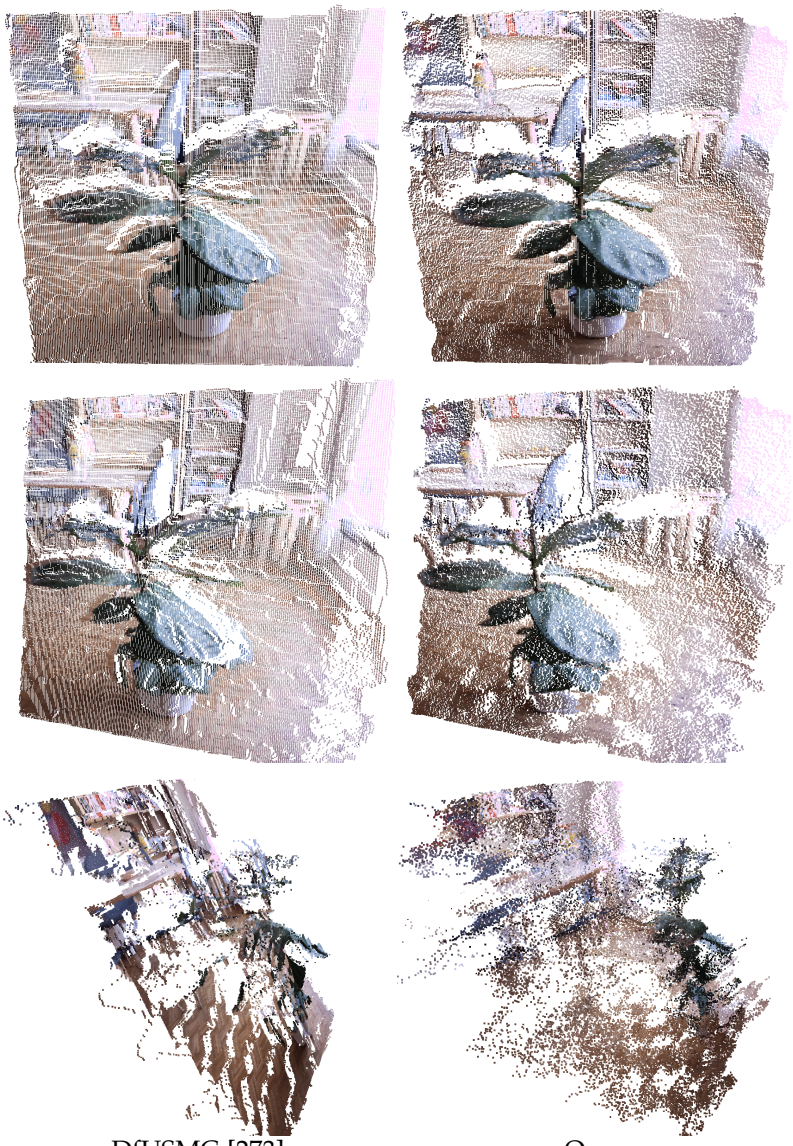


Figure 7.12: Point cloud estimation from real bursts.

Figure 7.13: Qualitative alignment results of our method on a real burst. Images are generated by superimposing the warped source images on the target image.

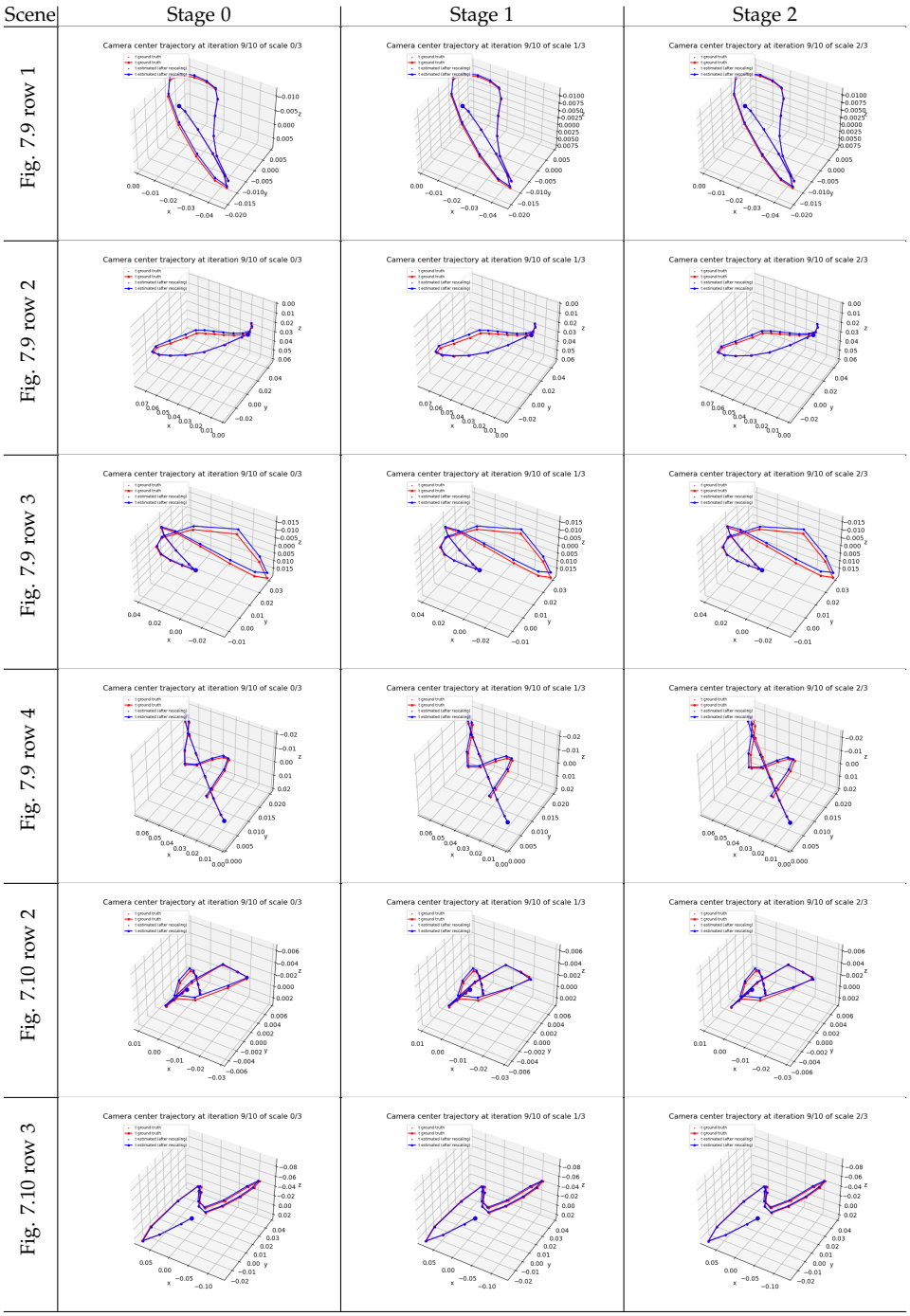


Figure 7.14: Trajectory at different scales of the coarse to fine approach for all the scenes shown in Fig. 7.9 and Fig. 7.10.

Chapter 8

Conclusion, Industrialization, and Perspectives

Chapter abstract:
This chapter is the conclusion of this thesis. It begins with a summary of our contributions in Section 8.1. In section 8.2, we present the limitations we observed for each method we experimented. The super-resolution algorithms we designed led to a startup aiming to provide software solutions for enhancing image quality. Moving on the section 8.4, we discuss from a technical standpoint the new challenges we had to face for the industrialization of the algorithms, Finally, in Section 8.5, we explore a handful of promising research directions that we believe would serve as exciting extensions of this thesis.

Contents

8.1 Summary of this Thesis

This thesis explores hybrid methods for inverse problems, focusing on their practical implementation in burst photography for real-world applications. We presented in the first part the physics of imaging systems and leveraged its knowledge to enhance learning-based methods for image restoration. We used, for instance, the knowledge of the image formation model, the sensor's noise model and the camera's ISP. In the first part, we studied learned inverse problems methods for single-image restoration based on unrolling. We proposed a trainable inverse problem regularized with a non-local sparse image prior, which uses a differentiable relaxation of the group lasso solver. Then, a framework providing differentiable relaxations of convex non-smooth optimization solvers for classic image priors is studied. The models proposed in this first part demonstrate comparable performance to larger neural networks with fewer parameters and less training data. They also have increased interpretability and faster training times. The second part of the thesis delves into integrating hybrid methods for multi-frame image restoration for real-world scenarios. The design of plug-and-play algorithms for burst photography is explored, with efforts directed toward practical implementation for mobile devices. Finally, the last part of this thesis tackles image registration for image bursts. We propose a new dense multi-frame registration algorithm enabling 3-D scene reconstruction from image bursts with tiny baselines.

8.2 Limitations

8.2.1 Data Quality

With the exception of the final chapter, all the methods proposed in this study are data-driven. While hybrid approaches demonstrate increased stability when applied to real-world data, our observations still highlight their dependency on the quality of the input data. Consequently, the accuracy of simulated data consistently emerges as a bottleneck, a challenge we particularly encountered during the industrialization phase. When implementing these methods on new camera systems, a significant portion of our efforts is directed toward refining data simulation for the specific imaging devices.
The estimation of required dataset quality, encompassing factors like accuracy, diversity, and dataset size, for tasks involving image restoration remains an underexplored area. And so far, we have limited solutions to offer in response to this issue.

Adaptability to New Hardware. The models we developed are tailored to specific imaging systems. In theory, a comprehensive process involving camera calibration to infer the camera's parameters and subsequent model retraining is essential to address camera defects. Nevertheless, our practical observations, as emphasized in the concluding remarks of Chapter 5, show this is not a major limitation in practice. The same model gives satisfactory results on a broad range of cameras. This outcome is unsurprising, given that our approach primarily targets the enhancement of the camera's sensor. While sensors exhibit a narrower range of limitations, optical limitations manifest more diversely. Therefore, we anticipate more complexities when designing models that address optical limitations.
However, it is still important to acknowledge that our experimentation has not covered scenarios involving different sensor types, such as CCD sensors. We have

also not delved into the effects of a rolling shutter mechanism.

8.2.2 Learned Inverse Problems

GPU memory footprint during the training phase. Memory consumption is the main limitation of the LIP methods proposed in this thesis, which rely on unrolled optimization. For unrolled methods, the intermediate results computed during the forward pass of the iterative solver must be stored in memory to accumulate gradients during the backward pass. To compute the full gradient, see Chapter 2 for more details. This can result in memory issues on the GPU if the number of iterations is too large. In our experiments, we typically required top-end GPUs (at the time of this Thesis) with approximately 24Gb of memory. See Chapters 3 and 4 for more details on GPUs used for our experiments. Note that solutions exist to address this limitation, for example, truncated backpropagation through time [71]. Another option discussed in the Background Chapter 2 is to compute the exact hypergradient leveraging the implicit function theorem.

FLOPS. Even though unrolled methods result in compact models with few trainable parameters, it does not always come with a reduced number of operations performed (FLOPS) compared with traditional one-pass neural networks. Indeed, due to the iterative nature of the algorithm during the inference procedure, the number of operations can stay important, especially for a large number of iterations. Consequently, studied hybrid models are not always superior to their neural network counterparts regarding the number of operations. Note that the number of operations generally correlates with the algorithm's latency, even though latency depends on many factors, including implementation and hardware.

Training instabilities. We also observed in our experiments that unrolled models may diverge during the training phase. As a rule of thumb, we observed that more sophisticated iterative solvers with automatic stepsizes (conjugate gradient, etc.) tend to exhibit more instabilities during the training phase than basic solvers such as gradient descent or ISTA. It is often easier to differentiate through simple solvers than more elaborate ones. However, we do not have theoretical insights for explaining this phenomenon.

8.2.3 Burst Methods

Optics limited cameras. In contrast with sensor-limited devices, restoring images from cameras with optical limitations poses a persistent challenge. We observed limited gains on cameras limited by the optics, such as telelens cameras on smartphones. The lens response of such a device is much more complex than the sensor to model; therefore, simulating accurate data for training is a challenge.

