Bayesian inference for image deblurring at a large scale

Yunshi Huang

École Doctorale Mathématiques et Sciences et Technologies de l'Information et de la Communication

UNIVERSITÉ GUSTAVE EIFFEL

PhD Thesis

presented by

Yunshi HUANG

BAYESIAN INFERENCE FOR IMAGE DEBLURRING AT A LARGE SCALE

Director	DR. EMILIE CHOUZENOUX	*Inria Saclay, France*
Co-supervisor	DR. VÍCTOR ELVIRA	*University of Edinburgh, UK*
Reviewer	DR. AURÉLIA FRAYSSE	*Université Paris Saclay, France*
Reviewer	PROF. ISMAIL BEN AYED	*ETS Montréal, Canada*
Examiner	PROF. FRANÇOIS SEPTIER	*Université Bretagne Sud, France*
Examiner/President	PROF. AMEL BENAZZA-BENYAHIA	*SUPCOM Tunis, Tunisia*

Contents

Résumé étendu

Le problème de la restauration d'une image à partir d'une version dégradée, floutée et bruitée, de celle-ci se pose dans de nombreux cadres applicatifs, tels que la microscopie, l'imagerie médicale et l'astronomie. La restauration non aveugle s'apparente à l'inversion d'un système linéaire, lié à l'opérateur de flou, en présence de mesures bruitées. Une simple inversion matricielle semble fournir une approche de résolution triviale, mais dans la plupart des cas, elle implique de calculer l'inverse d'une matrice de très grande dimension, mal conditionnée voir non carrée, et quand il existe, le résultat de l'inversion directe est rarement satisfaisant. Dans le cas de la restauration aveugle, le problème est encore plus compliqué puisque l'opérateur de flou est lui aussi inconnu, et on observe que l'image dégradée à partir de laquelle l'image et le flou sont à estimer. De nombreuses approches ont été proposées pour la restauration d'image. Par exemple, les méthodes traditionnelles basées sur des approches d'optimisation, minimisant une fonction de coût appropriée par un algorithme d'optimisation itératif adapté. Ces approches sont fiables et montrent des résultats satisfaisant. Cependant, un réglage fin des hyper-paramètres de la fonction de coût (e.g., paramètre de régularisation) peut être fastidieux. Récemment, des approches dites supervisées basées sur l'apprentissage profond ont été proposées pour la restauration d'images. Elles reposent sur des modèles non-linéaires complexes dont les paramètres sont appris à partir de bases de données d'images. Ces méthodes sont flexibles et peuvent arriver à d'excellents résultats pourvu que la base de données soit de qualité et de dimension suffisante. De plus, malgré leur complexité, ces approches peuvent être implémentées efficacement sur des cartes GPU, grâce à des environnements de programmation adaptés (e.g., PyTorch). En plus du point mentionné ci-dessus lié à l'importance de la base de données, un autre inconvénient est le manque d'interprétabilité de ces méthodes. Enfin, aussi bien les techniques d'optimisation ou d'apprentissage, fournissent généralement une estimation ponctuelle de la solution. En particulier, elles permettent rarement d'extraire des statistiques d'ordre supérieur, et donc ne peuvent pas quantifier l'incertitude sur le résultat obtenu.

Dans cette thèse, nous considérons une troisième famille d'approche, appelée l'approche bayésienne, pour traiter le problème de restauration d'image. Par rapport aux méthodes mentionnées précédemment, le cadre bayésien considère les paramètres recherchés (image/noyau dans notre cas) comme des grandeurs aléatoires plutôt que déterministes. L'objectif devient l'estimation de leur distribution *a posteriori*, sachant les données observées (i.e., l'image dégradée). L'application du cadre bayésien présente deux avantages principaux. Tout d'abord, la statistique du bruit, et des informations sur *a priori* sur les grandeurs inconnues peuvent être prises en compte explicitement, via la formule de Bayes. Deuxièmement, étant donné que toute la distribution *a posteriori* est estimée, il est facile d'obtenir n'importe quel moment, tels que la moyenne et la covariance, ou un intervalle de confiance pour une prise de décision ultérieure.

Néanmoins, une tâche complexe réside dans la mise en oeuvre de l'approche bayésienne, à savoir le calcul de la distribution *a posteriori*. En effet, dans la plupart des

formulations bayésiennes pour la restauration d'image, la constante de normalisation de cette distribution ne peut pas être calculée analytiquement même si le modèle *a priori* et le modèle d'observation sont simples (e.g., modèles gaussiens). De nombreuses méthodes algorithmiques ont été envisagées pour fournir une bonne approximation de la distribution *a posteriori* recherchée. Dans cette thèse, nous apportons des contributions à cette problématique, dans le cadre de deux problèmes issus de la restauration d'images : (i) l'identification de flou non stationnaire, (ii) la déconvolution aveugle. Pour le premier problème, nous proposons une approche de simulation stochastique, de type filtrage particulaire (Chapitre 3). Pour le second problème, nous proposons une méthode d'approximation par technique bayésienne variationnelle (Chapitre 4). Puis nous en proposons une version supervisée via une technique de déroulement de réseaux de neurones (Chapitre 5). Nos propositions méthodologiques sont étayées par un grand nombre de résultats numériques, impliquant des modèles de flou et bruit variés. Les contributions détaillées de chaque chapitre sont décrites ci-après.

Chapitre 3: Approche de filtrage particulaire pour l'identification de flou variant spatialement

Les images produites par des instruments optiques souffrent souvent de flou causé par la diffraction de la lumière ou le mouvement de l'objet. La présence du flou provoque la propagation d'une source ponctuelle infinitésimale dans l'image acquise définissant la "Point Spread Fonction" (PSF). Lorsque la PSF est inconnue, on peut recourir à une stratégie de déconvolution aveugle pour restorer l'image. Une autre approche consiste d'abord à identifier le flou puis à l'éliminer de l'image dégradée par une stratégie de restauration non aveugle. Cette méthode demande une bonne précision pour l'estimation de la PSF, qui peut être atteinte efficacement par une étape préalable d'acquisition d'objets normalisés et calibrés, tels que des microbilles sphériques fluorescentes en microscopie ou des diagrammes de résolution pour l'étalonnage de l'appareil photo numérique. L'identification de la PSF est typiquement formulé sous la forme des moindres carrés. En plus de servir pour la phase de restauration d'image, la PSF identifiée peut également être utilisée pour ajuster un modèle paramétrique (souvent non linéaire) afin déterminer les caractéristiques d'un système optique. Dans la plupart des scénarios réalistes, un modèle de PSF stationnaire n'est pas adapté en raison de la profondeur de champ, du mouvement des objets ou de la caméra, d'abérrations optiques anisotropes de la lentille, ou de la turbulence atmosphérique. De telles sources de dégradation d'image donnent lieu à ce que l'on appelle flou variable spatialement. L'identification de la PSF dans ce contexte nécessite d'estimer une carte de plusieurs PSFs, décrivant le noyau de flou à chaque emplacement du plan optique. Dans ce chapitre, nous abordons ce problème de l'estimation bayésienne des paramètres d'une PSF variable spatialement à partir d'acquisitions d'images calibrées. Nous considérons un modèle séquentiel paramétrique pour la carte des PSFs qui suppose des faibles variations entre les PSFs agissant sur des régions voisines de l'image. Nous formulons un modèle d'espace d'états où chaque pas de temps correspond à un emplacement de patch dans l'image. Nous adoptons une approche probabiliste visant à produire la distribution *a posteriori* des paramètres

inconnus. Cela nous permet de traiter l'incertitude de manière systématique et l'inclusion de connaissances *a priori* sur les inconnues. En conséquence, nous sommes en mesure non seulement de fournir des mesures d'incertitude sur les paramètres à estimer, mais aussi de propager cette incertitude à des tâches où ces paramètres estimés sont utilisés (ici, la tâche de restauration). Dans notre approche, la densité de probabilité *a priori* fonction (pdf) des paramètres prend une structure markovienne, ce qui nous permet d'exploiter des approches d'inférence existantes pour l'inférence séquentielle. Les récursions bayésiennes obtenues ne donnent pas lieu à des formes explicites et nous recourons au filtrage particulaire ("particle filtering", PF) pour une résolution approchée du problème d'inférence. Nous commençons par la méthode classique bootstrap PF (BPF). Le BPF parvient à obtenir des résultats satisfaisants dans certains de nos exemples mais il présente certaines lacunes, par exemple dans des scénarios avec un faible bruit d'observation où se pose un problème de dégénérescence des particules. Nous proposons donc une nouvelle méthode appelée GIAnPF ("Generalized Interacting Annealed PF"). GIAnPF gère la dégénérescence des particules en considérant une séquence de distributions intermédiaires, qui sont des versions "recuites" de la distribution de filtrage initiale. Nous démontrons la bonne performance de l'algorithme proposé ainsi que sa robustesse à travers plusieurs expériences numériques, incluant trois formes différentes de flous paramétriques variables spatialement. Nous illustrons en Figure 1 les résultats d'identification obtenus par différentes méthodes à partir d'acquisitions simulées sur une image de microscopie décomposée en 64 patchs et avec un écart-type de bruit égal à 10^{-2}. Sur cet exemple, les deux méthodes BPF et GIAnPF basée sur la méthodologie PF, fournissent de meilleurs résultats visuels, que les autres approches. De plus, en terme quantitatif, notre méthode GIAnPF atteint une plus faible erreur, au sens de la métrique RMSE. Nous avons également illustré la validité des estimations obtenues, et l'interêt de la quantification d'incertitude, pour résoudre des problèmes de restauration d'image. Le travail présenté dans ce chapitre a fait l'objet de deux publications [Huang *et al.* 2019, Huang *et al.* 2021a].

Chapitre 4: Approche bayésienne variationnelle pour la déconvolution aveugle

Nous considérons le problème de déconvolution aveugle, où une image floutée et bruitée doit être restaurée sans connaissance du modèle de flou. Nous proposons dans ce chapitre un algorithme bayésien varationnel visant à fournir une approximation optimale (au sens de la divergence de Kullback-Leibler) de la distribution *a posteriori* des paramètres à estimer (ici, l'image, le flou, et des hyper-paramètres du modèle). Nous supposons la PSF stationnaire, et satisfaisant une contrainte d'égalité linéaire (e.g., somme à un, symmétrie axiale). Nous adoptons de plus un modèle *a priori* de type SAR sur la PSF, dont la moyenne est définie à l'avance, et une régularisation de type variation totale sur l'image. Nous cherchons ensuite à minimiser la divergence de Kullback-Leibler (KL) entre la distribution *a posteriori* obtenue et un modèle séparable (de type "champs moyens") de celle-ci, par un algorithme de minimisation alternée. Même dans le cas d'un bruit gaussien, la mise à jour n'a pas une forme explicite et nous proposons donc d'adopter

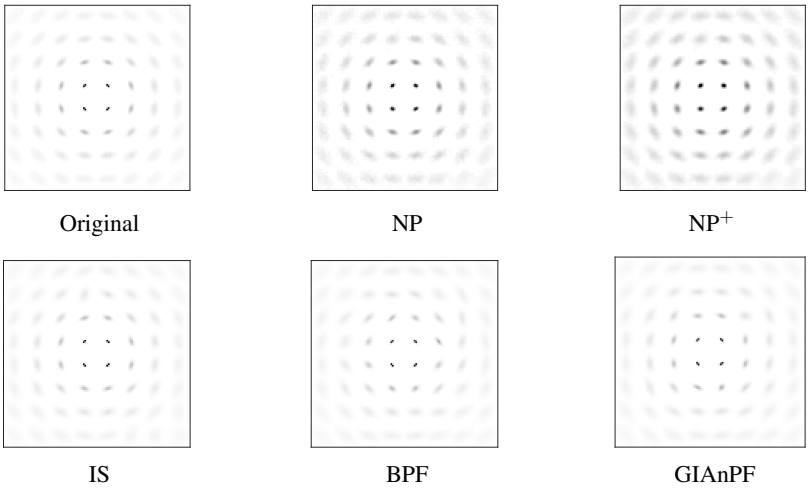


Figure 1: Noyaux originaux et estimés avec les méthodes d'optimisation NP, NP$^+$, et les méthodes d'échantillonnage IS, BPF, GIAnPF. La métrique RMSE (Relative Mean Square Error) vaut, respectivement, 0.2117, 0.1756, 0.1258, 0.1246 et 0.0680. Ce résultat est extrait de la publication [Huang *et al.* 2021a].

une stratégie d'augmentation de données inspirée des méthodes semi-quadratiques en optimisation. Nous obtenons alors l'algorithme *VBA* (variational Bayesian algorithm), dont nous donnons une première forme adaptée à un bruit gaussien, puis une extension de celui-ci à un modèle de bruit plus général. L'algorithme obtenu est rapide et performant. Néanmoins, des tests préliminaires ont montrés une grande sensibilité des résultats au réglage d'hyper-paramètres tels que le poids de la régularisation sur le noyau, et la variance du bruit. Cela nous conduit à la contribution suivante, présentée dans le prochain chapitre.

Chapitre 5: Algorithme VBA supervisé par déroulement profond ("deep unrolling")

Nous considérons ici le problème de déconvolution aveugle dans le cas d'un bruit gaussien. Nous proposons l'algorithme *unfoldedVBA*, qui applique le paradigme du déroulement profond, à l'approche VBA décrite dans le chapitre précédent. Les itérations VBA sont intégrées à une architecture de réseau neuronal profond, chaque itération devenant une couche du réseau. Cela permet (i) d'apprendre les hyperparamètres (en particulier, le niveau de bruit) de VBA de manière automatique et supervisée, (ii) d'améliorer la qualité des résultats en entraînant le modèle avec une fonction de coût liée à la métrique de qualité d'image, (iii) de mettre en oeuvre la méthode en tirant pleinement parti des ressources GPU disponibles, et donc de considérablement réduire le temps de restauration d'image pendant la phase de test. Contrairement aux méthodes standard d'apprentissage profond pour la déconvolution aveugle, notre méthode est l'interprétable, grâce à la technique de déroulement. Soulignons que les méthodes bayésiennes variationnelles ont déjà été employées dans un contexte d'apprentissage profond. Notamment, ce sont des méthodes standards pour l'entrainement des auto-encodeurs variationnels, et des réseaux de neurones bayésiens. Cependant, à notre connaissance, notre travail est le premier à étudier le

déroulement profond d'une technique bayésienne variationnelle. En complément des itérations VBA déroulées, nous intégrons à notre architecture des couches supplémentaires, permettant d'apprendre les hyper-paramètres et d'améliorer la qualité de l'image en sortie via un post-traitement. L'architecture complète est présentée en Figure 2, pour le cas d'images en niveaux de gris et couleur. Nous montrons la supériorité de notre approche par rapport à l'état de l'art sur plusieurs jeux de données, impliquant des flous variés (e.g., gaussien, mouvement). Nous montrons dans la Figure 3 un exemple de restauration pour une image couleur dégradée par un flou de focus. Nous voyons que l'approche proposée permet d'améliorer la qualité visuelle et quantitative de l'image restaurée, par rapport à l'algorithme initial VBA. Les contributions des chapitres 4 et 5 ont fait l'objet de la publication [Huang *et al.* 2021b].

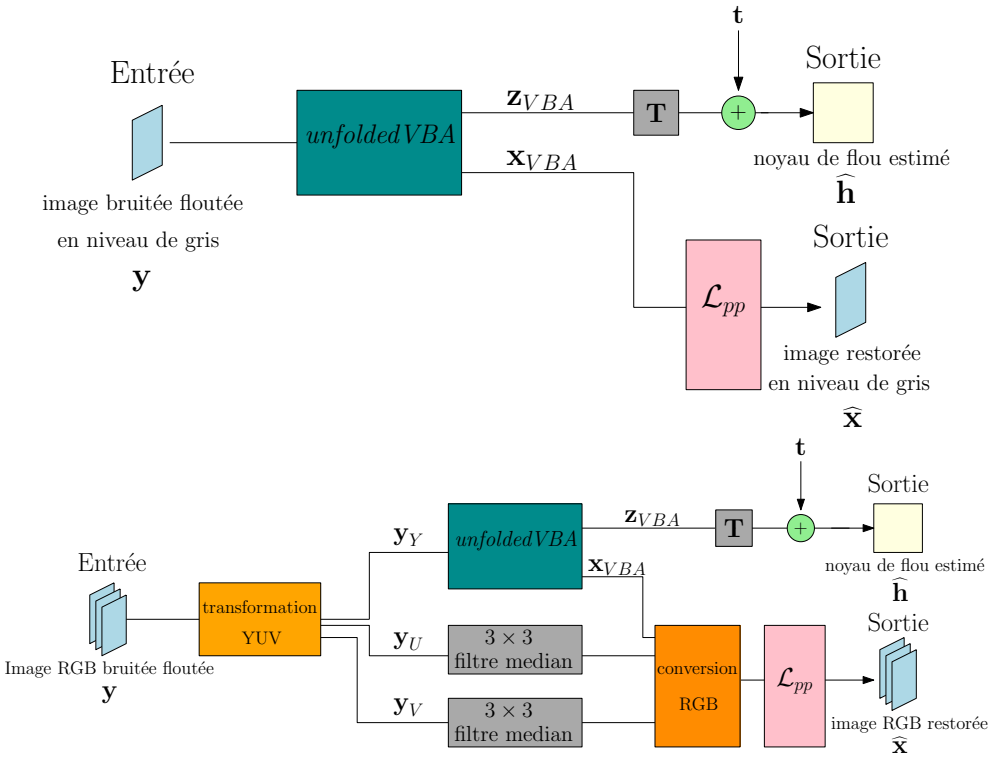
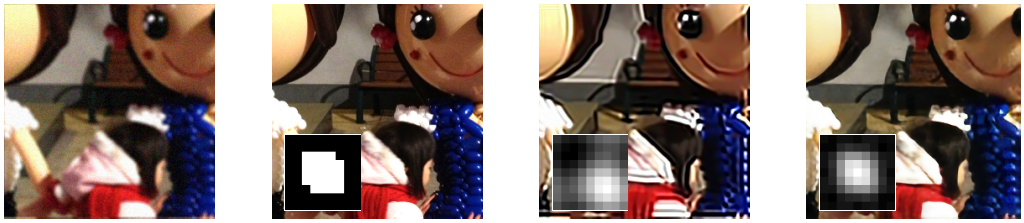


Figure 2: Architecture proposée pour la déconvolution aveugle, dans le cas d'images en niveaux de gris (haut) et couleur (bas).



Dégradée **Originale** **VBA** MSE= 0.0293 **Méthode proposée** MSE = 0.0046

Figure 3: Image dégradée, image/flou originaux, et résultats d'estimation par VBA et par la méthode VBA déroulée proposée. Nous indiquons la valeur de la métrique MSE (Mean Square Error) pour les deux résultats. Cet exemple est extrait de [Huang *et al.* 2021b].

Abstract/Résumé

Abstract

Image deblurring is an essential image restoration problem arising in several fields from astronomy to medical science. It amounts to restoring an image from a degraded, blurry and noisy, version of it. Bayesian image deblurring seeks for the posterior distribution of the image (and blur kernel, when unknown) given an observation model and some prior knowledge on the unknowns. Closed form for the posterior distribution can rarely be computed analytically, so Bayesian approximation tools are deployed to derive an estimation of it. This thesis brings novel contributions in that topic, by introducing novel Bayesian methods for tackling two important scenarios of image deblurring problems. First, we are interested in the so-called blur identification problem, of estimating spatially varying blur kernels given a clean image and its degraded version. We construct a probabilistic state-space model accounting for the smoothness among the neighboring blur kernels, and propose a novel algorithm based on bootstrap PF (BPF) to sample weighted particles, and thus construct the sought posterior distribution. Numerical experiments on various spatially-variant blur maps and images illustrate the benefits and good stability of our approach. Second, we focus on the blind image deblurring problem of jointly estimating the image and blur kernel given the blurry noisy image. We adopt the variational Bayesian approach, to build an appropriate approximation of the posterior distribution. We introduce majorization steps to maintain closed form updates even for non conjugate priors and non-Gaussian noise. This yields the variational Bayesian algorithm (VBA). We then propose to unfold VBA over neural network layers, following the recently introduced deep unrolling paradigm. This yields the unfolded VBA, benefiting from reduced parameter tuning, fast computations on GPU architecture, and improved quantitative restoration results. The superiority of unfolded VBA over state-of-the-art blind deblurring techniques is illustrated on three datasets involving color/grayscale natural images and various blur shapes.

Résumé

La déconvolution d'image est un problème essentiel de restauration d'image qui se pose dans plusieurs domaines allant de l'astronomie à la médecine. Cela consiste à restituer une image à partir d'une version dégradée, floue et bruitée de celle-ci. La déconvolution d'image dans un cadre bayésien s'intéresse à rechercher la distribution postérieure de l'image (et du flou, lorsqu'il est inconnu) étant donné un modèle d'observation et des connaissances préalables sur les inconnues. La forme explicite de la distribution postérieure peut rarement être calculée analytiquement, de sorte que des outils d'approximation bayésienne sont déployés pour en obtenir une estimation. Cette thèse apporte de nouvelles contributions dans ce domaine, en introduisant de nouvelles méthodes bayésiennes pour aborder deux scénarios importants de déconvolution d'image. Tout d'abord, nous nous intéressons au problème d'identification du flou, consistant à estimer des noyaux de flou variant

spatialement étant donné une image originale et sa version dégradée. Nous construisons un modèle d'espace d'état probabiliste tenant compte de la régularité entre les noyaux de flou voisins et proposons un nouvel algorithme basé sur le filtrage de particules bootstrap pour échantillonner des particules pondérées, et ainsi construire la distribution postérieure recherchée. Des expériences numériques sur divers exemples de flou spatialement variants illustrent les avantages et la stabilité de notre approche. Ensuite, nous nous concentrons sur le problème de déconvolution aveugle d'image, consistant à estimer conjointement l'image et le noyau de convolution à partir d'une image bruitée et floue. Nous adoptons une approche bayésienne variationnelle, pour construire une approximation appropriée de la distribution postérieure. Nous introduisons des étapes de majoration pour permettre des mises à jour explicites des variables même pour des a prioris non conjugués et un bruit non gaussien. Cela donne l'algorithme bayésien variationnel. Nous proposons ensuite de dérouler cet algorithme sur des couches de réseaux de neurones, en suivant le paradigme de déroulement profond récemment introduit dans la littérature. L'algorithme obtenue bénéficie d'un nombre réduit de paramètres à régler, d'une exécution rapide sur carte GPU et fournit des meilleurs résultats de restauration. La supériorité du la méthode est illustrée sur trois jeux de données impliquant des images naturelles en couleur ou niveaux de gris et diverses formes de flous.

Notation

$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	set of real vector of n rows
$\mathbb{R}^{m\times n}$	set of real matrices of m rows and n columns
(a,b)	real interval excluding a and b
s	scalars will be denoted by lowercase letters
$\mathbf{v}$	vectors will be denoted by lowercase bold letters
$\mathbf{M}$	matrices will be denoted by uppercase bold letters
$\mathbf{M}^\top$	transportation of the matrix $\mathbf{M}$
$\mathbf{M}^{-1}$	inverse of the matrix $\mathbf{M}$
$\text{trace}(\mathbf{M})$	trace of the matrix $\mathbf{M}$
$\mathbf{I}_m$	square identity matrix in $\mathbb{R}^{m\times m}$
v_i	$i-$th entry of the vector $\mathbf{v}$
$M_{i,j}$	element in $i-$th row and $j-$th column of the matrix $\mathbf{M}$
$\|s\|$	absolute value of the scalar s
$\|\mathbf{M}\|$	determinant of the matrix $\mathbf{M}$
$\|\|\mathbf{v}\|\|$	ℓ_2 norm of the vector $\mathbf{v}$
$\|\|\mathbf{M}\|\|$	ℓ_2 norm of the matrix $\mathbf{M}$
$\text{diag}(\mathbf{M})$	vector composed of the diagonal entries of the matrix $\mathbf{M}$
$\text{Diag}(\mathbf{v})$	diagonal matrix whose entries are given by the vector $\mathbf{v}$
$\frac{\delta f(\mathbf{v})}{\delta v_i}$	functional derivatives of the function f with respect to v_i
$\frac{\partial f(\mathbf{v})}{\partial v_i}$	partial derivatives of the function f with respect to v_i
ln	natural logarithm
exp	natural exponential
$\mathcal{KL}(f\|\|g)$	Kullback Leibler divergence between functions f and g
$\mathbf{x}_t$	the state of $\mathbf{x}$ at time step t
$\mathbf{x}_t^{(i)}$	the $i-$th sample of the state of $\mathbf{x}$ at time step t
$p(\mathbf{x})$	prior distribution of $\mathbf{x}$
$p(\mathbf{x}\|\mathbf{y})$	conditional distribution of $\mathbf{x}$ given $\mathbf{y}$
$\mathbb{E}_{\mathbf{x}}(f(\mathbf{x}))$	expectation of the function $f(\mathbf{x})$ with respect to $p(\mathbf{x})$
$\mathbb{E}_{\mathbf{x}\|\mathbf{y}}(f(\mathbf{x}))$	expectation of the function $f(\mathbf{x})$ with respect to $p(\mathbf{x}\|\mathbf{y})$
$\mathcal{U}([a,b])$	uniform distribution over the interval $[a,b]$
$\mathcal{N}(\mu,\Sigma)$	Gaussian distribution with mean vector μ and covariance matrix Σ
$\mathcal{N}_{[a,b]}(\mu,\Sigma)$	truncated Gaussian distribution with mean vector μ and covariance matrix Σ in the support $[a,b]$

List of acronyms

3MG majorize-minimize memory-gradient 38

AIS adaptive importance sampling 14

APF auxiliary particle filter 17

BPF bootstrap PF 16

CNN convolutional neural network 69

DL dual-lens 70

EKF extended Kalman filter 27

ESS effective sample size 35

GIAnPF Generalized Interacting Annealed PF 20

IS importance sampling 13

KF Kalman filter 27

KL Kullback-Leibler 13

M-H Metropolis-Hastings 29

MAE mean absolute error 78

MAP maximum a posteriori 8

MCMC Markov chain Monte Carlo 12

MIS multiple IS 14

MLE maximum likelihood estimate 8

MM majoration-minimization 55

MMSE minimum mean squared error 9

MSE mean squared error 39

NESS normalized effective sample size 35

NN neural network 90

pdf probability density function 20

PF particle filtering 15

CHAPTER 1

General introduction

Contents

1.1 Motivation

The problem of restoring an image from a degraded, blurred and noisy version thereof arises in many application frameworks, such as microscopy, medical imaging and astronomy. Non-blind image deblurring reads as a linear inverse problem requiring to invert the effect of a blur operator, in the presence of noisy measurements. Simple (pseudo)inversion of the blur degradation operator might seem as an appealing solution approach, but in most cases it involves computing the inverse of a very high-dimensional, ill-conditioned or even non-square matrix, and whenever it exists, the result of the direct inversion is rarely satisfactory. In the case of blind restoration, the problem is even more complicated since the blur operator is also unknown, and we observe that the degraded image from which the image and the blur are to be estimated. Many approaches have been proposed for image restoration. Traditional methods rely on optimization approaches, minimizing an appropriate cost function by a suitable iterative optimization algorithm. These approaches are reliable and demonstrate satisfactory results. However, finetuning the hyperparameters of the cost function (e.g., regularization parameter) or of the optimization solver (e.g., stepsize) can be tedious. Supervised approaches based on deep learning have also been proposed for image restoration. They rely on complex nonlinear models whose parameters are learned from image databases. These methods are flexible and can achieve excellent results assuming a training database of sufficient quality and size. Despite their inherent model complexity, these approaches can be implemented efficiently on GPU cards, thanks to suitable programming environments (for example, PyTorch). In addition to requiring large database, another disadvantage is the lack of interpretation of these methods. Both optimization and learning techniques generally offer a point estimate of the solution. In particular, they rarely allow to extract higher order statistics, and therefore cannot quantify the uncertainty on the result obtained.

In this thesis, we consider a third family of approaches, called the Bayesian approach, to deal with the image restoration problem. Compared to the methods mentioned previously, the Bayesian framework considers the sought parameters (image/blur in our case) as random quantities rather than deterministic ones. The goal becomes the estimation of their distribution a posteriori, given the observed data (i.e., the degraded image). The application of the Bayesian framework has two main advantages. First of all, the noise statistic and a priori information on the unknown quantities can be taken into account explicitly, via the Bayes' formula. Second, since the entire a posteriori distribution is estimated, it is easy to obtain any moment, such as the mean and covariance, or a confidence interval for further decision making.

Nevertheless, a complex task lies in the implementation of the Bayesian approach, namely the calculation of the a posteriori distribution. Indeed, in most Bayesian formulations for image restoration, the normalization constant of this distribution cannot be analytically computed even if the model a priori and the observation model are simple (e.g., Gaussian models). Many algorithmic methods have been considered to provide a good approximation of the desired a posteriori distribution. In this thesis, we make contributions to this question, in the context of two problems arising from image restoration, namely (i) non-stationary blur identification, and (ii) blind deconvolution. For the first problem, we propose a sequential stochastic simulation approach based on the particle filtering framework (Chapter 3). For the second problem, we propose an approximation method through a variational Bayesian technique (Chapter 4). Then we build a supervised version of this method using a neural network unfolding methodology (Chapter 5). Our methodological contributions are supported by a large number of numerical results, involving various models of blur and noise. The detailed contributions of each chapter are described below.

1.2 Main contributions

Chapter 3: Particle filtering method for space-varying blur identification

Images produced by optical instruments often suffer from blur caused by light diffraction or object motion. The presence of the blur causes the spread of an infinitesimal point source in the acquired image defining the "Point Spread Function" (PSF). When the PSF is unknown, a blind deconvolution strategy can be used to restore the image. Another approach is to firstly identify the blur and then remove it from the degraded image by a non-blind restoration strategy. This method requires good precision for the estimation of the PSF. This can be effectively achieved by a prior step of acquiring standardized and calibrated objects, such as fluorescent spherical microbeads in microscopy or resolution diagrams for calibration of digital cameras. The identification of the PSF is typically formulated as a (constrained) least squares minimization problem. The identified PSF can be further used to fit a parametric model (often nonlinear) to determine the characteristics of an optical system. In most realistic scenarios, a stationary PSF model is not suitable due to depth of field, object or camera movement, anisotropic optical aberrations of the lens, or atmospheric turbulence. Such sources of image degradation give rise to so-called spatially variant blur. Identifying the PSF in such context requires estimating a map of several PSFs, describing

the blur kernel at each location in the optical plane. In this chapter, we address the problem of the Bayesian estimation of the parameters of a spatially variant PSF from calibrated image acquisitions. We consider a parametric sequential model for the map of PSFs which assumes small variations between PSFs acting on neighboring regions of the image. We formulate a state-space model where each time step corresponds to a patch location in the image. We adopt a probabilistic approach aiming to produce the a posteriori distribution of the unknown parameters. This allows us to deal with uncertainty in a systematic way as well as the introduction of a priori knowledge about the unknowns. Consequently, we are able not only to provide measures of uncertainty on the parameters to be estimated, but also to propagate this uncertainty to tasks where these estimated parameters are used (here, the restoration task). In our approach, the probability density a priori function of the parameters takes a Markovian structure, which allows us to exploit existing inference approaches for sequential inference. The Bayesian recursions obtained do not give rise to explicit forms and we resort to particle filtering (PF) for an approximate resolution of the inference problem. We start with the classic method bootstrap PF (BPF). BPF manages to reach satisfactory results in some of our examples but it has some shortcomings, for example it fails in scenarios with low observation noise where a problem of particle degeneracy arises. We therefore propose a new method called *Generalized Interacting Annealed PF* (GIAnPF). GIAnPF handles particle degeneracy by considering a sequence of intermediate distributions, which are "annealed" versions of the initial filtering distribution. We demonstrate the good performance of the proposed algorithm as well as its robustness through several numerical experiments, including three different forms of spatially variable parametric fuzziness. We also illustrate the validity of the estimates obtained, and the interest of uncertainty quantification, to solve image restoration problems.

Chapter 4: Variational Bayesian algorithm for blind image deconvolution under Gaussian and non-Gaussian noise

We consider the problem of blind deconvolution, where a blurred and noisy image must be restored without assuming the knowledge of the blur kernel. We propose in this chapter a variational Bayesian algorithm aiming to provide an optimal approximation (in the sense of the Kullback-Leibler divergence) of the a posteriori distribution of the parameters to be estimated (here, the image, the blur, and hyperparameters of the model). We assume the PSF is stationary and satisfies a linear equality constraint (e.g., sum-to-one, axial symmetry). We also adopt an a priori model of the SAR type on the PSF, whose mean is defined in advance, and a total-variation (TV) regularization on the image. We then seek to minimize the Kullback-Leibler (KL) divergence between the a posteriori distribution obtained and a separable model (of the "mean field" type) of it, by an alternating minimization algorithm. Even in the simpler case of Gaussian noise, the updates do not have closed forms and we therefore propose a data augmentation strategy inspired by semi-quadratic methods used in optimization. This yields our algorithm *VBA* (variational Bayesian algorithm), for which we give a first form adapted to a Gaussian noise, then an extension of it to a more general noise model. The algorithm obtained is fast and efficient. Nevertheless, preliminary tests have shown its high sensitivity to the tuning of hyperparameters such as the weight

of the regularization on the kernel, and the variance of the noise. This leads us to our next contribution, presented in the next chapter.

Chapter 5: Unrolled variational Bayesian algorithm for blind image deconvolution

We focus here on the blind image deconvolution problem in the case of Gaussian noise. We propose the *unfoldedVBA* algorithm, which applies the deep unfolding paradigm, to the VBA approach described in the previous chapter. VBA iterations are embedded into a deep neural network architecture, with each iteration becoming a layer of the network. This allows (i) to learn the hyperparameters (in particular, the noise level) of VBA in an automatic and supervised way, (ii) to improve the quality of the results by training the model with a cost function linked to the metric image quality, and (iii) to implement the method by taking advantage of the available GPU resources, and therefore to considerably reduce the image restoration time during the test phase. Unlike standard deep learning methods for blind deconvolution, our method is interpretable, thanks to the unrolling paradigm. Note that variational Bayesian methods have already been used in a deep learning context. In particular, they are standard methods for training variational auto-encoders, and Bayesian neural networks. However, to our knowledge, our work is the first to investigate the deep unrolling of a variational Bayesian technique. In addition to the VBA layers, we integrate additional layers into our architecture, allowing us to learn the hyperparameters and to improve again the quality of the output image via post-processing. We show the superiority of our approach compared to the state-of-the-art on several datasets, involving various blurs (e.g., Gaussian blurs, real-world motion blurs), and grayscale/color natural images.

1.3 Publications

Journal papers

- Y. Huang, E. Chouzenoux and V. Elvira. Probabilistic Modeling and Inference for Sequential Space-Varying Blur Identification. IEEE Transactions on Computational Imaging, vol. 7, pages 531–546, 2021. [Huang *et al.* 2021a]

- Y. Huang, E. Chouzenoux and J.-C. Pesquet. Unrolled Variational Bayesian Algorithm for Image Blind Deconvolution. In major revisions in IEEE Transactions on Image Processing, 2022. https://arxiv.org/abs/2110.07202 [Huang *et al.* 2021b]

Conference papers

- Y. Huang, E. Chouzenoux and V. Elvira. Particle Filtering for Online Space-Varying Blur Identification. In Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2019), pages 544–548, Le Gosier, Guadeloupe, French West Indies, 2019. [Huang *et al.* 2019]

Invited talks

- Particle Filtering for Online Space-Varying Blur Identification. Journées Imagerie Optique Non Conventionnelle (JIONC 2019), Institut Langevin, Paris, 18 March 2019.
- Particle Filtering for Online Space-Varying Blur Identification. SigMA seminar, CRIStAL, Lille, 21 February 2022.

Softwares

- Unrolled Variational Bayesian Algorithm for Image Blind Deconvolution (PyTorch implementation). https://github.com/yunshihuang/unfoldedVBA.

1.4 Outline

- **Chapter** 1 presents the motivation and context of our research, and introduces our contributions.
- **Chapter** 2 gives an introduction to the field of image restoration, with a specific focus on Bayesian techniques.
- **Chapter** 3 presents our first contribution, that is the proposition of a sequential particle filtering algorithm for the probabilistic identification of the parameters of spatially-variant blurs.
- **Chapter** 4 introduces our second contribution, that is the construction of two variational Bayesian algorithms, to deal with image blind deconvolution problems under Gaussian and non-Gaussian noises.
- **Chapter** 5 states our third contribution, which is the introduction of the unrolling paradigm within the VBA method proposed in the previous chapter, and its experimental validation onto various datasets of blind image deblurring.
- **Chapter** 6 concludes the thesis and raises some perspectives.

CHAPTER 2

Background

Contents

The so-called Bayesian approach in statistical analysis is based on Bayes' rule. The main advantage of Bayesian approach is that instead of deriving only point estimate of unknowns like other methods (e.g., optimization-based), it provides a full distribution and thus yields, for instance, uncertainty quantification of the estimation [Jeffreys 1939, Robert *et al.* 2009, Mattei 2020]. Bayesian methods have been applied to different problems related to image processing. The rest of this chapter is organized as follows. In Section 2.1, we introduce the Bayesian paradigm, and provide typical Bayesian formulations encountered in image restoration. In Section 2.2 and Section 2.3, we review two main classes of Bayesian inference methods (static and dynamic ones) respectively, which will be at the core of this thesis developments. In Section 2.4, we conclude this chapter.

2.1 Bayesian framework for imaging science

2.1.1 Bayesian paradigm

In the field of statistical inference, there are two mainstreams called frequentist [Bailer-Jones 2017] and Bayesian [Bernardo & Smith 1994, Gelman *et al.* 2004, Robert 2007] statistics. For some unknown parameters to be estimated, the frequentist approach assumes them to be fixed and uses only the information from the current experiment. The Bayesian or probabilistic approach allows to quantify the uncertainty and to incorporate prior information. We first review some basic concepts of Bayesian approaches.

Let Θ be the collection of unknown parameters to be estimated, and $\mathbf{y}$ the given data. Both of them are random variables, thus we can get access to the posterior distribution of the unknowns $p(\Theta|\mathbf{y})$ according to the Bayes' rule, given the likelihood $p(\mathbf{y}|\Theta)$ and the prior distribution $p(\Theta)$ as follows:

$$p(\Theta|\mathbf{y}) = \frac{p(\Theta)p(\mathbf{y}|\Theta)}{p(\mathbf{y})}. \tag{2.1}$$

Hereabove,

$$p(\mathbf{y}) = \int p(\Theta)p(\mathbf{y}|\Theta)\mathrm{d}\Theta \tag{2.2}$$

is the marginal density of the variable Θ and can be seen as a normalization constant of $p(\Theta|\mathbf{y})$. This means that we have the proportionality relation:

$$p(\Theta|\mathbf{y}) \propto p(\Theta)p(\mathbf{y}|\Theta). \tag{2.3}$$

Eq. (2.3) is especially useful since in most cases the constant (2.2) cannot be computed analytically.

2.1.2 MAP/MMSE estimator

Let us review some popular point-wise estimators. Since the exact value of the parameter Θ is unknown, we need to resort to estimation methods. The relevant maximum likelihood estimate (MLE) estimator is derived as

$$\hat{\Theta}_{\mathrm{MLE}} = \mathrm{argmax}_{\Theta}\ p(\mathbf{y}|\Theta), \tag{2.4}$$

by obtaining the estimate of the unknown parameters Θ only based on the observation model $p(\mathbf{y}|\Theta)$. Moreover, if we have access to the prior distribution $p(\Theta)$, we can then estimate it by the posterior distribution $p(\Theta|\mathbf{y})$.
When maximizing this posterior distribution $p(\Theta|\mathbf{y})$, we obtain the maximum a posteriori (MAP) estimator,

$$\hat{\Theta}_{\mathrm{MAP}} = \mathrm{argmax}_{\Theta}\ p(\Theta|\mathbf{y}). \tag{2.5}$$

Note that if the prior distribution, $p(\Theta)$, is chosen as uniform, the MAP estimator coincides with the MLE estimator since

$$\begin{aligned}\hat{\Theta}_{\mathrm{MAP}} &= \mathrm{argmax}_{\Theta}\ p(\Theta|\mathbf{y}) \\ &= \mathrm{argmax}_{\Theta}\ p(\mathbf{y}|\Theta)p(\Theta).\end{aligned} \tag{2.6}$$

Equivalently,

$$\hat{\Theta}_{\mathrm{MAP}} = \mathrm{argmin}_{\Theta}\left(-\ln p(\mathbf{y}|\Theta) - \ln p(\Theta)\right). \tag{2.7}$$

However, the closed-form solution is intractable in most applications, so we resort to iterative optimization methods to compute the estimator numerically.

On the other hand, we can also choose to minimize the mean square error between the estimation $\hat{\Theta}$ and the true value Θ,

$$\hat{\Theta}_{\text{MMSE}} = \text{argmin}_{\Theta}\, \mathbb{E}_{\Theta}[(\hat{\Theta} - \Theta)^2]. \tag{2.8}$$

Thus the Bayesian estimator is the mean of the posterior distribution of Θ,

$$\hat{\Theta}_{\text{MMSE}} = \mathbb{E}_{\Theta|\mathbf{y}}(\Theta) = \int \Theta p(\Theta|\mathbf{y}) \mathrm{d}\Theta. \tag{2.9}$$

This posterior mean estimator is often called minimum mean squared error (MMSE) estimator of the parameter Θ. Note that if the posterior distribution is Gaussian, these two Bayesian estimators coincide since the mean and the mode of a normal distribution are identical. However, in most cases, as soon as the posterior distribution is not symmetric, MAP estimator and MMSE estimator would give different results. In general, both MAP estimator and MMSE estimator are point estimates instead of Bayes estimators unless Θ is discrete variable, which also demand a credible interval for the estimation. In conclusion, this shortage of point-wise estimators forces to find alternative Bayesian approaches with the ability to provide the uncertainty of the unknown parameters.

2.1.3 Image restoration

Image restoration is a key problematic of the field of computer vision. Blur is a main cause of image degradation that commonly appears during the process of acquisition of the images due to many reasons such as camera shake [Fergus *et al.* 2006], fast moving of the object [Rozumnyi *et al.* 2021], and out of focus of the object [Chan & Nguyen 2011]. Undoubtedly, the blur deteriorates valuable image information [Shi *et al.* 2014], thus image deblurring becomes a hot topic in many areas, for example, astronomical imaging [Murtagh *et al.* 2007], microscopy [Holmes *et al.* 2006] and medical imaging [Michailovich & Tannenbaum 2007], where the images are usually required to be in high quality. We display some examples of images degraded by different kinds of blurs in Fig. 2.1.

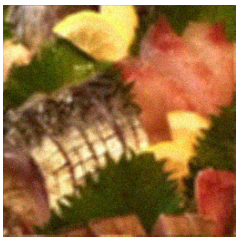
On top of blur degradation, images also suffer from noise coming from the sensor [Boyat & Joshi 2013], for example, the popular additive Gaussian noise [Russo 2003] is a side effect of sensor heat. The salt and pepper noise [Alqadi 2018], also known as impulse noise [Jayamanmadharao *et al.* 2010], is caused by sharp and sudden disturbances in the image signal. Poisson noise [Bertero *et al.* 2009] appears because of the effects of electromagnetic waves. Multiplicative speckle noise [Guadalupe Sanchez *et al.* 2012] is present in low level luminance images such as ultrasound images. We display some examples of images corrupted by different kinds of noise in Fig. 2.2.

In image deblurring (also called image deconvolution), given a degraded, blurred, and noisy image, the goal is to restore a clean image along with an estimation of corresponding blur kernel (if the latter is unknown). Mathematically, the observation model is defined as follows:

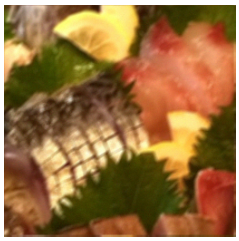
$$\mathbf{y} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \mathbf{n}, \tag{2.10}$$

a) Original image

b) Degraded image with Gaussian blur



c) Degraded image with uniform blur



d) Degraded image with defocus blur

e) Degraded image with motion blur

f) Degraded image with disk blur

Figure 2.1: Examples of images degraded by different blurs.

where $\mathbf{y} \in \mathbb{R}^N$ is the degraded image that is observed; $\tilde{\mathbf{x}} \in \mathbb{R}^N$ is the clean image to be restored; $\mathbf{n} \in \mathbb{R}^N$ corresponds to some additive random noise (more sophisticated noise models could arise, see [Marnissi *et al.* 2017, Dupe *et al.* 2008]); and $\tilde{\mathbf{H}} \in \mathbb{R}^{N \times N}$ is a linear operator related to the blur kernel $\tilde{\mathbf{h}} \in \mathbb{R}^M$. According to whether the blur kernel is known or unknown, one can divide the image restoration problem into two different categories: non-blind image deconvolution or blind image deconvolution. In the former case, the blur kernel is known and the image must be restored given the blurry noisy image and the blur kernel [Marnissi *et al.* 2017, Sun *et al.* 2014, Schuler *et al.* 2013]. As for the blind deconvolution problem, one needs to restore the image and estimate the blur kernel given the observed degraded image [Kundur & Hatzinakos 1996, Huang *et al.* 2021a, Likas & Galatsanos 2004]. Let us make a special focus on this problem that will be at the core of our thesis. There are mainly two approaches to tackle image blind deconvolution. The first one is called sequential identification process [Huang *et al.* 2021a, Carasso 2001] where the blur kernel is derived first through some kernel estimation methods applied on calibrated images, then non-blind image deconvolution is conducted to restore the image using the previously approximated blur kernel. This problem will be investigated in Chapter 3. A second method is the joint estimation approach [Babacan *et al.* 2009, Ren *et al.* 2020], which estimates the image and blur kernel simultaneously. Since this inverse problem is severely ill-posed, an infinite number of combinations of the pairs $(\mathbf{H}, \mathbf{x})$ might exist. This demands to adopt suitable prior assumptions on the sought image and on the sought blur kernel. We will tackle this

a) Original image

b) Degraded image with Gaussian noise

c) Degraded image with salt and pepper noise

d) Degraded image with speckle noise

Figure 2.2: Examples of images degraded by different noises.

problem in Chapters 4 and 5.

In Chapter 3 of this thesis, we will focus on the blur identification problem. This amounts to identifying an estimate $\hat{\mathbf{h}}$ of an original blur kernel $\tilde{\mathbf{h}}$, the given $\mathbf{y}$, and the original image $\mathbf{x}$. In the case of stationary blur, the observation model is

$$\mathbf{y} = \mathbf{X}\tilde{\mathbf{h}} + \mathbf{n}, \tag{2.11}$$

where $\mathbf{X}$ is the suitable block Toeplitz matrix related to the original image $\mathbf{x}$, and $\mathbf{n}$ models some noise. In calibration, it is useful to introduce a parametric model for the blur kernel depending on unknown parameters Θ,

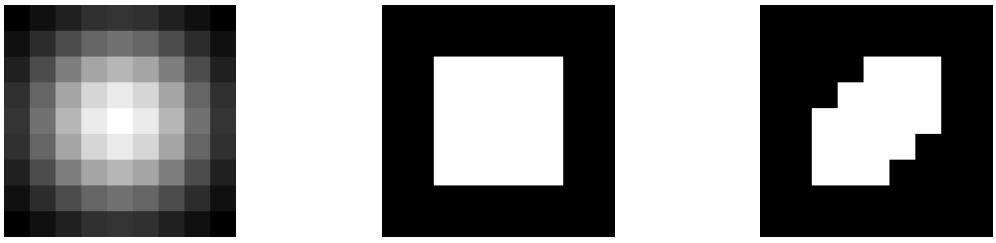
$$\tilde{\mathbf{h}} = h(\tilde{\Theta}). \tag{2.12}$$

The goal is thus to find estimate for $\tilde{\Theta}$ given the knowledge of the pair $(\mathbf{x},\mathbf{y})$. In Chapter 3, we will consider the more challenging case of a spatially-variant blur, requiring the estimation of a set of some parameters Θ_t, associated to image positions $t = 1,\cdots,T$. A complete review of the state-of-the-art for this problem will be provided in Chapter 3.

In Chapter 4 and 5, we focus on the joint image blind deconvolution problem. Given the observation model (2.10), we seek for an estimate $(\hat{\mathbf{x}},\hat{\mathbf{h}})$ of the pair of variables $(\tilde{\mathbf{x}},\tilde{\mathbf{h}})$ given $\mathbf{y}$. Following a Bayesian approach, our target is to find an entire distribution describing the sought parameters (i.e., image, kernel, hyperparameters). Many approaches

have been proposed to solve this task, a complete review will be provided in Chapter 4.

For both calibration and blind deconvolution problems, a key aspect is the definition of suitable priors for the unknowns. One can impose smoothness on the image by applying total-variation (TV) [Marnissi *et al.* 2017, Babacan *et al.* 2009] or simultaneous autoregressive (SAR) prior [Tzikas *et al.* 2009, Shi *et al.* 2017]. For specific images such as US images, one can rely on a sparsity prior [Zhao *et al.* 2016, Besson *et al.* 2017]. For the blur kernel, it is usually assumed to satisfy a normalization constraint (e.g., sum-to-one) [Zhou *et al.* 2015] to avoid the scaling ambiguity, as well as a non-negativity constraint [Mateos *et al.* 2016]. Sometimes, for special shapes of blurs, symmetry might be also assumed, i.e., the values of the blur are symmetrical according to the anti-diagonal line of 2D blur kernel, as shown in Fig. 2.3. The estimation of spatially-variant blurs can also benefit from priors accounting for their smooth variations in the image space [Denis *et al.* 2015].



a) Gaussian blur b) Uniform blur c) Defocus blur

Figure 2.3: Three examples of blurs that follow the diagonal symmetry.

2.2 Bayesian inference methods

In the following, we review two families of algorithms that perform approximate Bayesian inference, namely variational Bayesian algorithms (Section 2.2.1) and importance sampling (Section 2.2.2). Let us point out that another important family of methods are Markov chain Monte Carlo (MCMC) methods that approximate the target distribution by relying on Markov chain theory and Monte Carlo integration [Robert & Casella 2004, Gamerman & Lopes 2006, Robert *et al.* 2018]. We refer the reader to [Pustelnik *et al.* 2016] for a review of the usage of these techniques in the field of image restoration. Examples of applications in the context of image deblurring can be found in [Marnissi *et al.* 2020, Marnissi *et al.* 2018].

2.2.1 Variational Bayesian algorithm

The variational Bayesian algorithm (VBA) attempts to find a more tractable approximation $p(\Theta)$ of the unknown distribution $p(\Theta|\mathbf{y})$. This is a powerful tool especially when we want to marginalize the unknowns without explicitly computing them because the involved integral is intractable. The variational Bayesian paradigm mainly has two advantages, first

it provides us an analytical approximation of the posterior distribution of the unknowns. Secondly it enables us to get access to the high-order moments of the unknowns that allows us to obtain the uncertainty of the estimation, which is essentially important for a Bayesian method. Many applications of variational Bayesian principle have been arising to solve image deblurring problems [Marnissi *et al.* 2017, Likas & Galatsanos 2004, Babacan *et al.* 2009, Zheng *et al.* 2015c]. Among different types of approximation, mean field approximation is the most used one that makes use of a factorized structure of $q(\Theta)$ to approximate the posterior distribution,

$$p(\Theta|\mathbf{y}) \approx q(\Theta) = \prod_{r=1}^{R} q_r(\Theta_r), \tag{2.13}$$

with R being the number of factorized unknowns in the set of Θ. Then, one searches for a distribution $q(\Theta)$ with form (2.13) that approximates the posterior distribution $p(\Theta|\mathbf{y})$ [Blei *et al.* 2017]. The quality of the approximation is evaluated using the Kullback-Leibler (KL) divergence, defined as

$$\mathcal{KL}(q(\Theta)||p(\Theta|\mathbf{y})) = \int q(\Theta) \ln \left(\frac{q(\Theta)}{p(\Theta|\mathbf{y})} \right) \mathrm{d}\Theta \tag{2.14}$$

where the equality only holds when $q(\Theta) = p(\Theta|\mathbf{y})$. It is a non-symmetric measure of the difference between the target posterior distribution and its approximation. The optimal distribution $q(\Theta)$ is the one that minimizes the KL divergence. One can show that the solution of each $q_r(\Theta_r)$ minimizing the KL divergence reads,

$$q_r(\Theta_r) \propto \exp\left(< \ln p(\Theta, \mathbf{y}) >_{\prod_{i \neq r} q_i(\Theta_i)} \right). \tag{2.15}$$

The so-called VBA consists in alternating over the updates of the factors q_r, using the expression (2.15), while keeping others unchanged. An application of this methodology to image blind deconvolution will be presented in Chapters 4 and 5.

2.2.2 Importance sampling and adaptive importance sampling

Importance sampling (IS) is a Monte Carlo methodology for the approximation of intractable integrals [Robert & Casella 2004, Liu 2004]. While IS is a useful approximation technique in many settings, here we focus again on the Bayesian inference problem. Following the already defined notation, the posterior distribution is

$$p(\Theta|\mathbf{y}) = \frac{p(\mathbf{y}|\Theta)p(\Theta)}{p(\mathbf{y})} \propto \pi(\Theta), \tag{2.16}$$

where $\pi(\Theta) = p(\mathbf{y}|\Theta)p(\Theta)$ and $p(\mathbf{y}) = \int \pi(\Theta)\mathrm{d}\Theta$ is a constant. In many applications, we are interested in computing a moment w.r.t. the posterior distribution,

$$I = \mathbb{E}_{\Theta|\mathbf{y}}[\mathbf{g}(\Theta)] = \int \mathbf{g}(\Theta) p(\Theta|\mathbf{y}) \mathrm{d}\Theta \tag{2.17}$$

$$= \frac{1}{p(\mathbf{y})} \int \mathbf{g}(\Theta)\pi(\Theta)\mathrm{d}\Theta, \tag{2.18}$$

where $\mathbf{g}$ is any integrable function w.r.t. $p(\Theta|\mathbf{y})$. However, Eq. (2.18) cannot be computed, since in most cases, this integral is intractable and $p(\mathbf{y})$ is also not available.
IS [Elvira & Martino 2021] is a family of methods that generate samples from the so-called *proposal* or *importance* distribution and assign each sample with an importance weight, accounting for the mismatch between target and proposal distributions. The main difference between this approach and direct Monte Carlo sampling is that, in IS, the particle approximation of the distribution is composed of weighted Dirac masses. The standard IS algorithm is shown in Table. 2.1.

Table 2.1: IS algorithm

1. Draw M i.i.d. samples from the importance distribution $q(\Theta)$,
$$\Theta^{(m)} \sim q(\Theta),\ m = 1, \cdots, M \tag{2.19}$$
2. Compute the associated importance weights
$$w^{(m)} = \frac{\pi(\Theta^{(m)})}{q(\Theta^{(m)})},\ m = 1, \cdots, M \tag{2.20}$$
and compute the normalized weights over all the samples,
$$\overline{w}^{(m)} = \frac{w^{(m)}}{\sum_{i=1}^{M} w^{(i)}},\ m = 1, \cdots, M \tag{2.21}$$
3. Output: It provides M weighted particles $\{\Theta^{(m)}, \overline{w}^{(m)}\}_{m=1}^{M}$ for the approximation of the posterior distribution.

IS methods allow for the approximation of the expectation above via the self-normalized importance sampling (SNIS) estimator,

$$\tilde{I} = \sum_{m=1}^{M} \overline{w}^{(m)} \mathbf{g}(\Theta^{(m)}). \tag{2.22}$$

IS-based methods are powerful approximation techniques, but their performance strongly relies on the choice of the proposal distribution. Therefore, many approaches are proposed in order to tackle this challenge. One alternative is the usage of several proposals, under the so-called multiple IS (MIS) methodology [Elvira *et al.* 2015, Elvira *et al.* 2016a, Sbert *et al.* 2018, Elvira *et al.* 2019b]. Adaptive importance sampling (AIS) goes one step further and consists in iteratively adapting a set of proposals (see [Bugallo *et al.* 2017] for a review). One of the main families of adaptive importance sampling methods is the population Monte Carlo (PMC) algorithms [Iba 2001, Cappé *et al.* 2004, Cappé *et al.* 2008, Elvira *et al.* 2017a, Elvira *et al.* 2017b]. In PMC, the location parameters of the proposals are adapted following resampling-based schemes. Few PMC methods adapt also the scale parameters, e.g., O-PMC [Elvira & Chouzenoux 2022]. Other families of AIS methods include the moment-matching AMIS algorithm [Cornuet *et al.* 2012, El-Laham *et al.* 2019b, El-Laham *et al.* 2019a] or the layered AIS algorithm (e.g., LAIS [Martino *et al.* 2017b, Martino *et al.* 2017a, Mousavi *et al.* 2021]).

As a result, AIS methods can approximate the expectation of any function $\mathbf{g}(\cdot)$ as

$$\tilde{I}_t = \sum_{\tau=1}^{t} \sum_{m=1}^{M} \overline{\rho}_{\tau}^{(m)} \mathbf{g}(\Theta_{\tau}^{(m)}). \tag{2.23}$$

where

$$\overline{\rho}_t^{(n)} = \frac{w_t^{(n)}}{\sum_{\tau=1}^{t} \sum_{m=1}^{M} w_{\tau}^{(m)}}, \quad t = 1, \cdots, T, n = 1, \cdots, M. \tag{2.24}$$

Note that this estimator re-normalizes all importance weights over iterations. Other alternatives for combining the partial estimators of each iteration are provided in [Nguyen *et al.* 2015] and [Cappé *et al.* 2008].

2.3 Sequential Bayesian inference

In this section, we address the Bayesian inference problem in state-space models, a class of probabilistic models that allow for sequential processing of the observations. In the following, we describe this class of models and particle filtering (PF), a family of Monte Carlo methods for the approximation of posterior distributions in this setting.

2.3.1 State-space models

State-space models (SSMs) allow to describe dynamic systems that evolve over time/space. More precisely, a general probabilistic SSM is given by

$$\boldsymbol{\theta}_0 \sim p(\boldsymbol{\theta}_0), \tag{2.25}$$

$$\boldsymbol{\theta}_t \sim p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1}), \tag{2.26}$$

$$\mathbf{y}_t \sim p(\mathbf{y}_t | \boldsymbol{\theta}_t), \tag{2.27}$$

where $\boldsymbol{\theta}_t \in \mathbb{R}^n$ is the state at time step t and $\mathbf{y}_t \in \mathbb{R}^m$ is the observation, $t = 1, \cdots, T$; $p(\boldsymbol{\theta}_0)$ is the prior distribution of the hidden state $\boldsymbol{\theta}_0$ at the initial time step $t = 0$; $p(\boldsymbol{\theta}_t | \boldsymbol{\theta}_{t-1})$ is the state model, and $p(\mathbf{y}_t | \boldsymbol{\theta}_t)$ is the observation model given the current state. The three equations above are in general parametric, and the model parameters are either known or estimated through point-wise estimators [Chouzenoux & Elvira 2020] or via probabilistic techniques [Andrieu *et al.* 2010].

The sequence of states $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_T\}$ is unknown, but can be estimated through the observations $\{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_T\}$. Using the Bayes' rule, we can compute the full joint posterior distribution of all the states given all the observations as follows:

$$p(\boldsymbol{\theta}_{0:T} | \mathbf{y}_{1:T}) = \frac{p(\mathbf{y}_{1:T} | \boldsymbol{\theta}_{0:T}) p(\boldsymbol{\theta}_{0:T})}{p(\mathbf{y}_{1:T})}, \tag{2.28}$$

where $p(\boldsymbol{\theta}_{0:T})$ is the prior distribution, $p(\mathbf{y}_{1:T} | \boldsymbol{\theta}_{0:T})$ is the likelihood distribution of the observation model and $p(\mathbf{y}_{1:T})$ is the normalization constant defined as

$$p(\mathbf{y}_{1:T}) = \int p(\mathbf{y}_{1:T} | \boldsymbol{\theta}_{0:T}) p(\boldsymbol{\theta}_{0:T}) \mathrm{d}\boldsymbol{\theta}_{0:T}. \tag{2.29}$$

The previous problem is called the smoothing problem and requires batch processing of all observations. The main disadvantage of the full posterior approach in (2.28) is that (a) we have to reprocess all other observations when we obtain a new observation, and (b) the dimension of the inferential problem grows with t. Since computing the full joint posterior distribution at all time steps is computationally expensive and unnecessary in real-time applications, we consider building the sequence of the so-called filtering posterior distributions instead, defined as

$$p(\boldsymbol{\theta}_t|\mathbf{y}_{1:t}),\ t = 1,\cdots,T, \tag{2.30}$$

which describes the marginal distribution of the current state $\boldsymbol{\theta}_t$ given all the observations up to time t, $\mathbf{y}_{1:t} = \{\mathbf{y}_1,\cdots,\mathbf{y}_t\}$. In SSMs, the efficient and sequential computation of those filtering distributions is a problem of key importance, allowing to compute as well integrals of the form,

$$I = \int \mathbf{g}(\boldsymbol{\theta}_t)p(\boldsymbol{\theta}_t|\mathbf{y}_{1:t})\mathrm{d}\boldsymbol{\theta}_t, \tag{2.31}$$

where $\mathbf{g}(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ is an arbitrary function.

2.3.2 Particle filtering

The computation of these filtering distributions is called the optimal Bayesian filtering problem. Unfortunately, for most realistic state-space models, these distributions cannot be obtained analytically, thus we resort to their approximations. PFs, also called sequential Monte Carlo (SMC) methods, are Monte Carlo algorithms that are well suited for approximate Bayesian inference in SSMs. PFs are based on IS, reviewed in Chapter 3, and they approximate the distributions (and integrals) of interest through weighted particles. Table 2.2 describes the bootstrap PF (BPF) algorithm, the most standard and widely used PF. Step 2(a) describes the simulation of the particles and Step 2(b) details the computation of the normalized weights. While not being essential, Step 2(c) implements a resampling step that attenuates the particle degeneracy problem (see for instance [Djurić *et al.* 2003]).

Table 2.2: BPF algorithm

1. Initialization. Draw M i.i.d. samples, $\{\boldsymbol{\theta}_0^{(m)}\}_{m=1}^M$ from the prior $p(\boldsymbol{\theta}_0)$.
2. Filtering step. For $t = 1,...,T$:
   (a) Simulate
   $$\overline{\boldsymbol{\theta}}_t^{(m)} \sim p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(m)}), \quad m = 1,...,M \tag{2.32}$$
   (b) Compute the normalized weights by
   $$\overline{w}_t^{(m)} \propto p(\mathbf{y}_t|\overline{\boldsymbol{\theta}}_t^{(m)}, \mathbf{X}_t), \quad m = 1,...,M \tag{2.33}$$
   (c) Resample M times from $\{\overline{\boldsymbol{\theta}}_t^{(m)}\}_{m=1}^M$ with associated probabilities $\{\overline{w}_t^{(m)}\}_{m=1}^M$, i.e., for $m = 1,...,M$, let $\boldsymbol{\theta}_t^{(m)} = \overline{\boldsymbol{\theta}}_t^{(j)}$ with probability $\overline{w}_t^{(j)}$, $j = 1,...,M$
3. Output: The filtering step provides M weighted particles $\{\overline{\boldsymbol{\theta}}_t^{(m)}, \overline{w}_t^{(m)}\}_{m=1}^M$ at each t, for the approximation of the filtering distribution.

As a result of the BPF, for each time step t, we can approximate the filtering distribution $p(\boldsymbol{\theta}_t|\mathbf{y}_{1:t})$ by the M generated particles $(\overline{\boldsymbol{\theta}}_t^{(m)})_{1\leq m\leq M}$ and associated normalized weights $(\overline{w}_t^{(m)})_{1\leq m\leq M}$ computed in Step 2(a)-(b) as follows:

$$p(\boldsymbol{\theta}_t|\mathbf{y}_{1:t}) \approx \sum_{m=1}^{M} \overline{w}_t^{(m)} \delta(\boldsymbol{\theta}_t - \overline{\boldsymbol{\theta}}_t^{(m)}). \tag{2.34}$$

Then, the approximation of the expectation of any arbitrary function $\mathbf{g}(\cdot)$ is

$$\mathbb{E}[\mathbf{g}(\boldsymbol{\theta}_t)|\mathbf{y}_{1:t}] \approx \sum_{m=1}^{M} \overline{w}_t^{(m)} \mathbf{g}(\overline{\boldsymbol{\theta}}_t^{(m)}). \tag{2.35}$$

For example, we can compute the mean and the second order moment of the posterior distribution by taking $\mathbf{g}(\boldsymbol{\theta})$ as $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^2$, respectively.

The BPF is just an instance of particle filtering, but many other filters have been proposed. For instance, auxiliary particle filters (APFs) were originally proposed [Pitt & Shephard 1999] and have gained popularity in the last years due to their flexibility to implicitly adapt the multiple importance sampling proposals [Klaas *et al.* 2005, Elvira *et al.* 2018a, Elvira *et al.* 2019a, Branchini & Elvira 2021]. Also, multiple PFs [Djurić *et al.* 2007], Gaussian PFs [Kotecha & Djurić 2003], and Rao-Blackwellized PFs [Särkkä *et al.* 2007] have shown to provide superior performance in several applications. In all cases, the performance of the filter depends on the number of particles, M. The adaptation of M has been recently proven to be essential to attain successful approximations in challenging problems [Elvira *et al.* 2016b, Elvira *et al.* 2021]. Finally, note that the filtering problem is challenging whenever the observations are very informative, which leads to a steep or *peaky* likelihood. In order to attenuate this issue, [Deutscher *et al.* 2000] proposed an annealed particle filter by performing one annealing run at each time step. Another related work is [Gall *et al.* 2005], which combines a PF and the interacting annealing

algorithm. The application of this methodology to the problem of blur identification will be presented in Chapter 3.

2.4 Conclusion

In this chapter, we have provided an overview of the Bayesian paradigm and introduced an inference problem of interest in the field of image restoration. Then we have described two main classes of Bayesian inference strategies, namely static and dynamic ones, and the main methods for each of these classes. This thesis will mostly focus on VBA and BPF strategies. Our goal is to bring novel methodological contributions in these two methods, so as to derive efficient Bayesian-based strategies for solving two major problems of image restoration, namely blur identification and blind deconvolution.

CHAPTER 3

Probabilistic modeling and inference for sequential space-varying blur identification

Contents

3.1 Introduction

Images produced by optical instruments often suffer from blur caused by light diffraction or object motion. The presence of the blur provokes an infinitesimal point-source to be spread in the acquired image defining the so-called Point Spread Function (PSF). When the PSF is unknown, one can resort to a blind deconvolution strategy [Levin *et al.* 2011, Carasso 2001, Kenig *et al.* 2010, Reeves & Mersereau 1992] to jointly retrieve the image and blur. Another approach consists in first identifying the blur and then removing it from the degraded images using a non-blind restoration strategy [Pustelnik *et al.* 2016, Berisha & Nagy 2014]. This demands a good accuracy for

the PSF estimation, which can be efficiently reached by a preliminary acquisition step of normalized and calibrated objects, such as fluorescent spherical microbeads in microscopy [Kirshner *et al.* 2012b, Hortholary *et al.* 2021] or resolution charts in digital camera calibration [Mannan & Langer 2016, Bell *et al.* 2016]. The PSF identification problem is typically formulated as a least-squares one [Tezaur *et al.* 2015, Mourya 2016]. On top of serving for image restoration purposes, the identified PSF can also be fitted into a parametric non-linear model in order to determine characteristics of the optical system [Anthony & Granick 2009, Chouzenoux *et al.* 2019, Kirshner *et al.* 2012a, Xue & Blu 2014]. In most realistic scenarios, the stationary PSF model is not suitable due to extended depth of field [Trouvé *et al.* 2011, Zhang *et al.* 2016], moving objects or cameras [Zhe *et al.* 2014, Zhang & Hirakawa 2016], anisotropic optical lens aberrations [Jezierska *et al.* 2018], or atmospheric turbulence [Li *et al.* 2007, Schmitz *et al.* 2020, Chimitt & Chan 2020]. Such image degradation sources give rise to a so-called spatially variant blur [Chakrabarti *et al.* 2010, Hirsch *et al.* 2010]. The PSF identification then requires the recovery of a PSF map, describing the blur kernel at each location of the spatial plane [Tezaur *et al.* 2015].

In this chapter, we address the problem of estimating in a sequential manner the parameters of spatially variant PSFs from calibrated image acquisitions. We consider a flexible piecewise constant parametric model for the space-varying PSF map that allows us to describe smooth variations among PSFs acting on neighbor regions. We formulate a state-space model where each time step corresponds to a different patch location. We adopt a probabilistic approach, aiming at producing posterior distributions of the unknowns. The probabilistic approach allows for dealing with the uncertainty in a systematic manner and the inclusion of prior knowledge about the unknowns. As a consequence, we are able not only to provide uncertainty measures on the unknown parameters, but also to propagate this uncertainty to useful tasks where those estimated parameters are used (e.g., in deblurring). In our approach, the prior probability density function (pdf) of the parameters has a Markovian structure, which allows us to inherit existing inference approaches for the sequential inference. Note that more complicated local dependencies could be modeled, e.g., via Markov random fields, at the expense of losing the aforementioned simplicity in the inference task. Our flexible formulation includes the consideration of potential non-standard observation and transition models. More precisely, we can operate virtually with any non-linear and non-Gaussian model. Due to the intractability of the Bayesian recursions, we resort to PF for an approximate inference solution. We start by considering existing PF methods, such as the well-known BPF [Gordon *et al.* 1993]. While the BPF can obtain adequate results in some models, it presents some deficiencies, for instance in scenarios with low observation noise, where the peaky likelihood challenges the diversity among particles (see for instance [Djurić *et al.* 2003] about the particle degeneracy problem). Due to these limitations, we propose a new PF method that we call Generalized Interacting Annealed PF (GIAnPF). The new PF tackles the particle degeneracy by considering a sequence of intermediate distributions, that are annealed versions of the filtering distribution. The considered sequential Bayesian framework provides us with three key advantages, namely (i) low computational cost and limited memory load due to a sequential processing; (ii) a flexible choice of the state-space models that enables us to consider non-linear relations and

non-Gaussian noise; and (iii) a measure of statistical uncertainty on the estimated parameters of kernels. The sequential approach can be exploited to build online implementations in applications involving very large scale images available in batches. Dealing with the image in a sequential manner also alleviates the need of large memory capacity, since the information of previously processed patches does not need to be stored.
We demonstrate the good performance of the proposed approach as well as its robustness through several sets of numerical experiments for three representative parametric blur shapes. We also illustrate the validity of the resulting estimations, to solve image restoration problems.
The rest of Chapter 3 is organized as follows. Section 3.2 introduces the problem of spatially variant blur identification. We review the literature around this topic and discuss the construction of parametric blur models. Section 3.3 presents the state-space models for parametric PSF estimation and proposes our algorithm for Bayesian inference. Section 3.4 shows abundant experimental validation, and Section 3.5 closes the section with some final remarks.

3.2 Problem statement

Let us assume the clean image is $\mathbf{x} \in \mathbb{R}^N$ and its corresponding blurred and noisy image $\mathbf{y} \in \mathbb{R}^N$ that is corrupted by spatially variant blur and noise with given level. Each pair of image $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ are decomposed into a partition of T patches $(\mathbf{x}_t)_{t=1}^T \in \mathbb{R}^P$, $PT = N$, and $(\mathbf{y}_t)_{t=1}^T \in \mathbb{R}^P$. We assume that each patch of the observed image is given by:

$$\mathbf{y}_t = \mathbf{X}_t \mathbf{h}_t + \mathbf{n}_t. \tag{3.1}$$

Hereabove, $\mathbf{y}_t \in \mathbb{R}^P$ represents the blurry noisy patch, $\mathbf{h}_t \in \mathbb{R}^L$ is the blur kernel to be estimated at patch t, and $\mathbf{n}_t \in \mathbb{R}^P$ models an additive noise. Moreover, $\mathbf{X}_t \in \mathbb{R}^{P \times L}$ is a suitable matrix related to the blur of a patch t of original image $\mathbf{x}$. The model in Eq. (3.1) identifies with the PSF-interpolation model from [Denis *et al.* 2015, Eq. (22)], where the interpolation strategy depends on the choice made for $\mathbf{X}_t$ (see also [Hadj *et al.* 2014, Hirsch *et al.* 2010]). In all our practical experiments, for simplicity and complexity reasons, we will adopt the piecewise constant PSF model from [Denis *et al.* 2015, Eq.20], that is the order 0 (i.e., nearest neighbor) PSF-interpolation model. Note that several other strategies are available for modeling the space-varying blur operator, for instance in [Escande & Weiss 2015, Denis *et al.* 2015, Sroubek *et al.* 2016, Hirsch *et al.* 2010, Nagy *et al.* 2004], with various modeling accuracy and computational costs. Our choice in Eq. (3.1) is particularly well suited for fast blur identification due to the separability over patches. This model was also considered in [Bar *et al.* 2007, Hadj *et al.* 2014, Mourya 2016, Tezaur *et al.* 2015].
In many practical situations, a prior over the blur shape is available. It is thus convenient to introduce a parametric model for those blur kernels, with the advantage of reducing the number of unknowns to estimate in the blur identification task. Each kernel is then assumed to read:

$$\mathbf{h}_t = h(\boldsymbol{\rho}_t), \quad t = 1, \ldots, T \tag{3.2}$$

where h is a known function, not depending on t, that describes the general shape of the kernels, specified by parameters $(\boldsymbol{\rho}_t)_{t=1}^T \in \mathbb{R}^K$, with $K \geq 1$ typically much lower than L. In this section, we focus on the problem of sequential parametric blur estimation, which amounts to retrieve, for every $t = 1, \ldots, T$, an estimate for $\boldsymbol{\rho}_t$, given the knowledge of the past and present observations $(\mathbf{X}_i, \mathbf{y}_i)_{i=1}^t$ and the function $h(\cdot)$.The main underlying assumption in our approach is the smooth variation of the kernel shape parameters when progressing sequentially along the patch indexes $t = 1, \ldots, T$ (see discussion in Section 3.3.2 about patch ordering). Therefore, our approach is particularly well suited for space-varying blurs arising from optical aberrations (e.g., phase aberration [Schuler *et al.* 2012], atmospheric aberration [Chimitt & Chan 2020]), and smooth motion (e.g., camera motion blur [Šorel & Flusser 2008]). One can also refer to the smooth varying PSF maps used as illustrative examples in [Escande & Weiss 2015, Denis *et al.* 2015]. As described in Section 3.3, the spatial dependence is encoded in the transition density among consecutive patches, which acts as a prior pdf. We note that, as it is often the case in Bayesian inference, vague enough priors are usually adequate particularly when enough informative observations are available. In contrast, this section does not cover PSF maps with abrupt changes, such as those considered in [Zhe *et al.* 2014, Trouvé *et al.* 2011]. As we will show below, we exploit the similarity of kernels that are spatially close during the inference/estimation task. More precisely, unlike in other approaches, here the data of one patch is implicitly used to better estimate the parameter of other patches, which explains the good performance of the proposed modeling and methodology.

3.2.1 Related literature and contributions

The problem of space-varying blur identification has been widely studied in the literature of image processing. It is important to distinguish two types of approaches. First, there exists a bunch of methods, for performing image deblurring in the presence of an unknown space-varying blur degradation, thus corresponding to blind image restoration [Zhang *et al.* 2016, Hadj *et al.* 2014, Chakrabarti *et al.* 2010, Zhang & Hirakawa 2016] in the case of motion blur. We also refer the reader to the recent work [Alam *et al.* 2019] for a review on this topic. Let us also mention [Hirsch *et al.* 2010] for the case of multi-frame blind image restoration. Most of these methods are focused on the restoration task, and not on the quantitative estimation of the blur map itself. This is at the exception of [Zhang *et al.* 2016, Chakrabarti *et al.* 2010, Zhang & Hirakawa 2016], that jointly restore the image and a piecewise-constant map, obtained through a segmentation-based strategy, describing the non-stationary blur effects. Though image restoration is a problem of high interest, in certain applications such as microscopy or astronomical imaging, an accurate qualitative and quantitative knowledge of the blur effects is key for a better understanding and improving (e.g., through calibration) of the imaging device [Jezierska *et al.* 2018, Schmitz *et al.* 2020, Pankajakshan *et al.* 2012, Lefort *et al.* 2020]. For performing blur identification, it is necessary to make structural assumptions on the blur map to be estimated. For e.g., [Trouvé *et al.* 2011] assumes a finite set of candidate defocus blurs, to be tested in each location of the image, or [Rajagopalan & Chaudhuri 1999] considers parametric blurs depending only on a single parameter. The blur estimation can also

be facilitated by making use of a calibrated image (e.g., fluorescent bead in microscopy, distant star in astronomical imaging) in order to reduce the ill-posedness of the identification problem. When the PSF is stationary, and no parametric model of it is further assumed, its estimation from calibrated image acquisition can be easily solved by a penalized least-squared algorithm (see for instance [Chouzenoux & Pesquet 2017] for an efficient method in the case of large size images). The problem raised in this case is similar to the one arising in high dynamic range image fusion, for which wavelet-based methods have shown their efficiency [Tico & Pulli 2009, Tico *et al.* 2010]. If, additionally, a parametric model of the PSF is available, non-linear least squares method, such as Levenberg-Macquart [Kirshner *et al.* 2012a], or more recent proximity-based algorithms [Chouzenoux *et al.* 2019], can be employed for estimating directly the sought parameters. However, when PSF is non stationary, the accurate identification of its shift-variant evolution becomes much more challenging to resolve, even when the image is calibrated (i.e., known). This problem is typically addressed using optimization-based methods [Tezaur *et al.* 2015, Mourya 2016]. We also refer the reader to [Schmitz *et al.* 2020] and references therein, for a review of the problem and recent insights, in the particular context of astronomical imaging. In the aforementioned works, no parametric model of the PSF was assumed though it is highlighted in [Schmitz *et al.* 2020] as a promising research direction. Moreover, most available techniques address the problem in a batch manner, requiring the loading of the full image before the starting of the identification process, which can be at the price of a high memory cost in high resolution imaging or even incompatible with an on-the-fly image acquisition (e.g., satellite imaging). Furthermore, up to our knowledge, no Bayesian-based techniques have been proposed so far, so that available strategies provide only point-wise estimators for the PSF field. We conclude this state-of-the-art review by emphasizing that several works exploit the smooth variability of the PSFs within the field of view, with the aim to reduce the cost of the non-stationary blur operator [Denis *et al.* 2015, Escande & Weiss 2015] or to perform PSF field interpolation [Ngolé Mboula & Starck 2017, Chen *et al.* 2005].