Registration. We also faced difficulties in aligning real-world frames. Real images involve complex motions that are difficult to model with simple parameterizations such as rigid motions. Rigid motions fail to model motions induced by parallax effects and/or motions within the scene. It also fails to capture nonrigid motions. The block parameterizations allow for more flexibility but trade with instability issues as it requires aligning smaller patches and, therefore, requires fitting transformations on fewer data points. The Lucas-Kanade algorithm we implemented exhibits instabilities for surfaces with few textures and tends to produce wrong alignments.

Finally, the tiles-based motion model fails to capture large motions exceeding block size.

8.2.4 Multiframe registration

Regarding the work introduced in Chapter 7, we observed, after working with real bursts, that the registration is subject to a noticeable performance gap between synthetic and real data. The performances achieved in idealized synthetic scenarios may not be fully replicated when working with real data. We still do not understand the reasons for this behavior; some hypotheses and directions of amelioration are presented in the future work Section 8.5.

8.3 Challenges of the Industrialization

As highlighted in the introduction section, the super-resolution algorithm introduced in this thesis has led to the creation of a startup aiming to provide software solutions for enhancing image quality in various contexts, including smartphones and scientific imaging. We confronted several technical challenges to transition from academic research to a robust industrial product. The difficulties we faced were three-fold: **(1)** the management of significantly more complex data; secondly, **(2)** the necessity to adapt to new evaluation criteria distinct from our accustomed standards; **(3)** and lastly, much more Contrast computational resources.

Data quality variability and heterogeneity. A noticeable shift in emphasis is observed in the context of industrial applications for our algorithm. Here, the priority rests on stability and robustness—ensuring that the algorithm minimizes artifacts even in the most challenging scenarios. This stands in contrast to focusing solely on reconstruction quality under optimal conditions. The less favorable scenarios for burst super-resolution include bursts characterized by large camera motions, scenes with limited textures, abundant non-rigid and object motions within the scene, and pronounced parallax effects from nonplanar scenes. Additionally, the algorithm must accommodate data from cameras featuring subpar optics, including distortions, optical blurring defects, and/or lens flare. As discussed in this thesis, these defects are hard to model and correct.

Different evaluation criterions. Secondly, academic research prioritizes quantitative standardized benchmarks, whereas industries place greater significance on subjective rankings derived from proprietary datasets. As pointed out in the introduction, quantifying image quality is highly intricated to human perception and remains an open question. To the best of our knowledge, in the industry, perceptual evaluations made by experts often dictate image rankings. And this is especially the case in the smartphone industry, where pleasant photographs are paramount.

Embarkability. Lastly, we encountered significant challenges concerning the integration of mobile devices. Implementing the algorithm on the GPU/DSP of mobile devices encounters a less mature ecosystem with more prevalent issues. It lacks the comprehensive documentation found in the well-established PyTorch ecosystem. The table below outlines the disparities observed while transitioning from academic to industrial evaluations.

Academia	Industry
reconstruction quality	robustness
best cases	worst cases
quantitative results	qualitative results
	computational cost

8.4 Example of Add-Ons to the Super-Resolution Algorithm

These emerging challenges prompted us to propose some technical solutions. In this concluding Section, we succinctly explore a few of these add-ons to the burst super-resolution method.

8.4.1 Hiearchical Lucas Kanade

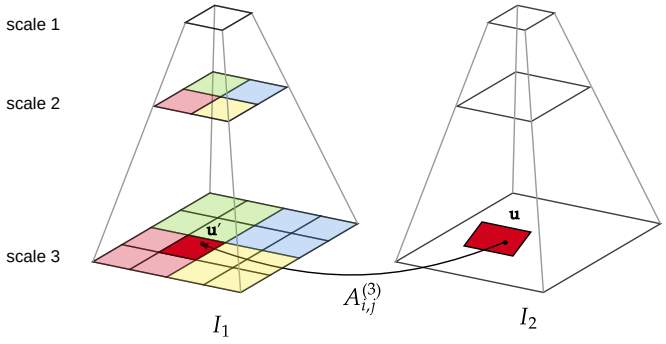


Figure 8.1: Hiearchical block affine registration. Registration of a pair of images with coarse-to-fine alignment. At the coarsest scale, the motion is estimated between Next, we subdivide the tiles at a finer scale and use the alignment parameters from the ancestor tile from the coarser scale as an initial guess. The method supports displacements up to $2^{N_{\text{scale}}}$ pixels.

Registration is a bottleneck for the super-resolution method. We developed improved registration algorithms to handle the limitations of the block-affine model to handle large displacements and complex motions featuring parallax effects. Instead of a tile-based registration method, we developed a Hierarchical implementation of the parametric Lucas Kanade algorithm [74, 73].
We were inspired by the hierarchical alignment proposed in HDR+ paper [81]. The authors performed a coarse-to-fine alignment on a multiscale pyramid. They proposed a translation tile-based alignment for each pyramid level, using the alignments from the coarser scale as an initial guess. The tile density is increased at each scale to predict a finer motion. Similarly, we developed a Hierarchical implementation of the parametric Lucas Kanade to handle more complex motions than pure translations for aligning the tiles. Likewise to [81], we infer tile-based affine motions at each scale and subdivide into more tiles at the following scale, using the previous scale's predicted motion as an initial guess. We are not aware of existing implementations of a hierarchical version of the parametric Lucas Kanade algorithm in the literature. The general principle of the algorithm is presented in Figure 8.1.

Results. Aligned patches are warped from the large image, hindering the border effects we experienced with the first block version. We can align larger motions and

manage to decrease the tile sizes. Figure 8.2 shows the result of our algorithm on synthetic data.

Figure 8.2: Alignement on synthetic data with the hierarchical implementation of the Lucas Kanade. The target image $\mathbf{I}_1$is synthetically warped with a rotation centered in the middle of the image. The algorithm performs registration on 3 scales and successfully aligns each tile at the finest level. **Left:** Illustration of each tile and its estimated motion parameters. **Right:** Warped templated image $\mathcal{W}_{\mathbf{p}}\mathbf{I}_2$ that successfully matchs the target image.

8.4.2 Fast Gradient Approximation and Fusing Operators

Numerous avenues have been explored to reduce the memory requirements and latency of the burst super-resolution technique introduced in [3] for an efficient implementation on mobile devices. One approach involves optimizing the architecture of the neural networks, while other enhancements have been made in the optimization phase itself. The subsequent section outlines some of the optimization concepts we worked on.
From Chapter 5, recall that the sequence of low resolution frames are obtained through the linear forward model

$$\mathbf{y}_k = DBW_{\mathbf{p}_k}\mathbf{x} + \boldsymbol{\epsilon}_k \quad \text{for} \quad k = 1 \cdots K, \tag{8.1}$$

where the operator $W_{\mathbf{p}_k}$ parameterized by $\mathbf{p}_k$ warps $\mathbf{x}$ to compensate for misalignments between $\mathbf{x}$ and $\mathbf{y}_k$ induced by camera motion between frames, B is a blurring operator accounting for photons integration over pixels, D downsamples the image in both the spatial and spectral domains and ϵ_k is some additive noise. We rewrite the formation model

$$\mathbf{Y} = U_{\mathbf{P}}\mathbf{x} + \boldsymbol{\epsilon}, \quad \text{where} \quad U_{\mathbf{P}} = \begin{bmatrix} DBW_{\mathbf{p}_1} \\ \vdots \\ DBW_{\mathbf{p}_K} \end{bmatrix}, \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{bmatrix}, \mathbf{P} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_K \end{bmatrix}, \boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{\epsilon}_1 \\ \vdots \\ \boldsymbol{\epsilon}_K \end{bmatrix}. \tag{8.2}$$

In the method introduced in [3], the super-resolution inverse problem is solved with half quadratic spliting; and to minimize the data fitting term $L(\mathbf{x}) = \|U_{\mathbf{P}}\mathbf{x} - \mathbf{Y}\|^2$, a plain gradient descent is used.

Effecient gradient implementation in Pytorch. For better performances, the gradient of the data fitting term is implemented by hand rather than relying on auto diff. The analytical gradient is given by

$$\nabla_{\mathbf{x}} L(\mathbf{x}) = \sum_{i=1}^{K} U_{\mathbf{P}}^{\top} \left(U_{\mathbf{P}}\mathbf{x} - \mathbf{Y} \right), \tag{8.3}$$

and can be implemented efficiently with Pytorch operators. Note that we used the fact that Pytorch provides the adjoint operator for many image processing operators, including convolutions and warps (`gridsample` in Pytorch), as Pytorch's auto diff engine requires it. The adjoints are not directly available in Pytorch Python library. Still, they can be linked from the c++ libtorch library with some bindings[1]. However, the gradient computation is costly as it requires computing and storing in memory K high-resolution images $W_{\mathbf{p}_i}\mathbf{x}$.

Commuting B and W. As pointed out in [29, 30] under pure translation and arbitrary blurring kernel, or rigid motion and radially symmetric blurring kernel, the operators B and W commute. Leveraging this assumption, we have the following formation model approximation

$$\tilde{\mathbf{Y}} = V_{\mathbf{P}} B\mathbf{x} + \boldsymbol{\epsilon}, \quad \text{where} \quad V_{\mathbf{P}} = \begin{bmatrix} DW_{\mathbf{p}_1} \\ \vdots \\ DW_{\mathbf{p}_K} \end{bmatrix}, \tilde{\mathbf{Y}} = \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_K \end{bmatrix}, \tag{8.4}$$

where $\tilde{\mathbf{y}}_k$ denotes the approximated low resolution image.

Fusing ops. From the equation 8.4, further optimization is possible by fusing operations. Indeed, the spatial and spectral downsampling and the warping can be fused in a single operation $V_{\mathbf{P}}$ performing proper resampling straight on low-resolution grids and avoiding the cost of computing and storing K high-resolution images. Similar optimization can be achieved for the adjoint operator $V_{\mathbf{P}}^{\top}$. Therefore, it is possible to approximate the gradient with the expression

$$\nabla_{\mathbf{x}} L(\mathbf{x}) \approx B^{\top} V_{\mathbf{P}}^{\top} \left(V_{\mathbf{P}} B\mathbf{x} - \mathbf{Y} \right) \tag{8.5}$$

Benchmarks. We show the profiling of the two forward models, using the Pytorch integrated Profiler, performed on the CPU. Note that the adjoint operations to compute the gradient have the same order of complexity. We also computed the discrepancy between the inferred lr frames with the approximated forward model and the true forward model, i.e., $\|\tilde{\mathbf{Y}} - \mathbf{Y}\|_2^2$, reported in terms of mean square error (MSE).
The experiment was performed with an hr gray image $\mathbf{x} \in \mathbb{R}^{400\times400}$, applying a downscaling factor of $\times 1/2$, generating 10 low resolution observations frames $\mathbf{y}_i \in \mathbb{R}^{200\times200}$. The motions randomly sampled rigid translations with translations in the range $[-10\text{px}, 10\text{px}]$ and rotations in the range $[-1^{\circ}, 1^{\circ}]$. We reported millisecond latency (ms) and memory usage in megabytes (Mb).

Forward model	$U\mathbf{x}$	$VB\mathbf{x}$
Speed (CPU)	41.337 ms	3.981 ms
Memory	24.41 Mb	2.13 Mb
MSE	0	$1.06 \cdot 10^{-2}$

Significant gains are obtained on the CPU. This also improves performance on GPU. Further optimization is also possible by implementing a dedicated cpp kernel to avoid computing some values twice (especially some quantities for the reverse and forward warping).

[1]See for more details `pytorch.org/tutorials/advanced/torch_script_custom_ops.html#building-the-custom-operator`

8.5 Future Work

In conclusion, the last chapter of this thesis presents exciting and promising directions for future research that we are eager to explore.

8.5.1 Joint Optical Deconvolution and Super-Resolution

Throughout our research, we have employed a standard linear camera model to describe the imaging system utilized in [29, 30] Notably, our study deliberately omitted the incorporation of optical components within the degradation model. Our analysis solely encompassed the blurring effects of pixel spatial integration, modeled with pixel binning. Our super-resolution work focused on sensor-constrained devices with highly aliased frames.

Spatially varying point spread functions. An exciting research topic for further investigation lies in the integration of optical modeling, aimed at exploring potential enhancements in instances of moderate blurs in the imaging system. Modeling of blur induced by the optics would be needed to do so. A first step toward that goal would be to model diffraction and lens aberrations with spatially varying point spread functions (PSFs). That spatially varying PSF could be calibrated for several distances, leveraging prior works. Alternatively, another research direction would be to jointly estimate the blur parameters of the imaging systems.