In this chapter, we propose a Bayesian sampling method, for the identification of parametrized space-varying blurs in the context of calibrated images. Our contribution is threefold: (i) the particle filtering strategy employed for sampling leads to a sequential and fast estimation result; (ii) the proposed method is versatile and easily adapted to diverse parametric blur shapes; and (iii) the Bayesian framework allows us to provide a posterior estimation, including uncertainty quantification, for the PSF map parameters.

3.2.2 Relevant parametric PSF models

The general model of Eq. (3.2) enables to encompass various blur shapes. Hereafter, we present three interesting classes of parametric blur models, that will be used thereby in the experiments of this chapter. To simplify the notation, let us ignore the subscript for patch t in this subsection. Without loss of generality, we consider kernels with square support parametrized by a grid with length $\sqrt{L}$ (assumed to be an odd integer), centered and regularly spaced, denoted $\mathcal{G} = \{-\frac{\sqrt{L}-1}{2}, -\frac{\sqrt{L}-1}{2}+1, \ldots, 0, 1, \ldots, \frac{\sqrt{L}-1}{2}\}$. The blur shape is determined by the values of a given function $\mathrm{h}(c_1, c_2)$ for $(c_1, c_2) \in \mathcal{G}^2$. Otherwise stated,

(c_1, c_2) are the (signed) distances, in pixels, from the origin (i.e., the center position of the squared kernel) in the horizontal and vertical axis, respectively. The vector $h(\boldsymbol{\rho}) \in \mathbb{R}^L$ is then simply deduced by ordering the entries of matrix $[\mathrm{h}(c_1, c_2; \boldsymbol{\rho})]_{(c_1,c_2)\in\mathcal{G}^2} \in \mathbb{R}^{\sqrt{L}\times\sqrt{L}}$, following the lexicographic order[1].

Generalized Gaussian blur space-varying generalized Gaussian blur shapes have been employed for instance in [Liu *et al.* 2021b] in the context of out-of-focus image deblurring. Such blur is parametrized by $\boldsymbol{\rho} = (\theta, \mathbf{s})$ with $\theta \in \mathbb{R}$ and $\mathbf{s} = (s_1, s_2) \in (0, +\infty)^2$ the orientation and width parameters, respectively. The parametric model then reads:

$$(\forall (c_1, c_2) \in \mathcal{G}^2), \quad \mathrm{h}(c_1, c_2; \boldsymbol{\rho}) = \lambda \exp\left(-\frac{1}{2}\frac{([c_1\ c_2]\boldsymbol{\Sigma}^{-1}(\theta, \mathbf{s})[c_1\ c_2]^\top)^p}{\alpha^p}\right), \tag{3.3}$$

with $p > 0$ and $\alpha > 0$ the power and the scale of generalized Gaussian model, respectively, and λ the normalization constant reads

$$\lambda = \frac{p}{\pi\Gamma(\frac{1}{p})2^{\frac{1}{p}}\alpha} \frac{1}{|\boldsymbol{\Sigma}(\theta, \mathbf{s})|^{\frac{1}{2}}}, \tag{3.4}$$

where $\Gamma(\cdot)$ is the Gamma function. The support size is:

$$s_{\max} = \sqrt{L}\left(2\alpha(-2\ln(a))^{1/p}\right)^{-1/2}. \tag{3.5}$$

Moreover,

$$\Sigma(\theta, \mathbf{s}) = \mathbf{R}_\theta \begin{bmatrix} s_1^2 & 0 \\ 0 & s_2^2 \end{bmatrix} \mathbf{R}_\theta^\top \tag{3.6}$$

with $\mathbf{R}_\theta \in \mathbb{R}^{2\times 2}$ the rotation matrix defined by the orientation angle θ,

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{3.7}$$

We can ensure that the kernel value in its support corner is less or equal than $a\,\mathrm{h}(0, 0; \boldsymbol{\rho})$, for some $a \in (0, 1)$, as soon as $s_1 \leq s_{\max}$, $s_2 \leq s_{\max}$ with $s_{\max}$ given in (3.5) above. Note that when $p = \alpha = 1$ in (3.3), we recover the common Gaussian blur model, used for instance in fluorescence microscopy in [Zhang *et al.* 2007]. An example is displayed in Fig. 3.1(left).

Defocus blur Defocus space-varying blurs have been considered for example in [Trouvé-Peloux *et al.* 2018] in the context of depth estimation in digital camera images. Defocus blur kernel can be parametrized by $\boldsymbol{\rho} = (\theta, \mathbf{s})$ with $\theta \in \mathbb{R}$ and $\mathbf{s} \in (0, +\infty)^2$. The parametric model is,

$$(\forall (c_1, c_2) \in \mathcal{G}^2), \quad \mathrm{h}(c_1, c_2; \boldsymbol{\rho}) = \begin{cases} \lambda & \text{if} \quad (c_1, c_2) \in \mathbf{C}(\theta, \mathbf{s}) \\ 0 & \text{otherwise} \end{cases}, \tag{3.8}$$

[1] The lexicographic ordered vector associated to a matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is the column vector $[a\ b\ c\ d]^\top$.

with the normalization constant λ defined as

$$\lambda = \pi s_1 s_2 L. \tag{3.9}$$

The support size is:

$$s_{\max} = \max \mathcal{G} = \frac{\sqrt{L}-1}{2}. \tag{3.10}$$

and the elliptical domain:

$$\mathbf{C}(\theta, \mathbf{s}) = \Big\{ (c_1, c_2) \in \mathcal{G}^2 \text{ such that } \frac{(c_1 \cos\theta + c_2 \sin\theta)^2}{s_1^2} + \frac{(c_1 \sin\theta - c_2 \cos\theta)^2}{s_2^2} \leq 1 \Big\}. \tag{3.11}$$

The latter can easily be ensured to be included into the kernel support, under the condition that $s_1 \leq s_{\max}$, $s_2 \leq s_{\max}$ with $s_{\max}$ given above. An example of such blur is displayed in Fig. 3.1(middle).

Skew-normal blur The skew-normal blur shape [Azzalini & Valle 1996] was first considered in [Hansen & Jensen 2008] to build synthetic image deblurring problems. Its ability for modeling shift-variant PSFs in real digital imaging sensors was then illustrated in [Simpkins & Stevenson 2014]. This blur model depends of five positive parameters $\boldsymbol{\rho} = (s_1, s_2, w, \alpha_1, \alpha_2)$, as follows:

$$(\forall (c_1, c_2) \in \mathcal{G}^2), \mathrm{h}(c_1, c_2; \boldsymbol{\rho}) = \lambda \exp\left(-\frac{1}{2} \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}^\top \mathbf{S}^{-1}(w) \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \right) \times \int_{-\infty}^{\alpha_1 \gamma_1 + \alpha_2 \gamma_2} \exp\left(-\frac{z^2}{2} \right) \mathrm{d}z, \tag{3.12}$$

where

$$\gamma_1 = \frac{c_1}{s_1}, \quad \gamma_2 = \frac{c_2}{s_2}, \quad \mathbf{S}(w) = \begin{bmatrix} 1 & w \\ w & 1 \end{bmatrix}, \tag{3.13}$$

and the normalization constant λ is given by,

$$\lambda = \frac{2}{(2\pi)^{\frac{3}{2}} |\mathbf{S}(w)|^{\frac{1}{2}}}. \tag{3.14}$$

The support size is:

$$s_{\max} = \frac{\sqrt{L}}{2\sqrt{-\ln(a)}}. \tag{3.15}$$

As shown in [Azzalini & Valle 1996, Eq. (2.4)], parameters (w, α_1, α_2) must ensure that the matrix below is positive definite:

$$\begin{bmatrix} 1 & \frac{w - \delta_1 \delta_2}{(1-\delta_1^2)^{\frac{1}{2}} (1-\delta_2^2)^{\frac{1}{2}}} \\ \frac{w - \delta_1 \delta_2}{(1-\delta_1^2)^{\frac{1}{2}} (1-\delta_2^2)^{\frac{1}{2}}} & 1 \end{bmatrix}. \tag{3.16}$$

which amounts to satisfying the following technical conditions:

$$\begin{aligned}
&\delta_1\delta_2 - \sqrt{(1-\delta_1^2)(1-\delta_2^2)} < w < \delta_1\delta_2 + \sqrt{(1-\delta_1^2)(1-\delta_2^2)},\\
&\alpha_1 = \frac{\delta_1 - \delta_2 w}{\sqrt{(1-w^2)(1-w^2-\delta_1^2-\delta_2^2+2\delta_1\delta_2 w)}},\\
&\alpha_2 = \frac{\delta_2 - \delta_1 w}{\sqrt{(1-w^2)(1-w^2-\delta_1^2-\delta_2^2+2\delta_1\delta_2 w)}},
\end{aligned} \tag{3.17}$$

for some $(\delta_1, \delta_2) \in (-1,1)^2$. Hereagain, the maximal proportion $a \in (0,1)$ reached on the corner of the kernel support can be imposed by setting s_1 and s_2 no greater than $s_{\max}$ given in (3.15) above. The skew-normal blur reduces to the Gaussian blur when $\alpha_1 = \alpha_2 = 0$. An example is displayed in Fig. 3.1(right).

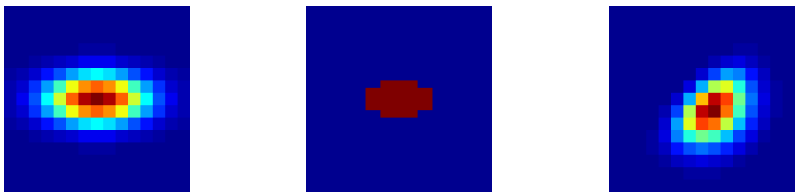


Figure 3.1: (left) Gaussian blur with $\theta = 0, \mathbf{s} = [0.2; 0.1]\sqrt{L}, \alpha = 1, p = 1$ and $L = 15^2$; (middle) defocus blur with $\theta = 0, \mathbf{s} = [0.2; 0.1]\sqrt{L}$ and $L = 25^2$; (right) skew-normal blur with $\mathbf{s}_1 = 0.15\sqrt{L}, s_2 = 0.15\sqrt{L}, w = 0, \alpha_1 = 1.5, \alpha_2 = 1.5$ and $L = 15^2$.

3.3 Space-varying blurs modeling and inference

In this section, we first present our novel modeling approach for space-varying blur maps based on state-space models. We describe the standard BPF for inference in such models. After pointing the limitations of the BPF in this context, we propose the GIAnPF. Finally, we discuss the properties of GIAnPF and the connections with other methods of the literature.

3.3.1 State-space modeling for blur identification

We start by considering a generative SSM, where the hidden state represents the hidden (hence unknown) parameters of the variant PSFs. This generative modeling allows for a systematic Bayesian estimation of the unknown parameters, which are considered hidden states in the SSM literature [Särkkä 2013]. It allows to explicitly model the spatial smoothness of the PSF field. For example, the variation among neighbor patches can be a small rotation and change of width (see hereafter for a discussion regarding the order of

the patches). Let us consider a state-space model given by

$$p(\boldsymbol{\rho}_0), \tag{3.18}$$
$$p(\boldsymbol{\rho}_t|\boldsymbol{\rho}_{t-1}), \tag{3.19}$$
$$p(\mathbf{y}_t|\boldsymbol{\rho}_t, \mathbf{X}_t), \tag{3.20}$$

for $t = 1, \ldots, T$, where $p(\boldsymbol{\rho}_0)$ is the prior distribution, $p(\boldsymbol{\rho}_t|\boldsymbol{\rho}_{t-1})$ is the transition model that generates the patch parameters t given the previous patch $t-1$, and $p(\mathbf{y}_t|\boldsymbol{\rho}_t, \mathbf{X}_t)$ is the observation model of patch t (that can be seen as the likelihood function when $\mathbf{y}_t$ is observed). The goal is to approximate probabilistically and sequentially the unknown parameters $\boldsymbol{\rho}_t$ for each patch t given subsets of data $\{\mathbf{X}_t, \mathbf{y}_t\}_{t=1}^{T}$. In particular, one can obtain the so-called filtering posterior distribution $p(\boldsymbol{\rho}_t|\mathbf{X}_{1:t}, \mathbf{y}_{1:t})$, i.e., the posterior on $\boldsymbol{\rho}_t$ conditioning on all data up to t.

3.3.2 Patch ordering for a sequential processing

The proposed approach requires setting the order in which the data will be sequentially processed. The parameters of consecutively processed patches must keep certain similarity in such a way the spatial information can be exploited. The underlying assumption in our approach is that the blurs affecting consecutive patches in the sequence have similar shape parameters. Otherwise stating, the parameters of the blurs should not change abruptly when going from patch t to patch $t-1$. Smooth PSF maps can then be identified, as soon as consecutive patches are spatially close. This is actually not the case if the patches are numbered naively following the lexicographic order, as illustrated in Figure 3.2(left). More suitable ordering must be adopted. If no additional structural assumption is available on the PSF map, we would suggest the zig-zag order, reminiscent from the one used in DCT-based image compression [Rabbani & Jones 1991] (Fig. 3.2(middle)). Circular blur maps, as those modeling phase aberrations ([Denis *et al.* 2015, Sec.3.4],[Schuler *et al.* 2012, Simpkins & Stevenson 2014],[Mourya 2016, Chap.4]), would be identified better by using a spiral ordering (Fig. 3.2(right)). Phase aberrations (and thus, the PSF width) tend to increase when getting further from the center. The spiral ordering is thus more suitable as it implicitly promotes changes of the PSF parameters depending mostly on the distance to the image center. The spiral ordering will be retained in most experiments of the section.

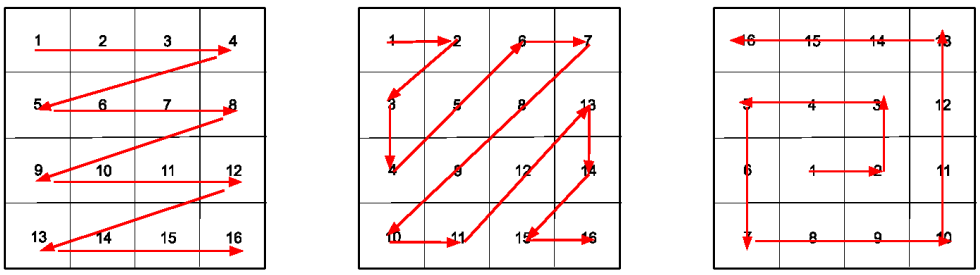


Figure 3.2: Patch ordering (in the case of 4×4 structure): (left) lexicographic ; (middle) zig-zag ; (right) spiral.

The smooth variations between consecutive kernels can be easily encoded using a Gaussian distribution for $p(\boldsymbol{\rho}_t|\boldsymbol{\rho}_{t-1})$. In order to ensure that the sampled blurs do not have degenerate shapes, range constraints can furthermore be enforced on some parameters (e.g., strict positivity for the width parameter, bounded PSF support), simply by considering truncated Gaussian distributions (see more details in Section 3.4). Note that it would be also possible to learn an optimal pattern to scan/process the data. Intuitively, the modeling/inference results of our proposed method would benefit for finding an order where consecutive kernels would be as similar as possible. This could be done for instance by first running a fast (but inaccurate) optimization-based strategy to choose the ordering in which the SSM would be constructed.

3.3.3 Bayesian inference in SSMs through particle filtering

Linear-Gaussian state-space models are often used since they allow for closed-form filtering and smoothing distributions via the well-known Kalman filter (KF) [Kalman 1960]. However, in this section we focus on more complex models which broadens the flexibility and applicability of our approach. Unfortunately, in these models the targeted pdf are intractable and approximations are required. Extensions of the KF exist, e.g., the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) although they can greatly deviate from the true solution when the model is heavily non-linear non-Gaussian. In order to tackle these limitations, we consider an approach based on Monte Carlo approximations, and in particular on PF [Gordon *et al.* 1993].

We start by considering the BPF in Table 3.1, the first and arguably the most relevant PF [Gordon *et al.* 1993]. The BPF departs by simulating M samples (or particles) from the prior distribution. Then, for each patch t, it simulates M particles $\{\overline{\boldsymbol{\rho}}_t^{(m)}\}_{m=1}^M$ from the transition model by conditioning on the previous particles (Step 2a). Then, normalized importance weights are computed as the evaluation of the likelihood at each $\overline{\boldsymbol{\rho}}_t^{(m)}$ in such a way all weights sum up to one (Step 2b). A resampling step is performed (Step 2d) in order to avoid particle degeneracy (see more details in [Tiancheng *et al.* 2015]). As a result, the BPF approximates the filtering posterior distribution $p(\boldsymbol{\rho}_t|\mathbf{X}_{1:t},\mathbf{y}_{1:t})$ for each patch t from the set of M weighted particles $\{\overline{\boldsymbol{\rho}}_t^{(m)},\overline{w}_t^{(m)}\}_{m=1}^M$.

Note that other existing PFs can be also used in this problem, e.g., the APFs [Pitt & Shephard 2001, Elvira *et al.* 2018a, Elvira *et al.* 2019a] or the Rao-Blackwellized PF [Djurić *et al.* 2003, Doucet & Johansen 2009, Särkkä 2013]. The research on PF has been very active in the last decade, and new PFs have been proposed in order to overcome existing challenges.

In the specific application we tackle here, we are facing the well-known particle degeneracy problem, appearing due to very informative observations (hence, peaky likelihood). In particular, typical suitable parametric PSF models are very sensitive w.r.t. some of their parameters and the variance of the observation noise can be relatively low. Particle degeneracy refers to the (undesirable) effect where few particles (or even only one) remain after the resampling step. This reduces the diversity of the particle approximation and endangers the estimation of the blur parameters in the following patches. In the next section, we present a new PF that addresses the shortcomings of off-the-shelf PFs in such challenging

Table 3.1: BPF algorithm for space-varying blur identification.

1. Initialization. Draw M i.i.d. samples, $\{\boldsymbol{\rho}_0^{(m)}\}_{m=1}^M$ from the prior $p(\boldsymbol{\rho}_0)$.
2. Filtering step. For $t = 1,...,T$:
   (a) Simulate
   $$\overline{\boldsymbol{\rho}}_t^{(m)} \sim p(\boldsymbol{\rho}_t|\boldsymbol{\rho}_{t-1}^{(m)}), \quad m = 1,...,M \tag{3.21}$$
   (b) Compute the normalized weights by
   $$\overline{w}_t^{(m)} \propto p(\mathbf{y}_t|\overline{\boldsymbol{\rho}}_t^{(m)}, \mathbf{X}_t), \quad m = 1,...,M. \tag{3.22}$$
   (c) Resample M times from $\{\overline{\boldsymbol{\rho}}_t^{(m)}\}_{m=1}^M$ with associated probabilities $\{\overline{w}_t^{(m)}\}_{m=1}^M$, i.e., for $m = 1,...,M$, let $\boldsymbol{\rho}_t^{(m)} = \overline{\boldsymbol{\rho}}_t^{(j)}$ with probability $\overline{w}_t^{(j)}$, $j = 1,...,M$
3. Output: The filtering step provides M weighted particles $\{\overline{\boldsymbol{\rho}}_t^{(m)}, \overline{w}_t^{(m)}\}_{m=1}^M$ at each t, for the approximation of the filtering distribution.

scenario.

3.3.4 GIAnPF

In this section we propose the novel GIAnPF algorithm. It implements an annealing mechanism (see for instance [Neal 2001]) that overcomes the aforementioned limitations. Instead of directly approximating the targeted distribution, a sequence of modified distributions is considered. In such way, the algorithm starts by considering a more convenient distribution (i.e., sufficiently spread) while the last distribution is the true posterior of interest (in our case, the filtering distribution).

The GIAnPF algorithm is described in Table 3.2. It starts by simulating M particles from the prior distribution in Step 1. However, unlike in the BPF or the APF, GIAnPF considers a sequence of Q intermediate steps in the processing of the data at each patch t. In the prediction step, M particles of the previous annealing layer $\{\boldsymbol{\rho}_{t,Q+1}^{(m)}\}_{m=1}^M$ are generated from the transition model by conditioning on the particles at time $t-1$ (Step 2a). Considering a sequence $0 = \beta_{Q+1} < \beta_Q < \cdots < \beta_1 < \beta_0 = 1$ and starting from $q = Q$ with a relatively small β_Q, we first compute the weights by evaluating the tempered likelihood at $\boldsymbol{\rho}_{t,q+1}^{(m)}$ (Step 2b), then each particle is resampled from the whole set of M particles with probability equal to the associated normalized weights (Step 2c). Finally, the new set of particles is simulated according to a transition kernel (Step 2d). We perform one step of Metropolis-Hastings (M-H) algorithm to sample from the transition kernel $\mathcal{T}_{t,q+1}(\cdot)$ with proposal $\mathcal{N}(\cdot, \kappa_{q+1}^2\mathbf{I})$, where κ_{q+1}^2 is the variance of the kernel [Robert & Casella 2004], as it is common in sequential Monte Carlo samplers [Del Moral *et al.* 2006]. Note that in the last step, with $q = 0$, the exponent is $\beta_0 = 1$, and therefore the M-H targets the true posterior pdf of the kernel parameters, which ensures the invariance of the particle approximation.

Rationale and parameter selection. The rationale of GIAnPF is as follows. The first exponent, β_Q, is chosen in such a way the targeted distribution is a sufficiently flattened version of the likelihood. Then, for each annealing layer $q = Q,\ldots,1$, a larger β_{q-1} than the

Table 3.2: Generalized Interacting Annealed PF for space-varying blur identification.

1. Initialization. Draw M i.i.d. samples, $\{\boldsymbol{\rho}_{0,0}^{(m)}\}_{m=1}^{M}$ from the prior $p(\boldsymbol{\rho}_0)$. Set $Q \in \mathbb{N}$.
2. Bayesian recursion. For $t = 1, ..., T$:

   a) Propagate the particles as

$$\boldsymbol{\rho}_{t,Q+1}^{(m)} \sim p(\boldsymbol{\rho}_{t,Q+1}|\boldsymbol{\rho}_{t-1,0}^{(m)}), \quad m = 1, ..., M \tag{3.23}$$

   For $q = Q, Q-1, \ldots, 0$,

   b) Compute the normalized tempered weights as:

$$\xi^{(m)} \propto p(\mathbf{y}_t|\boldsymbol{\rho}_{t,q+1}^{(m)}, \mathbf{X}_t)^{\beta_q - \beta_{q+1}}, \quad m = 1, \ldots, M \tag{3.24}$$

   where $0 = \beta_{Q+1} < \beta_Q < \cdots < \beta_1 < \beta_0 = 1$

   c) resample

$$\overline{\boldsymbol{\rho}}_{t,q+1}^{(m)} = \boldsymbol{\rho}_{t,q+1}^{(j)}, \text{with probability } \xi^{(j)},$$

   for each $m = 1, ..., M$

   d) sample as

$$\boldsymbol{\rho}_{t,q}^{(m)} \sim \mathcal{T}_{t,q+1}(\overline{\boldsymbol{\rho}}_{t,q+1}^{(m)}), \quad m = 1, \ldots, M \tag{3.25}$$

3. Output: At each t, we provide a set of M unweighted particles $\{\boldsymbol{\rho}_{t,0}^{(m)}\}_{m=1}^{M}$, for the approximation of the filtering distribution.

previous one is adopted so that the likelihood used when evaluating the weights becomes increasingly closer to the original one. The final particles at this time step t, $\{\boldsymbol{\rho}_{t,0}^{(m)}\}_{m=1}^{M}$, are then those used for the approximation of the filtering distribution.

This annealing procedure is especially helpful when the likelihood varies a lot among the particles, so that only few particles would have been chosen if the exact likelihood was used to do the sampling. In Step 2b), we compute the tempered weights to deal with the particle degeneracy problem. In Step 2c), the particle $\overline{\boldsymbol{\rho}}_{t,q+1}^{(m)}$ is resampled from the set of simulated particles, $\{\boldsymbol{\rho}_{t,q+1}^{(j)}\}_{j=1}^{M}$, with associated probabilities $\{\xi^{(j)}\}_{j=1}^{M}$. In Step 2d), we apply one step of the M-H to the re-sampled particles, to ensure that the particle approximation converges to the true targeted distribution (i.e., without the tempering exponent). Note that GIAnPF can be seen as a generalized version of the BPF, since it contains this algorithm as particular case when we set $Q = 0$ and the M-H iteration (Step 2d) is avoided, i.e., with $\kappa_{q+1} = 0$.

Connections to the literature. The GIAnPF connects with several algorithms in the literature. As stated above, it can be seen as a generalization of the BPF [Gordon *et al.* 1993]. It also holds clear links with annealing schemes such as [Neal 2001, Deutscher *et al.* 2000, Filippone 2014, Ulker *et al.* 2011]. In the context of PF, [Liu *et al.* 2021b] conducts interacting MCMC sampling procedure with the particles obtained from the BPF as the starting point. In [Madapura & Li 2007], the authors incorporate the annealing strategy within a Rao-Blackwellised Particle Filter and update the sample size using Kullback-Leibler Di-

vergence transformation. The algorithm proposed in [Gall *et al.* 2005] also bears some similarities with GIAnPF in its use of an annealing scheme in a PF. However, unlike in [Gall *et al.* 2005], our proposed method keeps closer ties to the standard BPF by modifying the sampling/weighting and adding an M-H update, which allows the particle approximation to converge to the filtering distribution with M (for any value of Q), and hence constructing consistent estimators of the sought intractable integrals.

3.4 Experimental results

3.4.1 Construction of an experimental database

We first describe the experimental database. It is composed of a set of several pairs of clean/degraded images associated to various choices for the patch decomposition, shift variant blur maps, and noise levels.

3.4.1.1 Images

We consider three images of size $N = 512 \times 512$ (displayed in Fig. 3.4), namely, `Chart`, `Cells`, and `Hubble`. These are representative images of calibrated objects (e.g., resolution charts, fluorescent beads and distant stars) typically used to estimate spatially variant blur parameters. In all our experiments, patches of size 64×64 are used and the blurry noisy version $\mathbf{y}$ is generated following Eqs. (3.1)-(3.2), using i.i.d. zero-mean Gaussian noise with standard deviation $\sigma_n > 0$, so that $p(\mathbf{y}_t|\boldsymbol{\rho}_t) = \mathcal{N}(\mathbf{X}_t h(\boldsymbol{\rho}_t), \sigma_n^2 \mathbf{I})$.

3.4.1.2 Shift-variant blur maps

Five realistic space-varying blur maps, associated to the three blur shapes presented in Section 3.2.2, are constructed. To this aim, we take inspiration from the examples provided in [Escande & Weiss 2015, Fig.8], [Berisha & Nagy 2014, Fig.7], [Berisha & Nagy 2014, Fig.7] and [Simpkins & Stevenson 2014, Fig.4]. In most of our models, we will consider the existence of an optical center for the device, and that the blur width/orientation depends on the relative position of the patch with respect to this center. Instead of considering deterministic maps, as in the aforementioned works, we include some randomized settings in the expressions so as to simulate more realistic scenarios of an optical device whose effect on the image may slightly vary with respect to external conditions (temperature, laser power, planarity of the disposal). For a pre-specified abscissa-ordinate system in the 2D image plan, we denote, for a given patch t, $\mathbf{o}_t = (o_{1,t}, o_{2,t})$ the coordinates of the patch center, $\hat{\mathbf{o}} = (\hat{o}_1, \hat{o}_2)$ the coordinates of the chosen optical center of the device, and $\ell_t = \|\mathbf{o}_t - \hat{\mathbf{o}}\|$ the Euclidian distance between both. The associated parameters, gathered in vector $\boldsymbol{\rho}_t$, are defined as follows, for the five considered models. Figure 3.3 displays examples of three realizations of each of those, illustrating the variability among patches, and also among different realizations for the same model.

Map 1 We assume Gaussian blurs, with size $L = 15^2$, with width and orientation parameters $\mathbf{s}_t$ and θ_t at each patch t. We opt for the spiral patch ordering and we set, for every t,

$$\theta_t = \arctan\left(\frac{o_{1,t} - \hat{o}_1}{o_{2,t} - \hat{o}_2}\right) + \varepsilon_t, \tag{3.26}$$

where $\varepsilon_t \sim \mathcal{N}(\frac{\pi}{2}, \sigma_\varepsilon^2)$,

$$s_{1,t} = \delta_{1,t}\ell_t, \quad s_{2,t} = \delta_{2,t}\ell_t. \tag{3.27}$$

where $\delta_{1,t} \sim \mathcal{U}([\delta_{1,\min}, \delta_{1,\max}]), \delta_{2,t} \sim \mathcal{U}([\delta_{2,\min}, \delta_{2,\max}])$ with $(\sigma_\varepsilon, \delta_{1,\min}, \delta_{1,\max}, \delta_{2,\min}, \delta_{2,\max})$ positive scalars.

Map 2 We consider Gaussian blurs with size $L = 15^2$ with width and orientation parameters $\mathbf{s}_t$ and θ_t at each patch t. We use the spiral patch ordering and, for every t, we set

$$\theta_t = \arctan\left(\frac{o_{1,t} - \hat{o}_1}{o_{2,t} - \hat{o}_2}\right) + \varepsilon_t, \tag{3.28}$$

where $\varepsilon_t \sim \mathcal{N}(\frac{\pi}{2}, \sigma_\varepsilon^2)$,

$$\begin{cases} s_{1,t} = \frac{1}{8}(\delta_{1,t}|o_{1,t} - \hat{o}_1| + 1) \\ s_{2,t} = \frac{1}{8}(\delta_{2,t}|o_{2,t} - \hat{o}_2| + 1) \end{cases} \tag{3.29}$$

with $\delta_{1,t} \sim \mathcal{U}([\delta_{1,\min}, \delta_{1,\max}]), \delta_{2,t} \sim \mathcal{U}([\delta_{2,\min}, \delta_{2,\max}])$ and $(\sigma_\varepsilon, \delta_{1,\min}, \delta_{1,\max}, \delta_{2,\min}, \delta_{2,\max})$ some positive scalars.

Map 3 We consider Gaussian blurs with size $L = 15^2$ with width and orientation parameters $\mathbf{s}_t$ and θ_t at each patch t. The patches are ordered in lexicographic order and, for every t, we set

$$\theta_t \sim \mathcal{U}([0, \frac{\pi}{8}]), \tag{3.30}$$

and

$$\begin{cases} s_{1,t} = \frac{1}{8}(\delta_{1,t} - o_{1,t}), \\ s_{2,t} = \frac{1}{15}(\delta_{2,t} - o_{2,t}), \end{cases} \tag{3.31}$$

where $\delta_{1,t} \sim \mathcal{U}([\delta_{1,\min}, \delta_{1,\max}]), \delta_{2,t} \sim \mathcal{U}([\delta_{2,\min}, \delta_{2,\max}])$, and $(\delta_{1,\min}, \delta_{1,\max}, \delta_{2,\min}, \delta_{2,\max})$ are some positive scalars.

Map 4 We assume blurs with defocus shape, with size $L = 25^2$ parametrized by width and orientation parameters $\mathbf{s}_t$ and θ_t at each patch t. We order the patches in lexicographic order and, for every t, we set

$$\theta_t \sim \mathcal{U}([0, \frac{\pi}{8}]), \tag{3.32}$$

and

$$\begin{cases} s_{1,t} = \frac{1}{3}(\delta_{1,t} - o_{1,t}), \\ s_{2,t} = \frac{1}{6}(\delta_{2,t} - o_{2,t}), \end{cases} \tag{3.33}$$

with $\delta_{1,t} \sim \mathcal{U}([\delta_{1,\min}, \delta_{1,\max}]), \delta_{2,t} \sim \mathcal{U}([\delta_{2,\min}, \delta_{2,\max}])$, and $(\delta_{1,\min}, \delta_{1,\max}, \delta_{2,\min}, \delta_{2,\max})$ some positive scalars.

Map 5 We specify here a generative model for the particular case of the skew-normal blur, with size $L = 15^2$, with the aim to mimic the shift-variant blur map that was considered in the experiments of [Simpkins & Stevenson 2014]. We order the patches in spiral order and we set

$$\omega_t = \operatorname{sign}\left(\frac{o_{1,t} - \hat{o}_1}{o_{2,t} - \hat{o}_2}\right)\left(1 - 2\big||o_{1,t} - \hat{o}_1| - |o_{2,t} - \hat{o}_2|\big|\right). \tag{3.34}$$

We then create a shift-variant map with skew normal shapes of parameters

$$\begin{cases} s_{1,t} = \frac{1}{3}(|o_{2,t} - \hat{o}_2| + \frac{1}{2})(\ell_t + \frac{1}{5}) + \delta_{1,t}, \\ s_{2,t} = \frac{1}{3}(|o_{1,t} - \hat{o}_1| + \frac{1}{2})(\ell_t + \frac{1}{5}) + \delta_{2,t}, \\ w_t = \frac{7}{5}\omega_t \ell_t + \delta_{3,t}, \\ \alpha_{1,t} = \delta_{4,t}(1 - \ell_t), \\ \alpha_{2,t} = \delta_{5,t}(1 - \ell_t), \end{cases} \tag{3.35}$$

with $\delta_{1,t} \sim \mathcal{U}([0, \delta_{1,\max}])$, $\delta_{2,t} \sim \mathcal{U}([0, \delta_{2,\max}])$, $\delta_{3,t} \sim \mathcal{U}([0, \delta_{3,\max}])$, $\delta_{4,t} \sim \mathcal{U}([\delta_{4,\min}, \delta_{4,\max}])$, $\delta_{5,t} \sim \mathcal{U}([\delta_{5,\min}, \delta_{5,\max}])$, where $(\delta_{1,\max}, \delta_{2,\max}, \delta_{3,\max}, \delta_{4,\min}, \delta_{4,\max}, \delta_{5,\min}, \delta_{5,\max})$ are positive scalars.

Table 3.3 shows the settings for the realistic blur maps considered in Section 3.4.1.2.

Blur maps	Parameters
Map 1	$\sigma_\varepsilon = 0.01, \delta_{1,\min} = 0.595, \delta_{1,\max} = 0.605,$ $\delta_{2,\min} = 0.295, \delta_{2,\max} = 0.305$
Map 2	$\sigma_\varepsilon = 0.01, \delta_{1,\min} = 2.95, \delta_{1,\max} = 3.05,$ $\delta_{2,\min} = 2.95, \delta_{2,\max} = 3.05$
Map 3	$\delta_{1,\min} = 2.45, \delta_{1,\max} = 2.55,$ $\delta_{2,\min} = 1.95, \delta_{2,\max} = 2.05$
Map 4	$\delta_{1,\min} = 1.45, \delta_{1,\max} = 1.55,$ $\delta_{2,\min} = 1.45, \delta_{2,\max} = 1.55$
Map 5	$\delta_{1,\max} = 0.01, \delta_{2,\max} = 0.01, \delta_{3,\max} = 0.1,$ $\delta_{4,\min} = 1.5, \delta_{4,\max} = 2, \delta_{5,\min} = 2, \delta_{5,\max} = 2.5$

Table 3.3: Numerical settings for the generation of the experimental shift-variant blur maps.

3.4.1.3 Quantitative comparison methodology

We will run several experiments of blur map estimation, using BPF and GIAnPF methods, both on toy examples and on our realistic dataset. Table 3.4 lists the transition models assumed when running BPF and GIAnPF, where $\mathcal{N}_{[s_{\min},s_{\max}]}(s_{t-1},\sigma_s^2)$ denotes a truncated normal distribution, i.e., the pdf is proportional to a normal pdf with mean s_{t-1} and variance σ_s^2 in the support $[s_{\min},s_{\max}]$, and 0 otherwise. More precisely, Model A will be

	Parameters settings	
Model A	$\begin{cases}\theta_t \sim \mathcal{N}(\theta_{t-1},\sigma_\theta^2)\\ s_{1,t} \sim \mathcal{N}_{[s_{\min},s_{\max}]}(s_{t-1},\sigma_s^2)\\ s_{2,t} \sim \mathcal{N}_{[s_{\min},s_{\max}]}(s_{t-1},\sigma_s^2)\end{cases}$	$s_{\min}=10^{-2}$, $s_{\max}$ set as in (3.5).
Model B	$\begin{cases}\theta_t \sim \mathcal{N}(\theta_{t-1},\sigma_\theta^2)\\ s_{1,t} \sim \mathcal{N}_{[s_{\min},s_{\max}]}(s_{t-1},\sigma_s^2)\\ s_{2,t} \sim \mathcal{N}_{[s_{\min},s_{\max}]}(s_{t-1},\sigma_s^2)\end{cases}$	$s_{\min}=4\cdot 10^{-2}$, $s_{\max}$ as in (3.10).
Model C	$\begin{cases}s_{1,t} \sim \mathcal{N}_{[s_{\min},s_{\max}]}(s_{t-1},\sigma_s^2)\\ s_{2,t} \sim \mathcal{N}_{[s_{\min},s_{\max}]}(s_{t-1},\sigma_s^2)\\ w \sim \mathcal{N}_{[w_{\min},w_{\max}]}(s_{t-1},\sigma_w^2)\\ \alpha_{1,t} \sim \mathcal{N}_{[\alpha_{\min},\alpha_{\max}]}(s_{t-1},\sigma_\alpha^2)\\ \alpha_{2,t} \sim \mathcal{N}_{[\alpha_{\min},\alpha_{\max}]}(s_{t-1},\sigma_\alpha^2)\end{cases}$	$s_{\min}=5\cdot 10^{-2}$, $s_{\max}$ as in (3.15), $\{w_{\min},w_{\max}\}=\{-0.9,0.9\}$, $\{\alpha_{\min},\alpha_{\max}\}=\{0.5,2\}$, so that (3.17) holds.