Improved solver for deconvolution problems. We focus on the case of sensor-limited cameras. In that setting, gradient descent has proven sufficient in terms of performance. However, a tailored inverse solver would be needed for blur with a larger PSF (i.e.) conjugate gradient descent or resolution in the Fourier domain.

Improved camera model. It would also be relevant to use work done in 7 to jointly estimate high-resolution images and scene structure. This way, it would be possible to use a refined camera model taking into account the structure of the scene when forming images (taking into account defocus blur or edge effects).

8.5.2 Ray-Tracing Based Data Simulations

It has been widely acknowledged that the realism of the training data was a major criterion for the performance of models based on deep learning for image restoration. However, we recognized that the quality of simulated data plays a crucial role. To further improve our approach, we propose exploring more accurate optics modeling. We saw in the background section that PSF models are limited and are far from accounting for all physical phenomena. A more effective way to simulate realistic data would be to leverage ray tracing. Ray tracing-based simulations and compound lens simulations have the potential to improve the quality of noisy/clean image pairs significantly. This way, we could generate more realistic low-resolution/high-resolution data pairs useful for training imaging devices. Depending on the scene structure and lighting, we could model complex phenomena such as defocus blurs. It would also be very relevant to study how to estimate the needed dataset quality in terms of accuracy, diversity, and size for image restoration tasks.

8.5.3 Differentiable Camera Model

A differentiable camera model considers the scene's structure for forming an image on the sensor's plane. This way, complex optic effects such as defocus blur can be rendered. This paves the way for complex image reconstruction from scenes shot with different settings, including focus plane and position. Complex tasks like focus stacking can be performed.

8.5.4 Diffusion-Based Priors on Image and Formation Model

Image diffusion-based priors. The diffusion prior has demonstrated impressive results in image restoration. Its ability to produce visually appealing images with sharp details, especially in challenging scenarios with high-resolution factors, holds great promise. Integrating diffusion models into the Plug-and-Play framework as alternative denoisers can be a fruitful exploration avenue. More generally, combining PnP/deep unfolding with diffusion-based image priors.

Parametric formation model. A research direction that has been neglected is the optimization of camera parameters and extrinsic parameters/scene structure jointly. We perform camera calibration and motion field estimation and then solve an inverse problem to estimate a clean image, with the exception of [3], where we jointly optimize motion and image reconstructions. An exciting research direction would be to simulate realistic camera models and form a prior on camera models and motion fields.

Extrinsic parameters and scene's structure. We are also interested in exploring the application of diffusion models to learn priors on camera forward operators, such as depth-varying PSFs and geometrical deformations. Finally, a similar prior could be used to model motion fields induced by camera shakes.

8.5.5 Implementation on GPU/DSP for Mobile Devices

While our current implementation successfully operates on the CPU within smartphones, we recognize the compelling need to extend its capabilities to GPU/DSP platforms because (1) CPU is devoted to more priority tasks on smartphones and (2) it is expected to improve latency significantly. The challenge entails the task of aligning the algorithmic architecture with the specific demands of these hardware configurations, thereby harnessing the complete computational potential of each platform effectively[2]. Substantial research has been dedicated to tailoring neural network architectures for deployment on embedded devices. However, the domain of optimized architectures for handling extensive image processing tasks on embedded platforms remains relatively unexplored, particularly with large images. Exploration to Halide[3,4] is also promising.

8.5.6 Improved Multi-Frame Registration

Finally, for our last work dealing with multi-frame registration and 3D reconstructions, the research directions include **(1)** incorporating a geometrical distortion

[2]For additional insights, refer to `https://lnstadrum.github.io/mva/`
[3]See `halide-lang.org/`
[4]See blog.minhazav.dev/

model induced by the optics to bridge the gap between simulated and real-world data. **(2)** employing a better initialization strategy for depth maps instead of relying on smartphones' depth. For instance, one could perform initialization with a sparse structure from motion method such as the one proposed in the DFUMSC algorithm [273]. Finally, **(3)** a faster implementation using closed-formed gradient rather than auto diff for normal map optimization could also drastically improve speed.

Appendix

Appendix A

Multi Frames Registration Algorithm for HDR Images

Problem setting. We focus on multi-frame registration in the case of frames with heterogeneous content. This can be, for instance, a set of images with bracketed exposures. In that case, the saturated regions and the SNR vary for each frame. This can also be the case for frames shot with varying plane, in the case of focus stacking to reconsruct images with extended depth-of-field.

Method. We propose to fit of a global transformation between the frames by utilizing a collection of pairwise affine transformations that have been individually estimated. These transformations are represented as edges of a graph where each node is associated with a frame. Note that for bursts with heterogeneous content, we only sample the frames with the nearest settings. The idea being that they have the maximum overlap of content and the matching is then simplified. For the case of a burst with bracketed exposures, this would imply to only align frames with similar exposures and avoid matching of the darkest frame with the lighter frame. Examples of such sampling graphs are shown in Figure A.1.

Optimization problem. Let us consider a set of affine transformations T_{ij} for $(i,j) \in E$. We want to find the transformationsH_j, L_i, for $i = 1, \cdots, n$, to fit approximately each T_{ij} and that $H_i L_i \approx I$. If we assume that $(i,i) \in E$ for all nodes i,we are looking for transformations H_i and L_i for $i = 1, \ldots, n$ that minimizes

$$\min_{H_i, L_i} \sum_{(i,j) \in E} \frac{w_{ij}}{2} \| T_{ij} - H_j L_i \|,$$

where $w_{ij} = 1$ if $i \neq j$ and $\gamma \geqslant 1$ otherwise.

Solver. We minimize with respect to each transformation H and L with block coordinate descent by minimizing a collection of 2×2 linear systems. Note that we increase the value of γ along the iterations.

Coarse-to-fine. We integrate this filtering on the multi-scale Lucas-Kanade algorithm. After performing optimization of motion parameters for each scale, we perform a filtering step. Note that it is possible to use different edge sampling at each

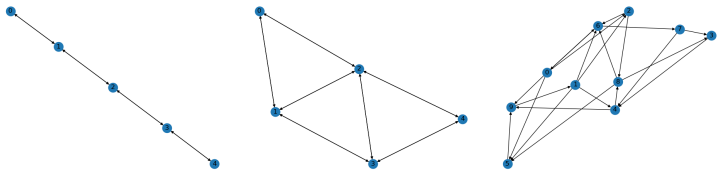


Figure A.1: Differnent transformations graphs, each node represents a frame of the burst. Each edge represents an estimated transformation T_{ij}. **Left:** sampling adjacent frames in the burst. **Middle:** sampling 2-nearest neighbors. Sampling random edges in the case of frames with homogeneous content.

Procedure 5 Coarse-to-fine transformation filtering

Input: $\mathbf{I}_1, \cdots, \mathbf{I}_N \in \mathbb{R}^n$, N_{scales}, $\{E_s\}_{s\in[1,N_{\text{scales}}]}$
  $H_i^0 \leftarrow I, \quad \text{for} \quad i = 1, \cdots, n$
  for s in $[1, N_{\text{scales}}]$ **do**
    $T_{i,j}^s \leftarrow \text{ParametricLucasKanade}\big(\mathbf{I}_i, \mathbf{I}_j, (H_i^s H_j^s)\big) \quad \text{for}(i,j) \in E_s$
    $\{H_i^s\}_{s\in[1,N]} \leftarrow \text{Filtering}(\{\text{T}_{\text{i,j}}^{\text{s}} \quad \text{for(i,j)} \in \text{E}_\text{s}\})$
  end for
  return $\{H_i^s\}_{s\in[1,N]}$

scale of the pyramid. As the first step of the pyramid is less costly in terms of optimization, we sample the set of edges more densely at the top of the pyramid. The procedure is summarized in Algorithm 5.

Experiments. We show in Figure A.2 the results of our algorithm on synthetic data. We generated warps of an image with random motions. Warped images are corrupted with some additive noise of variance $\sigma = 25$. We then sample a random graph of edges. We sample 25% of all edges. We then run our algorithm by performing optimization across four scales, with a filtering step at each scale. Figure A shows results of our experiments in terms of 4-corners geometric errors calculated as $\frac{1}{4}\sum_{i=1}^{4} \left\|\mathbf{x}_i^{\text{gt}} - \mathbf{x}_i\right\|_2$, where $\mathbf{x}_i$ denotes the corner coordinates in the image plane of the estimated geometric transformation, while $\mathbf{x}^{\text{gt}}$ denotes coordinates of the ground truth transformation. Figure A.2 shows results of our algorithm on synthetic data by sampling only nearest neighbors.

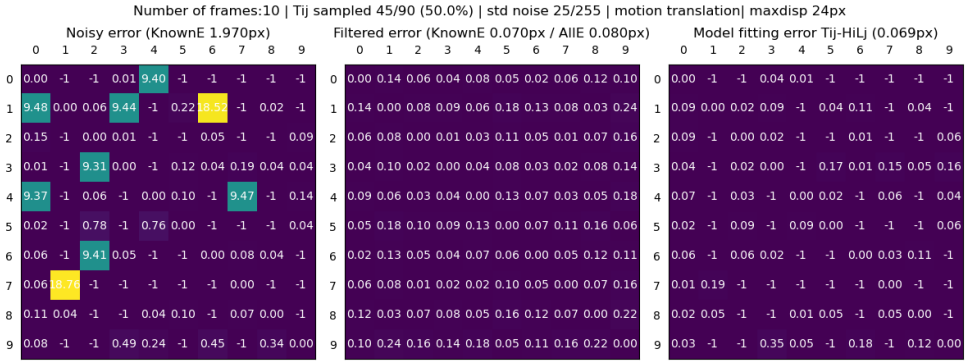


Figure A.2: Mutli frame image registration. **Left:** Geometric error on the sampled edges. Negative values account for unsampled edges. **Middle:** Pairwise geometric error on the burst. Our method successfully eliminates outliers and finds transformation parameters for unsampled edges. **Right:** Geometric model fitting error.

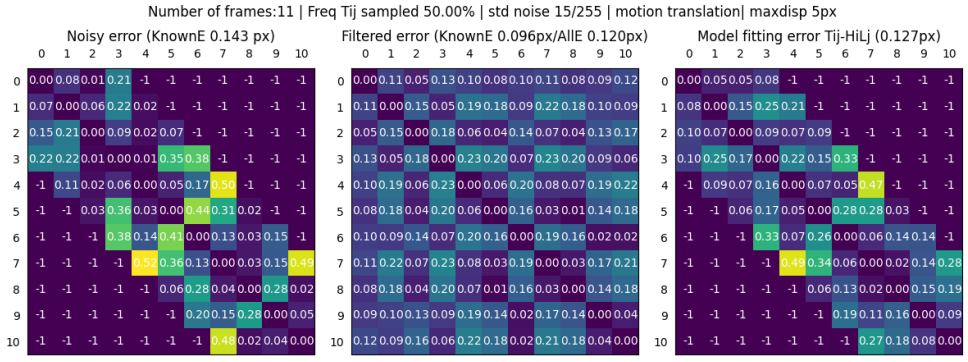


Figure A.3: Mutli frame image registration. In this example, only the 4 nearest frames are samples. **Left:** Geometric error on the sampled edges. **Middle:** Pairwise geometric error on the burst. The method aligns frames that were not aligned directly. **Right:** Geometric model fitting error.

Appendix B

Résumé Long en Français

B.1 Modèle Parcimonieux Non-Local Différentiable

La communauté du traitement d'image s'est longtemps concentrée sur la conception de modèles façonnés à la main pour les images naturelles afin de résoudre des problèmes inverses, menant, par exemple, aux approches basées sur les opérateurs différentiels, la variation totale ou la parcimonie des ondelettes. Plus récemment, les paradigmes de restauration d'image se sont orientés vers des approches pilotées par les données. Par exemple, les moyennes non-locales exploitent les autosimilarités, et de nombreuses approches réussies se sont appuyées sur des méthodes non supervisées telles que les modèles parcimonieux appris, les mélanges d'échelles gaussiennes, ou les champs d'experts. Des modèles plus puissants comme le BM3D ont également été obtenus en combinant plusieurs priors, en particulier les autosimilarités et les représentations parcimonieuses.
Ces méthodes sont maintenant souvent surpassées par les modèles d'apprentissage profond, capables d'exploiter des paires d'images corrompues/nettes pour l'apprentissage supervisé, dans des tâches telles que le débruitage, le dématriçage, l'augmentation d'échelle ou la suppression d'artefacts. Cependant, elles souffrent également d'un manque d'interprétabilité et du besoin d'apprendre un grand nombre de paramètres. Améliorer ces deux aspects est l'une des motivations clés de cet article. Notre objectif est de concevoir des algorithmes qui comblent le fossé de performance entre les approches antérieures, efficaces en termes de paramètres et interprétables, et les modèles profonds actuels.
Plus précisément, nous proposons une relaxation différentiable du modèle parcimonieux non-local LSSC. La relaxation nous permet d'obtenir des modèles pouvant être entraînés de bout en bout, et qui admettent une interprétation simple en termes de codage parcimonieux conjoint de patches similaires. Le principe de l'entraînement de bout en bout pour le codage parcimonieux a été introduit précédemment, et plus tard combiné pour la super-résolution avec des variantes de l'algorithme LISTA. Une variante basée sur le codage parcimonieux convolutif a ensuite été proposée pour le débruitage d'image, et une autre basée sur l'algorithme K-SVD a été introduite.
En contraste, notre contribution principale est d'étendre l'idée des algorithmes différentiables aux modèles parcimonieux structurés, qui est un concept clé derrière les approches LSSC, CSR, et BM3D. À notre connaissance, c'est la première fois que les modèles parcimonieux non-locaux sont démontrés efficaces dans un cadre d'apprentissage supervisé. Comme argumenté précédemment, combler les écarts entre les priors d'image classiques réussis au sein des cadres d'apprentissage pro-

fond est la clé pour surmonter les limitations des modèles à l'état de l'art actuels. Un fait marquant est notamment la performance des modèles résultants étant donné leur faible nombre de paramètres.
Par exemple, notre méthode pour le débruitage d'image est comparable à la baseline d'apprentissage profond DnCNN avec 8 fois moins de paramètres, surpasse significativement la variante en couleur CDnCNN avec 6 fois moins de paramètres, et atteint des résultats à l'état de l'art pour le débruitage aveugle et le déblocage jpeg. Pour ces deux dernières tâches, s'appuyer sur un modèle interprétable est important ; la plupart des paramètres sont consacrés à la reconstruction d'image et peuvent être partagés par des modèles dédiés à différents niveaux de bruit. Seul un petit sous-ensemble de paramètres peut être considéré comme des paramètres de régularisation, et peut être rendu dépendant du bruit, supprimant ainsi le fardeau de l'entraînement de plusieurs grands modèles indépendants pour chaque niveau de bruit. Pour le dématriçage d'image, nous obtenons des résultats similaires à l'approche à l'état de l'art RNAN, tout en réduisant le nombre de paramètres de 76 fois.

B.2 Un Cadre pour la Conception de Priors Entraînables

Malgré les succès incontestables de l'apprentissage profond dans des domaines aussi variés que le traitement d'images et la reconnaissance, le traitement du langage naturel, la parole ou la bioinformatique, les réseaux de neurones feed-forward sont souvent critiqués pour être des "boîtes noires" qui, à part peut-être pour leurs couches de classification ou de régression supérieures, sont difficiles ou même impossibles à interpréter. Dans les applications d'imagerie, par exemple, les opérations élémentaires consistent généralement en des convolutions et des non-linéarités ponctuelles, avec de nombreux paramètres ajustés par rétropropagation, et aucune interprétation fonctionnelle évidente.
Dans cet article, nous considérons à la place des architectures de réseau explicitement dérivées d'un algorithme d'optimisation, et donc interprétables d'un point de vue fonctionnel. La première instance de cette approche dont nous avons connaissance est LISTA, qui fournit une approximation rapide du codage parcimonieux. Cependant, nous ne nous contentons pas de concevoir une architecture qui fournit une approximation rapide à un problème d'optimisation donné, mais nous voulons également apprendre une représentation des données pertinente pour la tâche correspondante. Cela donne lieu à un paradigme d'apprentissage automatique inhabituel, où l'on apprend les paramètres d'une fonction objective paramétrique utilisée pour représenter les données, tout en concevant un algorithme d'optimisation pour la minimiser efficacement.
Même si l'interprétabilité n'est pas toujours nécessaire pour obtenir une bonne prédiction, ce point de vue, parfois appelé déroulement d'algorithme, s'est avéré efficace pour résoudre des problèmes d'imagerie inverse, fournissant des modèles efficaces et économes en paramètres. Cette approche permet l'utilisation de priors spécifiques au domaine au sein de modèles profonds entraînables, conduisant à un grand nombre d'applications telles que l'imagerie compressive, le dématriçage, le débruitage et la super-résolution.
Cependant, les approches existantes sont souvent limitées à des priors d'image simples tels que la parcimonie induite par la norme l_1 , ou des fonctions de régularisation différentiables, et un cadre algorithmique général pour combiner des fonctions de régularisation complexes, éventuellement non lisses, manque encore. Notre article aborde cette question et est capable de tirer parti d'une large classe

de priors d'image tels que la variation totale, la norme l_1 , le codage parcimonieux structuré, ou la régularisation laplacienne, où des problèmes d'optimisation locaux interagissent entre eux. L'interaction peut être locale parmi les voisins directs sur une grille d'image, ou non locale, capturant par exemple des similarités entre des patches d'image spatialement distants. Dans ce contexte, nous adoptons un point de vue plus général et flexible que le paradigme d'optimisation convexe standard, et considérons des formulations pour représenter les données basées sur des jeux non coopératifs potentiellement impliquant des termes non lisses, qui sont abordés en utilisant la technique de régularisation de Moreau-Yosida. Le déroulement de l'algorithme d'optimisation résultant se traduit par une architecture de réseau qui peut être entraînée de bout en bout et capturer n'importe quelle combinaison des priors spécifiques au domaine mentionnés ci-dessus. Cette approche inclut et améliore certains modèles de codage parcimonieux entraînables basés sur la norme l_1 par exemple. Plus important peut-être, elle peut être utilisée pour construire plusieurs nouveaux priors d'image intéressants : en particulier, nous montrons qu'une variante entraînable de la variation totale et sa variante non locale basée sur des auto-similarités sont compétitives avec l'état de l'art dans les tâches d'imagerie, malgré l'utilisation jusqu'à 50 fois moins de paramètres, avec des gains correspondants en vitesse. Nous démontrons l'efficacité et la flexibilité de notre approche sur plusieurs tâches d'imagerie, à savoir le débruitage, la reconstruction compressée de l'IRMf et l'appariement stéréo.
En résumé, nos contributions sont les suivantes : Premièrement, nous fournissons un nouveau cadre pour construire des variantes entraînables d'une large classe de priors d'image spécifiques au domaine. Deuxièmement, nous montrons que plusieurs de ces priors égalent ou surpassent même les techniques existantes qui utilisent un nombre beaucoup plus important de paramètres et de données d'entraînement. Enfin, nous présentons un ensemble d'astuces pratiques pour rendre les couches guidées par l'optimisation faciles à entraîner.

B.3 Super-Résolution à partir de Séquences d'Images Brutes

Le problème de la reconstruction d'images haute résolution (HR) à partir d'images de basse résolution (LR) se décline en plusieurs variantes, qui peuvent différer considérablement les unes des autres tant dans les détails techniques que dans les objectifs globaux. Lorsqu'une seule image LR est disponible, le problème inverse correspondant est fortement mal posé, nécessitant des hypothèses très fortes sur le type d'image considéré. Pour les images naturelles, les méthodes basées sur les données et utilisant des réseaux de neurones convolutionnels (CNN) se sont avérées très efficaces. Les réseaux antagonistes génératifs (GAN) ont également été utilisés pour synthétiser des images HR impressionnantes qui peuvent cependant contenir des détails haute fréquence "hallucinés".
Dans le vrai contexte de la super-résolution, où plusieurs cadres LR sont disponibles, les détails HR sont présents dans les données, mais ils sont répartis parmi de multiples images désalignées, avec des défis techniques tels que la récupération de l'enregistrement sub-pixel, mais aussi la promesse de récupérer des informations véridiques dans des applications allant de la photographie amateur à l'astronomie, l'imagerie biologique et médicale, l'imagerie microscopique et la télédétection.
Les vidéos sont bien sûr une source riche de multiples images étroitement liées de la même scène, avec plusieurs approches récentes de la super-résolution dans ce domaine, combinant souvent des priors basés sur les données des CNN avec

des auto-similarités entre les cadres. Cependant, la plupart des vidéos numériques sont produites par un pipeline complexe mappant les données brutes du capteur à des cadres éventuellement compressés et de résolution inférieure, entraînant une perte de détails haute fréquence et un bruit spatiallement corrélé qui peut être très difficile à inverser. Avec la capacité des smartphones modernes et des caméras de niveau amateur avancé d'enregistrer des séquences d'images brutes, d'autre part, il y a une nouvelle opportunité de restaurer les cadres correspondants avant que le processeur de signal d'image (ISP) de la caméra ne produise des dommages irrémédiables. C'est le problème abordé dans cette présentation, et il est difficile pour plusieurs raisons : (i) les images contiennent généralement des mouvements inconnus dus au tremblement de la main, rendant l'alignement sub-pixel difficile ; (ii) la conversion des données brutes du capteur bruyantes en images couleur complètes est en soi un problème difficile connu sous le nom de dématriçage ; et (iii) les priors d'image efficaces sont souvent basés sur les données, nécessitant donc une procédure d'estimation différentiable pour un apprentissage de bout en bout.
Dans cet article, nous abordons conjointement ces problèmes et proposons une nouvelle approche qui conserve l'interprétabilité des formulations classiques de problèmes inverses tout en permettant un apprentissage de bout en bout des paramètres des modèles. Cela peut être vu comme un pont entre le "vieux monde" du traitement du signal et le "nouveau monde courageux" des boîtes noires pilotées par les données, sans sacrifier l'interprétabilité : D'une part, nous abordons un problème inverse avec une procédure d'optimisation basée sur un modèle alternant les étapes d'estimation du mouvement et de l'image HR, en s'appuyant directement sur des travaux classiques des années 1980 et 1990. D'autre part, nous exploitons pleinement la technologie moderne sous la forme d'un prior "plug-and-play" qui mélange élégamment les réseaux neuronaux profonds avec des approches variationnelles. À son tour, le déroulement de la procédure d'optimisation nous permet d'apprendre les paramètres du modèle de bout en bout en utilisant des données d'entraînement avec des mouvements synthétiques.
Puisque l'aliasing produit des artefacts de basse fréquence associés à des composants haute fréquence sous-échantillonnés du signal original, il est généralement considéré comme un problème, motivant les fabricants d'appareils photo à ajouter des filtres anti-aliasing (optiques) devant le capteur. Pourtant, les images aliénées portent des informations haute fréquence, qui peuvent être récupérées à partir de plusieurs mesures décalées. De manière surprenante, l'aliasing est donc un allié dans le contexte de la super-résolution, un fait déjà noté dans des références antérieures. Comme le montre le reste de cette présentation, notre approche de la super-résolution de séquences d'images brutes exploite également cette idée et atteint un nouvel état de l'art sur plusieurs benchmarks standards qui utilisent un mouvement synthétique pour la vérité terrain. Elle donne également d'excellents résultats qualitatifs sur des données réelles obtenues avec des smartphones et des caméras de niveau amateur avancé. Intéressamment, comme illustré, notre méthode s'est avérée étonnamment robuste au bruit compte tenu du cadre particulièrement difficile de la super-résolution d'images brutes, qui implique simultanément le débruitage aveugle, le dématriçage, l'enregistrement et l'augmentation d'échelle.
Résumé des contributions :
À notre connaissance, nous proposons la première architecture basée sur un modèle, apprenable de bout en bout pour l'alignement d'images conjoint et la super-résolution à partir de séquences d'images brutes. Nous introduisons un nouveau module d'enregistrement d'image différentiable qui peut être appliqué à des images de différentes résolutions, est facilement intégrable dans des architec-

tures neuronales, et peut trouver d'autres utilisations au-delà de la super-résolution. Nous montrons que notre approche donne d'excellents résultats à la fois sur de vraies séquences d'images (avec jusqu'à × 4 ×4 d'augmentation d'échelle pour les images brutes) et sur des synthétiques (jusqu'à × 16 ×16 pour les images RGB).