Table 3.4: Considered transitions models as in Eq. (3.19).

employed to retrieve Gaussian blur maps (e.g., those resulting from Maps 1, 2 and 3), Model B to retrieve defocus blur maps (e.g., in case of Map 4) and Model C to estimate skew normal blur maps (e.g., Map 5). The upper bounds $s_{\max}$ defined are used, to preserve consistency of the estimated shapes with the considered blur support width L. An analysis of robustness to the setting for L will be presented in Section 3.4.4.4. In order to quantify the gain of exploiting the spatial structure of the problem, we will compare the filtering pdf approximated by BPF and GIAnPF algorithms, and the posterior distribution that only considers the data corresponding to each patch independently. In the latter case, we use IS [Elvira *et al.* 2019b] to approximate the intractable posterior. Note that both PF and IS have a comparable computational complexity per sample/particle, while PF allows to exploit the information of previously processed data (i.e., patches), due to an implicit sequential IS structure. We also perform comparisons with two optimization-based methods considering non-parametric models for the kernels. In the so-called NP formulation, the kernels $(\mathbf{h}_t)_{t=1}^T$ are estimated by minimizing a least-squares function under simplex and smoothness constraints, in a fully parallel manner for each patch t. We also compared with a more sophisticated non-parametric formulation where we included, in addition to the NP cost function, the total-variation based spatial regularization among kernels of neighboring patches from [Mourya 2016, Chap.4], yielding NP$^+$ method. Note that both resulting constrained convex minimization problems are solved with the FISTA algorithm [Beck & Teboulle 2009]. In all experiments, the relative mean squared error (RMSE) of blur kernel (RMSE$_h$) averaged over patches is used to evaluate the numerical performance of the blur estimation, i.e., $\text{RMSE}_h = \frac{1}{T}\sum_{t=1}^T \frac{||\overline{\mathbf{h}}_t-\mathbf{h}_t||_2}{||\mathbf{h}_t||_2}$ with $\mathbf{h}_t$ and $\overline{\mathbf{h}}_t$ the original kernels and the estimators re-

spectively at patch t. For methods providing a posterior estimation (i.e., BPF, GIAnPF and IS), we calculate $\overline{\mathbf{h}}_t = h(\overline{\boldsymbol{\rho}}_t)$ with $\overline{\boldsymbol{\rho}}_t$ the mean estimator of the unknown parameters, while for NP and NP$^+$, $\overline{\mathbf{h}}_t$ is directly the solution to the optimization problem, in patch t. We also include the standard deviation as well as the 95% credible interval of the estimators of BPF, GIAnPF and IS methods, to evaluate the statistical accuracy of the results. All the presented results are averaged over 100 random runs, and all the hyperparameters are tuned so as to minimize the RMSE (averaged over 5 random trials). The best results will be marked in bold cases in the tables. The numerical experiments are conducted in a Matlab environment on a computer with an Xeon(R) W-2135 processor (3.7 GHz clock frequency) and 12 GB of RAM.

3.4.2 Validation of the proposed method

We first discuss the settings and properties of the proposed method in an illustrative toy synthetic example. Our tests are conducted on the image `Chart` decomposed into $T = 64$ patches, and corrupted by skew-normal blurs whose parameters $(\boldsymbol{\rho}_t)_{t=1}^T$ are generated following the transition model C, with $(\sigma_s, \sigma_w, \sigma_\alpha) = (10^{-1}, 2\cdot 10^{-1}, 10^{-1})$ and, except otherwise stated, a noise standard deviation $\sigma_n = 0.05$. The blur parameter estimations are then conducted with BPF and GIAnPF, assuming transition Model C with known hyperparameters.

Setting particle and layer numbers The BPF method, as well as the proposed GIAnPF method, requires the setting of the number of particles. For GIAnPF, the number of layers Q plays also a role. This is necessary to reach a compromise between a precise target reconstruction and a reasonable computational time, for setting those parameters. We display in Fig. 3.5 the averaged RMSE on 100 runs with respect to the computational time, for different settings of the parameters of BPF and GIAnPF. The associated values for the parameters $(\beta_q)_{0\leq q\leq Q}$ are chosen on a logarithmic grid between 1 and 10^{-2}, and $\beta_{Q+1} = 0$. For a given number of particles M, the GIAnPF algorithm always generates a better approximation of the unknown blurs than BPF, although at the expense of a slightly increased. Note that, for the same time budget, GIAnPF outperforms BPF. In all the upcoming experiments, we will set $M = 3000$ for BPF, IS, and $Q = 2$ and $M = 500$ for GIAnPF, so that they have comparable time requirement.

Alleviating the particle degeneracy One of the advantages of the novel GIAnPF method is the promotion of the diversity among the particles. In Fig. 3.6, we display the normalized effective sample size (NESS), computed as $\text{NESS} = \frac{1}{M\sum_{m=1}^M \overline{w}_t^{(m)}}$, where $\overline{w}_t^{(m)}$ are the normalized weights (i.e., $0 \leq \text{NESS} \leq 1$). We average the effective sample size (ESS) (see [Kong 1992, Kong *et al.* 1994, Elvira *et al.* 2018b]) over 100 runs and all the patches, divided by the corresponding number of particles M, for various level of noise σ_n. The results are also averaged over 10 independent generations of the data. It can be seen that GIAnPF algorithm has higher NESS, thus preserves much more diversity in the particles for all analyzed noise levels. Similarly, Fig. 3.7 displays, for the same experiment, the RMSE of the

blur estimate using the mean of the approximate posterior of each filter. We can see that a larger diversity (larger NESS) translates into a smaller error.

3.4.3 Comparative performance on the experimental database

In this section, we compare the performance of GIAnPF and other approaches in different scenarios arising from our experimental database.

Quantitative performance Table 3.6 displays the RMSE_h obtained when estimating kernels of Maps 1 to 5, from observations of `Cells` image, degraded by two noise levels, namely $\sigma_n = 0.01$ and $\sigma_n = 0.05$. Illustrative examples of results are also provided in Figure 3.8. In all the Bayesian methods, we display the blur maps resulting from kernel parameters equal to the MMSE estimator, i.e., the mean of the posterior distribution given the available data. The methods IS and NP, which do not exploit the spatial smoothness among neighboring kernels, reach the worst performance. A visual inspection of Figure 3.8 confirms this result. Among the spatially regularized methods, i.e., NP+, BPF and GIAnPF, the former is still far from reaching the quality of estimation of two latter methods though it improves over NP. Such behavior is expected as BPF and GIAnPF incorporate explicit knowledge about the parametric shape of the blur, leading to less diversity and thus less error in the restored blur maps. This can be seen in Figure 3.8. An analysis of the sensitivity of the methods over an error in the assumed parametric shape will be discussed in Section 3.4.4.1. In all cases presented in Table 3.6, the proposed method GIAnPF performs similarly or better than the standard BPF. The benefits from the annealing procedure are particularly noticeable in the case of $\sigma_n = 0.01$. This improved performance can be explained by the annealed approach of GIAnPF, which is particularly effective when the posterior pdf of the parameters has the probability mass concentrated in small regions of the space, as it happens with low values of σ_n (due to a *peaky* likelihood).

Complexity comparison We perform a scalability analysis of the different methods, by applying those to the estimation of Map 5 blur map, from resized versions of the large scale aerial image from [Chouzenoux & Pesquet 2017]. The same hyperparameter settings as in Table 3.6 were used. In Table 3.5, we report the averaged computational time over 10 random trials. For IS and NP, that are fully separable methods onto patches, we also report the time per patch, that could be reached using parallel implementation on T cores. Both of these methods benefit from fast complexity cost assuming parallel processing of the patches. It is worthy to notice that the cost per patch of NP still increases for the largest images, probably due to memory saturation issues, while IS has a constant time requirement per patch. However, despite their rapidity, let us recall that the qualitative results for both these methods were rather poor on our experiments. The complexity burden dramatically increases for the regularized optimization-based method NP^+. In the latter, the data is processed in a batch way. Parallelization over patches for this method is not straightforward, up to our knowledge, due to the non-separable structure of the underlying optimization problem. In contrast, the complexity of the proposed method stays very reasonable even for the largest image example, though involving the probabilistic estimation

Image size	$N = 256^2$	$N = 512^2$	$N = 1024^2$	$N = 2048^2$
Patches	$T = 16$	$T = 64$	$T = 256$	$T = 1024$
NP	36.16 (2.26)	135.5 (2.11)	4987 (19.48)	82273 (80.34)
NP$^+$	35.80	127.3	4473	71913
IS	5.21 (0.32)	21.85 (0.34)	91.29 (0.36)	374.7 (0.37)
BPF	5.09	20.15	85.53	345.4
GIAnPF	4.67	19.03	78.81	329.9

Table 3.5: Averaged computational time in seconds, for the different methods, in the case of Map 5 (i.e., $L = 225$ and $K = 5$) applied to various images of size N, decomposed into T patches. Time per patch is also reported for IS and NP, as they process patches independently.

of $T \times K = 5120$ parameters. As in the previous experiment, BPF and GIAnPF again show a similar computational cost.

Stability to the noise level We display in Fig. 3.9 a comparison of the RMSE_h reached by the methods, when estimating kernels from observations degraded by increasing noise level σ_n. One can observe that NP and NP+ performance rapidly deteriorates when the noise increases. In contrast, IS, BPF and GIAnPF perform in a relatively stable way. Moreover, the proposed GIAnPF method again outperforms the other competitors. As expected, its superiority over the standard BPF is less visible as σ_n becomes higher, since GIAnPF is particularly dedicated to problems with peaky likelihoods, which arise when the noise level is low.

3.4.4 Robustness analysis

In practical contexts, it is frequent that the assumed parametric PSF model is erroneous (e.g., over-simplified with wrong support size) and/or that the calibrated image is only partially known (e.g., in digital camera imaging, a noisy non-blurry version can be acquired using short exposure settings). Furthermore, the noise level in the blurred image is usually estimated and not known with perfect accuracy. It is thus of main importance to quantify the robustness of our method to such model mismatch. To this aim, we conducted four experiments, aiming at reproducing different realistic scenarios of imperfect knowledge of the acquisition model. Except if specified otherwise, in all the examples of this section, we used the image `Cells`. The degraded image is obtained by applying the space-varying Gaussian blur Map 1, and then adding a noise with level equals to $\sigma_n = 0.05$.

3.4.4.1 Choice of the parametric blur model

We first evaluated the influence of a mismatch between the assumed blur shape, and its actual one. We ran the different methods assuming instead a generalized Gaussian blur shape with different powers p, where $p = 1$ corresponds to the ground truth. The retrieved average RMSE_h of BPF, GIAnPF, IS when taking different values for p are displayed in

Fig. 3.10. We also displayed the results of NP and NP$^+$ approaches, that remain unchanged since no parametric model for the kernels is considered in those methods. As expected, the best performance are obtained for the correct $p = 1$ setting. More interestingly, we can observe that, even if a wrong parametric model is adopted, i.e., using $p \neq 1$, our proposed method GIAnPF still performs better than its competitors, and in particular the non parametric method NP$^+$, for a wide range of values for p.

3.4.4.2 Noisy calibrated image

In the previous experiments, we assumed that the original image $\mathbf{x}$ is known in a perfect manner. However, in reality, it is commonly not the case. Here, we consider the situation when only a noisy version of $\mathbf{x}$ is provided in the estimation process, denoted $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{w}$ where $\mathbf{w}$ is an additive Gaussian noise follows $\mathcal{N}(0, \sigma_x^2 \mathbf{I})$. Eleven different levels for the noise on $\mathbf{x}$ are considered, namely $\sigma_x = \{0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10\}$. Note that this range of values covers from low noise level values (where traditionally the particle-based filters struggle to operate due to the mass being concentrated in small regions; see for instance [Del Moral *et al.* 2006]) to relatively large noise values (which makes the inference less accurate). We display in Fig. 3.11 the averaged RMSE over 10 noisy realizations for each σ_x, for the different approaches. The improvement of our proposed method GIAnPF with respect to BPF is more significant at low noise level, since particle degeneracy mostly arises in such context. Moreover, Fig. 3.11 shows that GIAnPF still outperforms its competitors when the noise level on $\mathbf{x}$ increases.

3.4.4.3 Setting of noise level

In the previous analysis, the exact observation noise variance σ_n was assumed to be known and applied explicitly in our blur identification method. In more realistic situations, such noise level would have to be learned, leading to an estimate $\hat{\sigma}_n$. We ran the different blur identification methods for the five different values $\hat{\sigma}_n = \{0.01, 0.025, 0.05, 0.07, 0.1\}$ (recall that the true observation noise level here is $\sigma_n = 0.05$). Note that the optimization-based methods are not sensitive to such change, so that the results for NP and NP$^+$ are unchanged. We display in Fig. 3.12 the average RMSE, for the different values for $\hat{\sigma}_n$. We can notice that GIAnPF gives the best performance even when the noise level is poorly estimated. As expected, it reaches minimal mean and variance values when the ground truth noise level is used in the estimation process.

3.4.4.4 Setting of kernel size

In blur estimation, especially when the blur is non-stationary, a challenge lies in the setting a priori, of the kernel support width L. In this experiment we propose to assess the different approaches in the case where an erroneous L is assumed. In particular, we estimate the blur from the image `Cells`, when it is degraded by Map 1, and a noise level $\sigma_n = 0.05$. We run the different methods/models, by setting the size for the kernel width as $L \in \{5, 7, 9, 11, 13, 15, 19, 23, 27\}$, while the groundtruth is $\overline{L} = 15$. All the true/estimated

kernels have been extended to the largest tested size $L = 27$ by zero-padding, to allow valid computation of estimation errors. We display in Fig. 3.13 the obtained RMSE values. One can notice that all methods have stable estimation error, as long as the assumed L is greater or equal to $\overline{L}$, which is indeed expected. In contrast, when $L < \overline{L}$, the results deteriorate for all methods. We observe that the GIAnPF method outperforms its competitors in almost all tested values for L.

3.4.5 Image restoration

We conclude this experimental section by illustrating the ability of the obtained kernel estimates to lead to satisfying and interpretable image restoration results. We assume that we have access to a pair $(\mathbf{x}, \mathbf{y})$ of clean/degraded version of a given calibrated image. We focus on the estimation of blurs resulting from our synthetic blurred maps, and then the use of those results for the restoration of several non-calibrated images. Through this example, we also illustrate and discuss the advantage of providing probabilistic blur estimates.
We consider the image `Chart`, and its degraded version corrupted by the space-varying Map 4 with $T = 64$ and noise with standard deviation $\sigma_n = 0.1$. We ran GIAnPF method and NP$^+$ approaches, identified as the two best blur identification procedures in the previous section. As shown in Fig. 3.14(left), we then perform the restoration of a set of four natural images[2], from their degraded version in Fig. 3.14(right), following the same blur model, and noise level $\sigma_x = 0$, 0.01, 0.05 and 0.1. The restoration is performed in a non-blind manner, by making use of the fast majorize-minimize memory-gradient (3MG) algorithm from [Chouzenoux *et al.* 2013], that was recently adapted to the problem of image retrieval under shift-varying blur [Cadoni *et al.* 2016, Chalvidal & Chouzenoux 2020]. Note that the variational formulation adopted in 3MG is very similar to the one in [Denis *et al.* 2015]. The regularization parameters of 3MG are finetuned so as to maximize $\mathrm{SNR}_x = 20\log_{10}\left(\frac{||\mathbf{x}||}{||\mathbf{x}-\hat{\mathbf{x}}||}\right)$, with $\mathbf{x}$ the original image, and $\hat{\mathbf{x}}$ the restored one. We present the results in Table 3.7, in terms of SNR_x between $\mathbf{x}$ and the restored images obtained with the blur estimates of either GIAnPF or NP$^+$. In the case of GIAnPF, we ran the restoration with 100 samples from the estimated posterior distribution of the blur maps, for a single run of GIAnPF, which allows us to compute the mean and the 95% confidence intervals for both image quality metrics. In Fig. 3.15(left), we show the mean of the pixel-wise squared error between the true kernel and these 100 samples obtained from the posterior distribution. This metric, called Bayesian mean squared error (MSE), allows to assess the uncertainty of GIAnPF on each kernel estimate. One can identify patches with larger estimation errors, corresponding to flat zones in the `Chart` images, thus making the blur estimation less accurate. In contrast, the patches localized in zones with significant content (sharp edges, in particular) benefit from very good estimation quality for the associated blur (pixel-wise error lower than 10^{-6}). Standard deviation is shown in Fig. 3.15(right), illustrating that the estimated distribution is more spread for difficult patches, while it is rather peaky for the patches that are better estimated. One can see from Table 3.7 that GIAnPF outperforms NP$^+$ in almost all examples, with up to 0.4 improvement in signal-to-noise

[2] http://sipi.usc.edu/database/

ratio (SNR) score. Examples of visual results obtained from a given GIAnPF sample, are also displayed in Fig. 3.14(right).
We then illustrate the usefulness of the probabilistic estimation with associated uncertainty quantification that GIAnPF provides, for the image restoration task. Specifically, we focus on the restoration of `Boat` image degraded by Map 4 and no noise (i.e., $\sigma_x = 0$). We display in Fig. 3.16(left) the uncertainty quantification, defined as the standard deviation per pixel of the 100 restored images, obtained using the 100 samples from the estimated posterior of the blur maps, in a single run of GIAnPF. This uncertainty map can be compared with Fig. 3.16(right). The latter displays the mean square error map, obtained by averaging the results of 100 restorations performed with the MMSE estimator of the kernel parameters (mean of the approximated posterior), obtained by 100 different independent runs of GIAnPF. It is noticeable that our probabilistic approach with the inference performed by GIAnPF; is able to quantify large uncertainty (see Fig. 3.16(left)) in areas where the squared errors are also large (see Fig. 3.16(right)). Let us point out that the uncertainty map from Fig. 3.16(left), is obtained without the need of processing the data multiple times, and does not require the knowledge of the original `boat` image.

3.5 Conclusion

This chapter addressed the estimation of the PSF parameters for spatially-varying blurs from calibrated image acquisitions. We proposed an original statistical modeling of the problem, accounting for the spatial dependency among neighboring kernels, and we applied a sequential Bayesian inference technique in this context. In order to alleviate the particle degeneracy problem brought by the BPF in some cases, we also proposed a new sampling method called the GIAnPF. Our results in different scenarios illustrated the good performance of the approach, including a useful uncertainty quantification. The novel approach opens many possibilities beyond this work. For instance, different noise distributions could be immediately used. Moreover, other state-space models, not necessarily Markovian, could be considered. These perspectives will be discussed thoroughly in Chapter 6.

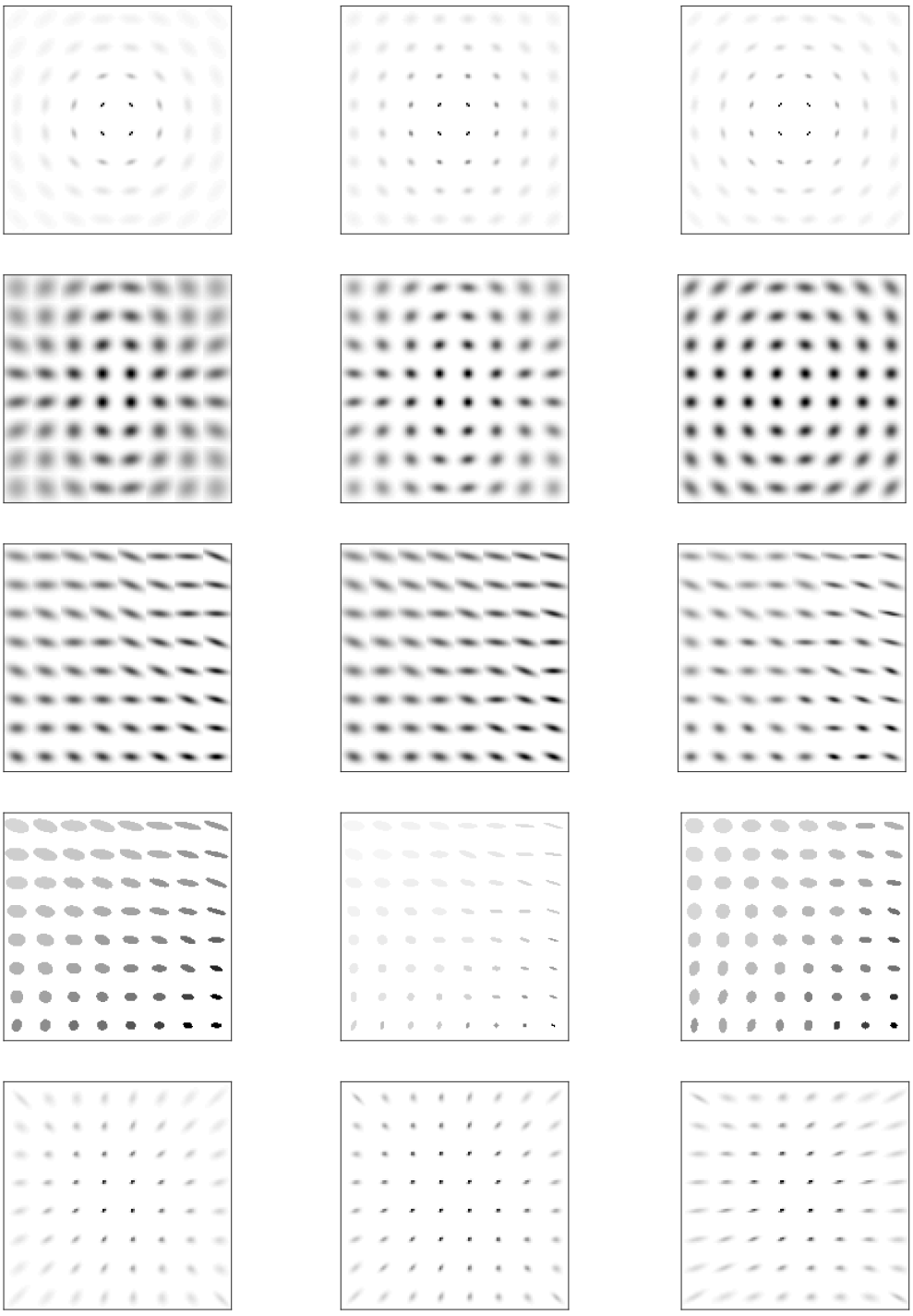


Figure 3.3: Three realizations (from left to right) of the space-varying blur Maps 1 to 5 (from top to bottom). Each column displays one realization from the generative model. One can notice that, for each given row (i.e., blur map), the three images displayed share the same global aspect (e.g., circular map in rows 1, 2 or 5) with some slight variability, for instance in the blur kernel widths and orientations.

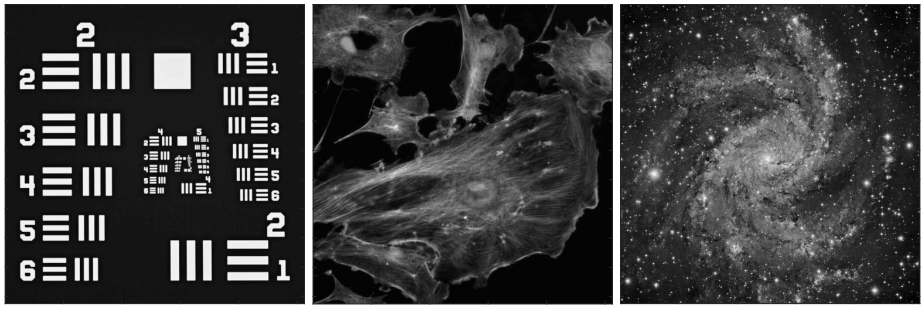


Figure 3.4: Test images `Chart`, `Cells` and `Hubble`.

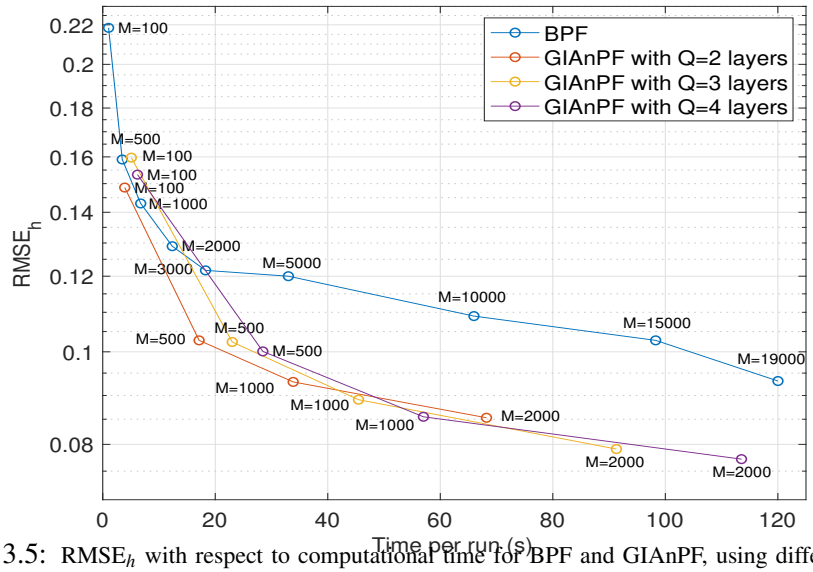


Figure 3.5: RMSE_h with respect to computational time for BPF and GIAnPF, using different settings of (M, Q).

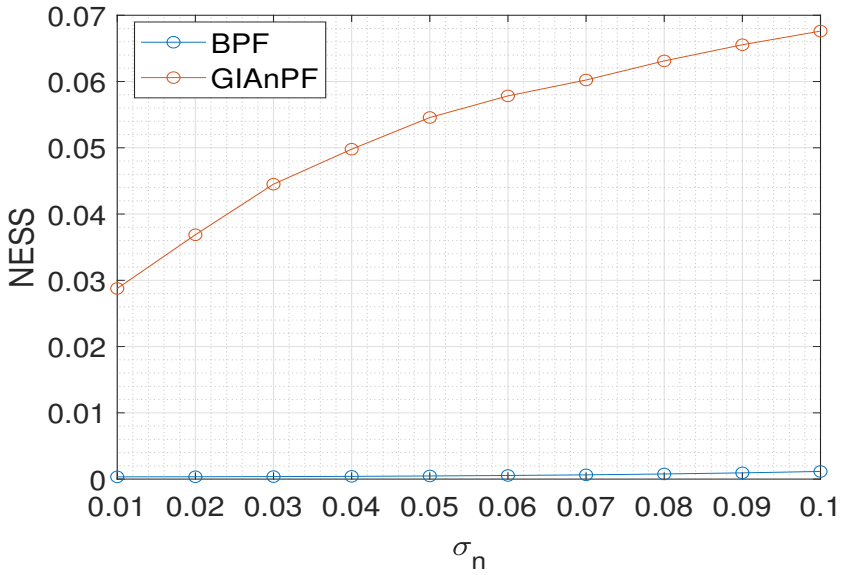


Figure 3.6: Normalized effective sample size (NESS) for BPF (blue) and GIAnPF (red) for various noise level σ_n of the blurry noisy image.

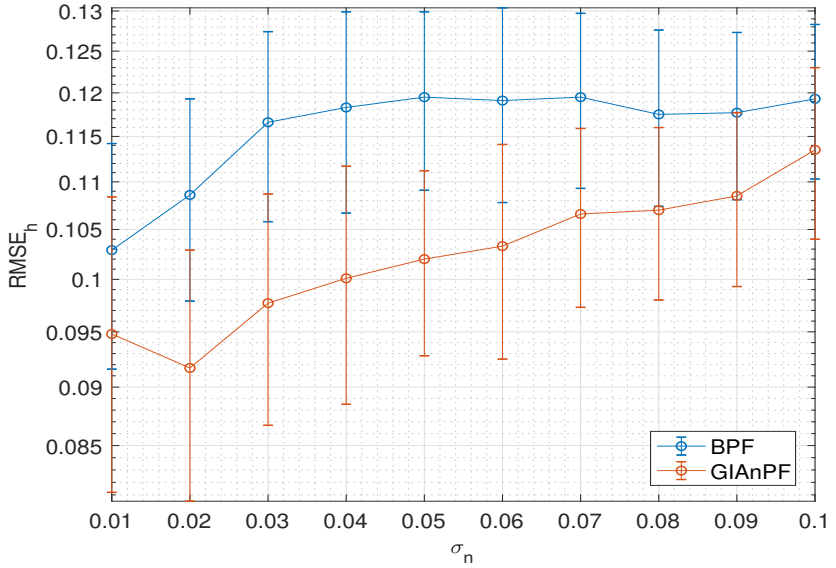


Figure 3.7: RMSE in the blur estimation for BPF (blue) and GIAnPF (red) for various noise level σ_n of the blurry noisy image.

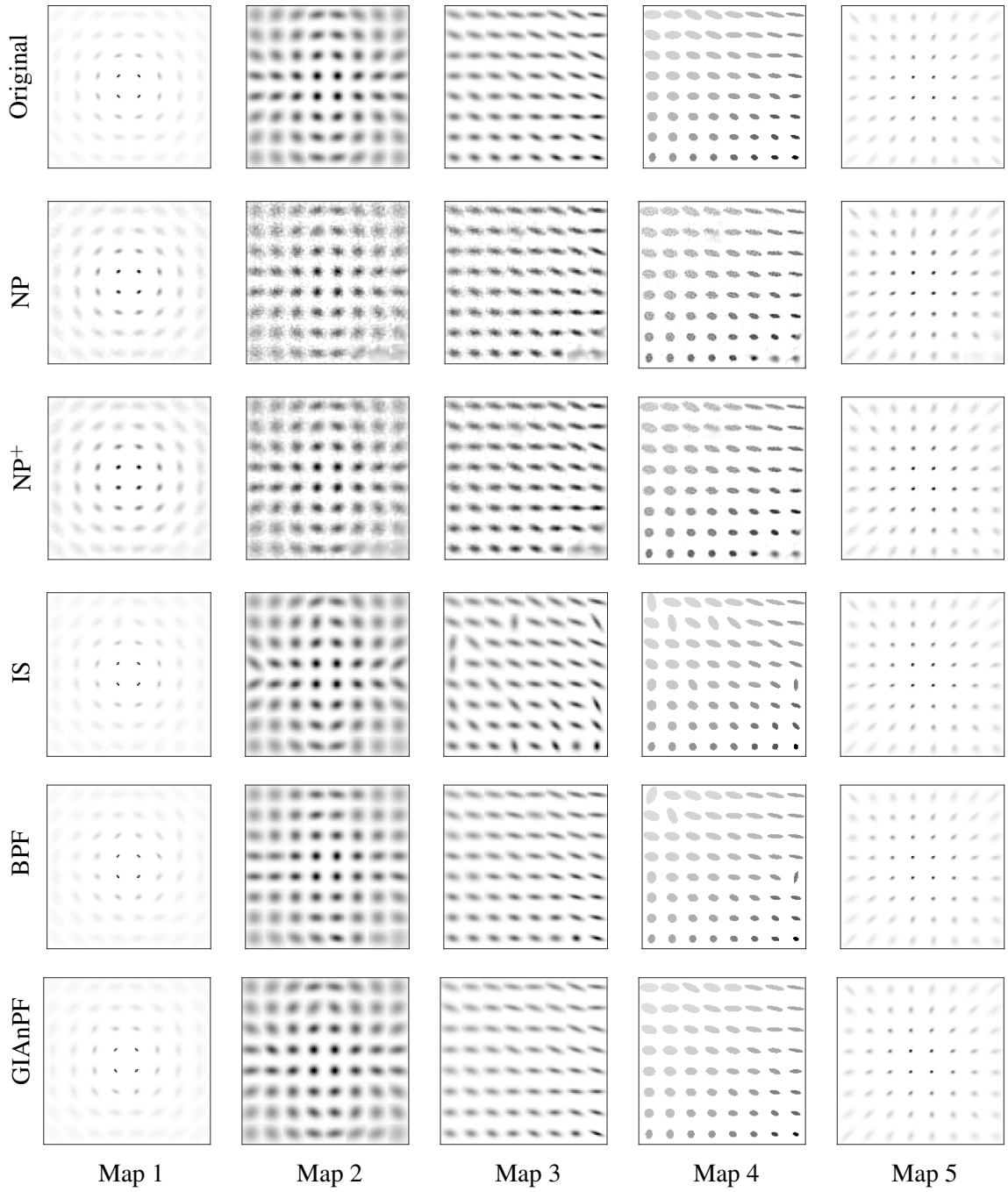


Figure 3.8: Original kernels and identification results using `Cells` image with $T = 64$ patches and $\sigma_n = 0.01$. From top to bottom : original kernels, restored kernels with NP, NP$^+$, IS, BPF, GIAnPF.

		Noise level $\sigma_n = 0.01$	Noise level $\sigma_n = 0.05$
Map 1	NP	0.2117	0.3209
	NP$^+$	0.1756	0.2543
	IS	0.1258 (0.0129) 0.1057-0.1545	0.3339 (0.0157) 0.3056-0.3620
	BPF	0.1246 (0.0080) 0.1109-0.1446	0.1964 (0.0204) 0.1696-0.2646
	GIAnPF	**0.0680** (0.0065) 0.0578-0.0837	**0.1564** (0.0122) 0.1347-0.1858
Map 2	NP	0.2474	0.2381
	NP$^+$	0.1383	0.1385
	IS	0.1653 (0.0072) 0.1471-0.1768	0.1653 (0.0072) 0.1471-0.1768
	BPF	0.0917 (0.0093) 0.0812-0.1211	0.1246 (0.0143) 0.1026-0.1661
	GIAnPF	**0.0452** (0.0044) 0.0375-0.0549	**0.1169** (0.0077) 0.1026-0.1307
Map 3	NP	0.1905	0.2723
	NP$^+$	0.1480	0.1608
	IS	0.1380 (0.0128) 0.1143-0.1626	0.4089 (0.0194) 0.3717-0.4503
	BPF	0.0968 (0.0049) 0.0897-0.1063	0.1854 (0.0550) 0.1145-0.3611
	GIAnPF	**0.0396** (0.0034) 0.0345-0.0486	**0.1288** (0.0114) 0.1116-0.1544
Map 4	NP	0.3263	0.4286
	NP$^+$	0.2988	0.3772
	IS	0.2746 (0.0187) 0.2443-0.3185	0.5554 (0.0204) 0.5208-0.6021
	BPF	0.1360 (0.0098) 0.1174-0.1584	0.2847 (0.0236) 0.2459-0.3411
	GIAnPF	**0.0701** (0.0089) 0.0521-0.0860	**0.2640** (0.0100) 0.2449-0.2833
Map 5	NP	0.1830	0.3578
	NP$^+$	0.1415	0.2428
	IS	0.1331 (0.0066) 0.1198-0.1469	0.1852 (0.0079) 0.1247-0.1407
	BPF	0.0910 (0.0067) 0.0799-0.1066	0.1332 (0.0043) 0.1247-0.1407
	GIAnPF	**0.0688** (0.0075) 0.0587-0.0880	**0.1322** (0.0061) 0.1214-0.1470

Table 3.6: RMSE$_h$ for NP, NP$^+$, IS, BPF, GIAnPF. `Cells` image blurred by space-varying kernels generated following Maps 1-5.

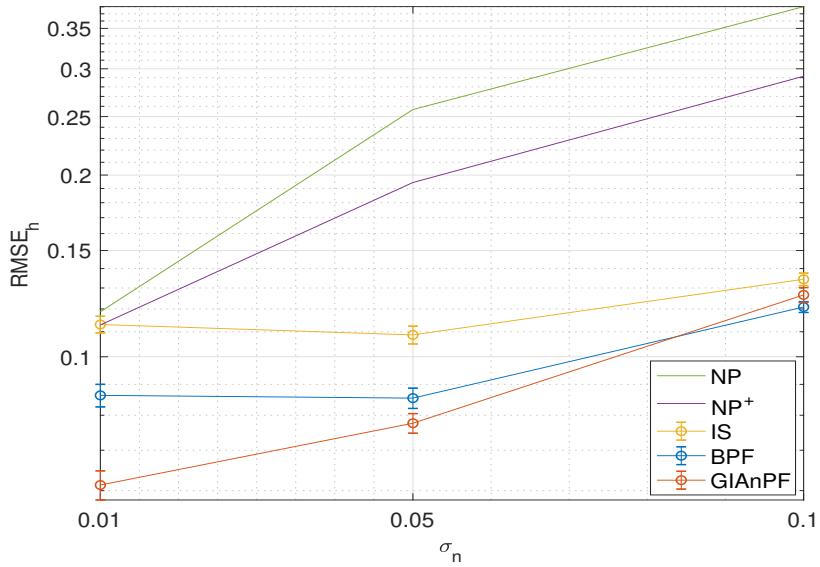


Figure 3.9: Mean/variance of the RMSE$_h$ reached by the different methods. `Hubble` image blurred by space-varying kernels generated following Map 5 and three different values for σ_n.

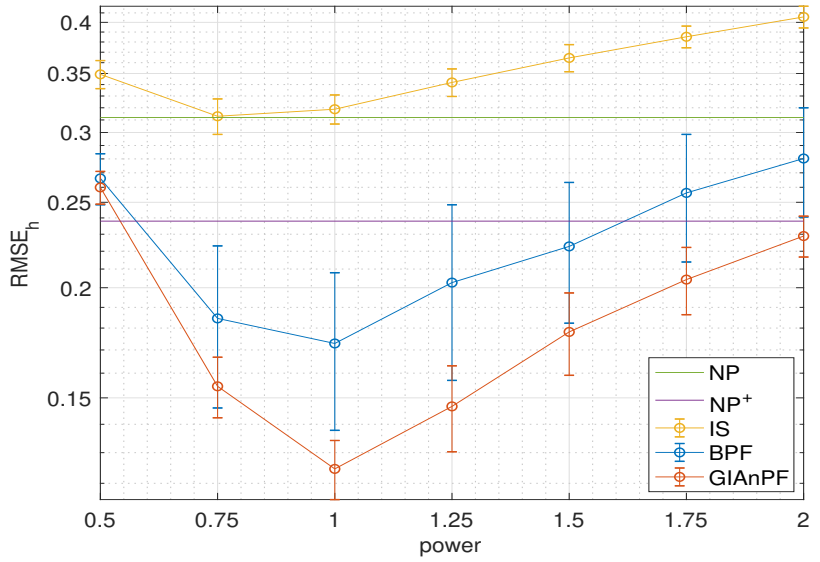


Figure 3.10: Mean/variance of the RMSE$_h$ with the different methods assuming different powers within the generalized Gaussian blur shape (the ground truth is $p = 1$).