B.4 HDR et Super-Résolution Conjoints à partir de Séquences Bracketées Brutes

Les facteurs clés limitant le niveau de détail des photographies capturées par les appareils photo numériques sont leur résolution spatiale et leur gamme dynamique : une haute résolution est nécessaire pour zoomer sur de petites régions de l'image, et une haute gamme dynamique est nécessaire pour révéler les détails cachés dans les zones sombres (par exemple, les ombres) et éviter les artefacts de couleur dus à la saturation dans les zones lumineuses (par exemple, les reflets). Pour une taille de capteur donnée, une résolution plus élevée signifie également une taille de pixel plus petite, avec moins de lumière atteignant chaque photorécepteur, résultant en une gamme dynamique plus faible et un bruit accru dans les régions sombres, un effet exacerbé dans les smartphones en raison de leur petite taille de capteur. Il est naturel, et désormais assez courant, d'utiliser plusieurs photographies pour reconstruire une image avec une résolution spatiale plus élevée, un processus connu sous le nom de super-résolution (ou SR en abrégé dans cette présentation, voir par exemple) ou de gamme dynamique, un processus connu sous le nom d'imagerie à haute gamme dynamique (ou HDR) (voir par exemple).
Nous proposons dans cet article une nouvelle méthode pour l'imagerie SR et HDR conjointe à partir de séquences d'images brutes présentant une gamme de différentes expositions qui peuvent désormais être capturées par la plupart des smartphones et des caméras de gamme moyenne (Figure 1). Un défi majeur relevé par notre algorithme est l'alignement automatique avec une précision sub-pixel des éléments de la séquence nécessaire pour compenser le bougé de la caméra et éventuellement le mouvement (modéré) de l'objet, malgré les variations de saturation et de rapport signal/bruit dues aux différentes expositions utilisées dans la séquence. Parmi les autres difficultés notables figurent les forts contrastes et niveaux de bruit rencontrés dans les scènes nocturnes par exemple, où une photo peut présenter à la fois des régions très sombres et bruyantes et des régions saturées près des sources de lumière, ainsi que le fait qu'une caméra numérique ne capture qu'un canal de couleur à chaque pixel selon le tableau de filtres de couleur correspondant (ou CFA, souvent un motif Bayer). Malgré ce dernier défi, il semble désormais clair qu'il est préférable de travailler directement avec les données d'image brutes plutôt qu'avec les images sRGB produites par le processeur de signal d'image (ou ISP) de la caméra puisque leur construction implique plusieurs étapes, y compris l'équilibrage des blancs, le débruitage, le dématriçage, la correction gamma, la compression du contenu de chaque canal de couleur à 8 bits, etc., qui entraînent une perte inévitable d'informations dans les hautes fréquences spatiales et la gamme dynamique.
L'approche proposée dans le reste de cette présentation étend l'algorithme de super-résolution multi-cadres à la reconstruction conjointe d'images HDR et de super-résolution à partir de séquences d'images brutes. Ses principales caractéristiques peuvent être résumées comme suit :
Notre méthode utilise un modèle physiquement précis de formation d'image qui prend en compte les transformations successives appliquées à l'image d'irradiance analogique originale, y compris la quantification du signal, le bruit, l'exposition

et la quantisation spatiale. Nous combinons un algorithme d'optimisation itératif pour résoudre le problème inverse correspondant avec une représentation d'image apprise pour un alignement robuste et un prior d'image naturelle appris. C'est la première nouveauté technique principale de notre article, nous permettant d'aborder la reconstruction conjointe d'images couleur à haute résolution et à haute gamme dynamique à partir de séquences photographiques brutes capturées par un appareil photo portable avec bracketing d'exposition. L'algorithme proposé est rapide, avec des exigences de mémoire faibles par rapport aux approches basées sur l'apprentissage pour la restauration d'image, et des caractéristiques qui sont apprises de bout en bout à partir de données synthétiques mais réalistes, générées à nouveau à l'aide de notre modèle de formation d'image. Nous introduisons une méthode d'alignement d'image pour compenser le bougé de la caméra qui est robuste aux mouvements d'objets (modérés) et une technique de fusion d'images qui est elle-même tolérante aux erreurs d'alignement. Ensemble, ils forment la deuxième nouveauté technique principale de notre article, et ils sont des facteurs clés dans la robustesse de notre algorithme dans les tâches d'imagerie SR et HDR, avec notamment une amélioration significative par rapport à la super-résolution. De nombreuses expériences démontrent l'excellente performance de l'approche proposée avec des facteurs de super-résolution allant jusqu'à ×4 sur de véritables photographies prises en extérieur avec des appareils photo portables, et une grande robustesse aux conditions de faible luminosité, au bruit, au bougé de la caméra et au mouvement modéré des objets. Ces résultats sont confirmés par des comparaisons quantitatives et qualitatives avec l'état de l'art dans les tâches d'imagerie super-résolution et HDR sur des séquences d'images synthétiques et réelles.

B.5 Alignement d'Image Dense et Reconstruction 3D à partir de Rafales d'Images

Nous abordons le problème de l'enregistrement *dense*, de l'estimation de pose et de la reconstruction 3D à partir de séquences d'images capturées par un appareil photo portable, avec de petits mouvements. Notre méthode estime les flux optiques, les poses de caméra et les cartes de profondeur qui peuvent être utilisées pour de multiples applications, y compris le débruitage, la super-résolution, l'imagerie HDR et les reconstructions 3D.

En photographie en rafale, un appareil photo capture une courte séquence d'images (par exemple, 10 images), en succession rapide (par exemple, une seconde), éventuellement avec différents réglages de l'appareil photo. Exploitant le fait que, pour les appareils photo portables, les images sont prises de points de vue légèrement différents, les rafales peuvent être utilisées pour l'amélioration des images comme le démontrent des approches récentes de l'imagerie à haute gamme dynamique avec des expositions bracketées, la photographie nocturne, le défloutage ou la super-résolution. Les rafales peuvent également être exploitées pour récupérer la structure 3D des scènes capturées, montrant qu'il est possible d'exploiter la parallaxe dans la scène pour en déduire sa structure 3D.

Une large famille de méthodes peut effectuer l'enregistrement d'images et la reconstruction 3D à partir de séquences d'images, allant de l'alignement d'images dense à la structure à partir du mouvement. Cependant, ces méthodes ne tirent généralement pas pleinement parti de la spécificité de la photographie en rafale à leur avantage et peuvent donc être sous-optimales pour la tâche. Par exemple, l'enregistrement précis est un défi pour les vraies rafales capturées dans la nature, et les méthodes typiques alignent les images par paires et indépendamment avec

un cadre de référence. Néanmoins, la qualité de l'alignement peut sérieusement affecter la qualité de l'image améliorée en créant des artefacts de fantômes ou de fermeture éclair lorsque l'algorithme d'enregistrement n'aligne pas les cadres avec une précision suffisante. La reconstruction 3D à partir de rafales est également un défi en raison des petits déplacements (c'est-à-dire, le déplacement de la caméra entre les images consécutives), et nécessite donc un alignement très fin. Cependant, les petits mouvements offrent également des opportunités : cela facilite l'appariement et permet l'utilisation d'une carte de profondeur comme représentation compacte et pratique de la scène 3D. Cet article introduit un nouvel algorithme basé sur l'optimisation spécifiquement adapté pour aligner et inférer la structure 3D à partir de rafales.

Nous proposons une nouvelle approche pour l'enregistrement dense de rafales qui tire parti du paramètre multi-images en modélisant directement la structure de la scène et les poses de la caméra. Pour ce faire, un bon choix de paramétrisation pour aligner les cadres est vital. Les homographies sont largement utilisées car elles sont à la fois simples et efficaces pour les scènes planes ou éloignées ou lorsque le mouvement consiste principalement en une rotation pure autour du centre optique, avec peu de parallaxe. Cependant, elles sont limitées aux scènes avec peu de relief 3D. Les mouvements complexes peuvent bien sûr être approximés par des transformations paramétriques simples définies sur de petits carreaux ou un flux optique, atténuant le problème de parallaxe. Mais le prix à payer est un grand nombre de paramètres à ajuster, ce qui peut affecter la robustesse. Notre méthode basée sur l'optimisation estime la pose de la caméra et la structure 3D de la scène en minimisant conjointement les erreurs de reprojection photométrique dans un cadre de référence. Les paramètres de pose sont ajustés individuellement pour chaque cadre, tandis que les paramètres de structure sont ajustés localement dans les images mais partagés entre toutes les vues. Le flux entre les cadres peut ensuite être calculé en reprojetant les points dans d'autres vues. Cette modélisation nécessite un nombre de paramètres beaucoup plus petit que les transformations paramétriques par bloc ou le flux optique tout en fournissant un niveau d'expressivité suffisant pour représenter avec précision les scènes statiques.

Comme aucun ensemble de données stéréo multi-vues existant dont nous avons connaissance ne couvre exactement nos cas d'utilisation typiques, nous validons notre approche avec des rafales synthétiques construites avec des logiciels de rendu (l'ensemble de données proposé sera rendu public). Notre modèle s'avère très polyvalent dans le contexte de rafales d'images avec de petits mouvements ; il donne des performances de pointe par rapport aux méthodes convaincantes conçues spécifiquement pour le flux optique, l'estimation de pose et l'estimation de profondeur. Nos résultats suggèrent également que, dans le cas de petits mouvements, une formulation dense est très bénéfique car de nombreuses méthodes concurrentes sont basées - au moins partiellement - sur des points clés épars. Enfin, pour valider notre approche avec des données réelles, nous démontrons des applications avec de vraies rafales capturées avec un smartphone Pixel 6 pro pour le débruitage de la photographie nocturne, la super-résolution et les reconstructions 3D denses.

Contributions : Dans le contexte de l'imagerie en rafale, nous proposons une méthode d'enregistrement multi-cadres polyvalente qui excelle dans diverses tâches. (1) notre algorithme donne des métriques d'alignement dense de pointe sur des données synthétiques : nous surpassons les méthodes basées sur l'apprentissage profond de pointe. Les flux estimés sur des données réelles peuvent être utilisés pour des tâches nécessitant des alignements fins, telles que la super-résolution en rafale. (2) Notre algorithme donne également des métriques de

pointe pour l'estimation de la pose de la caméra. (3) Notre méthode est également compétitive pour l'estimation de la profondeur avec de petits mouvements. Nous parvenons à capturer la structure des scènes 3D en exploitant uniquement des rafales avec de petits déplacements. (4) Enfin, nous présentons également un nouvel algorithme à point fixe pour inférer des cartes de profondeur dans de nouvelles positions de caméra. Nous utilisons cet algorithme pour estimer les flux optiques inversés et déformer les vues de référence sur d'autres vues, ce qui est requis par des tâches en aval telles que la super-résolution ou la photographie en basse lumière.

Bibliography

[1] B. Lecouat, J. Ponce, and J. Mairal, "Fully trainable and interpretable non-local sparse models for image restoration," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, Springer, 2020, pp. 238–254.

[2] B. Lecouat, J. Ponce, and J. Mairal, "A flexible framework for designing trainable priors with adaptive smoothing and game encoding," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 664–15 675, 2020.

[3] B. Lecouat, J. Ponce, and J. Mairal, "Lucas-kanade reloaded: End-to-end super-resolution from raw image bursts," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2370–2379.

[4] B. Lecouat, T. Eboli, J. Ponce, and J. Mairal, "High dynamic range and super-resolution from raw image bursts," *arXiv preprint arXiv:2207.14671*, 2022.

[5] B. Lecouat, Y. D. de Mont-Marin, T. Bodrito, J. Mairal, and J. Ponce, "Fine dense alignment of image bursts through camera pose and depth estimation," *arXiv preprint arXiv:2312.05190*, 2023.

[6] M. F. Land and D.-E. Nilsson, *Animal eyes*. OUP Oxford, 2012.

[7] E. Hecht, "Optics 4th edition," *Optics 4th edition by Eugene Hecht Reading*, 2001.

[8] F. Heide, "Structure-aware computational imaging," Ph.D. dissertation, University of British Columbia, 2016.

[9] S. Ray, *Applied photographic optics*. Routledge, 2002.

[10] C. Ware and I. Zaquine, *Micro et nano-physique*. 2014.

[11] S. Nayar, "Itroduction to Computer Vision," in *Monograph FPCV-0-1, First Principles of Computer Vision*, Columbia University, New York, Feb. 2022.

[12] R. P. Feynman, *QED: The strange theory of light and matter*. Princeton University Press, 2006, vol. 90.

[13] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[14] B. Ciechanowski, *Lights and shadows*, `ciechanow.ski/lights-and-shadows/`, 2020.

[15] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High dynamic range imaging: acquisition, display, and image-based lighting*. Morgan Kaufmann, 2010.

[16] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

[17] S. Nayar, "Image Formation," in *Monograph FPCV-1-2, First Principles of Computer Vision*, Columbia University, New York, Feb. 2022.

[18] J. Koenderink, A. van Doorn, B. Pinna, and R. Pepperell, "On right and wrong drawings," *Art & Perception*, vol. 4, no. 1-2, pp. 1–38, 2016.

[19] W.-S. Lai, Y. Shih, C.-K. Liang, and M.-H. Yang, "Correcting face distortion in wide-angle videos," *IEEE Transactions on Image Processing*, vol. 31, pp. 366–378, 2021.

[20] Y. Shih, W.-S. Lai, and C.-K. Liang, "Distortion-free wide-angle portraits on camera phones," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[21] O. Liba *et al.*, "Handheld mobile photography in very low light.," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 164–1, 2019.

[22] A. Hertzmann, *How does perspective work in pictures*, `aaronhertzmann.com/2022/02/28/how-does-perspective-work.html`, 2022.

[23] S. W. Hasinoff, "Variable-aperture photography," Ph.D. dissertation, 2008.

[24] T. Eboli, "Hybrid non-blind image deblurring for real scenarios," Ph.D. dissertation, Université PSL, 2021.

[25] T. Eboli, J.-M. Morel, and G. Facciolo, "Fast two-step blind optical aberration correction," in *European Conference on Computer Vision*, Springer, 2022, pp. 693–708.

[26] T. Goossens, Z. Lyu, J. Ko, G. C. Wan, J. Farrell, and B. Wandell, "Ray-transfer functions for camera simulation of 3d scenes with hidden lens design," *Optics Express*, vol. 30, no. 13, pp. 24 031–24 047, 2022.

[27] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International journal of computer vision*, vol. 98, pp. 168–186, 2012.

[28] X. Zhu and P. Milanfar, "Removing atmospheric turbulence via space-invariant deconvolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 157–170, 2012.

[29] P. Milanfar, *Super-resolution imaging*. CRC press, 2017.

[30] T. Köhler, "Multi-frame super-resolution reconstruction with applications to medical imaging," *arXiv preprint arXiv:1812.09375*, 2018.

[31] B. A. Wandell, *Foundations of vision*. Sinauer Associates, 1995.

[32] M. Delbracio, "Two problems of digital image formation: Recovering the camera point spread function and boosting stochastic renderers by auto-similarity filtering," Ph.D. dissertation, École normale supérieure de Cachan-ENS Cachan, 2013.

[33] J. E. Farrell, P. B. Catrysse, and B. A. Wandell, "Digital camera simulation," *Applied optics*, vol. 51, no. 4, A80–A90, 2012.

[34] M. Bauer, V. Volchkov, M. Hirsch, and B. Schcölkopf, "Automatic estimation of modulation transfer functions," in *2018 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2018, pp. 1–12.

[35] M. Delbracio, P. Musé, and A. Almansa, "Non-parametric sub-pixel local point spread function estimation," *Image Processing on Line*, vol. 2, pp. 8–21, 2012.

[36] M. Delbracio, A. Almansa, J.-M. Morel, and P. Musé, "Subpixel point spread function estimation from two photographs at different distances," *SIAM Journal on Imaging Sciences*, vol. 5, no. 4, pp. 1234–1260, 2012.

[37] E. Tseng *et al.*, "Differentiable compound optics and processing pipeline optimization for end-to-end camera design," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 2, pp. 1–19, 2021.

[38] C. Garnier, R. Collorec, J. Flifla, C. Mouclier, and F. Rousee, "Infrared sensor modeling for realistic thermal image synthesis," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, IEEE, vol. 6, 1999, pp. 3513–3516.

[39] J.-P. Colinge and C. A. Colinge, *Physics of semiconductor devices*. Springer Science & Business Media, 2005.

[40] M. Grundmann, *Physics of semiconductors*. Springer, 2010, vol. 11.

[41] E. Martinec, *Noise, dynamic range and bit depth in digital slrs*, `https://homes.psd.uchicago.edu/~ejmartin/pix/20d/tests/noise/`, 2008.

[42] S. W. Hasinoff, F. Durand, and W. T. Freeman, "Noise-optimal capture for high dynamic range photography," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 553–560.

[43] P. Hanji, F. Zhong, and R. K. Mantiuk, "Noise-aware merging of high dynamic range image stacks without camera calibration," in *Proceedings of the workshops of the European Conference on Computer Vision (ECCVW)*, 2020, pp. 376–391.

[44] E. Martinec, *Noise, dynamic range and bit depth in digital SLRs*, "`https://photonstophotos.net/EmilMartinec/noise.html`", 2008.

[45] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE transactions on information theory*, vol. 44, no. 6, pp. 2325–2383, 1998.

[46] R. Mantiuk, G. Krawczyk, D. Zdrojewska, R. Mantiuk, K. Myszkowski, and H.-P. Seidel, *High dynamic range imaging*. na, 2015.

[47] S. T. McHugh, *Understanding photography: master your digital camera and capture that perfect photo*. No starch press, 2018.

[48] T. Briand, "Image formation from a large sequence of raw images: Performance and accuracy," Ph.D. dissertation, Université Paris-Est, 2018.

[49] S. Mallat, *A Wavelet Tour of Signal Processing, Second Edition*. Academic Press, New York, 1999.

[50] C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949. DOI: `10.1109/jrproc.1949.232969`. [Online]. Available: `https://doi.org/10.1109/jrproc.1949.232969`.

[51] S. Nayar, "Image Sensing," in *Monograph FPCV-1-2, First Principles of Computer Vision*, Columbia University, New York, Feb. 2022.

[52] B. Lecouat, J. Ponce, and J. Mairal, "Lucas-kanade reloaded: End-to-end super-resolution from raw image bursts," in *Proc. International Conference on Computer Vision (ICCV)*, 2021.

[53] B. Wronski *et al.*, "Handheld multi-frame super-resolution," vol. 38, no. 4, 28:1–28:18, 2019.

[54] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Deep burst super-resolution," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9209–9218.

[55] A. Ford and A. Roberts, "Colour space conversions," *Westminster University, London*, vol. 1998, pp. 1–31, 1998.

[56] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE transactions on image processing*, vol. 6, no. 12, pp. 1646–1658, 1997.

[57] M. Delbracio, D. Kelly, M. S. Brown, and P. Milanfar, "Mobile computational photography: A tour," *Annual Review of Vision Science*, vol. 7, pp. 571–604, 2021.

[58] R. Sumner, "Processing raw images in matlab," *Department of Electrical Engineering, University of California Sata Cruz*, vol. 2, 2014.

[59] E. Reinhard, M. M. Stark, P. Shirley, and J. A. Ferwerda, "Photographic tone reproduction for digital images," vol. 21, no. 3, pp. 267–276, 2002.

[60] J. Mairal, F. Bach, J. Ponce, *et al.*, "Sparse modeling for image and vision processing," *Foundations and Trends® in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014.

[61] U. S. Kamilov, C. A. Bouman, G. T. Buzzard, and B. Wohlberg, "Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 40, no. 1, pp. 85–97, 2023.

[62] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, T. Pock, *et al.*, "An introduction to total variation for image analysis," *Theoretical foundations and numerical methods for sparse recovery*, vol. 9, no. 263-340, p. 227, 2010.

[63] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D image denoising with shape-adaptive principal component analysis," 2009.

[64] N. Parikh and S. P. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[65] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[66] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proceedings of the Global Conference on Signal and Information Processing*, 2013, pp. 945–948.

[67] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE transactions on computational imaging*, vol. 5, no. 1, pp. 52–67, 2018.

[68] F. Pedregosa, "Hyperparameter optimization with approximate gradient," in *International conference on machine learning*, PMLR, 2016, pp. 737–746.

[69] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. International Conference on Machine Learning (ICML)*, 2010.

[70] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, "Truncated back-propagation for bilevel optimization," in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 1723–1732.

[71] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, "Truncated back-propagation for bilevel optimization," 2019.

[72] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[73] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, pp. 221–255, 2004.

[74] J. Sanchez, "The inverse compositional algorithm for parametric registration," *Image Processing On Line*, 2016.

[75] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H.-P. Seidel, and H. P. Lensch, "Optimal hdr reconstruction with linear digital cameras," in *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 215–222.

[76] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Shechtman, "Robust patch-based hdr reconstruction of dynamic scenes.," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 203–1, 2012.

[77] J. Hu, O. Gallo, K. Pulli, and X. Sun, "Hdr deghosting: How to deal with saturation?" In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1163–1170.

[78] O. Gallo, A. Troccoli, J. Hu, K. Pulli, and J. Kautz, "Locally non-rigid registration for mobile hdr photography," in *Proceedings of the IEEE conference on computer vision and pattern recognition Workshops*, 2015, pp. 49–56.

[79] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE transactions on image processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[80] B. Wronski *et al.*, "Handheld multi-frame super-resolution," *ACM Transactions on Graphics (ToG)*, vol. 38, no. 4, pp. 1–18, 2019.

[81] S. W. Hasinoff *et al.*, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–12, 2016.

[82] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 12, no. 7, pp. 629–639, 1990.

[83] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[84] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[85] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[86] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[87] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[88] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[89] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2012.

[90] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[91] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration.," in *Proc. International Conference on Computer Vision (ICCV)*, 2009.

[92] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[93] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[94] T. Plötz and S. Roth, "Neural nearest neighbors networks," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[95] 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, no. 7, pp. 3142–3155, 2017.

[96] F. Kokkinos and S. Lefkimmiatis, "Iterative joint image demosaicking and denoising using a residual denoising network," *IEEE Transactions on Image Processing*, 2019.

[97] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[98] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," in *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[99] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 38, no. 2, pp. 295–307, 2016.

[100] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[101] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2011.

[102] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[103] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[104] J. Liu, X. Chen, Z. Wang, and W. Yin, "Alista: Analytic weights are as good as learned weights in lista," *Proc. International Conference on Learning Representations (ICLR)*, 2019.

[105] D. Simon and M. Elad, "Rethinking the CSC model for natural images," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[106] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[107] M. Scetbon, M. Elad, and P. Milanfar, "Deep k-svd denoising," *arXiv preprint arXiv:1909.13164*, 2019.

[108] J. Sun, H. Li, Z. Xu, *et al.*, "Deep admm-net for compressive sensing mri," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2016.

[109] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Global Conference on Signal and Information Processing*, IEEE, 2013, pp. 945–948.

[110] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2777–2824, 2011.

[111] B. A. Olshausen and D. J. Field., "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[112] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[113] S. Lefkimmiatis, "Universal denoising networks: A novel cnn architecture for image denoising," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[114] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4608–4622, 2018.

[115] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[116] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Transactions on Pattern Analysis and Nachine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2016.

[117] C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, and M. Prato, "Deep unfolding of a proximal interior point method for image restoration," *Inverse Problems*, 2019.

[118] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2011.

[119] A. K. Fletcher, P. Pandit, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[120] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[121] X. Liu, M. Tanaka, and M. Okutomi, "Single-image noise level estimation for blind denoising," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5226–5237, 2013.

[122] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.

[123] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research (JMLR)*, vol. 11, no. Jan, pp. 19–60, 2010.

[124] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007.

[125] K. Ma *et al.*, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2016.

[126] F. Kokkinos and S. Lefkimmiatis, "Deep image demosaicking using a cascade of convolutional residual denoising networks," in *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[127] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–12, 2016.

[128] D. Martin, C. Fowlkes, D. Tal, J. Malik, *et al.*, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," 2001.

[129] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[130] K. Yu, C. Dong, C. C. Loy, and X. Tang, "Deep convolution networks for compression artifacts reduction," 2015.

[131] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2013.

[132] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Asilomar Conference on Signals, Systems & Computers*, 2003.

[133] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[134] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. International Conference on Machine Learning (ICML)*, 2008.