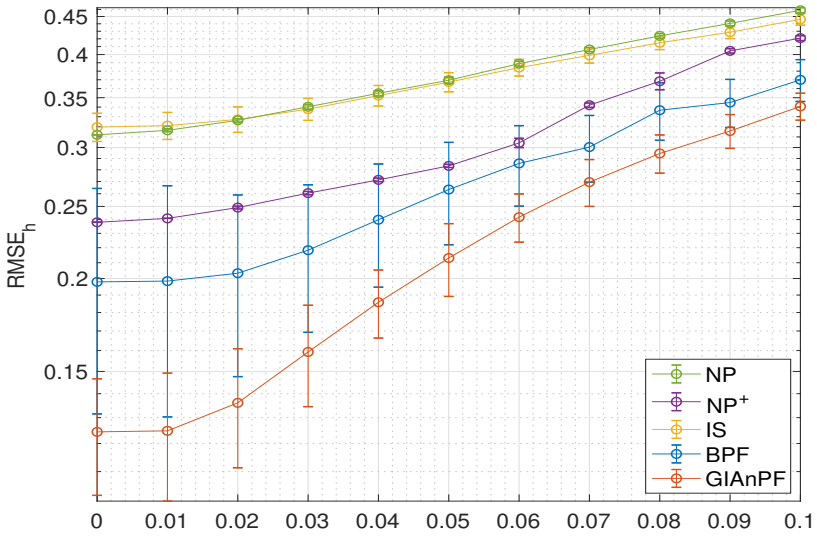


Figure 3.11: Mean/variance of the RMSE_h reached by the different methods, for various noise level σ_x deteriorating the input calibrated image.

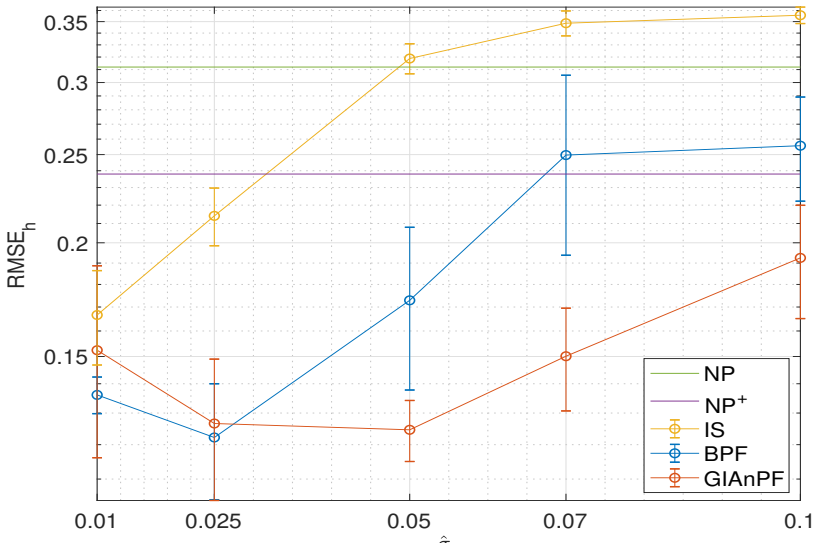


Figure 3.12: Mean/variance of the RMSE_h reached by the different methods, when using an erroneous noise level estimate $\hat{\sigma}_n$ (the ground truth is $\sigma_n = 0.05$).

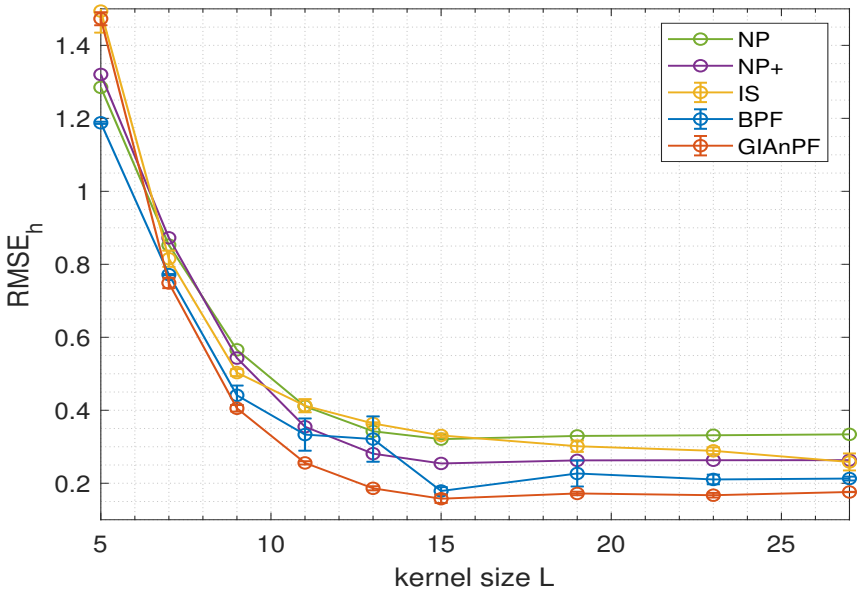


Figure 3.13: Mean/variance of the RMSE for various kernel support width L (ground truth size: $\overline{L} = 15$).

Image	σ_x	NP$^+$	GIAnPF
Boat	0	20.1051	**20.5247**[20.3207-20.7045]
	0.01	**18.8952**	18.0044 [17.8957-18.1138]
	0.05	17.9529	**17.9610**[17.9485-17.9714]
	0.1	17.2432	**17.2755** [17.2674-17.2848]
Goldhill	0	20.7009	**20.9611**[20.6529-21.1522]
	0.01	19.1551	**20.2494** [20.1838-20.3108]
	0.05	18.9202	**18.9990**[18.9728-19.0205]
	0.1	18.2412	**18.3510** [18.3356-18.3630]
Plane	0	23.2597	**23.6970** [23.5257-23.8888]
	0.01	**22.1416**	21.8859 [21.7679-21.9899]
	0.05	20.4632	**20.5290**[20.5132-20.5487]
	0.1	19.6010	**19.6535** [19.6445-19.6654]
Cameraman	0	20.4253	**20.4839** [19.7894-20.9030]
	0.01	19.3293	**19.8935** [19.7158-20.0245]
	0.05	18.3764	**18.6403** [18.5687-18.6794]
	0.1	17.4996	**17.7183** [17.6861-17.7376]

Table 3.7: SNR$_x$ values of restored images using estimated blur derived from NP$^+$ and GIAnPF applied on the calibrated image Chart.

Figure 3.14: (left) Original images ; (middle) blurry noisy versions with $\sigma_n = 0.01$; (right) example of restored image using the mean blur estimate from one run of GIAnPF.

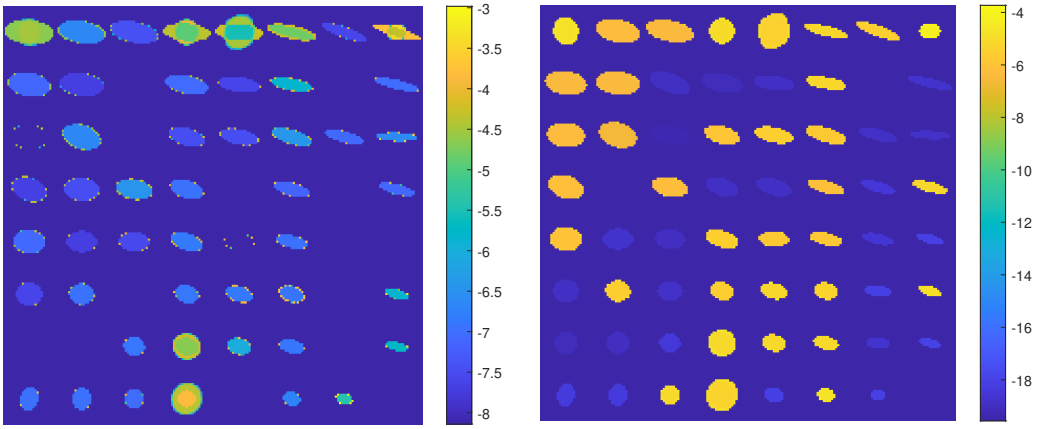


Figure 3.15: Results of 100 samples from the posterior obtained with single run of GIAnPF when applied to image `Chart` degraded by Map 4 and $\sigma_n = 0.1$: (left) $\log_{10}$ of the pixel-wise mean squared error over the 100 samples. (right) $\log_{10}$ of the pixel-wise standard deviation over the 100 samples.

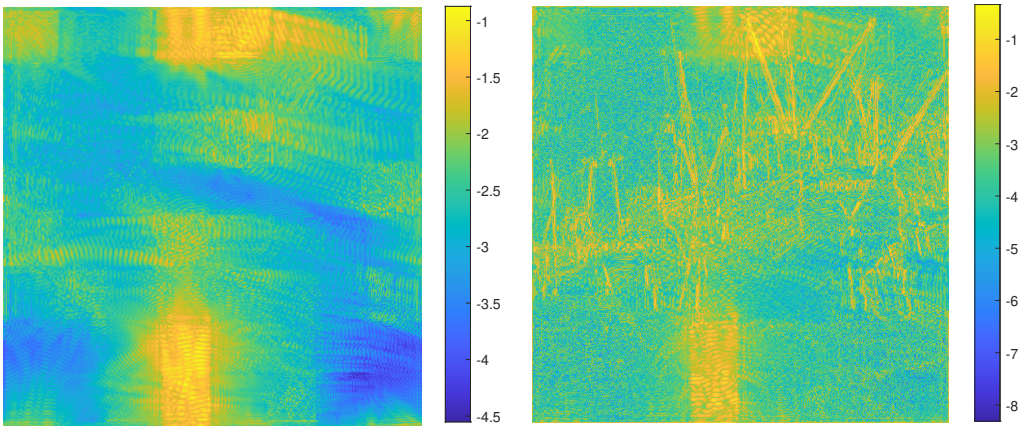


Figure 3.16: Uncertainty quantification vs error in the estimation of the kernels by GIAnPF on the image `boat` ; (left) $\log_{10}$ of the pixel-wise standard deviation of the restored image, when using 100 samples from the posterior of the blur parameters, approximated by a single run of GIAnPF. (right) $\log_{10}$ of the mean pixel-wise quadratic error between the true and the restored image when using the posterior mean estimator of the parameters approximated by GIAnPF, averaged over 100 independent runs.

CHAPTER 4

Variational Bayesian algorithm for image blind deconvolution

Contents

4.1 Introduction

Image blind deconvolution problem arises in many fields of image processing such as astronomy [Murtagh *et al.* 2007], biology [Holmes *et al.* 2006] and medical imaging [Michailovich & Tannenbaum 2007]. Given a degraded, blurred and noisy image, the aim is to restore a clean image along with an estimate of the blur kernel. Blind deconvolution is a severely ill-posed problem as there exists an infinite number of pairs (image/blur) that lead to the same observed image. Blind deconvolution methods available in literature adopt either a sequential identification process [Carasso 2001], or a joint estimation approach [Levin *et al.* 2009]. In the former, the blur kernel is identified first, possibly through a calibration step [Bell *et al.* 2016, Huang *et al.* 2021a, Chouzenoux *et al.* 2019]. Then the unknown image is inferred using a non-blind image restoration method. This approach was the one considered in our Chapter 3. In the latter, the blur kernel and unknown image are simultaneously estimated. This joint approach is retained in this chapter, and Chapter 5.

Since the problem of joint estimation is highly ill-posed, it is mandatory to incorporate prior knowledge on the sought unknowns. The retained prior strongly influences the choice for the solver. Let us distinguish two main classes of joint blind deconvolution approaches. A first option consists of formulating the problem as the minimization of a cost function gathering a data fidelity term (e.g., least-squares discrepancy) and penalties/constraints acting on the image and kernel variables. In such a way, it is quite standard to impose normalization and sparsity enhancing constraints on the kernel coefficients to avoid scale ambiguity inherent to the blind deconvolution model [Komodakis & Paragios 2012, Kotera *et al.* 2013, Krishnan *et al.* 2011]. One can also easily impose the smoothness of the image, by adopting total-variation based regularization [Chan & Wong 1998]. Several other efficient choices have been proposed in the literature, along with suitable iterative optimization methods to solve the resulting problems [Levin 2006, Shan *et al.* 2008, Joshi *et al.* 2008, Abboud *et al.* 2019, Bolte *et al.* 2010, Liu *et al.* 2021a]. The main advantage of this family of methods is probably its flexibility. But this comes at the price of heavy parameter tuning. The second option is to resort to a Bayesian formulation to express the model and a priori knowledge on the variables. The estimates are then defined from the estimation of the moments (typically, the mean) of a posterior distribution given the observed data and prior. As this typically involves the evaluation of intractable integrals, sampling [Robert & Casella 2004, Luengo *et al.* 2020] or approximation [Evans & Swartz 1995] strategies are used. MCMC methods have been widely used for blind deconvolution involving 1D sparse signals [Rosec *et al.* 2003, Ge *et al.* 2011, Kail *et al.* 2012], but it is up to our knowledge scarcely employed in large-scale problems [Bishop *et al.* 2008], probably for computational time reasons. Another family of approach consists in adopting the so-called variational Bayesian approximation paradigm [Fox & Roberts 2012, Blei *et al.* 2017, Zheng *et al.* 2015a, Gharsalli *et al.* 2012]. Then, a simpler (usually separable) approximation to the posterior is built through the minimization of a suitable divergence. This approach leads to fast Bayesian-based algorithms, whose great performance has been assessed in the context of non-blind [Marnissi *et al.* 2017, Zheng *et al.* 2015b] and blind [Babacan *et al.* 2009] image restoration. Bayesian-based techniques usually require less parameters than optimization-based ones. Moreover, they can provide higher-order moments estimates, such as covariance matrices, which are of high interest for assessing probabilistically the uncertainty of the results. However, dealing with complex noise models and priors in such methods may be tricky, and the algorithms may be quite computationally heavy, hereagain with hyperparameters to be tuned.

A recent trend is to insert optimization-based steps in Bayesian sampling/approximation methods for a more versatile modality and faster computations. See, for example, [Marnissi *et al.* 2017, Pereyra *et al.* 2016, Marnissi *et al.* 2020] for applications of such ideas in the context of large-scale image processing. Following this strategy, in this chapter, we introduce an algorithm called VBA, based on the Bayesian variational strategy (see Section 2.2.1), enhanced by optimization-based ideas inherited from [Marnissi *et al.* 2017], with the advantages to cope with a large set of priors on the kernel and the image, and to present a reduced computational cost. The rest of Chapter 4 is organized as follows. In Section 4.2, we introduce the image degradation model and introduce our Bayesian modeling, and provide the background for deriving our algorithm.

Section 4.3 explicitly describes the iterative updates of the proposed VBA in the case of Gaussian noise. Section 4.4 explicitly describes the iterative updates of the proposed VBA in the case of non-Gaussian noise. Section 4.5 concludes this chapter. The next Chapter 5 will be dedicated to the extension of VBA using the deep unrolling paradigm, and to the numerical experiments.

4.2 Problem statement

4.2.1 Observation model

We focus on the restoration of an original image $\widetilde{\mathbf{x}} \in \mathbb{R}^N$, from a degraded version of it $\mathbf{y} \in \mathbb{R}^N$, related to $\widetilde{\mathbf{x}}$ according to the following model:

$$(\forall i \in \{1,...,N\}) \;\; y_i = \mathcal{D}([\widetilde{\mathbf{H}}\widetilde{\mathbf{x}}]_i). \tag{4.1}$$

Hereabove, $\mathbf{n} \in \mathbb{R}^N$ models some additive random perturbation on the observation. Moreover, $\widetilde{\mathbf{H}} \in \mathbb{R}^{N\times N}$ is a linear operator modeling the effect of a blur kernel $\widetilde{\mathbf{h}} \in \mathbb{R}^M$. $\mathcal{D}(\cdot)$ is a noise model that may depend on the data available. In the following paragraphs, we'll discuss the case of Gaussian noise and non-Gaussian noise respectively.
In this work, we focus on the generalized blind deconvolution problem where the matrix associated with a given kernel $\mathbf{h} = [h_1,\dots,h_M]^\top$ reads

$$\mathbf{H} = \sum_{m=1}^{M} h_m \mathbf{S}_m, \tag{4.2}$$

with $\{\mathbf{S}_1,\dots,\mathbf{S}_M\}$ is a set of M known sparse $N \times N$ real-valued matrices. This model allows to retrieve the standard image deblurring model, as a special case when $\mathbf{H}$ identifies with a 2D discrete convolution matrix with suitable padding. The considered problem amounts to retrieving an estimate $(\widehat{\mathbf{x}},\widehat{\mathbf{h}})$ of the pair of variables $(\widetilde{\mathbf{x}},\widetilde{\mathbf{h}})$ given $\mathbf{y}$. Due to the ill-posedness of this inverse problem, assumptions are required on the sought image / kernel and on the noise statistics to reach satisfying results. Furthermore, we introduce a linear equality constraint on the blur kernel estimate $\mathbf{h}$. A general expression of such a constraint is as follows:

$$\mathbf{h} = \mathbf{T}\mathbf{z} + \mathbf{t}, \tag{4.3}$$

where $\mathbf{T} = (T_{m,p})_{1\leq m\leq M, 1\leq p\leq P} \in \mathbb{R}^{M\times P}$ is a matrix of rank $P \in \{1,\dots,M\}$ and $\mathbf{t} = [t_1,\dots,t_M]^\top \in \mathbb{R}^M$ is some vector of predefined constants. Vector $\mathbf{z} = [z_1,\dots,z_P]^\top \in \mathbb{R}^P$ becomes the new unknown of the problem, along with the image $\mathbf{x}$. A typical linear equality constraint in such context is the sum-to-one constraint, i.e., $\sum_{m=1}^{M} h_m = 1$. Other examples will be provided in the experimental Section 5.2. We can thus rewrite (4.2) as

$$\mathbf{H} = \sum_{p=1}^{P} z_p \mathbf{K}_p + \mathbf{K}_0 = \mathcal{H}(\mathbf{z}), \tag{4.4}$$

with

$$(\forall p \in \{1,\dots,P\}) \quad \mathbf{K}_p = \sum_{m=1}^{M} T_{m,p}\mathbf{S}_m \in \mathbb{R}^{N\times N}, \tag{4.5}$$

and

$$\mathbf{K}_0 = \sum_{m=1}^{M} t_m \mathbf{S}_m \in \mathbb{R}^{N\times N}. \tag{4.6}$$

4.2.2 Hierarchical Bayesian Modeling

Let us now introduce the hierarchical Bayesian model on which our VBA method will be grounded.

4.2.2.1 Likelihood

First, we express the likelihood $p(\mathbf{y}|\mathbf{x},\mathbf{z})$ of the observed data, given the unknowns $(\mathbf{x},\mathbf{z})$. In general case, the likelihood can be expressed as follows:

$$p(\mathbf{y}|\mathbf{x},\mathbf{z}) \propto \exp\left[-\sum_{i=1}^{N} \phi_i([\mathbf{H}\mathbf{x}]_i; y_i)\right] = \exp\left[-\sum_{i=1}^{N} \phi_i\left([\mathcal{H}(\mathbf{z})\mathbf{x}]_i; y_i\right)\right]. \tag{4.7}$$

According to different noise models $\mathcal{D}(\cdot)$, ϕ_i can have various forms (see [Marnissi *et al.* 2017, Tab I] for examples). For instance, it reduces to a least square function in the case of additive Gaussian noise.

4.2.2.2 Prior

As already mentioned, it is necessary to incorporate suitable prior knowledge on the sought quantities to limit the problem ill-posedness. We here consider a wide range of sparsity enhancing prior for the image $\mathbf{x}$, by adopting the generic model,

$$p(\mathbf{x}|\gamma) \propto \gamma^{\frac{N}{2\kappa}} \exp\Big(-\gamma \sum_{j=1}^{J} ||\mathbf{D}_j\mathbf{x}||^{2\kappa}\Big), \tag{4.8}$$

with $\kappa \in (0,1]$ a scale parameter and $(\mathbf{D}_j)_{1\le j\le J} \in (\mathbb{R}^{S\times N})^J$ both assumed to be known. For instance, an isotropic total variation prior is obtained by setting $\kappa = 1/2$, $S = 2$, $J = N$ and for every $j \in \{1,\ldots,N\}$, $\mathbf{D}_j\mathbf{x} = [[\nabla^h\mathbf{x}]_j, [\nabla^v\mathbf{x}]_j] \in \mathbb{R}^2$ gathers the horizontal and vertical gradients of $\mathbf{x}$ at pixel j. Other relevant choices are discussed in [Marnissi *et al.* 2017]. Hereabove, $\gamma > 0$ is a regularization hyperparameter that we incorporate in our hierarchical model. We assume a Gamma distribution on γ,

$$p(\gamma) \propto \gamma^{\alpha-1}\exp(-\eta\gamma), \tag{4.9}$$

where $\alpha \ge 0$ and $\eta \ge 0$ are the (known) shape and inverse scale parameters of the Gamma distribution. Such choice for the hyperparameter is rather standard in the context of Bayesian image restoration.

Regarding the blur $\mathbf{h}$, we adopt the so-called SAR model, successfully used for Bayesian-based blind deconvolution in [Babacan *et al.* 2009]. The model relies on the following Gaussian model,

$$p(\mathbf{h}|\xi) \propto \xi^{\frac{M}{2}} \exp\Big(-\frac{\xi}{2}||\mathbf{A}(\mathbf{h}-\mathbf{m})||^2\Big), \tag{4.10}$$

where $\mathbf{A} \in \mathbb{R}^{Q \times M}$ with $Q \in \mathbb{N} \setminus \{0\}$ denotes a matrix of rank M. $\mathbf{m} \in \mathbb{R}^M$ is the mean of the underlying Gaussian distribution, and $\xi > 0$ is such that $\xi \mathbf{A}^\top \mathbf{A}$ is its inverse covariance matrix. If $\mathbf{h}$ follows this distribution, the projection of $\mathbf{h}$ onto the affine space defined by (4.3) is also Gaussian as well as the vector $\mathbf{z}$ associated with each projected vector. More precisely, $\mathbf{z}$ follows a Gaussian distibution with mean $\boldsymbol{\mu} = \mathbf{T}^{-1}(\mathbf{m} - \mathbf{t})$ and covariance matrix $\xi^{-1}\mathbf{T}^{-1}(\mathbf{A}^\top \mathbf{A})^{-1}(\mathbf{T}^{-1})^\top$ where $\mathbf{T}^{-1}$ is the left inverse of $\mathbf{T}$, i.e., $\mathbf{T}^{-1} = (\mathbf{T}^\top \mathbf{T})^{-1}\mathbf{T}^\top$. This yields the following prior for the variable of interest $\mathbf{z}$:

$$p(\mathbf{z}|\xi) \propto \xi^{\frac{P}{2}} \exp\Big(-\frac{\xi}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \mathbf{L}(\mathbf{z}-\boldsymbol{\mu})\Big), \tag{4.11}$$

where $\mathbf{L} = \mathbf{T}^\top \mathbf{T}(\mathbf{T}^\top \left(\mathbf{A}^\top \mathbf{A}\right)^{-1}\mathbf{T})^{-1}\mathbf{T}^\top \mathbf{T}$. We will consider $(\mathbf{L}, \boldsymbol{\mu})$ to be predefined by the user, so as to be adapted to the sought properties of the blur kernel to estimate. The hyperparameter ξ will be learned during a training phase, as we will explain in Section 5.1.

4.2.2.3 Hierarchical model

Let us assume that $(\mathbf{x}, \gamma)$ and $\mathbf{z}$ are mutually independent. According to Bayes formula, the posterior distribution of the unknowns $\Theta = (\mathbf{x}, \mathbf{z}, \gamma)$ given the observed data $\mathbf{y}$ is defined as

$$p(\Theta|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \mathbf{z}) p(\mathbf{x}|\gamma) p(\mathbf{z}|\xi) p(\gamma), \tag{4.12}$$

where the four factors on the right side have been defined above.

4.2.3 Variational Bayesian Inference

The Bayesian inference paradigm seeks for solving the blind restoration problem through the exploration of the posterior $p(\Theta|\mathbf{y})$. Typically, one would be interested in the posterior mean, its covariance, or its modes (i.e., maxima). Let us make (4.12) explicit:

$$\begin{aligned} p(\Theta|\mathbf{y}) \propto & \exp\left(-\gamma \sum_{j=1}^{J} ||\mathbf{D}_j \mathbf{x}||^{2\kappa} - \sum_{i=1}^{N} \phi_i\left([\mathcal{H}(\mathbf{z})\mathbf{x}]_i; y_i\right)\right) \\ & \times \gamma^{\frac{N}{2\kappa}+\alpha-1} \exp(-\eta\gamma) \xi^{\frac{P}{2}} \exp\Big(-\frac{\xi}{2}(\mathbf{z}-\boldsymbol{\mu})^\top \mathbf{L}(\mathbf{z}-\boldsymbol{\mu})\Big). \end{aligned} \tag{4.13}$$

Unfortunately, neither $p(\Theta|\mathbf{y})$, nor its moments (e.g., mean, covariance), nor its mode positions have a closed form. In particular $p(\mathbf{y})$, which acts as a normalization constant, cannot be calculated analytically. We thus resort to the variational Bayesian framework, introduced in Section 2.2.1, to approximate this distribution by a more tractable one, denoted by $q(\Theta)$, for which the estimators are easier to compute. The approximation is computed with the aim to minimize the KL divergence between the target posterior and its approximation, which amounts to determining

$$\begin{aligned} q^{\text{opt}}(\Theta) &= \text{argmin}_q \, \mathcal{KL}(q(\Theta)||p(\Theta|\mathbf{y})), \\ &= \text{argmin}_q \int q(\Theta) \ln\left(\frac{q(\Theta)}{p(\Theta|\mathbf{y})}\right) \mathrm{d}\Theta, \end{aligned} \tag{4.14}$$

where the equality holds only when $q(\Theta) = p(\Theta|\mathbf{y})$. In order to make the solution of the above minimization problem tractable, a typical strategy is to make use of a so-called mean field approximation of the posterior, combined with an alternating minimization procedure, which yields to the so-called VBA. More specifically, we consider the following factorization:

$$q(\Theta) = q_{\mathbf{X}}(\mathbf{x})q_{\mathbf{Z}}(\mathbf{z})q_{\Gamma}(\gamma). \tag{4.15}$$

Due to the intricate form of the chosen prior on the image and of the likelihood (in the case of non-Gaussian noise), we introduce extra approximation steps, relying on a majoration-minimization (MM) strategy, reminiscent from [Marnissi *et al.* 2017]. In addition, we propose a strategy to reduce the time complexity of VBA, so as to deal with medium to large size images. In Section 4.3 and 4.4, we describe our developments in the case of Gaussian and non-Gaussian noises, respectively. As we will emphasize, even for the simplest case of Gaussian noise, the method requires the setting of two cumbersome hyperparameters, namely the regularization weight ξ and the noise level denoted β. In Chapter 5.1, we will show how to unroll the VBA method as a neural network structure, so as to learn the parameters (ξ, β) in a supervised fashion. All the numerical experiments are reported in Chapter 5.1.

4.3 Case of Gaussian noise

Let us first focus on the resolution of the inverse problem (4.1) when $\mathcal{D}(\cdot)$ models additive Gaussian noise. First, we reformulate the observation model as

$$\mathbf{y} = \widetilde{\mathbf{H}}\widetilde{\mathbf{x}} + \mathbf{n}. \tag{4.16}$$

We then express the corresponding likelihood $p(\mathbf{y}|\mathbf{x},\mathbf{z})$ of the observed data, given the unknowns $(\mathbf{x},\mathbf{z})$. Since the noise is assumed to be Gaussian distributed, the likelihood can be expressed as follows:

$$p(\mathbf{y}|\mathbf{x},\mathbf{z}) = \beta^{\frac{N}{2}} \exp\left(-\frac{\beta}{2}||\mathbf{y} - \mathcal{H}(\mathbf{z})\mathbf{x}||^2\right), \tag{4.17}$$

where we recall that β denotes the inverse of the noise variance. In the sequel, we will assume that $\mathbf{n}$ is a realization of an additive Gaussian noise with zero mean and standard deviation σ. In the remainder of the section, it will be convenient to set $\beta = \sigma^{-2}$.
We now describe our proposed implementation of the VBA when applied to the approximation to the posterior in (4.13). We first present an MM-based procedure to handle the complicated form of the prior term on variable $\mathbf{x}$. Then, we give the explicit expressions of the updates performed in the alternating minimization method.

4.3.1 MM-based approximation

Let us focus on the prior term in (4.8). This distribution is difficult to deal with as soon as κ is different from 1 (in which case a Gaussian distribution is retrieved). We thus propose

to construct a surrogate for the prior on $\mathbf{x}$. We use the tangent inequality for concave functions, which yields the following majorant function for the ℓ_κ-function with $\kappa \in (0,1]$:

$$(\forall u > 0)(\forall v \geq 0) \quad v^\kappa \leq (1-\kappa)u^\kappa + \kappa u^{\kappa-1} v. \tag{4.18}$$

Let us introduce the vector of auxiliary positive variables $\boldsymbol{\lambda} = (\lambda_j)_{1\leq j\leq J}$. From the previous inequality, we then deduce the following majorant function for the negative logarithm of the prior distribution:

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad \gamma \sum_{j=1}^{J} ||\mathbf{D}_j \mathbf{x}||^{2\kappa} \leq \sum_{j=1}^{J} F_j(\mathbf{D}_j \mathbf{x}, \lambda_j; \gamma), \tag{4.19}$$

where, for every $j \in \{1,\ldots,J\}$,

$$F_j(\mathbf{D}_j \mathbf{x}, \lambda_j; \gamma) = \gamma \frac{\kappa ||\mathbf{D}_j \mathbf{x}||^2 + (1-\kappa)\lambda_j}{\lambda_j^{1-\kappa}}. \tag{4.20}$$

This majorant function can be understood as a Gaussian lower bound on the prior distribution on $\mathbf{x}$, which will appear more tractable in the VBA implementation. We will also show that the update of the auxiliary variables remains rather simple, thus not impacting the complexity of the whole procedure.
In a nutshell, using (4.13), and (4.19), we obtain the following inequality:

$$p(\Theta|\mathbf{y}) \geq \mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}) \tag{4.21}$$

where the lower bound on the posterior distribution is

$$\mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}) = C\gamma^{\frac{N}{2\kappa}} \exp\left(-\frac{\beta}{2} ||\mathbf{y} - \mathcal{H}(\mathbf{z})\mathbf{x}||^2 - F(\mathbf{x}, \boldsymbol{\lambda}; \gamma) \right) p(\gamma) p(\mathbf{z}|\xi). \tag{4.22}$$

Hereabove we have introduced the shorter notation

$$F(\mathbf{x}, \boldsymbol{\lambda}; \gamma) = \sum_{j=1}^{J} F_j(\mathbf{D}_j \mathbf{x}, \lambda_j; \gamma), \tag{4.23}$$

and C is a multiplicative constant independent from Θ. Inequality (4.21) leads to the following majorization of the $\mathcal{KL}$ divergence involved in (4.14):

$$\mathcal{KL}(q(\Theta)||p(\Theta|\mathbf{y})) \leq \mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda})). \tag{4.24}$$

By minimizing the upper bound in (4.24) with respect to $\boldsymbol{\lambda}$, we can keep it as tight as possible, so as to guarantee the good performance of the VBA. To summarize, we propose to solve Problem (4.14) through the following four iterative steps:

1. Minimizing $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}))$ w.r.t. $q_{\mathbf{X}}(\mathbf{x})$.
2. Minimizing the upper bound $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}))$ in (4.24) w.r.t. $q_{\mathbf{Z}}(\mathbf{z})$.
3. Update the auxiliary variables $(\lambda_j)_{1\leq j\leq J}$ to minimize $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}))$.
4. Minimizing $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}))$ w.r.t. $q_\Gamma(\gamma)$.

Subsequently, at a given iteration k of the proposed algorithm, the corresponding estimated variables will be indexed by k.

4.3.2 VBA updates

Let us now describe the four steps of the proposed VBA, starting from a given iteration k associated with the current approximated distributions $q_{\mathbf{X}}^k(\mathbf{x}), q_{\mathbf{Z}}^k(\mathbf{z})$, and $q_{\Gamma}^k(\gamma)$, and the auxiliary parameter estimate $\boldsymbol{\lambda}^k$. We also denote by $(\mathbf{x}^k, \mathbf{z}^k, \boldsymbol{\gamma}^k)$ the estimates of the means of $q_{\mathbf{X}}^k$, $q_{\mathbf{Z}}^k$, and q_{Γ}^k, and $(\mathbf{C}_{\mathbf{x}}^k, \mathbf{C}_{\mathbf{z}}^k)$ the covariance estimates for $q_{\mathbf{X}}^k$ and $q_{\mathbf{Z}}^k$.

4.3.2.1 Update of $q_{\mathbf{X}}(\mathbf{x})$

By definition,

$$q_{\mathbf{X}}^{k+1}(\mathbf{x}) = \text{argmin}_{q_{\mathbf{X}}} \mathcal{KL}(q_{\mathbf{X}}(\mathbf{x}) q_{\Gamma}^k(\gamma) q_{\mathbf{Z}}^k(\mathbf{z}) || \mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}^k)). \tag{4.25}$$

The standard solution provided by (2.15) remains valid, by replacing the joint distribution by a lower bound chosen proportional to $\mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}^k)$:

$$\begin{aligned} q_{\mathbf{x}}^{k+1}(\mathbf{x}) &\propto \exp\left(< \ln \mathcal{F}(\mathbf{x}, \mathbf{z}, \gamma \mid \mathbf{y}; \boldsymbol{\lambda}^k) >_{q_{\Gamma}^k(\gamma), q_{\mathbf{Z}}^k(\mathbf{z})} \right) \\ &\propto \exp\left(\int \int \ln \mathcal{F}(\mathbf{x}, \mathbf{z}, \gamma \mid \mathbf{y}; \boldsymbol{\lambda}^k) q_{\Gamma}^k(\gamma) q_{\mathbf{Z}}^k(\mathbf{z}) \mathrm{d}\gamma \mathrm{d}\mathbf{z} \right). \end{aligned} \tag{4.26}$$

By decomposing the different terms and using (4.4),

$$\begin{aligned} q_{\mathbf{x}}^{k+1}(\mathbf{x}) \propto \exp\Bigg\{ -\frac{1}{2}\mathbf{x}^\top \Bigg(& \beta \left(\mathbb{E}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{H})^\top \mathbb{E}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{H}) + \sum_{p=1}^{P} \sum_{q=1}^{P} \mathbf{e}_p^\top \mathrm{cov}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{z}) \mathbf{e}_q \mathbf{K}_p^\top \mathbf{K}_q \right) \\ & + 2\mathbb{E}_{q_{\Gamma}^k(\gamma)}(\gamma) \mathbf{D}^\top \Lambda^k \mathbf{D} \Bigg) \mathbf{x} + \beta \mathbf{x}^\top \mathbb{E}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{H})^\top \mathbf{y} \Bigg\} \end{aligned} \tag{4.27}$$

where

$$\mathbb{E}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{H}) = \sum_{p=1}^{P} \mathbf{e}_p^\top \mathbb{E}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{z}) \mathbf{K}_p + \mathbf{K}_0, \tag{4.28}$$

$$\mathbf{D} = [\mathbf{D}_1^\top, \ldots, \mathbf{D}_J^\top]^\top, \tag{4.29}$$

Λ^k is the block diagonal matrix whose diagonal elements are $(\kappa(\lambda_j^k)^{\kappa-1} \mathbf{I}_S)_{1 \leq j \leq J}$, and $(\mathbf{e}_1, \ldots, \mathbf{e}_P)$ is the canonical basis of $\mathbb{R}^P$. We thus obtain a Gaussian distribution:

$$q_{\mathbf{X}}^{k+1}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \check{\mathbf{x}}^{k+1}, \check{\mathbf{C}}_{\mathbf{x}}^{\ k+1}), \tag{4.30}$$

parametrized by

$$(\check{\mathbf{C}}_{\mathbf{x}}^{\ k+1})^{-1} = \beta \left((\mathbf{H}^k)^\top \mathbf{H}^k + \sum_{p=1}^{P} \sum_{q=1}^{P} \mathbf{e}_p^\top \mathbf{C}_{\mathbf{z}}^k \mathbf{e}_q \mathbf{K}_p^\top \mathbf{K}_q \right) + 2\gamma^k \mathbf{D}^{\mathrm{T}} \Lambda^k \mathbf{D}, \tag{4.31}$$

$$\check{\mathbf{x}}^{k+1} = \beta \check{\mathbf{C}}_{\mathbf{x}}^{\ k+1} (\mathbf{H}^k)^\top \mathbf{y}, \tag{4.32}$$

with $\mathbf{H}^k = \mathcal{H}(\mathbf{z}^k)$.
In image restoration applications, dimension N can be rather large (typically greater than

10^6 variables), so that the storage of the full covariance matrix $\check{\mathbf{C}}_{\mathbf{x}}^{k+1}$ is neither desirable nor usually possible. We thus propose to resort to a diagonal approximation to this matrix when required, so that the update finally reads:

$$q_{\mathbf{X}}^{k+1}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{x}^{k+1}, \mathbf{C}_{\mathbf{x}}^{k+1}), \tag{4.33}$$

with

$$\mathbf{C}_{\mathbf{x}}^{k+1} = \mathrm{Diag}\left(\delta_{\mathbf{x}}^{k+1}\right) \tag{4.34}$$

$$\mathbf{x}^{k+1} = \mathrm{CG}\left((\check{\mathbf{C}}_{\mathbf{x}}^{k+1})^{-1}, \beta(\mathbf{H}^k)^\top \mathbf{y}\right), \tag{4.35}$$

where $\delta_{\mathbf{x}}^{k+1} \in \mathbb{R}^N$ is the vector of the inverses of the diagonal elements of $(\check{\mathbf{C}}_{\mathbf{x}}^{k+1})^{-1}$, and $\mathrm{CG}(\mathtt{A}, \mathtt{b})$ denotes the application of a linear conjugate gradient solver to the linear system `Ax=b`.

4.3.2.2 Update of $q_{\mathbf{Z}}(\mathbf{z})$

According to the VBA principle,

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) = \mathrm{argmin}_{q_{\mathbf{Z}}} \mathcal{KL}(q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) q_{\mathbf{Z}}(\mathbf{z}) || \mathcal{F}(\Theta|\mathbf{y}; \boldsymbol{\lambda}^k)). \tag{4.36}$$

Using (2.15) and the previously introduced bound $\mathcal{F}(\Theta \mid \mathbf{y}; \boldsymbol{\lambda}^k)$, we have

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \propto \exp\left(\int\int \ln \mathcal{F}(\mathbf{x}, \mathbf{z}, \gamma \mid \mathbf{y}; \boldsymbol{\lambda}^k) q_{\Gamma}^{k}(\gamma) q_{\mathbf{X}}^{k+1}(\mathbf{x}) \mathrm{d}\gamma \mathrm{d}\mathbf{x}\right). \tag{4.37}$$

Replacing the involved quantities by their expression yields

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \propto \exp\left\{-\frac{1}{2}\mathbf{z}^\top \left(\beta \mathbf{B}^{k+1} + \xi \mathbf{L}\right)\mathbf{z} + \mathbf{z}^\top \left(\beta \mathbf{a}^{k+1} + \xi \mathbf{L}\boldsymbol{\mu}\right)\right\}, \tag{4.38}$$

where $\mathbf{a}^{k+1} = (a_p^{k+1})_{1 \leq p \leq P} \in \mathbb{R}^P$ and $\mathbf{B}^{k+1} = (B_{p,q}^{k+1})_{1 \leq p,q \leq P} \in \mathbb{R}^{P \times P}$ are such that, for every $(p,q) \in \{1, \ldots, P\}^2$,

$$\begin{aligned} a_p^{k+1} =& \mathbb{E}_{q_{\mathbf{X}}^{k+1}(\mathbf{x})}(\mathbf{x})^\top \mathbf{K}_p^\top \mathbf{y} - \mathbb{E}_{q_{\mathbf{X}}^{k+1}}(\mathbf{x}^\top \mathbf{K}_p^\top \mathbf{K}_0 \mathbf{x}) \\ =& (\mathbf{x}^{k+1})^\top \mathbf{K}_p^\top \mathbf{y} - \mathbf{B}_{p,0}^{k+1}, \end{aligned} \tag{4.39}$$

$$\begin{aligned} B_{p,q}^{k+1} =& \mathbb{E}_{q_{\mathbf{X}}^{k+1}}(\mathbf{x}^\top \mathbf{K}_p^\top \mathbf{K}_q \mathbf{x}) \\ =& \mathrm{trace}\left(\mathbf{K}_p \mathbf{C}_{\mathbf{x}}^{k+1} \mathbf{K}_q^\top\right) + (\mathbf{x}^{k+1})^\top \mathbf{K}_p^\top \mathbf{K}_q \mathbf{x}^{k+1} \end{aligned} \tag{4.40}$$

with

$$\begin{aligned} B_{p,0}^{k+1} =& \mathbb{E}_{q_{\mathbf{X}}^{k+1}}(\mathbf{x}^\top \mathbf{K}_p^\top \mathbf{K}_0 \mathbf{x}) \\ =& \mathrm{trace}\left(\mathbf{K}_p \mathbf{C}_{\mathbf{x}}^{k+1} \mathbf{K}_0^\top\right) + (\mathbf{x}^{k+1})^\top \mathbf{K}_p^\top \mathbf{K}_0 \mathbf{x}^{k+1}. \end{aligned} \tag{4.41}$$

Thus, the update for the distribution $q_{\mathbf{Z}}$ reads

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{z}^{k+1}, \mathbf{C}_{\mathbf{z}}^{k+1}), \tag{4.42}$$

with

$$(\mathbf{C}_{\mathbf{z}}^{k+1})^{-1} = \beta \mathbf{B}^{k+1} + \xi \mathbf{L}, \tag{4.43}$$

$$\mathbf{z}^{k+1} = \mathbf{C}_{\mathbf{z}}^{k+1} \left(\beta \mathbf{a}^{k+1} + \xi \mathbf{L} \boldsymbol{\mu} \right). \tag{4.44}$$

4.3.2.3 Update of $\boldsymbol{\lambda}$

Let us now express the update of the auxiliary variable. We aim at finding

$$\boldsymbol{\lambda}^{k+1} = \text{argmin}_{\boldsymbol{\lambda}} \mathcal{KL}(q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) || \mathcal{F}(\Theta | \mathbf{y}; \boldsymbol{\lambda})). \tag{4.45}$$

This amounts to finding, for every $j \in \{1, \ldots, J\}$,

$$\begin{aligned}
\lambda_j^{k+1} &= \text{argmin}_{\lambda_j \in [0,+\infty)} \int q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \times \log \frac{q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z})}{\mathcal{F}(\Theta | \mathbf{y}, \boldsymbol{\lambda})} \mathrm{d}\Theta, \\
&= \text{argmin}_{\lambda_j \in [0,+\infty)} \sum_{j=1}^{J} \int \int q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) \times F_j(\mathbf{D}_j \mathbf{x}, \lambda_j; \gamma) \mathrm{d}\mathbf{x} \mathrm{d}\gamma, \\
&= \text{argmin}_{\lambda_j \in [0,+\infty)} \frac{\kappa \mathbb{E}_{q_{\mathbf{x}}^{k+1}(\mathbf{x})} \left[||\mathbf{D}_j \mathbf{x}||^2 \right] + (1-\kappa) \lambda_j}{\lambda_j^{1-\kappa}}.
\end{aligned} \tag{4.46}$$

The explicit solution to the above minimization problem yields the following update:

$$\begin{aligned}
\lambda_j^{k+1} &= \mathbb{E}_{q_{\mathbf{x}}^{k+1}(\mathbf{x})} \left[||\mathbf{D}_j \mathbf{x}||^2 \right] \\
&= ||\mathbf{D}_j \mathbf{x}^{k+1}||^2 + \text{trace} \left(\mathbf{D}_j^{\top} \mathbf{D}_j \mathbf{C}_{\mathbf{x}}^{k+1} \right).
\end{aligned} \tag{4.47}$$

4.3.2.4 Update of $q_{\Gamma}(\gamma)$

Finally, the update related to the hyperparameter γ is expressed as

$$q_{\Gamma}^{k+1}(\gamma) = \text{argmin}_{q_{\Gamma}} \mathcal{KL}(q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) || \mathcal{F}(\Theta | \mathbf{y}; \boldsymbol{\lambda}^{k+1})). \tag{4.48}$$

Using (2.15), we have

$$q_{\Gamma}^{k+1}(\gamma) \propto \exp \left(\int \int \ln \mathcal{F}(\mathbf{x}, \mathbf{z}, \gamma \,|\, \mathbf{y}; \boldsymbol{\lambda}^{k+1}) \times q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \, \mathrm{d}\mathbf{x} \mathrm{d}\mathbf{z} \right). \tag{4.49}$$

The above integral has the following closed form expression:

$$q_{\Gamma}^{k+1}(\gamma) \propto \gamma^{\frac{N}{2\kappa} + \alpha - 1} \exp(-\eta \gamma) \times \exp \left(-\gamma \sum_{j=1}^{J} \frac{\kappa \mathbb{E}_{q_{\mathbf{x}}^{k+1}(\mathbf{x})} \left[||\mathbf{D}_j \mathbf{x}||^2 \right] + (1-\kappa) \lambda_j^{k+1}}{(\lambda_j^{k+1})^{1-\kappa}} \right). \tag{4.50}$$

It thus follows from (4.47) that the update of q_Γ is

$$q_\Gamma^{k+1}(\gamma) = \Gamma(d, b^{k+1}), \tag{4.51}$$

that is the Gamma distribution with parameters

$$d = \frac{N}{2\kappa} + \alpha, \;\; b^{k+1} = \sum_{j=1}^{J} (\lambda_j^{k+1})^\kappa + \eta. \tag{4.52}$$

The mean of q_Γ^{k+1} is finally given by

$$\gamma^{k+1} = \frac{d}{b^{k+1}}. \tag{4.53}$$

Note that parameter d is not iteration dependent and can thus be precomputed from the beginning of the VBA.

4.3.3 Overview of VBA with Gaussian noise

Algorithm 1 provides a summary of the resulting VBA for solving the blind deconvolution problem introduced in Section 4.2 under Gaussian noise. We also specify our initialization strategy. More practical details about the latter will be discussed in the experimental Section 5.2. As a result, the optimal posterior distributions for both variables $\mathbf{x}$ and $\mathbf{z}$ will be approximated as Gaussian distributions, while the one for hyperparameter γ is approximated by a Gamma distribution. In particular, after K iterations, it is direct to extract from VBA outputs an estimate for the posterior mean of the image and the kernel, through variable $\mathbf{x}^K$ and $\mathbf{T}\mathbf{z}^K + \mathbf{t}$. The associated covariance matrices are given by $\mathbf{C}_\mathbf{x}^K$ and $\mathbf{T}\mathbf{C}_\mathbf{z}^K\mathbf{T}^\top$. These matrices can be useful to perform uncertainty quantification of the results. The VBA also allows us to estimate easily the hyperparameter γ involved in the image prior.

Algorithm 1 VBA approach for image blind deconvolution under Gaussian noise

Initialization. Set hyperparameters $(\xi, \beta, \alpha, \eta)$. Define initial values for $(\mathbf{x}^0, \mathbf{C}_\mathbf{x}^0, \mathbf{z}^0, \mathbf{C}_\mathbf{z}^0)$. Compute $\boldsymbol{\lambda}^0$ and γ^0 using (4.47) and (4.53), respectively.

Iterative steps. For $k = 0, 1, \ldots, K$:

1: Update the mean $\mathbf{x}^{k+1}$ and the covariance matrix $\mathbf{C}_\mathbf{x}^{k+1}$ of $q_\mathbf{X}^{k+1}(\mathbf{x})$ using (4.34)-(4.35).
2: Update the mean $\mathbf{z}^{k+1}$ and the covariance matrix $\mathbf{C}_\mathbf{z}^{k+1}$ of $q_\mathbf{Z}^{k+1}(\mathbf{z})$ using (4.43)-(4.44).
3: Update λ_j^{k+1} using (4.47), for every $j \in \{1, \ldots, J\}$.
4: Update the mean γ^{k+1} of $q_\Gamma^{k+1}(\gamma)$ using (4.52)-(4.53).

4.4 Case of non-Gaussian noise

In this section, we extend our implementation of the VBA when the noise degrading the image is non-Gaussian. We can still adopt the MM-based approximation for the prior term on variable $\mathbf{x}$. The novelty here is to also apply an MM-based strategy to approximate the likelihood, which becomes more difficult for non-Gaussian noise. We detail below this new approximation, and give the explicit expressions of the updates performed in the resulting VBA.

4.4.1 MM-based approximation

4.4.1.1 Likelihood

Let us present our majorizing strategy to cope with the non-Gaussian noise case. Similarly as in [Marnissi *et al.* 2017, Ass.III.1], we assume that the noise model leads to a likelihood in (4.7) such that, for every $i \in \{1,\cdots,N\}$, ϕ_i is differentiable on $\mathbb{R}$ and there exists $\mu_i(y_i) > 0$ such that the function defined by $v \mapsto \frac{v^2}{2} - \frac{\phi_i(v;y_i)}{\mu_i(y_i)}$ in convex on $\mathbb{R}$. This assumption is satisfied when, for every $i \in \{1,\cdots,N\}$, ϕ_i is $\beta_i(y_i)$-Lipschitz differentiable. Examples of useful functions and noise models satisfying this assumption were displayed in [Marnissi *et al.* 2017, Tab.1]. It is worthy to note that functions $(\phi_i)_{1\leq i\leq N}$, and thus the parameters $(\mu_i(y_i))_{1\leq i\leq N}$ implicitly account for the noise level present in the image.

For every $i \in \{1,\cdots,N\}$ and $v \in \mathbb{R}$, let us define the following function:

$$\varsigma_i(v;y_i) = \sup_{t\in\mathbb{R}} \left(-\frac{1}{2}(v-t)^2 + \frac{\phi_i(t;y_i)}{\mu_i(y_i)} \right). \tag{4.54}$$

Then, according to [Marnissi *et al.* 2017, Prop.III.1], for every $i \in \{1,\cdots,N\}$,

$$(v \in \mathbb{R}) \quad \phi_i(v;y_i) = \inf_{w_i\in\mathbb{R}} \quad T_i(v,w_i;y_i) \tag{4.55}$$

where, for every $v \in \mathbb{R}$,

$$T_i(v,w_i;y_i) = \mu_i(y_i) \left(\frac{1}{2}(v-w_i)^2 + \varsigma_i(w_i;y_i) \right). \tag{4.56}$$

Moreover, the unique minimizer of $w_i \mapsto T_i(v,w_i;y_i)$ reads

$$\hat{w}_i(v) = v - \frac{1}{\mu_i(y_i)} \phi_i'(v;y_i). \tag{4.57}$$

This leads to construct an upper bound for the likelihood function.

$$\begin{aligned} (\forall \mathbf{x} \in \mathbb{R}^N) \;\; \phi(\mathbf{x},\mathbf{y}) &= \inf_{\mathbf{w}\in\mathbb{R}^N} T(\mathbf{x},\mathbf{z},\mathbf{w};\mathbf{y}) \\ &= \inf_{\mathbf{w}\in\mathbb{R}^N} \sum_{i=1}^{N} T_i\left([\mathcal{H}(\mathbf{z})\mathbf{x}]_i, w_i; y_i\right) \end{aligned} \tag{4.58}$$

with

$$(\forall i \in \{1,..,N\}) \;\; T_i(v,w_i;y_i) = \mu_i(y_i) \left(\frac{1}{2}(v-w_i)^2 + \varsigma_i(w_i;y_i) \right) \tag{4.59}$$

and ς_i defined in (4.54). Using (4.13), and (4.58), we obtain the following inequality:

$$p(\Theta|\mathbf{y}) \geq \mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}) \tag{4.60}$$

where the lower bound on the posterior distribution is

$$\mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}) = C\gamma^{\frac{N}{2\kappa}} \exp\left(-T(\mathbf{x},\mathbf{z},\mathbf{w};\mathbf{y}) - F(\mathbf{x},\boldsymbol{\lambda};\gamma)\right) p(\gamma) p(\mathbf{z}|\xi), \tag{4.61}$$

using again the notation (4.23). Inequality (4.60) leads to the majorization

$$\mathcal{KL}(q(\Theta)||p(\Theta|\mathbf{y})) \leq \mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda})). \tag{4.62}$$

By minimizing the upper bound in (4.62) with respect to $\boldsymbol{\lambda}$ and $\mathbf{w}$, we can keep it as tight as possible, so as to guarantee the good performance of the VBA. As we aforementioned, the other steps of VBA remain the same as in the Gaussian case. To summarize, we propose to solve Problem (4.14) through the following five iterative steps:

1. Minimizing the upper bound $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}))$ w.r.t. $q_{\mathbf{X}}(\mathbf{x})$.
2. Minimizing the upper bound $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}))$ w.r.t. $q_{\mathbf{Z}}(\mathbf{z})$.
3. Update the auxiliary variables $(w_i)_{1\leq i\leq N}$ to minimize $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}))$.
4. Update the auxiliary variables $(\lambda_j)_{1\leq j\leq J}$ to minimize $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}))$.
5. Minimizing the upper bound $\mathcal{KL}(q(\Theta)||\mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}))$ w.r.t. $q_{\Gamma}(\gamma)$.

At a given iteration k of the proposed algorithm, the corresponding estimated variables will be indexed by k.

4.4.2 VBA updates

Let us now describe the steps of the proposed VBA, starting from a given iteration k associated with the current approximated distributions $q^k_{\mathbf{X}}(\mathbf{x}), q^k_{\mathbf{Z}}(\mathbf{z})$, and $q^k_{\Gamma}(\gamma)$, and the auxiliary parameters $\boldsymbol{\lambda}^k$ and $\mathbf{w}^k$. We also denote by $(\mathbf{x}^k, \mathbf{z}^k, \gamma^k)$ the estimates of the means of $q^k_{\mathbf{X}}$, $q^k_{\mathbf{Z}}$, and q^k_{Γ}, and $(\mathbf{C}^k_{\mathbf{x}}, \mathbf{C}^k_{\mathbf{z}})$ the covariance estimates for $q^k_{\mathbf{X}}$ and $q^k_{\mathbf{Z}}$.

4.4.2.1 Update of $q_{\mathbf{X}}(\mathbf{x})$

By definition,

$$q^{k+1}_{\mathbf{X}}(\mathbf{x}) = \text{argmin}_{q_{\mathbf{X}}} \mathcal{KL}(q_{\mathbf{X}}(\mathbf{x}) q^k_{\Gamma}(\gamma) q^k_{\mathbf{Z}}(\mathbf{z}) || \mathcal{F}(\Theta|\mathbf{y};\mathbf{w}^k,\boldsymbol{\lambda}^k)). \tag{4.63}$$

Once again, we can use the solution provided by (2.15), by replacing the joint distribution by a lower bound chosen proportional to $\mathcal{F}(\Theta|\mathbf{y};\mathbf{w}^k,\boldsymbol{\lambda}^k)$:

$$\begin{aligned} q^{k+1}_{\mathbf{x}}(\mathbf{x}) &\propto \exp\left(< \ln \mathcal{F}(\mathbf{x},\mathbf{z},\gamma \mid \mathbf{y};\mathbf{w}^k,\boldsymbol{\lambda}^k) >_{q^k_{\Gamma}(\gamma), q^k_{\mathbf{Z}}(\mathbf{z})} \right) \\ &\propto \exp\left(\int\int \ln \mathcal{F}(\mathbf{x},\mathbf{z},\gamma \mid \mathbf{y};\mathbf{w}^k,\boldsymbol{\lambda}^k) q^k_{\Gamma}(\gamma) q^k_{\mathbf{Z}}(\mathbf{z}) \mathrm{d}\gamma \mathrm{d}\mathbf{z} \right). \end{aligned} \tag{4.64}$$

By decomposing the different terms and using (4.4),

$$\begin{aligned} & q^{k+1}_{\mathbf{x}}(\mathbf{x}) \\ & \propto \exp\Bigg\{ -\frac{1}{2}\mathbf{x}^\top \Bigg(\mathbb{E}_{q^k_{\mathbf{Z}}(\mathbf{z})}(\mathbf{H})^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y})) \mathbb{E}_{q^k_{\mathbf{Z}}(\mathbf{z})}(\mathbf{H}) + \sum_{p=1}^{P}\sum_{q=1}^{P} \mathbf{e}_p^\top \mathrm{cov}_{q^k_{\mathbf{Z}}(\mathbf{z})}(\mathbf{z}) \mathbf{e}_q \mathbf{K}_p^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y})) \mathbf{K}_q \\ & + 2\mathbb{E}_{q^k_{\Gamma}(\gamma)}(\gamma) \mathbf{D}^\top \Lambda^k \mathbf{D} \Bigg) \mathbf{x} + \mathbf{x}^\top \mathbb{E}_{q^k_{\mathbf{Z}}(\mathbf{z})}(\mathbf{H})^\top \mathbf{u} \Bigg\} \end{aligned} \tag{4.65}$$

where

$$\mathbb{E}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{H}) = \sum_{p=1}^{P} \mathbf{e}_p^\top \mathbb{E}_{q_{\mathbf{Z}}^k(\mathbf{z})}(\mathbf{z}) \mathbf{K}_p + \mathbf{K}_0, \tag{4.66}$$

$$\mathbf{D} = [\mathbf{D}_1^\top, \ldots, \mathbf{D}_J^\top]^\top, \tag{4.67}$$

Λ^k is the block diagonal matrix whose diagonal elements are $(\kappa(\lambda_j^k)^{\kappa-1}\mathbf{I}_S)_{1\leq j\leq J}$, and $(\mathbf{e}_1, \ldots, \mathbf{e}_P)$ is the canonical basis of $\mathbb{R}^P$. $\boldsymbol{\mu}(\mathbf{y}) = [\mu_1(y_1), ..., \mu_N(y_N)]^\top$, $\mathbf{u}$ is a $N \times 1$ vector whose $i-$th component is given by $u_i = \mu_i(y_i) w_i^k$. We thus obtain a Gaussian distribution:

$$q_{\mathbf{X}}^{k+1}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \check{\mathbf{x}}^{k+1}, \check{\mathbf{C}}_{\mathbf{x}}^{\,k+1}), \tag{4.68}$$

parametrized by

$$(\check{\mathbf{C}}_{\mathbf{x}}^{\,k+1})^{-1} = (\mathbf{H}^k)^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{H}^k + \sum_{p=1}^{P}\sum_{q=1}^{P} \mathbf{e}_p^\top \mathbf{C}_{\mathbf{z}}^k \mathbf{e}_q \mathbf{K}_p^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_q + 2\gamma^k \mathbf{D}^{\mathrm{T}} \boldsymbol{\Lambda}^k \mathbf{D}, \tag{4.69}$$

$$\check{\mathbf{x}}^{k+1} = \check{\mathbf{C}}_{\mathbf{x}}^{\,k+1} (\mathbf{H}^k)^\top \mathbf{u}, \tag{4.70}$$

with $\mathbf{H}^k = \mathcal{H}(\mathbf{z}^k)$.
Hereagain, we propose to resort to a diagonal approximation to this matrix when required, so that the update finally reads:

$$q_{\mathbf{X}}^{k+1}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{x}^{k+1}, \mathbf{C}_{\mathbf{x}}^{k+1}), \tag{4.71}$$

with

$$\mathbf{C}_{\mathbf{x}}^{k+1} = \mathrm{Diag}\left(\delta_{\mathbf{x}}^{k+1}\right) \tag{4.72}$$

$$\mathbf{x}^{k+1} = \mathrm{CG}\Big((\check{\mathbf{C}}_{\mathbf{x}}^{\,k+1})^{-1}, (\mathbf{H}^k)^\top \mathbf{u}\Big), \tag{4.73}$$

where $\delta_{\mathbf{x}}^{k+1} \in \mathbb{R}^N$ is the vector of the inverses of the diagonal elements of $(\check{\mathbf{C}}_{\mathbf{x}}^{\,k+1})^{-1}$.

4.4.2.2 Update of $q_{\mathbf{Z}}(\mathbf{z})$

According to the VBA principle,

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) = \mathrm{argmin}_{q_{\mathbf{Z}}} \mathcal{KL}(q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^k(\gamma) q_{\mathbf{Z}}(\mathbf{z}) || \mathcal{F}(\Theta | \mathbf{y}; \mathbf{w}^k, \boldsymbol{\lambda}^k)). \tag{4.74}$$

Using the VBA optimality condition (2.15) and the previously introduced bound $\mathcal{F}(\Theta \mid \mathbf{y}; \mathbf{w}^k, \boldsymbol{\lambda}^k)$, we have

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \propto \exp\left(\int\int \ln \mathcal{F}(\mathbf{x}, \mathbf{z}, \gamma \mid \mathbf{y}; \mathbf{w}^k, \boldsymbol{\lambda}^k) q_{\Gamma}^k(\gamma) q_{\mathbf{X}}^{k+1}(\mathbf{x}) \mathrm{d}\gamma \mathrm{d}\mathbf{x}\right). \tag{4.75}$$

Replacing the involved quantities by their expression yields

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \propto \exp\left\{-\frac{1}{2}\mathbf{z}^\top \left(\mathbf{B}^{k+1} + \xi \mathbf{L}\right)\mathbf{z} + \mathbf{z}^\top \left(\mathbf{a}^{k+1} + \xi \mathbf{L}\boldsymbol{\mu}\right)\right\}, \tag{4.76}$$

where $\mathbf{a}^{k+1} = (a_p^{k+1})_{1\leq p\leq P} \in \mathbb{R}^P$ and $\mathbf{B}^{k+1} = (B_{p,q}^{k+1})_{1\leq p,q\leq P} \in \mathbb{R}^{P\times P}$ are such that, for every $(p,q) \in \{1,\dots,P\}^2$,

$$\begin{aligned} a_p^{k+1} =& \mathbb{E}_{q_{\mathbf{X}}^{k+1}(\mathbf{x})}(\mathbf{x})^\top \mathbf{K}_p^\top \mathbf{u} - \mathbb{E}_{q_{\mathbf{X}}^{k+1}}(\mathbf{x}^\top \mathbf{K}_p^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_0\mathbf{x}) \\ =& (\mathbf{x}^{k+1})^\top \mathbf{K}_p^\top \mathbf{u} - \mathbf{B}_{p,0}^{k+1}, \qquad (4.77) \\ B_{p,q}^{k+1} =& \mathbb{E}_{q_{\mathbf{X}}^{k+1}}(\mathbf{x}^\top \mathbf{K}_p^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_q\mathbf{x}) \\ =& \mathrm{trace}\left(\mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_p\mathbf{C}_{\mathbf{x}}^{k+1}\mathbf{K}_q^\top\right) + (\mathbf{x}^{k+1})^\top \mathbf{K}_p^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_q\mathbf{x}^{k+1} \qquad (4.78) \end{aligned}$$

with

$$\begin{aligned} B_{p,0}^{k+1} =& \mathbb{E}_{q_{\mathbf{X}}^{k+1}}(\mathbf{x}^\top \mathbf{K}_p^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_0\mathbf{x}) \\ =& \mathrm{trace}\left(\mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_p\mathbf{C}_{\mathbf{x}}^{k+1}\mathbf{K}_0^\top\right) + (\mathbf{x}^{k+1})^\top \mathbf{K}_p^\top \mathrm{Diag}(\boldsymbol{\mu}(\mathbf{y}))\mathbf{K}_0\mathbf{x}^{k+1}. \qquad (4.79) \end{aligned}$$

Thus, the update for the distribution $q_{\mathbf{Z}}$ reads

$$q_{\mathbf{Z}}^{k+1}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{z}^{k+1}, \mathbf{C}_{\mathbf{z}}^{k+1}), \qquad (4.80)$$

with

$$\begin{aligned} (\mathbf{C}_{\mathbf{z}}^{k+1})^{-1} =& \mathbf{B}^{k+1} + \xi\mathbf{L}, \qquad (4.81) \\ \mathbf{z}^{k+1} =& \mathbf{C}_{\mathbf{z}}^{k+1}\left(\mathbf{a}^{k+1} + \xi\mathbf{L}\boldsymbol{\mu}\right). \qquad (4.82) \end{aligned}$$

4.4.2.3 Update of w

Let us now express the update of the auxiliary variable $\mathbf{w}$. We aim at finding

$$\mathbf{w}^{k+1} = \mathrm{argmin}_{\mathbf{w}} \mathcal{KL}(q_{\mathbf{X}}^{k+1}(\mathbf{x})q_{\Gamma}^{k}(\gamma)q_{\mathbf{Z}}^{k+1}(\mathbf{z}) || \mathcal{F}(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}^k)). \qquad (4.83)$$

which leads to

$$\begin{aligned} \mathbf{w}^{k+1} =& \mathrm{argmin}_{\mathbf{w}} \int q_{\mathbf{X}}^{k+1}(\mathbf{x})q_{\Gamma}^{k}(\gamma)q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \times \ln \frac{q_{\mathbf{X}}^{k+1}(\mathbf{x})q_{\Gamma}^{k}(\gamma)q_{\mathbf{Z}}^{k+1}(\mathbf{z})}{L(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}^k)} \mathrm{d}\Theta \qquad (4.84) \\ =& \mathrm{argmin}_{\mathbf{w}} \int q_{\mathbf{X}}^{k+1}(\mathbf{x})q_{\Gamma}^{k}(\gamma)q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \left(-\ln L(\Theta|\mathbf{y};\mathbf{w},\boldsymbol{\lambda}^k)\right) \mathrm{d}\Theta \qquad (4.85) \\ =& \mathrm{argmin}_{\mathbf{w}} \int q_{\mathbf{X}}^{k+1}(\mathbf{x})q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \sum_{i=1}^{N} T_i([\mathbf{H}\mathbf{x}]_i, w_i; y_i) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \qquad (4.86) \\ =& \mathrm{argmin}_{\mathbf{w}} \int q_{\mathbf{X}}^{k+1}(\mathbf{x})q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \sum_{i=1}^{N} T_i([\mathcal{H}(\mathbf{z})\mathbf{x}]_i, w_i; y_i) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z} \qquad (4.87) \\ =& \mathrm{argmin}_{\mathbf{w}} \int q_{\mathbf{X}}^{k+1}(\mathbf{x})q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \sum_{i=1}^{N} T_i\left(\left[\left(\sum_{p=1}^{P} z_p\mathbf{K}_p + \mathbf{K}_0\right)\mathbf{x}\right]_i, w_i; y_i\right) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}. \qquad (4.88) \\ & \qquad (4.89) \end{aligned}$$

It follows that, for every $i \in \{1,\dots,N\}$,

$$w_i^{k+1} = \operatorname{argmin}_{w_i} T_i\left(\left[\mathbf{H}^{k+1}\mathbf{x}^{k+1}\right]_i, w_i; y_i\right) \tag{4.90}$$

$$= \left[\mathbf{H}^{k+1}\mathbf{x}^{k+1}\right]_i - \frac{1}{\mu_i(y_i)}\phi_i'\left(\left[\mathbf{H}^{k+1}\mathbf{x}^{k+1}\right]_i; y_i\right). \tag{4.91}$$

4.4.2.4 Update of $\boldsymbol{\lambda}$

Let us now express the update of the auxiliary variable $\boldsymbol{\lambda}$. We aim at finding

$$\boldsymbol{\lambda}^{k+1} = \operatorname{argmin}_{\boldsymbol{\lambda}} \mathcal{KL}(q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) || \mathcal{F}(\Theta|\mathbf{y}; \mathbf{w}^{k+1}, \boldsymbol{\lambda})). \tag{4.92}$$

This amounts to finding, for every $j \in \{1,\dots,J\}$,

$$\begin{aligned}
\lambda_j^{k+1} &= \operatorname{argmin}_{\lambda_j \in [0,+\infty)} \int q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \times \log \frac{q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z})}{\mathcal{F}(\Theta|\mathbf{y}, \mathbf{w}^{k+1}, \boldsymbol{\lambda})} \mathrm{d}\Theta, \\
&= \operatorname{argmin}_{\lambda_j \in [0,+\infty)} \sum_{j=1}^{J} \int \int q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}^{k}(\gamma) \times F_j(\mathbf{D}_j\mathbf{x}, \lambda_j; \gamma) \mathrm{d}\mathbf{x}\mathrm{d}\gamma, \\
&= \operatorname{argmin}_{\lambda_j \in [0,+\infty)} \frac{\kappa \mathbb{E}_{q_{\mathbf{x}}^{k+1}(\mathbf{x})}\left[||\mathbf{D}_j\mathbf{x}||^2\right] + (1-\kappa)\lambda_j}{\lambda_j^{1-\kappa}}.
\end{aligned} \tag{4.93}$$

The explicit solution to the above minimization problem yields the following update:

$$\begin{aligned}
\lambda_j^{k+1} &= \mathbb{E}_{q_{\mathbf{x}}^{k+1}(\mathbf{x})}\left[||\mathbf{D}_j\mathbf{x}||^2\right] \\
&= ||\mathbf{D}_j\mathbf{x}^{k+1}||^2 + \operatorname{trace}\left(\mathbf{D}_j^{\top}\mathbf{D}_j\mathbf{C}_{\mathbf{x}}^{k+1}\right).
\end{aligned} \tag{4.94}$$

4.4.2.5 Update of $q_{\Gamma}(\gamma)$

Finally, the update related to the hyperparameter γ is expressed as

$$q_{\Gamma}^{k+1}(\gamma) = \operatorname{argmin}_{q_{\Gamma}} \mathcal{KL}(q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\Gamma}(\gamma) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) || \mathcal{F}(\Theta|\mathbf{y}; \mathbf{w}^{k+1}, \boldsymbol{\lambda}^{k+1})). \tag{4.95}$$

Using (2.15), we have

$$q_{\Gamma}^{k+1}(\gamma) \propto \exp\left(\int \int \ln \mathcal{F}(\mathbf{x}, \mathbf{z}, \gamma \mid \mathbf{y}; \mathbf{w}^{k+1}, \boldsymbol{\lambda}^{k+1}) \times q_{\mathbf{X}}^{k+1}(\mathbf{x}) q_{\mathbf{Z}}^{k+1}(\mathbf{z}) \, \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{z}\right). \tag{4.96}$$

The above integral has the following closed form expression:

$$q_{\Gamma}^{k+1}(\gamma) \propto \gamma^{\frac{N}{2\kappa}+\alpha-1} \exp(-\eta\gamma) \times \exp\left(-\gamma \sum_{j=1}^{J} \frac{\kappa \mathbb{E}_{q_{\mathbf{x}}^{k+1}(\mathbf{x})}\left[||\mathbf{D}_j\mathbf{x}||^2\right] + (1-\kappa)\lambda_j^{k+1}}{(\lambda_j^{k+1})^{1-\kappa}}\right). \tag{4.97}$$

It thus follows from (4.94) that the update of q_{Γ} is

$$q_{\Gamma}^{k+1}(\gamma) = \Gamma(d, b^{k+1}), \tag{4.98}$$

that is the Gamma distribution with parameters

$$d = \frac{N}{2\kappa} + \alpha, \quad b^{k+1} = \sum_{j=1}^{J} (\lambda_j^{k+1})^{\kappa} + \eta. \tag{4.99}$$

The mean of q_{Γ}^{k+1} is finally given by

$$\gamma^{k+1} = \frac{d}{b^{k+1}}. \tag{4.100}$$

4.4.3 Overview of VBA with non-Gaussian noise

Algorithm 2 provides a summary of the resulting VBA for solving the blind deconvolution problem introduced in Section 4 under non-Gaussian noise. As a result, the optimal posterior distributions for both variables $\mathbf{x}$ and $\mathbf{z}$ will be approximated as Gaussian distributions, while the one for hyperparameter γ is approximated by a Gamma distribution. We also provide the update formula for the hyperparameter $\mathbf{w}$.

Algorithm 2 VBA approach for image blind deconvolution under non-Gaussian noise

Initialization. Set hyperparameters (ξ, α, η). Define initial values for $(\mathbf{x}^0, \mathbf{C}_{\mathbf{x}}^0, \mathbf{z}^0, \mathbf{C}_{\mathbf{z}}^0)$. Compute $\boldsymbol{\lambda}^0$, γ^0 and $\mathbf{w}^0$ using (4.94), (4.100) and (4.91), respectively.

Iterative steps. For $k = 0, 1, \ldots, K$:

1: Update the mean $\mathbf{x}^{k+1}$ and the covariance matrix $\mathbf{C}_{\mathbf{x}}^{k+1}$ of $q_{\mathbf{X}}^{k+1}(\mathbf{x})$ using (4.72)-(4.73).
2: Update the mean $\mathbf{z}^{k+1}$ and the covariance matrix $\mathbf{C}_{\mathbf{z}}^{k+1}$ of $q_{\mathbf{Z}}^{k+1}(\mathbf{z})$ using (4.81)-(4.82).
3: Update $\mathbf{w}_i^{k+1}$ using (4.91), for every $i \in \{1, \ldots, N\}$.
4: Update λ_j^{k+1} using (4.94), for every $j \in \{1, \ldots, J\}$.
5: Update the mean γ^{k+1} of $q_{\Gamma}^{k+1}(\gamma)$ using (4.99)-(4.100).

4.5 Conclusion

In this chapter, we showed how to address the image blind deblurring problem through the variational Bayesian paradigm. We introduced a general hierarchical Bayesian model involving suitable priors for the image and the kernel. We introduced majorant functions to allow closed-form updates for each variables, for both Gaussian and non-Gaussian noises. Nonetheless, our preliminary experiments showed that it appears difficult to find an efficient manner to estimate the hyperparameter ξ, as this value highly fluctuates from one image/kernel pair to the other so that a simple prior modeling does not appear obvious. Moreover, in both considered noise settings, our VBA requires the knowledge of the noise level, through β (in Gaussian case) or $(\mu_i(y_i))_{1 \leq i \leq N}$ (in non-Gaussian case). This is limiting, and one might prefer to have this quantity estimated in an automatic manner. We thus propose in the next chapter, to resort to a supervised learning strategy to learn both ξ and β along the iterates of VBA, in the spirit of recent works [Bertocchi *et al.* 2019] on the unrolling (also called unfolding) of iterative algorithms.

CHAPTER 5

Deep unrolled variational Bayesian algorithm for image blind deconvolution

Contents

In Chapter 4, we address the problem of blind image deconvolution using the variational Bayesian strategy. This strategy, as well as the references cited in the introduction of Chapter 4, belong to the class of traditional restoration methods. They rely on explicit and explainable steps, grounded on solid optimization/Bayesian methodological tools. This somewhat contrasts with the recent trend in image restoration relying on the deep neural network machinery. The use of deep learning for image deblurring only goes back to the last decade [Kupyn *et al.* 2018, Dong *et al.* 2016, Albluwi *et al.* 2018]. These methods rely on the deep learning methodology. More precisely, a supervised learning strategy is adopted to learn (implicitly) some prior information on the image/kernel from a so-called training set. A highly non-linear and multi-layers architecture is built, and its parameters (i.e., neuron weights) are estimated by back-propagation to minimize a given loss function associated with the task at hand (e.g., image visual quality). Several recent works propose neural network architectures dedicated to the problem of image blind deconvolution. Let us mention DeblurGAN [Kupyn *et al.* 2018], based on conditional generative adversarial networks and a multi-component loss function, SRCNN [Dong *et al.* 2016] and its extended version, DBSRCNN [Albluwi *et al.* 2018], relying on a convolutional neural network (CNN) architecture. SelfDeblur [Ren *et al.* 2020] combines an optimization-based

method with two generative networks for modeling deep priors on image and blur kernel, respectively. Other image deblurring problems have been studied using neural network techniques, for example [Wen *et al.* 2021] presents an unsupervised method based on multi-adversarial CycleGAN [Zhu *et al.* 2017] for high-resolution image generation, and [Mohan *et al.* 2021] proposes a dynamic scene deblurring method for unconstrained dual-lens (DL) cameras. Most of these methods are supervised. They can perform quite well, as long as the training set is large and representative enough. Moreover, they are well suited to GPU-based implementation. However, they have traditionally suffered from lack of interpretability and robustness [Bietti *et al.* 2018].

Both traditional (optimization/Bayesian) approaches and deep learning methods are able to provide good performance. For most traditional image restoration methods, the common important challenge is the (hyper)parameter tuning. The parameters can arise from the model itself (e.g., prior weight, noise level) or from the resolution method (e.g., algorithm stepsize). Several strategies are available, based on empirical search, or more advanced statistical tools. However, these methods are often time-consuming without guaranteed success. While for deep learning methods, they lack theoretical guarantee and sometimes they demand numerous trials to find an optimal network structure. An emerging set of methods, in the field of inverse problems in signal/image processing, performs algorithm unrolling [Monga *et al.* 2021]. An iterative method (e.g., an optimization algorithm) is unrolled as layers of a neural network. The reduced set of parameters of this network are learnt by supervised training. Promising results have been obtained in the context of image deconvolution in [Li *et al.* 2020, Bertocchi *et al.* 2019, Pesquet *et al.* 2021, Galinier *et al.* 2020], as well as other image restoration applications [Zhang *et al.* 2020]. One key advantage is that cumbersome parameter tuning can be avoided, by simply embedding the sought parameters into the network and learning them at training phase. Theoretical results assessing the stability and robustness of unrolling techniques can be found in [Pesquet *et al.* 2021, Combettes & Pesquet 2020, Chouzenoux *et al.* 2021]. These methods are also closely related to plug-and-play techniques where a trained neural network is employed as the denoiser [Pesquet *et al.* 2021, Sun *et al.* 2021, Zhang *et al.* 2021]. Deep unrolling can also be combined with a bilevel optimization framework, still with the aim for efficient hyperparameter learning (see [Crockett & Fessler 2022] for a recent survey).