[135] A. v. d. Oord *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[136] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of dna-and rna-binding proteins by deep learning," *Nature biotechnology*, vol. 33, no. 8, pp. 831–838, 2015.

[137] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *arXiv preprint arXiv:1912.10557*, 2019.

[138] H. Nikaidô, K. Isoda, *et al.*, "Note on non-cooperative convex games," *Pacific Journal of Mathematics*, vol. 5, no. Suppl. 1, pp. 807–815, 1955.

[139] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms I: Fundamentals*. Springer science & business media, 2013, vol. 305.

[140] Y.-L. Yu, "Better approximation and faster algorithm using the proximal average," in *Adv. in Neural Information Processing Systems (NIPS)*, 2013.

[141] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE transactions on image processing*, vol. 21, no. 8, pp. 3467–3478, 2012.

[142] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[143] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2017.

[144] D. Belanger, B. Yang, and A. McCallum, "End-to-end learning for structured prediction energy networks," in *Proc. International Conference on Machine Learning (ICML)*, 2017.

[145] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *Proc. International Conference on Machine Learning (ICML)*, 2015.

[146] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. in Neural Information Processing Systems (NeurIPS)*, 2019.

[147] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. International Conference on Computer Vision (ICCV)*, 1998.

[148] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2009.

[149] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[150] B. Turlach, W. Venables, and S. Wright, "Simultaneous variable selection," *Technometrics*, vol. 47, no. 3, p. 349, 2005.

[151] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989.

[152] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[153] A. Juditsky, A. Nemirovski, and C. Tauvel, "Solving variational inequalities with stochastic mirror-prox algorithm," *Stochastic Systems*, vol. 1, no. 1, pp. 17–58, 2011.

[154] P. Mertikopoulos, B. Lecouat, H. Zenati, C.-S. Foo, V. Chandrasekhar, and G. Piliouras, "Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[155] G. Korpelevich, "The extragradient method for finding saddle points and other problems," *Matecon*, vol. 12, pp. 747–756, 1976.

[156] C. Lemaréchal and C. Sagastizábal, "Practical aspects of the moreau–yosida regularization: Theoretical preliminaries," *SIAM Journal on Optimization*, vol. 7, no. 2, pp. 367–385, 1997.

[157] J. J. Moreau, "Fonctions convexes duales et points proximaux dans un espace Hilbertien," *CR Acad. Sci. Paris Sér. A Math*, vol. 255, pp. 2897–2899, 1962.

[158] K. Yosida, *Functional analysis*, Springer-Verlag, Ed. 1964.

[159] A. Chambolle and J. Darbon, "On total variation minimization and surface evolution using parametric maximum flows," *International journal of computer vision*, vol. 84, no. 3, p. 288, 2009.

[160] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[161] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[162] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[163] S. Ravishankar and Y. Bresler, "Mr image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE transactions on medical imaging*, vol. 30, no. 5, pp. 1028–1041, 2010.

[164] J. Yang, Y. Zhang, and W. Yin, "A fast alternating direction method for tvl1-l2 signal reconstruction from partial fourier data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 288–297, 2010.

[165] X. Qu, Y. Hou, F. Lam, D. Guo, J. Zhong, and Z. Chen, "Magnetic resonance image reconstruction from undersampled measurements using a patch-based nonlocal operator," *Medical image analysis*, vol. 18, no. 6, pp. 843–856, 2014.

[166] E. M. Eksioglu, "Decoupled algorithm for mri reconstruction using nonlocal block matching model: Bm3d-mri," *Journal of Mathematical Imaging and Vision*, vol. 56, no. 3, pp. 430–440, 2016.

[167] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 5410–5418.

[168] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[169] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, arXiv:1512.02134, 2016. [Online]. Available: `http://lmb.informatik.uni-freiburg.de/Publications/2016/MIFDB16`.

[170] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3354–3361.

[171] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 26, no. 6, pp. 508–517, 1978.

[172] J. Yang and T. Huang, "Image super-resolution: Historical overview and future challenges," in *Super-resolution imaging*, P. Milanfar, Ed., CRC Press, 2011.

[173] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[174] X. Wang *et al.*, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *European Conference on Computer Vision (ECCV) workshop on Perceptual Image Restoration and Manipulation*, 2018.

[175] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

[176] P. Milanfar, *Super-resolution imaging*. CRC Press, 2011.

[177] R. Tsai and T. Huang, "Multiframe image restoration and registration," in *Advances in Computer Vision and Image Processing*, JAI Press Inc., 1984, pp. 317–339.

[178] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, "Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3224–3232.

[179] W. Li, X. Tao, T. Guo, L. Qi, J. Lu, and J. Jia, "Mucan: Multi-correspondence aggregation network for video super-resolution," in *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[180] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[181] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Deep burst super-resolution," *arXiv preprint arXiv:2101.10997*, 2021.

[182] R. Kimmel, "Demosaicing: Image reconstruction from color ccd samples," *IEEE Transactions on image processing*, vol. 8, no. 9, pp. 1221–1228, 1999.

[183] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of Imaging Understanding Workshop*, 1981.

[184] R. C. Hardie, K. J. Barnard, and E. E. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE transactions on Image Processing*, vol. 6, no. 12, pp. 1621–1633, 1997.

[185] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play admm for image restoration: Fixed-point convergence and applications," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2016.

[186] K. Zhang, L. V. Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[187] P. Vandewalle, L. Sbaiz, J. Vandewalle, and M. Vetterli, "Aliasing is good for you: Joint registration and reconstruction for super-resolution," Tech. Rep., 2006.

[188] R. Hardie, "A fast image super-resolution algorithm using an adaptive wiener filter," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2953–2964, 2007.

[189] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on image processing*, vol. 16, no. 2, pp. 349–366, 2007.

[190] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[191] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical models and image processing*, vol. 53, no. 3, pp. 231–239, 1991.

[192] C. Bercea, A. Maier, and T. Köhler, "Confidence-aware levenberg-marquardt optimization for joint motion estimation and super-resolution," in *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, pp. 1136–1140.

[193] P. Vandewalle, "Super-resolution from unregistered aliased images," EPFL, Tech. Rep., 2006.

[194] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[195] A. B. Molini, D. Valsesia, G. Fracastoro, and E. Magli, "Deepsum: Deep neural network for super-resolution of unregistered multitemporal images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3644–3656, 2019.

[196] M. Deudon *et al.*, "Highres-net: Recursive fusion for multi-frame super-resolution of satellite imagery," *arXiv preprint arXiv:2002.06460*, 2020.

[197] A. Lugmayr, M. Danelljan, and R. Timofte, "Ntire 2020 challenge on real-world image super-resolution: Methods and results," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[198] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[199] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," *Adv. in Neural Information Processing Systems (NIPS)*, 2009.

[200] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, 2006, Second edition.

[201] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE transactions on Image Processing*, vol. 4, no. 7, pp. 932–946, 1995.

[202] Y. He, K.-H. Yap, L. Chen, and L.-P. Chau, "A nonlinear least square technique for simultaneous image registration and super-resolution," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2830–2841, 2007.

[203] E. K. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," *Proc. International Conference on Machine Learning (ICML)*, 2019.

[204] A. Ignatov, L. Van Gool, and R. Timofte, "Replacing mobile camera isp with a single deep learning model," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

[205] G. Bhat, M. Danelljan, and R. Timofte, "Ntire 2021 challenge on burst super-resolution: Methods and results," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 613–626.

[206] T. Ehret, A. Davy, P. Arias, and G. Facciolo, "Joint demosaicking and denoising by fine-tuning of bursts of raw images," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[207] Z. Xia, F. Perazzi, M. Gharbi, K. Sunkavalli, and A. Chakrabarti, "Basis prediction networks for effective burst denoising with large kernels," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[208] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. European Conference on Computer Vision (ECCV)*, 2018.

[209] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," ACM, 1997, pp. 369–378.

[210] S. Mann and R. W. Picard, "On being 'undigital' with digital cameras: Extending dynamic range by combining differently exposed pictures," in *Proceedings of Is&T*, 1995, pp. 442–448.

[211] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H. Seidel, and H. P. A. Lensch, "Optimal HDR reconstruction with linear digital cameras," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2010, pp. 215–222.

[212] S. W. Hasinoff, F. Durand, and W. T. Freeman, "Noise-optimal capture for high dynamic range photography," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 553–560.

[213] C. Aguerrebere, J. Delon, Y. Gousseau, and P. Musé, "Best algorithms for HDR image generation. A study of performance bounds," *SIAM Journal on Imaging Science*, vol. 7, no. 1, pp. 1–34, 2014.

[214] O. Gallo, M. Tico, R. Manduchi, N. Gelfand, and K. Pulli, "Metering for exposure stacks," *Computer Graphics Forum*, vol. 31, no. 2, pp. 479–488, 2012.

[215] H. Zimmer, A. Bruhn, and J. Weickert, "Freehand HDR imaging of moving scenes with simultaneous resolution enhancement," *Computer Graphics Forum*, vol. 30, no. 2, pp. 405–414, 2011.

[216] O. Gallo, A. J. Troccoli, J. Hu, K. Pulli, and J. Kautz, "Locally non-rigid registration for mobile HDR photography," in *(CVPRW)*, IEEE Computer Society, 2015, pp. 48–55.

[217] K. Ma, H. Li, H. Yong, Z. Wang, D. Meng, and L. Zhang, "Robust multi-exposure image fusion: A structural patch decomposition approach," *IEEE Transactions on Image Processing (TIP)*, vol. 26, no. 5, pp. 2519–2532, 2017.

[218] P. Sen, N. K. Kalantari, M. Yaesoubi, S. Darabi, D. B. Goldman, and E. Shechtman, "Robust patch-based HDR reconstruction of dynamic scenes," vol. 31, no. 6, 203:1–203:11, 2012.

[219] O. T. Tursun, A. O. Akyüz, A. Erdem, and E. Erdem, "An objective deghosting quality metric for HDR images," *Computer Graphics Forum*, vol. 35, no. 2, pp. 139–152, 2016.

[220] S. W. Hasinoff *et al.*, "Burst photography for high dynamic range and low-light imaging on mobile cameras," vol. 35, no. 6, 192:1–192:12, 2016.

[221] M. Ernst and B. Wronski, *HDR+ with bracketing on pixel phones*, "`https://ai.googleblog.com/2021/04/hdr-with-bracketing-on-pixel-phones.html`", 2021.

[222] O. Liba *et al.*, "Handheld mobile photography in very low light," vol. 38, no. 6, 164:1–164:16, 2019.

[223] S. K. Nayar and T. Mitsunaga, "High dynamic range imaging: Spatially varying pixel exposures," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, vol. 1, 2000, pp. 472–479.

[224] A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia, "Convolutional sparse coding for high dynamic range imaging," in *Computer Graphics Forum*, Wiley Online Library, vol. 35, 2016, pp. 153–163.

[225] J. N. Martel, L. K. Mueller, S. J. Carey, P. Dudek, and G. Wetzstein, "Neural sensors: Learning pixel exposures for hdr imaging and video compressive sensing with programmable sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 7, pp. 1642–1653, 2020.

[226] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," vol. 36, no. 4, 144:1–144:12, 2017.

[227] S. Wu, J. Xu, Y. Tai, and C. Tang, "Deep high dynamic range imaging with large foreground motions," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 120–135.

[228] Q. Yan *et al.*, "Dual-attention-guided network for ghost-free high dynamic range imaging," *International Journal of Computer Vision (IJCV)*, pp. 1–19, 2021.

[229] Q. Yan *et al.*, "Deep HDR imaging via A non-local network," *IEEE Transactions on Image Processing (TIP)*, vol. 29, pp. 4308–4322, 2020.

[230] Y. Niu, J. Wu, W. Liu, W. Guo, and R. W. H. Lau, "HDR-GAN: HDR image reconstruction from multi-exposed LDR images with large motions," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 3885–3896, 2021.

[231] E. Pérez-Pellitero, S. Catley-Chandar, A. Leonardis, and R. Timofte, "NTIRE 2021 challenge on high dynamic range imaging: Dataset, methods and results," in *CVPR Workshops*, 2021, pp. 691–700.

[232] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, "HDR image reconstruction from a single exposure using deep cnns," vol. 36, no. 6, 178:1–178:15, 2017.

[233] Y. Endo, Y. Kanamori, and J. Mitani, "Deep reverse tone mapping," vol. 36, no. 6, 177:1–177:10, 2017.