In this chapter, we propose a novel approach for blind image deconvolution, that aims at gathering the best of the aforementioned methods. To be specific, we apply the unrolling paradigm to the algorithm presented in Chapter 4. We focus here on the case of Gaussian noise only. We create a deep neural network architecture, where VBA iterations proposed in Section 4.3 are integrated as layers. This allows us to (i) learn the hyperparameters (in particular, the noise level) of VBA in an automatic supervised fashion, (ii) improve further the quality of the results by choosing a dedicated loss in the training phase, and (iii) implement the method by taking full advantage of possible GPU resources, thus considerably reducing the processing time during the test phase. In contrast to standard deep learning methods for blind deconvolution, all these benefits come along with a preservation of the interpretability of the method, thanks to the unrolling technique. Let us emphasize that variational Bayesian methods often appear in deep learning context. Indeed, they are backbones of variational autoencoders [Kingma & Welling 2014] and also constitute methods

of choice for training Bayesian neural networks [Jospin *et al.* 2020]. However, up to our knowledge, our work is the first to investigate the unrolling of a variational Bayesian technique. The rest of Chapter 5 is organized as follows. The unrolling of VBA is presented in Section 5.1. Numerical results, including comparisons with various methods (in particular, our algorithm from Section 4.3), are presented in Section 5.2. Section 5.3 concludes this chapter.

5.1 Deep unrolled architecture

5.1.1 Overview

We introduce a supervised learning strategy to estimate the hyperparameter ξ and the inverse of the noise variance β, that are required to run VBA from Section 4.3. We adopt the so-called *unrolling* (or unfolding) methodology [Monga *et al.* 2021]. The idea is to view each iteration of an iterative algorithm as one layer of a neural network structure. Each layer can be parametrized by some quantities that are learned from a training database so as to minimize a task-oriented loss function. The advantage of the unrolling approach is threefold: (i) each layer mimics one iteration of the algorithm and thus it is highly interpretable, (ii) the choice of the loss is directly related to the task at the end, which is beneficial for the quality of the results, (iii) once trained, the network can be applied easily and rapidly on a large set of test data without any further tuning. In particular, its implementation can make use of GPU-accelerated frameworks. Several recent examples in the field of image processing have shown the benefits of unrolling [Gilton *et al.* 2021, Li *et al.* 2021, Tolooshams *et al.* 2021, Nan *et al.* 2020] when compared to standard black-box deep learning techniques or more classical restoration methods based on Bayesian or optimization tools. Let us in particular mention the works [Li *et al.* 2020, Li *et al.* 2019] for the application of unrolling in the context of blind image restoration.

Let us now specify the unrolling procedure in the context of VBA. Let $K > 0$ be the number of iterations of the VBA described in Algorithm 1, thus corresponding to K layers of a neural network architecture. Iteration $k \in \{0,\dots,K-1\}$ of our unrolled VBA can be conceptually expressed as

$$\begin{aligned}&(\mathbf{x}^{k+1},\mathbf{C}_{\mathbf{x}}^{k+1},\mathbf{z}^{k+1},\mathbf{C}_{\mathbf{z}}^{k+1},\boldsymbol{\lambda}^{k+1},\boldsymbol{\gamma}^{k+1})\\&=\mathcal{A}(\mathbf{x}^{k},\mathbf{C}_{\mathbf{x}}^{k},\mathbf{z}^{k},\mathbf{C}_{\mathbf{z}}^{k},\boldsymbol{\lambda}^{k},\boldsymbol{\gamma}^{k},\xi^{k},\beta^{k}).\end{aligned}\tag{5.1}$$

The initialization procedure for $(\mathbf{x}^0,\mathbf{C}_{\mathbf{x}}^0,\mathbf{z}^0,\mathbf{C}_{\mathbf{z}}^0,\boldsymbol{\lambda}^0,\boldsymbol{\gamma}^0)$ is detailed in Algorithm 1. For $k \in \{0,\dots,K-1\}$, the expressions of $(\mathbf{x}^{k+1},\mathbf{C}_{\mathbf{x}}^{k+1},\mathbf{z}^{k+1},\mathbf{C}_{\mathbf{z}}^{k+1},\boldsymbol{\gamma}^{k+1},\boldsymbol{\lambda}^{k+1})$ as a function of the input arguments of $\mathcal{A}(\cdot)$ are given respectively by (4.72)-(4.73), (4.81)-(4.82), (4.94), and (4.99)-(4.100). Furthermore, $(\xi^k,\beta^k)_{0\leq k\leq K-1}$ are now learned, instead of being constant and preset by the user. This leads to the *unfoldedVBA* architecture depicted in Fig. 5.1, which can be summarized into the composition of K layers $\mathcal{L}_{K-1}\circ\cdots\circ\mathcal{L}_0$. Each layer $\mathcal{L}_k$ with $k\in\{0,\dots,K-1\}$ is made of three main blocks, that are two neural networks, namely NN_{σ}^{k} and NN_{ξ}^{k}, and the core VBA block $\mathcal{A}(\cdot)$. There remains to specify our strategy for building the two inner networks, with the aim to learn $(\xi^k,\beta^k)_{0\leq k\leq K-1}$.

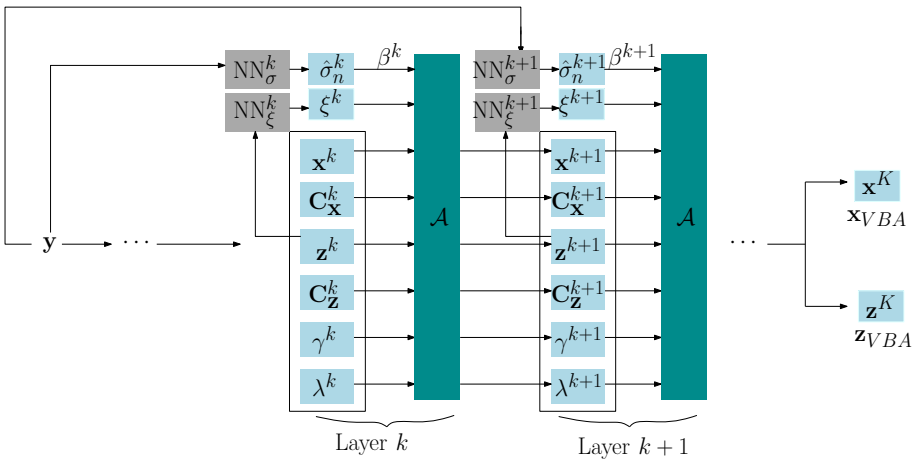


Figure 5.1: Architecture of *unfoldedVBA* network.

5.1.2 Learning hyperparameter ξ

For every $k \in \{0,...,K-1\}$, neural network NN_ξ^k takes as input the current kernel estimate $\mathbf{h}^k = \mathbf{T}\mathbf{z}^k + \mathbf{t}$ and delivers ξ^k as an output. The architecture of the neural network is shown in Fig. 5.2. Note that the Softplus function, defined as

$$(\forall x \in \mathbb{R}) \quad \mathrm{Softplus}(x) = \ln(1+\exp(x)), \tag{5.2}$$

is used as a last layer, in order to enforce the strict positivity of the output hyperparameter ξ^k.

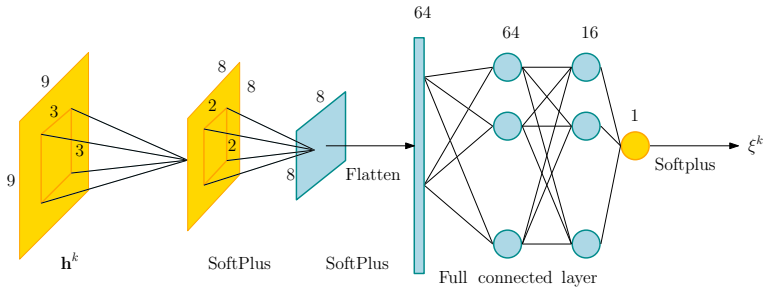


Figure 5.2: Neural network architecture NN_ξ^k for estimating ξ^k, for $k \in \{0,...,K-1\}$.

5.1.3 Learning noise parameter β

When the noise parameter β is unknown, it might be useful to include a procedure to learn it automatically, again in a supervised fashion. In this case, we propose to introduce simple nonlinear mappings such that, for every $k \in \{0,\ldots,K-1\}$,

$$\sigma^k = \mathrm{NN}_\sigma^k(\mathbf{y}) = \mathrm{Softplus}(\rho_k)\widehat{\sigma}(\mathbf{y}) + \mathrm{Softplus}(\tau_k), \tag{5.3}$$

and $\beta^k = (\sigma^k)^{-2}$. Hereabove, $\mathbf{y}$ is the observed degraded image, from which we deduce the wavelet-based variance estimator (also used in [Bertocchi *et al.* 2019]),

$$\widehat{\sigma}(\mathbf{y}) = \frac{\text{median}(|\mathbf{W}_H \mathbf{y}|)}{0.6745}, \tag{5.4}$$

where $|\mathbf{W}_H \mathbf{y}|$ gathers the absolute value of the diagonal coefficients of the first level Haar wavelet decomposition of the degraded image $\mathbf{y}$. Moreover, $(\rho_k, \tau_k)_{0 \leq k \leq K-1}$ are two scalar parameters to be learned during the training phase.

5.1.4 Complete architecture

We now present our complete blind deconvolution architecture for grayscale images and color images in Fig. 5.3. First, let us notice that VBA and its unrolled variant is designed for grayscale images. We thus generalized the architecture from Fig. 5.3(top), to process color images. To this end, we first transform the input RGB image to its YUV representation, which takes human perception into consideration. The network NN_σ^k is first applied to the luminance part $\mathbf{y}_Y$ of the image. After applying the *unfoldedVBA* network (see Fig. 5.1), we obtain $\mathbf{z}_{\text{VBA}}$ and $\mathbf{x}_{\text{VBA}}$ as outputs. The latter is a restored version of the luminance channel. The remaining (U,V) color channels are simply obtained by median filtering of $(\mathbf{y}_U, \mathbf{y}_V)$. Both architectures in Fig. 5.3 additionally involve post-processing layers. More precisely, we first include a linear layer so as to encode the linear transformation (4.3), and then deduce the estimated blur kernel $\widehat{\mathbf{h}}$. Second, we also allow a post-processing layer $\mathcal{L}_{\text{pp}}$ acting on the image, so as to reduce possible residual artifacts, finally yielding $\widehat{\mathbf{x}}$. In the case of color images, the post-processing is applied on the RGB representation to avoid chromatic artifacts.

5.1.5 Training procedure

The training of both proposed architectures from Fig. 5.3 requires to define a loss function, measuring the discrepancy between the output $(\widehat{\mathbf{x}}, \widehat{\mathbf{h}})$ and the ground truth $(\overline{\mathbf{x}}, \overline{\mathbf{h}})$, that we denote hereafter by $\ell(\widehat{\mathbf{x}}, \widehat{\mathbf{h}}, \overline{\mathbf{x}}, \overline{\mathbf{h}})$. In the blind deconvolution application, one can for instance consider a loss function related to the error reconstruction on the kernel, or to the image quality, or a combination of both. Two training procedures will be distinguished and discussed in our experimental section, namely:

Greedy training The parameters of the unfolded VBA are learned in a greedy fashion so as to minimize the kernel reconstruction error at each layer. Then, the post-processing network is learned in a second step, so as to maximize an image quality metric such as the structural similarity index measure (SSIM) [Wang *et al.* 2004].

End-to-end training The parameters of the complete architecture are learned end-to-end so as to maximize the image quality metric.

Whatever the chosen training procedure, it is necessary to make use of a back-propagation step, that is to differentiate the loss function with respect to all the parameters of the network. Most operations involved in Fig. 5.3 can be differentiated efficiently using standard auto-differentiation tools. However, we observed in our experiments that it is beneficial (and sometimes even necessary) for a stable training phase to avoid using such tools

for differentiating the VBA layer $\mathcal{A}(\cdot)$ involved in Fig. 5.1. In practice, we used the explicit expressions for the partial derivatives of it. Note that we followed the approach from [Charlier *et al.* 2021] to obtain the expression of the derivatives for the CG solver.

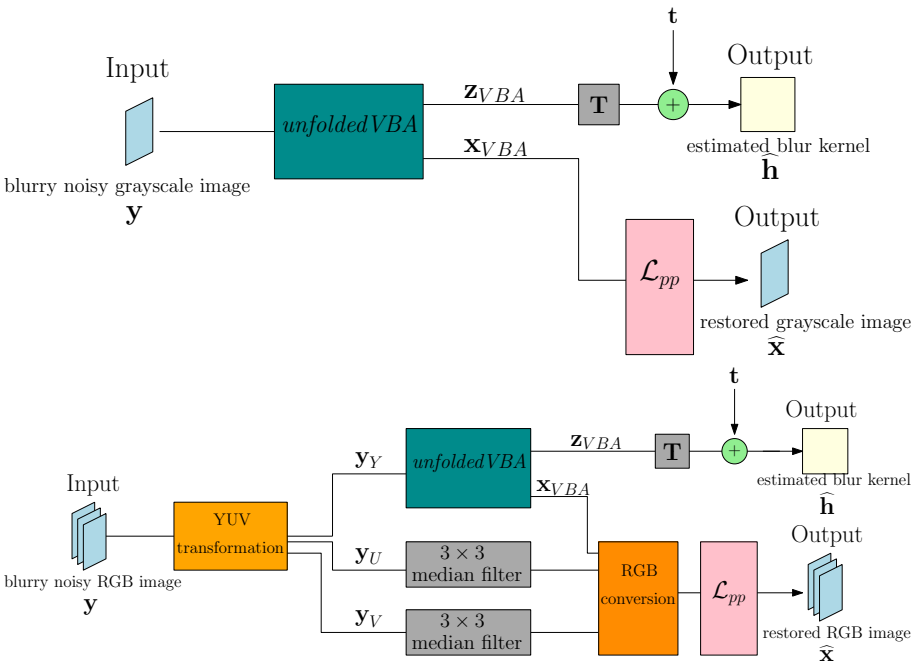


Figure 5.3: Proposed blind image restoration pipeline for grayscale (top) and color (bottom) images.

5.2 Experimental results

5.2.1 Problem formulation and settings

5.2.1.1 Problem overview

We focus on the resolution of the blind image deconvolution problem, where $\widetilde{\mathbf{x}} \in \mathbb{R}^N$ is an original image, either grayscale or colored one. We come back to the model presented in Section 4.2.1, where the linear operator $\widetilde{\mathbf{H}} \in \mathbb{R}^{N \times N}$ models the application of a blur kernel $\widetilde{\mathbf{h}} \in \mathbb{R}^M$ to the image. The noise $\mathbf{n}$ is assumed to be an additive white Gaussian noise with zero mean and standard deviation σ. In the case of color images, we assume that the same kernel, and the same noise level, is applied to each of the three RGB channels.

5.2.1.2 Datasets

Let us introduce the three datasets used to train and test our network, and compare our approach with state-of-the-art techniques. In all the cases, the training set is made of 100 images from the COCO training set. The validation set contains 40 images from the BSD500 validation set. For *Dataset 1* and *Dataset 2*, the test set consists of 30 images from the Flickr30 test set. For *Dataset 3*, we adopted the same testing strategy

as in [Ren *et al.* 2020], using the 4 images from [Levin *et al.* 2009]. Each image is center-cropped using a window of size $N = 256 \times 256$. Each original image $\widetilde{\mathbf{x}}$ is associated to a degraded version of it, $\mathbf{y}$, built from Model (4.1). Various blur kernels and noise levels are used, as detailed hereafter.

Dataset 1: All involved images are converted in grayscale. Each image of the database is blurred with 10 randomly generated Gaussian blurs, and then corrupted by additive noise. Thus in total, we have 1000 ($= 100 \times 10$) training images, 400 ($= 40 \times 10$) validation images, and 300 ($= 30 \times 10$) test images for Dataset 1. The Gaussian blurs are of size 9×9. Two of them are isotropic with standard deviation randomly generated following a uniform law within $[0.2, 0.4]$. Eight of them are anisotropic with orientation either $\pi/4$ or $3\pi/4$ (with equal probability) and vertical/horizontal widths (i.e., standard deviations of the 2D Gaussian shape) uniformly drawn within $[0.15, 0.4]$. On this dataset, the noise standard deviation is set to $\sigma = 0.01$, and assumed to be known (so that blocks $(\mathrm{NN}_\sigma^k)_{1 \leq k \leq K}$ of our architecture are overlooked).

Dataset 2: All the images are then colored ones. We degraded each of them with 15 different blurs, namely 10 Gaussian blurs (simulated using the same procedure as above), two uniform blurs with width 5×5 and 7×7 pixels, and 3 out-of-focus blurs. For the latter, the vertical and horizontal widths are set randomly within $[0.2, 0.5]$, and the orientation is either $\pi/4$ or $3\pi/4$ (with equal probability). Furthermore, for each blurred image, zero-mean Gaussian noise is added, with standard deviation σ randomly chosen, with uniform distribution over $[0.005, 0.05]$. The noise level is not assumed to be known and is estimated using the proposed NN_σ architecture. In total, we have 1500 ($= 100 \times 15$) training images, 600 ($= 40 \times 15$) validation images, and 450 ($= 30 \times 15$) test images, on this dataset. Examples of blurs involved in *Dataset 2* are depicted in Fig. 5.4.

Dataset 3: All the images are converted in grayscale. We degraded each of them with 8 different real-world motion blurs from[1] [Levin *et al.* 2009]. As the kernels have different sizes ranging from 13×13 to 27×27, we pad them with 0 values, so to reach a common large size of 27×27. For each blurred image, zero-mean Gaussian noise is added, with standard deviation σ randomly chosen, following a uniform distribution within $[5 \times 10^{-6}, 5 \times 10^{-5}]$. The noise level is not assumed to be known and is estimated using the proposed NN_σ architecture. In total, we have 800 ($= 100 \times 8$) training images, 320 ($= 40 \times 8$) validation images, and 32 ($= 4 \times 8$) test images, on this dataset. The motion blurs involved in *Dataset 3* are illustrated in Fig. 5.5.

5.2.1.3 VBA settings

Different settings are adopted for the implementation of the VBA step on each dataset, that we specified hereafter. For all considered datasets, we seek for kernels whose entries satisfy a sum-to-one constraint. For the datasets involving synthetic blurs (namely, *Dataset 1* and *Dataset 2*), we impose additionally an axial symmetry along the main diagonal axis. In all cases, the constraints can be easily translated into the affine constraint (4.3). For *Dataset*

[1] https://webee.technion.ac.il/people/anat.levin/

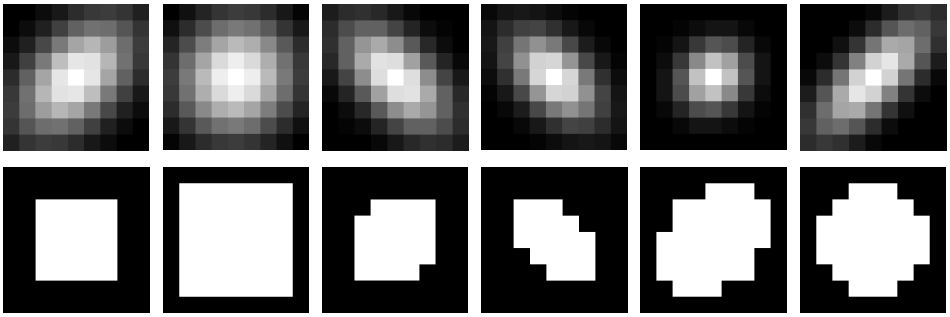


Figure 5.4: Examples of blur kernels used to construct *Dataset 2*.

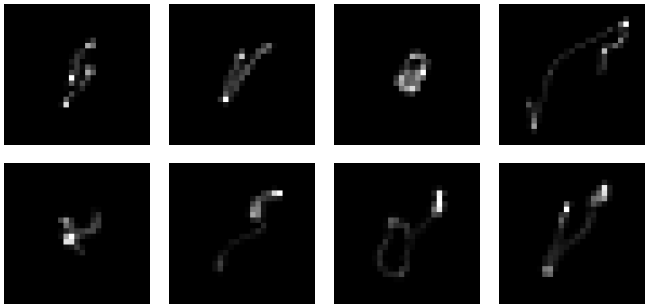


Figure 5.5: Motion blur kernels from *Dataset 3*.

1 and *Dataset 2*, the degree of freedom of the kernel model is equal to $P = \frac{(\sqrt{M}+1)\sqrt{M}}{2} - 1$, while we have $P = M - 1$ for *Dataset 3*. Regarding the choice of the blur prior, hereagain, a distinction is made between the first two scenarios and the last one. The synthetic blurs from *Dataset 1* and *Dataset 2* are either very smooth (e.g., Gaussian blurs) or present large constant areas (e.g., defocus blurs). In order to incorporate such prior knowledge, for both these datasets, $\mathbf{A} \in \mathbb{R}^{(2M+1)\times M}$ is set to the matrix that computes the horizontal and vertical differences between pixels, augmented with an additional first row corresponding to an averaging operation, i.e., $[1,\ldots,1]/M$. This choice allows us to promote smooth variations in the kernels, while satisfying the required full column rank assumption on $\mathbf{A}$. A constant vector with entries equal to $\frac{1}{M}$ is set for the prior mean $\mathbf{m}$. In contrast, the motion blurs of *Dataset 3* depicted in Fig. 5.5 present complex shapes, with no specific property on their gradient. We thus simply set $\mathbf{A}$ to the identity matrix in the prior, while a rough estimation of the blur obtained from [2] [Kotera *et al.* 2013] is used for $\mathbf{m}$. Matrices $(\mathbf{D}_j)_{1\leq j\leq J}$ and parameter κ, involved in (4.8), are set in such a way that the chosen prior on the image yields an isotropic total-variation regularization (see our comment in Section 4.2.2.2). We must now specify the initialization for VBA iterates/layers. Our initial guess $\mathbf{x}^0$ for the image is the degraded one. The associated covariance matrix $\mathbf{C}_\mathbf{x}^0$ is initialized using the identity matrix. For *Dataset 1* and *Dataset 2*, the blur is initialized with a 5×5 uniform kernel,

[2] http://zoi.utia.cas.cz/deconv_sparsegrad

from which we deduce the corresponding $\mathbf{z}^0$, while we use the prior mean for initializing the method for *Dataset 3*. In all VBA experiments, the covariance matrix $\mathbf{C}_{\mathbf{z}}^0$ is set to a multiple of identity matrix. As the blurs involved in the *Dataset 3* are of large size, a diagonal approximation of $\mathbf{C}_{\mathbf{z}}^k$ is used in (4.69) so as to cope with memory saturation issues. A similar strategy was adopted for instance in [Marnissi *et al.* 2017] and was not observed to yield any degradation of the resulting quality. The hyperparameters (α, η) involved in the prior law on parameter γ are set to zero in practice which is equivalent to impose a non-informative Jeffrey improper prior. Finally, the conjugate gradient solver used for the update of the image is run over 10 iterations which appears sufficient to reach practical stability. The solver is initialized with the degraded image.

5.2.1.4 Training specifications

We present results obtained by adopting the two training strategies described in Section 5.1.5. For the *greedy training*, we make use of the mean squared error on the estimated kernel, as a loss function for the *unfoldedVBA* layers, defined as $\ell(\widehat{\mathbf{x}}, \widehat{\mathbf{h}}, \widetilde{\mathbf{x}}, \widetilde{\mathbf{h}}) = \|\widetilde{\mathbf{h}} - \widehat{\mathbf{h}}\|^2$. The SSIM loss ([Wang *et al.* 2004]), between the output image $\widehat{\mathbf{x}}$ and the ground truth $\widetilde{\mathbf{x}}$ is used to train the post-processing layer $\mathcal{L}_{\text{pp}}$. For the *end-to-end training*, we use again SSIM between $\widehat{\mathbf{x}}$ and $\widetilde{\mathbf{x}}$. We use warm initialization for *end-to-end training*, that is we initialize with the weights learnt during the *greedy training* phase, associated with a weight decay procedure. ADAM optimizer, with mini-batch size equal to 10, is used for all the training procedures. Its parameters such as learning rate (lr), weight decay (wd) and epochs number are finetuned, so as to obtain stable performance on each validation set. The number of layers K (i.e., number of VBA iterations) is set during the *greedy training*, and kept the same for the *end-to-end training*. In practice, we increase K as long as a significant decrease in the averaged MSE over the training set was observed. We specify in Table 5.1 all the retained settings. The train/validation/test phase are conducted with a code implemented in Pytorch (version 1.7.0) under Python (version 3.6.10) environment, and run on an Nvidia DGX wokstation using one Tesla V100 SXM2 GPU (1290 MHz frequency, 32GB of RAM). Our code is made available at `https://github.com/yunshihuang/unfoldedVBA`.

5.2.1.5 Comparison to other methods

The proposed method is compared to several blind deconvolution approaches available in the literature:

Optimization-based methods: We first evaluate the VBA described in Section 4.3, in the favorable situation where the noise level σ is assumed to be known, and parameter ξ is finetuned empirically (see more details hereafter). VBA is run until reaching practical convergence, i.e., when the relative squared distance between two consecutive image iterates gets lower than 10^{-5}. We also test two optimization-based approaches for blind deconvolution. The first one is called *deconv2D*. It makes use of the proximal alternating algorithm from [Bolte *et al.* 2010], to minimize a least-squares data fidelity term combined with various priors, namely total variation and positivity constraint on the image, sum-to-one and

	Dataset 1	Dataset 2	Dataset 3
Greedy training	*UnfoldedVBA*		
	$K = 6$, epoch $= 10$ lr $= 5 \times 10^{-3}$	$K = 21$, epoch $= 10$ lr $= 5 \times 10^{-3}$ (for $\mathcal{L}_0$) lr $= 10^{-3}$ (for other layers)	$K = 10$, epoch $= 10$ lr $= 10^{-4}$ (for $\mathcal{L}_0$) lr $= 10^{-3}$ (for other layers)
	Post-processing $\mathcal{L}_{pp}$		
	U-net [Ronneberger *et al.* 2015] epoch $= 30$, lr $= 10^{-3}$	Residual network [Bertocchi *et al.* 2019, Fig.4] epoch $= 200$, lr $= 10^{-3}$	U-net [Ronneberger *et al.* 2015] epoch $= 200$, lr $= 10^{-3}$
End-to-end training	$K = 6$ epoch $= 6$ lr $= 5 \times 10^{-5}$ wd $= 10^{-4}$	$K = 21$ epoch $= 6$ lr $= 5 \times 10^{-5}$ wd $= 10^{-4}$	$K = 10$ epoch $= 6$ lr $= 5 \times 10^{-5}$ wd $= 10^{-4}$

Table 5.1: Settings for the training phases in our experiments

quadratic constraint on the kernel. This method is implemented in Matlab, and inherits some of the software accelerations discussed in [Abboud *et al.* 2019] for blind video deconvolution. The second competitor in this category is the *blinddeconv* approach [3] from [Krishnan *et al.* 2011]. For the sake of fair comparisons, for both datasets, we finetune the hyperparameters of these three methods on 40% of the training set and apply an average of the found values on the test set. Moreover, following the use of these three methods, we perform a non-blind deconvolution step BM3D-DEB [4] [Lebrun 2012], which uses their respective estimated blur kernel to restore the image.

Deep learning methods: We perform comparisons with three recent deep learning architectures for blind deconvolution. SelfDeblur [5] [Ren *et al.* 2020] is an unsupervised approach able to jointly perform the image restoration and kernel estimation tasks. DBSRCNN [6] [Albluwi *et al.* 2018] and DeblurGAN [7] [Kupyn *et al.* 2018] are two supervised deep learning techniques. In contrast with SelfDeblur, they both only provide the estimated image, but do not estimate the kernel. Both these methods have been retrained on each of our datasets, using the same settings as in their initial implementation. Moreover, we adapted DBSRCNN to color images using the same pipeline as for our method, that is applying DBSRCNN on the luminance channel while simply nonlinearly filtering the chrominance ones. Finally, the kernels provided as outputs of the SelfDeblur method are normalized as a post-processing step to satisfy the sum-to-one constraint, to make the comparisons more faithful.

[3] Matlab code: https://dilipkay.wordpress.com/blind-deconvolution/

[4] Matlab code: https://webpages.tuni.fi/foi/GCF-BM3D/index.html#ref_software

[5] Python/Pytorch code: https://github.com/csdwren/SelfDeblur

[6] Python/Pytorch code: https://github.com/Fatma-ALbluwi/DBSRCNN

[7] Python/Pytorch (training) and Matlab C-mex (testing) codes: https://github.com/KupynOrest/DeblurGAN

5.2.1.6 Evaluation metrics

All the methods are evaluated in terms of their performance on the blur kernel estimation (when available) and on the image restoration. Different metrics are used. For the blur kernels, we evaluate (*i*) the MSE, (*ii*) the so-called $\mathcal{H}_\infty$ error defined as the ℓ_∞ norm of the difference between the 2D discrete Fourier coefficients (with suitable padding) of the estimated and of the true kernel, and (*iii*) the mean absolute error (MAE) defined as the ℓ_1 norm of the difference between $\widetilde{\mathbf{h}}$ and $\widehat{\mathbf{h}}$. For evaluating the image quality, we compute (*i*) the SSIM, (*ii*) the peak-signal-to-noise ratio (PSNR), and (*iii*) the PieAPP value [Prashnani *et al.* 2018], between the estimated image $\widehat{\mathbf{x}}$ and the ground truth $\widetilde{\mathbf{x}}$.

5.2.2 Experimental results

5.2.2.1 Dataset 1

In Tables 5.2 and 5.3, we report the results of kernel estimation and image restoration, computed on the test set, using the different methods. As could be expected, the greedy approach tends to give more weight to the kernel quality than the end-to-end training. Our two training approaches yield great performance, when compared to all the other tested approaches. One can notice that the VBA with finetuned value for ξ performs quite well, showing the validity of our Bayesian formulation. The proposed unrolled VBA technique allows us to avoid a manual tuning of this parameter, and further increases the resulting quality. This is a direct outcome of the supervised training procedure aiming at maximizing quality scores, and also to the introduction of a post-processing step on the images. DBSRCNN has a good performance in terms of image restoration in this dataset. However, it is not capable of estimating the blur kernel, which might be useful for various applications. We display two examples of results in Fig 5.7, extracted from our test set. One can notice, by visual inspection of these results, the high quality of the restored images. No artifacts can be observed, which is confirmed by a low average value of the PieAPP index on the test set. Moreover, the kernels are generally estimated quite accurately, as shown by the low MSE score and the good retrieval of their general structure. In the few cases when the unfolded VBA algorithm fails to give a perfect recovery of the blur kernel as in Fig. 5.7(bottom), the estimation is still accurate enough to yield a good recovery of the image whatever *greedy training* or *end-to-end training* is used. One can also notice that our method tends to provide better contrasted images, compared to its closest competitor in the image restoration task that is DBSRCNN. We display in Fig. 5.6(left) the evolution of the SSIM loss during the end-to-end training of the proposed architecture, showing the increase of the loss, then its stabilization, for both training and validation set, thus confirming an appropriate setting of ADAM optimizer parameters. Finally, Table 5.4(left) displays the average test time for each methods, that is the computational time required to restore one example of the dataset, once the method is finetuned/trained. We displayed CPU time for a fair comparison between methods, for codes ran on a Dell workstation equipped with an Xeon(R) W-2135 processor (3.7 GHz clock frequency and 12 GB of RAM). GPU time is also indicated when available. The fastest method is DBSRCNN, though we must emphasize that, in contrast with all the other methods based on Matlab/Python softwares,

Method	MSE	$\mathcal{H}_\infty$ error	MAE
VBA	0.0017 (0.0022)	0.1674 (0.1003)	0.0472 (0.0317)
deconv2D	0.0025 (0.0035)	0.1483 (0.1037)	0.0489 (0.0395)
blinddeconv	0.0013 (0.0011)	0.1553 (0.0660)	0.0417 (0.0203)
SelfDeblur	0.0143 (0.0181)	0.3253 (0.1105)	0.1350 (0.0663)
Proposed (greedy)	**0.0008** (0.0012)	**0.1165** (0.0677)	**0.0281** (0.0168)
Proposed (end-to-end)	0.0009 (0.0013)	0.1188 (0.0672)	0.0289 (0.0170)

Table 5.2: Quantitative assessment of the restored kernels. Mean (standard deviation) values computed over the test sets of *Dataset 1*.

Method	SSIM	PSNR	PieAPP
Blurred	0.6542 (0.1072)	22.2254 (2.3779)	4.1794 (0.9005)
VBA	0.7603 (0.0814)	23.7332 (2.5672)	1.5109 (0.6184)
deconv2D	0.7668 (0.0912)	24.5970 (2.8656)	1.9289 (0.4959)
blinddeconv	0.7528 (0.0963)	23.9347 (2.4299)	1.9170 (0.6630)
SelfDeblur	0.6948 (0.1006)	22.2704 (2.1255)	3.3178 (0.7291)
DBSRCNN	0.7780 (0.0895)	**24.9561** (2.9800)	1.5959 (0.6463)
DeblurGAN	0.6613 (0.0731)	22.4388 (2.4074)	1.8937 (0.7630)
Proposed (greedy)	0.7945 (0.0890)	**24.7093** (2.9351)	1.4047 (0.6437)
Proposed (end-to-end)	**0.7989** (0.0886)	**24.6638** (3.0711)	**1.1976** (0.5433)

Table 5.3: Quantitative assessment of the restored images. Mean (standard deviation) values computed over the test sets of *Dataset 1*.

DBSRCNN makes use of an optimized C implementation, for its test phase on CPU. DeblurGAN is also very fast, but the resulting quality was quite poor in our experiments. Our method reaches a reasonable computational time on CPU. It becomes quite competitive when making use of GPU implementation, as the unrolled architecture is well suited for that purpose. This allows to drop the test time per image to few seconds, making it advantageous, with the addition benefit of better quality results in average, and of an available kernel estimate.

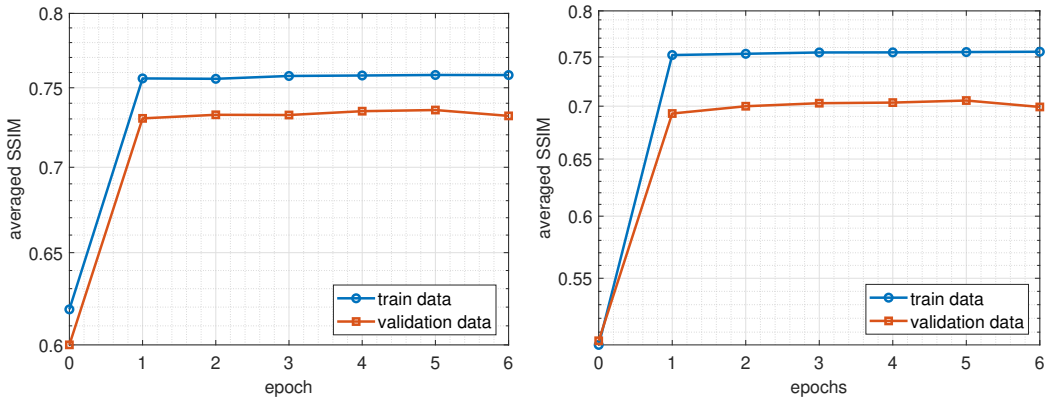


Figure 5.6: Evolution of SSIM loss along epochs of *end-to-end training* phase, averaged either on training or on validation sets of *Dataset 1* (left) and *Dataset 2* (right).

Method	Dataset 1	Dataset 2	Dataset 3
VBA	153s (15s)	156s (18s)	606s (62s)
deconv2D	16s	19s	333s
blinddeconv	19s	22s	188s
SelfDeblur	452s (51s)	455s (55s)	2232s (279s)
DBSRCNN	1s	2s	1s
DeblurGAN	2s (1s)	3s (2s)	2s (1s)
Proposed	36s (4s)	113s (12s)	286s (31s)

Table 5.4: Average test time per image, using CPU (resp. GPU).

5.2.2.2 Dataset 2

The results of kernel estimation and image restoration on Dataset 2 using the various methods are shown in Tables 5.5 and 5.6, respectively. This dataset is more challenging, as it includes color images, various blur shapes, and various noise levels. The latter are not assumed to be known anymore. Hereagain, we can observe that the *greedy training* yields the best performance in terms of kernel estimation for the three considered metrics. In contrast, *end-to-end training* tends to favor the restored image quality while still providing a good kernel quality compared to other methods. In this more complicated context, standard VBA does not perform very well, as setting ξ becomes tedious for such an heterogeneous dataset. Let us note that the noise level is assumed to be known for this particular method, putting it in a quite favorable situation, compared to the other competitors, including our proposed approach. DBSRCNN provides again a good image recovery, but our proposed approach still outperforms it for both SSIM and PieAPP metrics. We display two examples of restoration in Fig 5.8, when the sought blur is uniform, and out-of-focus, respectively. Such blur shapes are challenging and the MSE on the estimated blur might appear not excellent. Nevertheless, our method remains the best among the compared ones. The visual quality of the image generated by the proposed method is also very satisfying. We display in Fig. 5.6(right) the evolution of the SSIM loss during the *end-to-end training*, witnessing the absence of any overfitting issue. Moreover, we present in Fig 5.9 the evolution of the MSE loss on the kernel estimate, along the $K = 21$ layers of the architecture trained in an *end-to-end* manner. The MSE was averaged on test set examples associated to either Gaussian or out-of-focus blurs, respectively. These plots show that, for our choice of K (finetuned on the validation set), the MSE values are close to minimal. Larger K implied an increase of memory and training time, while not necessarily improving the results quality. One can also notice more fluctuations in the case of out-of-focus blur, which turns out to be more challenging to restore. A similar curve was obtained for uniform blurs, not shown by lack of space. Finally, Table 5.4(right) presents the average test time of the different methods. Again, our method appears competitive in terms of running time.

Method	MSE	$\mathcal{H}_\infty$ error	MAE
VBA	0.0148 (0.0139)	0.4492 (0.1638)	0.1339 (0.0627)
deconv2D	0.0099 (0.0160)	0.2796 (0.1692)	0.0869 (0.0576)
blinddeconv	0.0245 (0.0264)	0.3113 (0.1409)	0.1596 (0.1106)
SelfDeblur	0.4336 (0.3188)	0.8720 (0.2774)	0.6975 (0.3047)
Proposed (greedy)	**0.0037** (0.0079)	**0.1888** (0.1061)	**0.0570** (0.0414)
Proposed (end-to-end)	0.0039 (0.0079)	0.1960 (0.1056)	0.0588 (0.0411)

Table 5.5: Quantitative assessment of the restored kernels. Mean (standard deviation) values computed over the test sets of *Dataset 2*.