[234] M. S. Santos, T. I. Ren, and N. K. Kalantari, "Single image HDR reconstruction using a CNN with masked features and perceptual loss," vol. 39, no. 4, p. 80, 2020.

[235] Y. Liu *et al.*, "Single-image HDR reconstruction by learning to reverse the camera pipeline," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1648–1657.

[236] R. Dahl, M. Norouzi, and J. Shlens, "Pixel recursive super resolution," in *Proc. International Conference on Computer Vision (ICCV)*, 2017.

[237] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "PULSE: Self-supervised photo upsampling via latent space exploration of generative models," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[238] S. Farsiu, M. Elad, and P. Milanfar, "Multiframe demosaicing and super-resolution of color images," *IEEE Transactions on Image Processing (TIP)*, vol. 15, no. 1, pp. 141–159, 2006.

[239] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing (TIP)*, vol. 16, no. 2, pp. 349–366, 2007.

[240] G. Bhat, M. Danelljan, F. Yu, L. V. Gool, and R. Timofte, "Deep reparametrization of multi-frame super-resolution and denoising," pp. 2460–2470, 2021.

[241] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1981, pp. 674–679.

[242] Z. Luo *et al.*, "Ebsr: Feature enhanced burst super-resolution with deformable alignment," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops*, 2021, pp. 471–478.

[243] J. Choi, M. K. Park, and M. G. Kang, "High dynamic range image reconstruction with spatial resolution enhancement," *Computer Journal*, vol. 52, no. 1, pp. 114–125, 2009.

[244] B. K. Gunturk and M. Gevrekci, "High-resolution image reconstruction from multiple differently exposed images," *IEEE Signal Processing Letters*, vol. 13, no. 4, pp. 197–200, 2006.

[245] A. A. Rad, L. Meylan, P. Vandewalle, and S. Süsstrunk, "Multidimensional image enhancement from a set of unregistered and differently exposed images," in *Computational Imaging*, ser. SPIE Proceedings, vol. 6498, SPIE, 2007, p. 649 808.

[246] P. Vandewalle, S. Süsstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to super-resolution," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, 2006.

[247] Y. Traonmilin and C. Aguerrebere, "Simultaneous high dynamic range and superresolution imaging without regularization," *SIAM Journal on Imaging Science*, vol. 7, no. 3, pp. 1624–1644, 2014.

[248] S. Vasu, A. Shenoi, and A. N. Rajagopalan, "Joint HDR and super-resolution imaging in motion blur," in *Proceedings of the International Conference on Image Processing (ICIP)*, 2018, pp. 2885–2889.

[249] S. Y. Kim, J. Oh, and M. Kim, "Deep SR-ITM: joint learning of super-resolution and inverse tone-mapping for 4k UHD HDR applications," in *Proc. International Conference on Computer Vision (ICCV)*, 2019, pp. 3116–3125.

[250] X. Deng, Y. Zhang, M. Xu, S. Gu, and Y. Duan, "Deep coupled feedback network for joint exposure fusion and image super-resolution," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 3098–3112, 2021.

[251] R. N. Clark, *Digital camera reviews and sensor performance summary*, "`https://clarkvision.com/articles/digital.sensor.performance.summary/`", 2006.

[252] G. Luijk, *Dcraw tutorial*, "`http://guillermoluijk.com/tutorial/dcraw/index_en.htm`", 2007.

[253] A. Foi, M. Trimeche, V. Katkovnik, and K. O. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image raw-data," *IEEE Transactions on Image Processing (TIP)*, vol. 17, no. 10, pp. 1737–1754, 2008.

[254] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," vol. 26, no. 3, p. 70, 2007.

[255] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 036–11 045.

[256] T. Plötz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2750–2759.

[257] F. Heide *et al.*, "FlexISP: A flexible camera image processing framework," vol. 33, no. 6, 231:1–231:13, 2014.

[258] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Transactions on Image Processing (TIP)*, vol. 5, no. 7, pp. 932–946, 1995.

[259] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of Machine Learning Research (JMLR)*, vol. 18, pp. 1–43, 2018.

[260] P. E. Anuta, "Spatial registration of multispectral and multitemporal digital imagery using fast fourier transform techniques," *IEEE Transactions on Geoscience eletronics*, vol. 8, no. 4, pp. 353–368, 1970.

[261] G. Ward, "Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures," *Journal on Graphics, GPU, & Game Tools*, vol. 8, no. 2, pp. 17–30, 2003.

[262] C.-H. Chang, C.-N. Chou, and E. Y. Chang, "Clkn: Cascaded lucas-kanade networks for image alignment," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[263] A. Dudhane, S. W. Zamir, S. Khan, F. Khan, and M.-H. Yang, "Burst image restoration and enhancement," *arXiv preprint arXiv:2110.03680*, 2021.

[264] A. Monod, J. Delon, and T. Veit, "An analysis and implementation of the hdr+ burst denoising method," *Image Processing On Line*, vol. 11, pp. 142–169, 2021.

[265] H. S. Malvar, L. He, and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2004, pp. 485–488.

[266] T. O. Aydin, R. Mantiuk, and H. Seidel, "Extending quality metrics to full luminance range images," in *Proceedings of Human Vision and Electronic Imaging*, B. E. Rogowitz and T. N. Pappas, Eds., ser. SPIE Proceedings, vol. 6806, SPIE, 2008, 68060B.

[267] G. Eilertsen, S. Hajisharif, P. Hanji, A. Tsirikoglou, R. K. Mantiuk, and J. Unger, "How to cheat with metrics in single-image HDR reconstruction," in *Proceedings of the workshops of the International Conference on Computer Vision (ICCVW)*, 2021, pp. 3981–3990.

[268] M. Narwaria, R. K. Mantiuk, M. P. D. Silva, and P. L. Callet, "HDR-VDP-2.2: a calibrated method for objective quality prediction of high-dynamic range and standard images," *Journal on Electronic Imaging*, vol. 24, no. 1, p. 010 501, 2015.

[269] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. Srinivasan, and J. T. Barron, "Nerf in the dark: High dynamic range view synthesis from noisy raw images," *arXiv preprint arXiv:2111.13679*, 2021.

[270] M. Delbracio and G. Sapiro, "Burst deblurring: Removing camera shake through fourier burst accumulation," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*.

[271] I. Chugunov, Y. Zhang, Z. Xia, X. Zhang, J. Chen, and F. Heide, "The implicit values of a good hand shake: Handheld multi-frame neural depth refinement," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2852–2862.

[272] I. Chugunov, Y. Zhang, and F. Heide, "Shakes on a plane: Unsupervised depth estimation from unstabilized photography," *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[273] H. Ha, S. Im, J. Park, H.-G. Jeon, and I. S. Kweon, "High-quality depth from uncalibrated small motion clip," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5413–5421.

[274] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, Springer, 2003, pp. 363–370.

[275] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Proc. European Conference on Computer Vision (ECCV)*, Springer, 2020, pp. 402–419.

[276] J. Kopf, X. Rong, and J.-B. Huang, "Robust consistent video depth estimation," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1611–1621.

[277] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[278] C. Aguerrebere, M. Delbracio, A. Bartesaghi, and G. Sapiro, "Fundamental limits in multi-image alignment," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5707–5722, 2016.

[279] S. Farsiu, M. Elad, and P. Milanfar, "Constrained, globally optimal, multi-frame motion estimation," in *IEEE/SP 13th Workshop on Statistical Signal Processing, 2005*, IEEE, 2005, pp. 1396–1401.

[280] C. Aguerrebere, M. Delbracio, A. Bartesaghi, and G. Sapiro, "A practical guide to multi-image alignment," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 1927–1931.

[281] S. Im, H. Ha, G. Choe, H.-G. Jeon, K. Joo, and I. S. Kweon, "High quality structure from small motion for rolling shutter cameras," in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 837–845.

[282] J. Sánchez, "The Inverse Compositional Algorithm for Parametric Registration," *Image Processing On Line*, vol. 6, pp. 212–232, 2016, https://doi.org/10.5201/ipol.2016.153.

[283] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003, ISBN: 0521540518.

[284] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *CoRR*, vol. abs/1812.01537, 2018. arXiv: 1812.01537. [Online]. Available: http://arxiv.org/abs/1812.01537.

[285] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, 2022, ISSN: 1424-8220. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5353.

[286] C. Lei and Y.-H. Yang, "Optical flow estimation on coarse-to-fine region-trees using discrete optimization," in *Proc. International Conference on Computer Vision (ICCV)*, 2009, pp. 1562–1569. DOI: 10.1109/ICCV.2009.5459253.

[287] B. O. Community, *Blender - a 3d modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org.

[288] *Evermotion Archinteriors vol.43*, https://evermotion.org/shop/show_product/archinteriors-vol-43/12555.

[289] *Architecture topics*, https://www.youtube.com/watch?v=Gn1biEB5PbQ.

[290] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 1*. Birkhäuser Boston, 2009. DOI: 10.1007/978-0-8176-4803-9. [Online]. Available: https://doi.org/10.1007/978-0-8176-4803-9.

[291] E. Zacur, M. Bossa, and S. Olmos, "Left-invariant riemannian geodesics on spatial transformation groups," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1503–1557, 2014. DOI: 10.1137/130928352.

[292] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 3, pp. 1623–1637, 2020.

[293] G. Bradski, "The opencv library.," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

RÉSUMÉ

Cette thèse explore les méthodes hybrides pour les problèmes inverses, en se concentrant sur leur mise en œuvre pratique pour la photographie en rafale. Elle est divisée en deux parties principales.
La première partie est consacrée à l'étude de méthodes hybrides pour des applications de restauration d'images, en fournissant plusieurs outils méthodologiques. Notamment, un nouveau problème inverse appris régularisé avec un prior parcimonieux non locale est proposé, tirant parti d'une relaxation différentiable d'un optimiser du problème d'optimisation group lasso. Ensuite, un cadre fournissant des relaxations différentiables de solveurs d'optimisation convexes non lisses pour des priors d'images est étudié. Ces modèles présentent des performances comparables à celles de réseaux de neurones état de l'art plus grands, mais avec beaucoup moins de paramètres, une interprétabilité accrue, des durées d'entraînement plus courts et une plus petite quantité de données d'apprentissage.
La deuxième partie de la thèse se penche sur l'intégration de l'apprentissage automatique pour les techniques de restauration d'images multi-images, pour des applications sur des images réelles, pour des problèmes comme la super-résolution et la reconstruction HDR. La conception d'algorithmes plug-and-play (PnP) pour la photographie en rafale est explorée, avec des efforts dirigés vers la mise en œuvre pratique et l'optimisation de la mémoire pour une implémentation sur appareil mobile. Au cours de notre étude, la qualité de l'alignement des images a été identifié comme un élément bloquant. Pour contourner ce problème, nous proposons un nouvel algorithme de recalage multi-images dense, permettant également la reconstruction de scènes 3D à partir de rafales d'images avec de petits déplacements.

MOTS CLÉS

Traitement image, problèmes inverses, apprentissage, super-résolution, imagerie haute dynamique, reconstruction 3D.

ABSTRACT

This thesis explores hybrid methods for inverse problems, focusing on their practical implementation in burst photography for real-world applications. It is divided into two main parts.
The first part is dedicated to studying hybrid methods for single-image restoration applications, providing several methodological tools. Notably, a novel learned inverse problem regularized with a non-local sparse image prior is proposed, leveraging a differentiable relaxation of the group lasso. Then, a framework providing differentiable relaxations of convex non-smooth optimization solvers for classic image priors is studied. These models demonstrate comparable performance to larger neural networks but with significantly fewer parameters, increased interpretability, faster training times, and a smaller amount of training data.
The second part of the thesis delves into integrating machine learning into multi-frame image restoration techniques for real-world scenarios like burst super-resolution and HDR reconstruction. The design of plug-and-play (PnP) algorithms for burst photography is explored, with efforts directed toward practical implementation and memory optimization for mobile devices. Throughout our investigation, we have consistently identified registration quality as a prominent bottleneck. To effectively tackle this challenge, we propose a novel dense multi-frame registration algorithm, also enabling 3-D scene reconstruction from image bursts with tiny baselines.

KEYWORDS

Image processing, inverse problems, machine learning, super-resolution, high-dynamic range, 3D reconstruction.