Method	SSIM	PSNR	PieAPP
Blurred	0.5427 (0.1150)	21.7994 (2.1679)	4.2378 (0.8539)
VBA	0.4024 (0.1571)	16.0371 (4.1798)	2.4218 (0.5545)
deconv2D	0.6880 (0.1065)	23.1940 (2.8986)	2.2245 (0.6721)
blinddeconv	0.6961 (0.1034)	23.2663 (2.7229)	2.3259 (0.8080)
SelfDeblur	0.5107 (0.1305)	19.9943 (2.1467)	5.9269 (1.4066)
DBSRCNN	0.6948 (0.1688)	**23.6041** (4.2073)	1.9474 (0.7171)
DeblurGAN	0.3370 (0.0740)	17.2781 (1.2909)	3.6581 (1.0040)
Proposed (greedy)	0.7454 (0.1015)	23.2169 (2.4442)	**1.7250** (0.5324)
Proposed (end-to-end)	**0.7518** (0.1025)	**23.5631** (2.5959)	**1.7681** (0.5502)

Table 5.6: Quantitative assessment of the restored images. Mean (standard deviation) values computed over the test sets of *Dataset 2*.

5.2.2.3 Dataset 3

The results of kernel estimation and image restoration on *Dataset 3* using the various methods are shown in Tables 5.7 and 5.8, respectively. The level of complexity of the blind deconvolution task again increases compared to the former experiments, as the blurs are of larger size and have less smooth kernel. The noise level also has to be estimated. Regarding the performance of kernel estimation, we can observe that the best performance are reached by the optimization-based method blinddeconv, followed closely by our method. SelfDeblur and deconv2D are slightly behind, in terms of quantitative metrics. Regarding the image restoration, the results are more contrasted. Hereagain, the proposed method with *end-to-end training* gives the best performance regarding the quality of the restored image, and at the same time keeps the quality of blur kernel estimation compared to *greedy training*. The standard VBA procedure reaches rather fair quality on this dataset. We must howewer point out that, as for *Dataset 2*, this method was run under the simplifying assumption that the noise level is known. Both deep learning methods DBSRCNN and DeblurGAN fail to recover a good image quality. We display two examples of restoration in Fig 5.10 where the sought kernels correspond to two different motion blurs from the dataset. We can notice that most of the methods struggle in the estimation of the blur kernel. Although SelfDeblur yields fairly good results according to a visual inspection, this method seems to suffer from a shift ambiguity, typical in blind deconvolution, which explains its rather limited quantitative scores compared to the other methods. As for the image quality, our proposed method is the most satisfying visually, which is aligned with

Method	MSE	$\mathcal{H}_\infty$ error	MAE
VBA	0.0376 (0.0206)	0.6817 (0.2969)	0.2518 (0.0898)
deconv2D	0.0592 (0.0162)	1.0446 (0.3027)	0.2753 (0.0457)
blinddeconv	**0.0303** (0.0085)	**0.6249** (0.2088)	**0.2119** (0.0583)
SelfDeblur	0.0500 (0.0260)	0.7362 (0.3044)	0.2522 (0.0926)
Proposed (greedy)	**0.0320** (0.0159)	0.6564 (0.2918)	0.2189 (0.0697)
Proposed (end-to-end)	**0.0321** (0.0160)	0.6583 (0.2930)	0.2197 (0.0701)

Table 5.7: Quantitative assessment of the restored kernels. Mean (standard deviation) values computed over the test sets of *Dataset 3*.

Method	SSIM	PSNR	PieAPP
Blurred	0.5830 (0.1500)	21.1404 (2.8626)	3.4647 (0.6608)
VBA	0.7344 (0.1242)	24.3958 (3.0088)	1.7028 (0.4914)
deconv2D	0.5440 (0.1422)	20.4229 (2.6241)	2.2811 (0.6407)
blinddeconv	0.4847 (0.1669)	16.0690 (2.7140)	0.6957 (0.5241)
SelfDeblur	0.7381 (0.1897)	24.3973 (4.6249)	0.6893 (0.3982)
DBSRCNN	0.5327 (0.1111)	19.7017 (2.2976)	2.3438 (0.5672)
DeblurGAN	0.5536 (0.1727)	21.0929 (3.2151)	2.3692 (0.7155)
Proposed (greedy)	0.9197 (0.0353)	29.5730 (1.9817)	0.1778 (0.2408)
Proposed (end-to-end)	**0.9295** (0.0342)	**31.0749** (2.0658)	**0.2227** (0.2309)

Table 5.8: Quantitative assessment of the restored images. Mean (standard deviation) values computed over the test sets of *Dataset 3*.

its best PieAPP score by far among all the methods. We display in Fig. 5.11(left) the evolution of the SSIM loss during the *end-to-end training*. The evolution of the MSE loss on the kernel estimate is displayed in Fig. 5.11(right), across the $K = 10$ layers of the architecture trained in an *end-to-end* manner. Slight fluctuations can be observed compared to the plots obtained for both synthetic blur datasets, which shows the high challenge of estimating such complex motion blur kernels. Table 5.4(right) displays the average test time for each methods. With the exception of DBSRCNN and DeblurGAN (but yielding rather low quality results), all the methods see their computational time increasing, compared to the other two datasets. This is directly related to the increase of the blur kernel size. This being said, the computational time for our proposed method remains competitive, especially when making use of GPU.

5.3 Conclusion

This chapter proposed a novel method for blind image deconvolution that combines the variational Bayesian algorithm from Chapter 4 with a neural network architecture. Our experiments illustrated the excellent performance of this method on three datasets, comprising grayscale and color images, and degraded with various synthetic and real-world kernel types. Compared to state-of-the-art variational and deep learning approaches, our method delivered a more accurate estimation of both the image and the blur kernels. It also included an automatic noise estimation step, so that it required little hyperparameter tuning, com-

pared to its counterpart from Chapter 4. The proposed method is very competitive in terms of computational time during the test phase, while showing similar train time to its deep learning competitors. The main core of the proposed architecture is highly interpretable, as it implements unrolled iterates of a well sounded Bayesian-based blind deconvolution method. As a byproduct, it also outputs estimates for the covariance matrices of both sought quantities (image/kernel). This information could be of interest for uncertainty quantification and model selection tasks (see for instance [Huang *et al.* 2021a, Repetti *et al.* 2019]). More generally, our work demonstrates that unrolling VBA algorithms constitute a promising research direction for solving challenging problems arising in Data Science.

Figure 5.7: Ground-truth image/blur, degraded image, restored images (with PieAPP index) and estimated blurs (with MSE score) when available, for various methods, on two examples in the test set of *Dataset 1*.

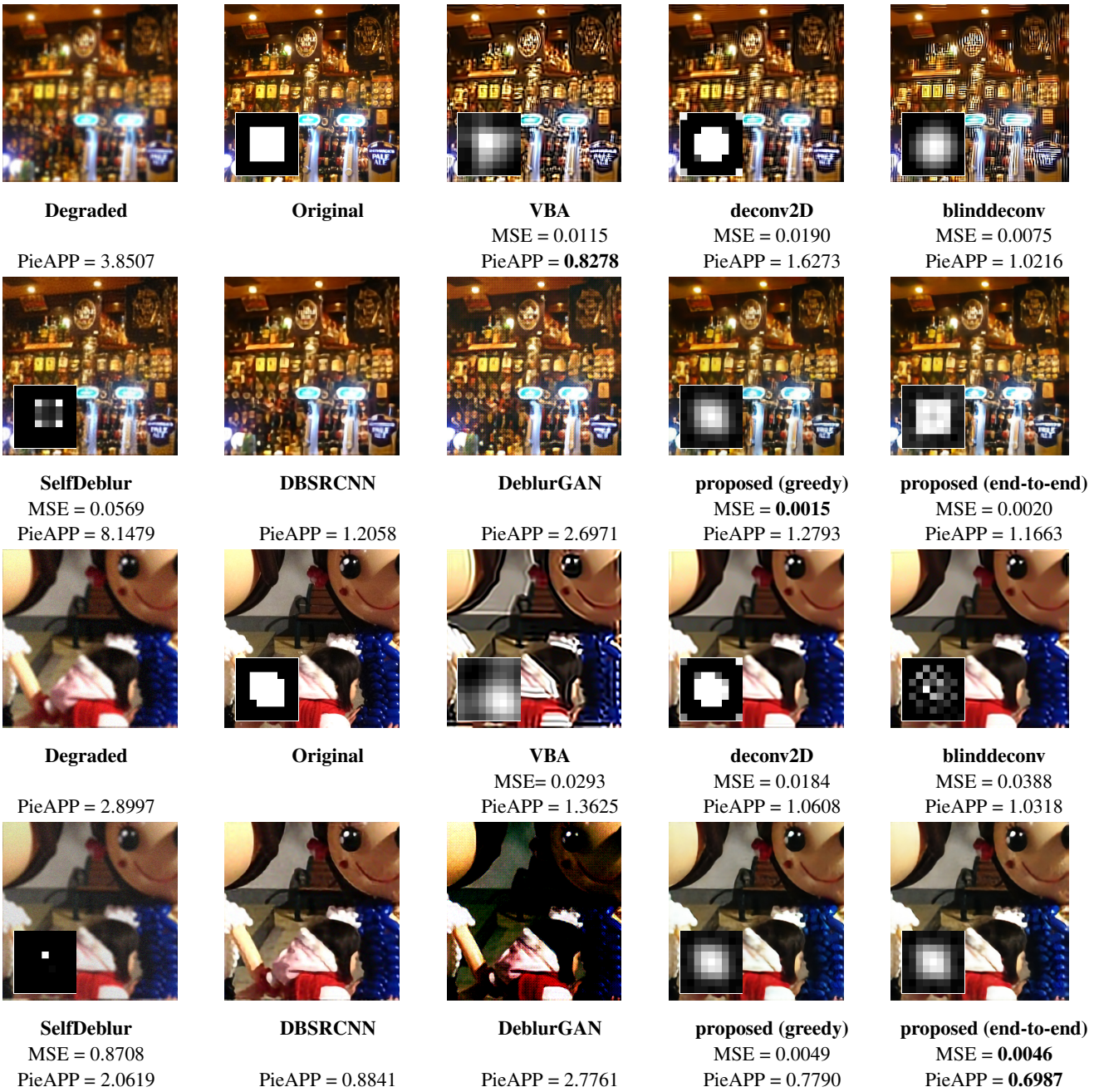


Figure 5.8: Ground-truth image/blur, degraded image, restored images (with PieAPP index) and estimated blurs (with MSE score) when available, for various methods, on two examples in the test set of *Dataset 2*.

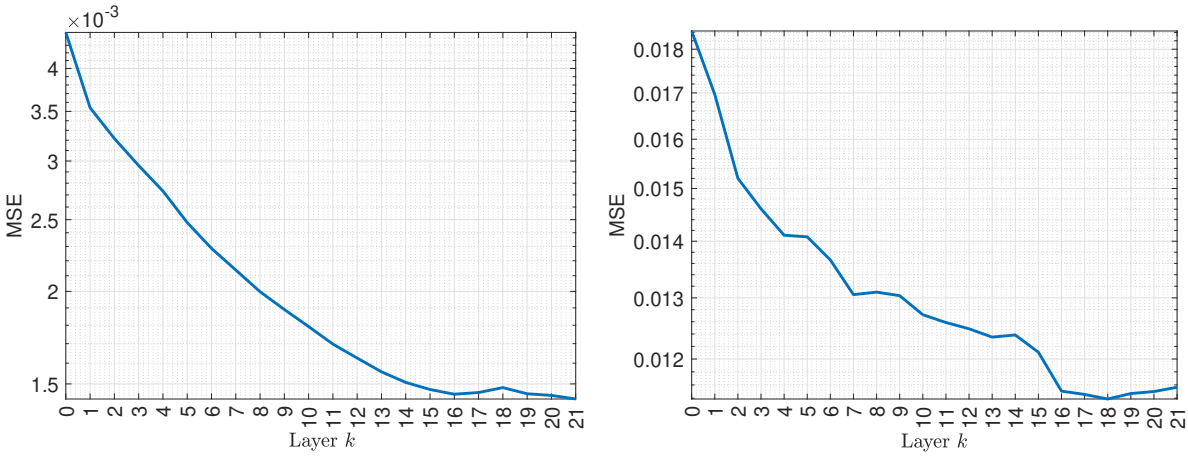


Figure 5.9: MSE loss along the layers for proposed method using *end-to-end training*. Average over *Dataset 2* test examples involving either Gaussian (left), or out-of-focus (right) blur shapes.

Figure 5.10: Ground-truth image/blur, degraded image, restored images (with PieAPP index) and estimated blurs (with MSE score) when available, for various methods, on two examples in the test set of *Dataset 3*.

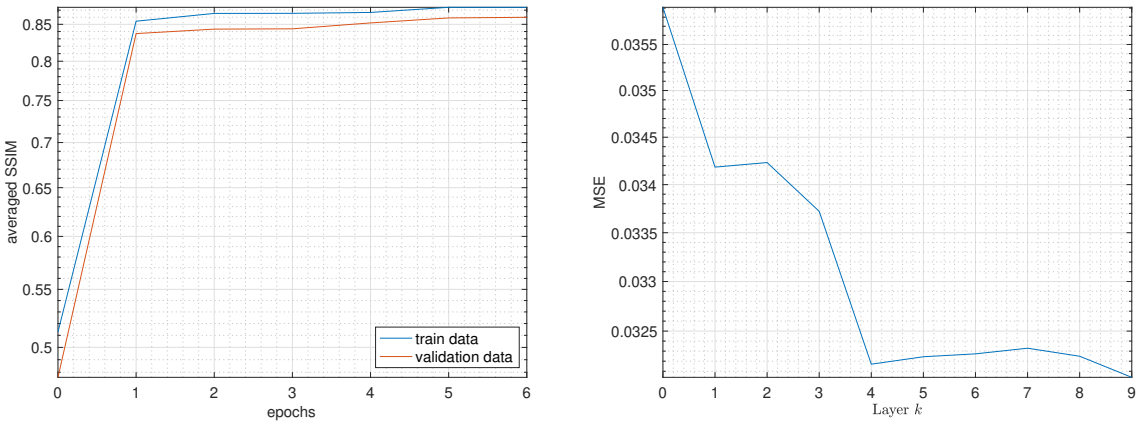


Figure 5.11: (left) Evolution of SSIM loss along epochs of *end-to-end training* phase, averaged either on training or on validation sets of *Dataset 3*. (right) MSE loss along the layers for proposed method using *end-to-end training*. Average over *Dataset 3* test examples involving motion blur shapes.

CHAPTER 6

Conclusion

Contents

Image deblurring is a challenging problem common to several areas from astronomy, biology, medical science to embedded camera systems. It becomes even more severe when the blur is unknown and/or spatially-variant. Most state-of-the-art deblurring methods, including traditional optimization-based methods and more recent deep learning strategies, rely on deterministic models and tools, providing point-wise image estimates. In contrast, in this thesis, we focus on image restoration methods relying on the Bayesian framework. The image degradation model, the priors and the resolution tools become probabilistic. The estimates are defined from the estimation of the moments (typically, the mean) of a posterior distribution given the observed data and prior. Higher order moments (typically, the covariance) can also be exhibited, allowing to quantify uncertainty about the estimates. Bayesian approaches are thus very appealing. However, they typically involve the evaluation of intractable integrals. This can be overcome through the use of sampling or approximation strategies. Among available Bayesian techniques, few can efficiently tackle large scale image deblurring problems involving complex (e.g., spatially-variant) blurs and non-Gaussian noise. Moreover, hyperparameter tuning can be cumbersome. This PhD thesis has brought novel contributions to these challenges by proposing advances along two classes of Bayesian methods, namely particle filtering and variational Bayesian algorithms.

6.1 Contributions

6.1.1 Chapter 3: A new BPF approach for spatially-variant PSFs identification

We addressed the problem of spatially-variant PSFs identification from calibrated images. We formulated a state-space model to describe the space-varying PSF map, accounting for

the spatial dependency among PSFs of the neighboring regions. We then proposed the algorithm GIAnPF, relying on the BPF paradigm, to solve the blur identification task in a probabilistic manner. Our algorithm implements an annealing mechanism to overcome the limitations of traditional BPF. The validity of the proposed method has been validated on numerous examples involving various shapes of blur kernels and PSF maps. This work let to the publications [Huang *et al.* 2019, Huang *et al.* 2021a].

6.1.2 Chapter 4: Novel VBA methods for blind image deconvolution

In [Marnissi *et al.* 2017], an efficient VBA strategy was proposed to tackle the case of non-blind image deconvolution. In this chapter, we proposed an extension of this methodology to solve the problem of blind image deconvolution. We introduced a generic linear equality constraints (e.g., sum-to-one) and a SAR prior on the unknown kernel. We then showed how to build suitable majorizing approximations of the original complicated posterior distribution, so as to yield a VBA method with simple updates while preserving its versatility. We provided two algorithms, respectively for addressing the case of Gaussian and non-Gaussian noise.

6.1.3 Chapter 5: Unrolled VBA for blind image deconvolution

Preliminary tests on the algorithms proposed in Chapter 4 showed high sensitivity to some hyperparameters. In particular, the parameter related to the kernel prior distribution is cumbersome to set. Moreover, the algorithms both require the knowledge of the noise level at hand. Finally, although reasonably good restoration results have been achieved, the methods were still far from the quality of some recent deep-learning methods. To address this challenge, we built an unrolling strategy where the iterations of VBA are recast as layers of neural network (NN) structures. The unrolling paradigm allowed to propose a supervised training approach where the NN parameters are learned so as to maximize image and kernel quality metrics. On top of solving the hyperparameter tuning problem, unrolling also led to fast algorithms amenable to implementations on GPU tools. Two unfolded VBA architectures have been proposed, able to process grayscale or color images, as well as to estimate the noise level. Our experiments on three datasets showed the superiority of our approach compared to several approaches from the state-of-the-art.

6.2 Perspectives

Our work raises several short and long term research paths that we list in this section.

6.2.1 Chapter 3

In this chapter, a BPF method has been proposed for spatially-variant blur identification. Several perspectives are foreseen, that we list herebelow, ranked from shorter term to longer term and open problems.

- Although our method is rather versatile to the noise and evolutionary laws assumed in our state-space model, we only tested it in scenarios involving Gaussian (or truncated Gaussian) laws. A first short-term perspective is thus to perform a more extensive numerical analysis, to cope with more sophisticated laws.

- Real-data experiments would certainly enforce the impact of this work. We did preliminary attempts, but we faced the issue of a mismatch between the piecewise constant spatially-variant blur degradation model considered in our approach, and the blur actually occurring in real-world data. Adapting our method to more realistic spatially-variant models (see [Denis *et al.* 2015, Escande & Weiss 2015]) would be an interesting path.

- Among realistic scenarios, we have considered the extension and application of our approach to the problem of blur calibration in radio interferometry [Birdi *et al.* 2020]. Discussions on this topic have been initiated this year with Dr. A. Repetti from Heriot-Watt University, Edinburgh. The main challenges are (i) the very large scale of the data, (ii) the complex-valued acquisitions in the Fourier domain, and (iii) the construction of a suitable SSM mimicking the neighboring interactions between the blur parameters.

- The robustness experiment considered in Section 3.4.4.2 is actually reminiscent from the image fusion problem [Tico & Pulli 2009]. In this inverse problem, one observes a pair of images, one being degraded with high noise level but no blur, the other being blurry and noisy with low level noise. The goal is to provide an estimation of the original image, jointly with the blur identification. Such problem arises typically in satellite imaging and in digital camera sensing, when pairs of short/long exposure images of the same scene can be acquired. Mapping such problem within our framework would require to include in our observation model the noise degradation of the image. Then, our idea would be to solve the resulting problem in an alternating fashion, using our GIAnPF for the blur identification step, while an optimization-based (e.g., total least squares) strategy would be employed to estimate, in a point-wise fashion, the original image.

- The main weakness of the model considered in Chapter 3 is that it is unidimensional. An important direction for future work is thus its extension to a multidimensional SSM to better encompass the variations among the PSFs. The increase of the dimensionality in the evolution equation of SSM is a challenging topic, as it requires to define suitable Markovian structures to preserve somewhat the causality relationships and thus to keep a sequential modeling/inference. A promising idea that we did not pursue by lack of time is to rely on 2D hidden Markov models as the ones described in [Champagnat *et al.* 1998]. The inference could be performed using tools described in [Lindsten & Schön 2013].

6.2.2 Chapters 4 and 5

- A first short-term goal is to extend the experiments from Chapter 5 to an example involving non-Gaussian noise, as it was done in the work by [Marnissi *et al.* 2017]. The VBA method in that case has been described in Chapter 4, but our numerical experiments on that algorithm stayed incomplete and were thus not reported in the manuscript. The extension of the unrolled architecture to the non-Gaussian case would also be an interesting avenue for short-term research.

- Another natural extension of these chapters is to tackle a real-world application. To that end, we are starting a collaboration with Dr. Nora Ouzir (CentraleSupélec), for extending and applying our method to the blind deconvolution of ultrasound imaging. This requires several changes in the noise and prior models, leading to a new structure for the unfolded VBA approach. We obtained promising preliminary results, and are confident that we will be able to finalize a publication on that topic by the end of the year.

- Our contributions in Chapter 4 would gain impact if they were assessed by theoretical convergence analysis results. Specifically, we would be interested in analyzing the convergence properties of VBA iterations, using recent results from the field of fixed point analysis [Combettes & Pesquet 2021]. To conduct such study, we would go back to simpler settings, such as non-blind image deblurring under Gaussian noise. Despite the high interest of this topic, the convergence properties of VBA schemes have been scarcely studied in the literature, and we believe that the analysis tools might be now mature enough to start this ambitious topic.

- Deep unrolled architectures are promoted as being easier to study theoretically. In particular, stability to noise in the NN inputs/parameters can be studied [Combettes & Pesquet 2020, Bertocchi *et al.* 2019]. Then, it would be very interesting to perform such analysis to the proposed architectures from Chapter 5.

Bibliography

[Abboud *et al.* 2019] F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot and L. Laborelli. *An Alternating Proximal Approach for Blind Video Deconvolution*. Signal Processing: Image Communication, vol. 70, pages 21–36, 2019. (Cited on pages 52 and 78.)

[Alam *et al.* 2019] M. Z. Alam, Q. Qian and B. K. Gunturk. *Space-variant blur kernel estimation and image deblurring through kernel clustering*. Signal Processing: Image Communication, vol. 76, pages 41–55, 2019. (Cited on page 22.)

[Albluwi *et al.* 2018] F. Albluwi, V. A. Krylov and R. Dahyot. *Image Deblurring and Super-Resolution Using Deep Convolutional Neural Networks*. In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2018), pages 1–6, Aalborg, Denmark, 17-20 Sept. 2018. (Cited on pages 69 and 78.)

[Alqadi 2018] Z. Alqadi. *Salt and Pepper Noise: Effects and Removal*. International Journal on Electrical Engineering and Informatics, vol. 2, Jul. 2018. (Cited on page 9.)

[Andrieu *et al.* 2010] C. Andrieu, A. Doucet and R. Holenstein. *Particle Markov Chain Monte Carlo methods*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 72, no. 3, pages 269–342, 2010. (Cited on page 15.)

[Anthony & Granick 2009] S. M. Anthony and S. Granick. *Image Analysis with Rapid and Accurate Two-Dimensional Gaussian Fitting*. Langmuir, vol. 25, no. 14, pages 8152–8160, 2009. (Cited on page 20.)

[Azzalini & Valle 1996] A. Azzalini and A. Valle. *The multivariate skew-normal distribution*. Biometrika, vol. 83, no. 4, pages 715–726, 1996. (Cited on page 25.)

[Babacan *et al.* 2009] S. D. Babacan, R. Molina and A. K. Katsaggelos. *Variational Bayesian Blind Deconvolution Using a Total Variation Prior*. IEEE Transactions on Image Processing, vol. 18, no. 1, pages 12–26, 2009. (Cited on pages 10, 12, 52 and 54.)

[Bailer-Jones 2017] C. Bailer-Jones. Practical Bayesian Inference: A Primer for Physical Scientists. Cambridge University Press, 2017. (Cited on page 7.)

[Bar *et al.* 2007] L. Bar, N. Sochen and N. Kiryati. *Restoration of Images with Piecewise Space-Variant Blur*. In Fiorella Sgallari, Almerico Murli and Nikos Paragios, editors, Scale Space and Variational Methods in Computer Vision, pages 533–544, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. (Cited on page 21.)

[Beck & Teboulle 2009] A. Beck and M. Teboulle. *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*. SIAM Journal on Imaging Sciences, vol. 2, pages 183–202, 2009. (Cited on page 34.)

[Bell *et al.* 2016] T. Bell, J. Xu and S. Zhang. *Method for out-of-focus camera calibration*. Applied Optics, vol. 55, no. 9, pages 2346–2352, 2016. (Cited on pages 20 and 51.)

[Berisha & Nagy 2014] S. Berisha and J. G. Nagy. *Iterative Methods for Image Restoration*. Academic Press Library in Signal Processing, vol. 4, pages 193–247, 2014. (Cited on pages 19 and 30.)

[Bernardo & Smith 1994] J. M. Bernardo and A. F. M. Smith. Bayesian Theory. John Wiley & Sons, 1994. (Cited on page 7.)

[Bertero *et al.* 2009] M. Bertero, P. Boccacci, G. Desidera and G. Vicidomini. *Image deblurring with Poisson data: From cells to galaxies*. Inverse Problems, vol. 25, page 123006, Nov. 2009. (Cited on page 9.)

[Bertocchi *et al.* 2019] 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, vol. 36, Sept. 2019. (Cited on pages 67, 70, 72, 77 and 92.)

[Besson *et al.* 2017] A. Besson, D. Perdios, F. Martinez, M. Arditi, Y. Wiauxy and J.-P. Thiran. *USSR: An ultrasound sparse regularization framework*. In Proceedings of the IEEE International Ultrasonics Symposium (IUS 2017), pages 1–4, Washington, D.C., USA, 6-9 Sept. 2017. (Cited on page 12.)

[Bietti *et al.* 2018] A. Bietti, G. Mialon and J. Mairal. *On regularization and robustness of deep neural networks*. arXiv:1810.00363, 2018. (Cited on page 70.)

[Birdi *et al.* 2020] J. Birdi, A. Repetti and Y. Wiaux. *Polca SARA - Full polarization, direction-dependent calibration and sparse imaging for radio interferometry*. Monthly Notices of the Royal Astronomical Society (MNRAS), vol. 492, no. 3, pages 3509–3528, Mar. 2020. (Cited on page 91.)

[Bishop *et al.* 2008] T. E. Bishop, R. Molina and J. R. Hopgood. *Blind restoration of blurred photographs via AR modelling and MCMC*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2008), pages 669–672, San Diego, CA, USA, 12-15 Oct. 2008. (Cited on page 52.)

[Blei *et al.* 2017] D. M. Blei, A. Kucukelbir and J. D. McAuliffe. *Variational Inference: A Review for Statisticians*. Journal of the American Statistical Association, vol. 112, no. 518, pages 859–877, 2017. (Cited on pages 13 and 52.)

[Bolte *et al.* 2010] J. Bolte, P. L. Combettes and J.-C. Pesquet. *Alternating proximal algorithm for blind image recovery*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2010), pages 1673–1676, Honk Kong, China, 26-29 Sept. 2010. (Cited on pages 52 and 78.)

[Boyat & Joshi 2013] A. Boyat and B. K. Joshi. *Image denoising using wavelet transform and median filtering*. In Proceedings of the Nirma University International Conference on Engineering (NUiCONE 2013), pages 1–6, Ahmedabad, India, 28-30 Nov. 2013. (Cited on page 9.)

[Branchini & Elvira 2021] N. Branchini and V. Elvira. *Optimized auxiliary particle filters: adapting mixture proposals via convex optimization*. In Uncertainty in Artificial Intelligence, pages 1289–1299. PMLR, 2021. (Cited on page 17.)

[Bugallo *et al.* 2017] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Míguez and P. M. Djurić. *Adaptive Importance Sampling: The past, the present, and the future*. IEEE Signal Processing Magazine, vol. 34, no. 4, pages 60–79, 2017. (Cited on page 14.)

[Cadoni *et al.* 2016] S. Cadoni, E. Chouzenoux, J.-C. Pesquet and C. Chaux. *A Block Parallel Majorize-Minimize Memory Gradient Algorithm*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2016), pages 3194–3198, Phoenix, Arizona, 25-28 Sept. 2016. (Cited on page 38.)

[Cappé *et al.* 2008] O. Cappé, R. Douc, A. Guillin, J.-M. Marin and C. P. Robert. *Adaptive importance sampling in general mixture classes*. Statistics and Computing, vol. 18, no. 4, pages 447–459, 2008. (Cited on page 14.)

[Cappé *et al.* 2004] O. Cappé, A. Guillin, J.-M. Marin and R. Christian. *Population Monte Carlo*. Journal of Computational and Graphical Statistics, vol. 13, no. 4, pages 907–929, 2004. (Cited on page 14.)

[Carasso 2001] A. S. Carasso. *Direct blind deconvolution*. SIAM Journal on Applied Mathematics, vol. 61, no. 6, pages 1980–2007, 2001. (Cited on pages 10, 19 and 51.)

[Chakrabarti *et al.* 2010] A. Chakrabarti, T. Zickler and W. T. Freeman. *Analyzing Spatially-varying Blur*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2010), pages 2512–2519, San Francisco, CA, USA, 13-18 Jun. 2010. (Cited on pages 20 and 22.)

[Chalvidal & Chouzenoux 2020] M. Chalvidal and E. Chouzenoux. *Block Distributed 3MG Algorithm and its Application to 3D Image Restoration*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2020), Abu Dhabi, United Arab Emirates, 25-28 Oct. 2020. (Cited on page 38.)

[Champagnat *et al.* 1998] F. Champagnat, J. Idier and Y. Goussard. *Stationary Markov random fields on a finite rectangular lattice*. IEEE Transactions on Information Theory, vol. 44, no. 7, pages 2901–2916, 1998. (Cited on page 91.)

[Chan & Nguyen 2011] S. H. Chan and T. Q. Nguyen. *Single image spatially variant out-of-focus blur removal*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2011), pages 677–680, Brussels, Belgium, 11-14 Sept. 2011. (Cited on page 9.)

[Chan & Wong 1998] T. Chan and C.-K. Wong. *Total variation blind deconvolution*. IEEE transactions on image processing, vol. 7, no. 3, pages 370–375, 1998. (Cited on page 52.)

[Charlier *et al.* 2021] B. Charlier, J. Feydy, J. Glaunès, F.-D. Collin and G. Durif. *Kernel Operations on the GPU, with Autodiff, without Memory Overflows*. Journal of Machine Learning Research, vol. 22, no. 74, pages 1–6, 2021. (Cited on page 73.)

[Chen *et al.* 2005] Y.-C. Chen, L. R. Furenlid, D. W. Wilson and H. H. Barrett. *Calibration of Scintillation Cameras and Pinhole SPECT Imaging Systems*. Small-Animal Spect Imaging, pages 195–201, 2005. (Cited on page 23.)

[Chimitt & Chan 2020] N. Chimitt and S. H. Chan. *Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients*. Optical Engineering, vol. 59, no. 8, page 083101, 2020. (Cited on pages 20 and 21.)

[Chouzenoux & Elvira 2020] E. Chouzenoux and V. Elvira. *GraphEM: EM algorithm for blind Kalman filtering under graphical sparsity constraints*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020), pages 5840–5844, Barcelona, Spain, 4-8 May 2020. (Cited on page 15.)

[Chouzenoux & Pesquet 2017] E. Chouzenoux and J.-C. Pesquet. *A Stochastic Majorize-Minimize Subspace Algorithm for Online Penalized Least Squares Estimation*. IEEE Transactions on Signal Processing, vol. 65, no. 18, pages 4770–4783, 2017. (Cited on pages 22 and 35.)

[Chouzenoux *et al.* 2013] E. Chouzenoux, A. Jezierska, J.-C. Pesquet and H. Talbot. *A Majorize-Minimize Subspace Approach for l2-l0 Image Regularization*. SIAM Journal on Imaging Science, vol. 6, pages 563–591, 2013. (Cited on page 38.)

[Chouzenoux *et al.* 2019] E. Chouzenoux, T. T.-K. Lau, C. Lefort and J.-C. Pesquet. *Optimal Multivariate Gaussian Fitting with Applications to PSF Modeling in Two-Photon Microscopy Imaging*. Journal of Mathematical Imaging and Vision, vol. 61, no. 7, pages 1037–1050, 2019. (Cited on pages 20, 22 and 51.)

[Chouzenoux *et al.* 2021] E. Chouzenoux, C. Della Valle and J.-C. Pesquet. *Inversion of Integral Models: a Neural Network Approach*. arXiv:2105.15044, 2021. (Cited on page 70.)

[Combettes & Pesquet 2020] P. L. Combettes and J.-C. Pesquet. *Deep neural network structures solving variational inequalities*. Set-Valued and Variational Analysis, vol. 28, pages 491–518, Sept. 2020. (Cited on pages 70 and 92.)

[Combettes & Pesquet 2021] P. L. Combettes and J.-C. Pesquet. *Fixed Point Strategies in Data Science*. IEEE Transactions on Signal Processing, vol. 69, pages 3878–3905, 2021. (Cited on page 92.)

[Cornuet *et al.* 2012] J.-M. Cornuet, J.-M. Marin, A. Mira and C. Robert. *Adaptive multiple importance sampling*. Scandinavian Journal of Statistics, vol. 39, no. 4, pages 798–812, 2012. (Cited on page 14.)

[Crockett & Fessler 2022] C. Crockett and J. A. Fessler. *Bilevel Methods for Image Reconstruction*. Foundations and Trends in Signal Processing, vol. 15, no. 2-3, pages 121–289, 2022. (Cited on page 70.)

[Del Moral *et al.* 2006] P. Del Moral, A. Doucet and A. Jasra. *Sequential monte carlo samplers*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 68, no. 3, pages 411–436, 2006. (Cited on pages 29 and 37.)

[Denis *et al.* 2015] L. Denis, E. Thiébaut, F. Soulez, J. M. Becker and R. Mourya. *Fast Approximations of Shift-Variant Blur*. International Journal of Computer Vision, vol. 115, pages 253–278, 2015. (Cited on pages 12, 21, 22, 23, 27, 38 and 91.)

[Deutscher *et al.* 2000] J. Deutscher, A. Blake and I. Reid. *Articulated body motion capture by annealed particle filtering*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2000), volume 2, pages 126–133, Hilton Head, SC, 13-15 Jun. 2000. (Cited on pages 17 and 30.)

[Djurić *et al.* 2003] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo and J. Míguez. *Particle Filtering*. IEEE Signal Processing Magazine, vol. 20, no. 5, pages 19–38, 2003. (Cited on pages 16, 20 and 28.)

[Djurić *et al.* 2007] P. M. Djurić, T. Lu and M. F. Bugallo. *Multiple particle filtering*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007), volume 3, pages III–1181–III–1184, Honolulu, HI, USA, 15-20 Apr. 2007. IEEE. (Cited on page 17.)

[Dong *et al.* 2016] 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, vol. 38, no. 2, pages 295–307, 2016. (Cited on page 69.)

[Doucet & Johansen 2009] A. Doucet and A. M. Johansen. *A tutorial on particle filtering and smoothing: Fifteen years later*. Handbook of Nonlinear Filtering, vol. 12, no. 656-704, page 3, 2009. (Cited on page 28.)

[Dupe *et al.* 2008] F.-X. Dupe, M. J. Fadili and J.-L. Starck. *Image deconvolution under poisson noise using sparse representations and proximal thresholding iteration*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), pages 761–764, Las Vegas, Nevada, USA, 30 Mar.- 4 Apr. 2008. (Cited on page 10.)

[El-Laham *et al.* 2019a] Y. El-Laham, V. Elvira and M. Bugallo. *Recursive shrinkage covariance learning in adaptive importance sampling*. In Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2019), pages 624–628, Le gosier, Guadeloupe, French West Indies, 15-18 Dec. 2019. IEEE. (Cited on page 14.)

[El-Laham *et al.* 2019b] Y. El-Laham, L. Martino, V. Elvira and M. F. Bugallo. *Efficient adaptive multiple importance sampling*. In Proceedings of the European Signal Processing Conference (EUSIPCO 2019), pages 1–5, A Coruña, Spain, 2-6 Sept. 2019. IEEE. (Cited on page 14.)

[Elvira & Chouzenoux 2022] V. Elvira and E. Chouzenoux. *Optimized Population Monte Carlo*. IEEE Transactions on Signal Processing (to appear in), 2022. (Cited on page 14.)

[Elvira & Martino 2021] V. Elvira and L. Martino. *Advances in Importance Sampling*. arXiv, pages 1–14, Feb. 2021. (Cited on page 13.)

[Elvira *et al.* 2015] V. Elvira, L. Martino, D. Luengo and M. F. Bugallo. *Efficient Multiple Importance Sampling Estimators*. IEEE Signal Processing Letters, vol. 22, no. 10, pages 1757–1761, 2015. (Cited on page 14.)

[Elvira *et al.* 2016a] V. Elvira, L. Martino, D. Luengo and M. F. Bugallo. *Heretical Multiple Importance Sampling*. IEEE Signal Processing Letters, vol. 23, no. 10, pages 1474–1478, 2016. (Cited on page 14.)

[Elvira *et al.* 2016b] V. Elvira, J. Míguez and P. M. Djurić. *Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment*. IEEE Transactions on Signal Processing, vol. 65, no. 7, pages 1781–1794, 2016. (Cited on page 17.)

[Elvira *et al.* 2017a] V. Elvira, L. Martino, D. Luengo and M. F. Bugallo. *Improving Population Monte Carlo: Alternative Weighting and Resampling Schemes*. Signal Processing, vol. 131, no. 12, pages 77–91, 2017. (Cited on page 14.)

[Elvira *et al.* 2017b] V. Elvira, L. Martino, D. Luengo and M. F. Bugallo. *Population Monte Carlo schemes with reduced path degeneracy*. In Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2017), pages 1–5, Curacao, 10-13 Dec. 2017. IEEE. (Cited on page 14.)

[Elvira *et al.* 2018a] V. Elvira, L. Martino, M. F Bugallo and P. M. Djurić. *In Search for Improved Auxiliary Particle Filters*. In Proceedings of the European Signal Processing Conference (EUSIPCO 2018), pages 1637–1641, Rome, Italy, 3-7 Sept. 2018. (Cited on pages 17 and 28.)

[Elvira *et al.* 2018b] V. Elvira, L. Martino and C. P. Robert. *Rethinking the effective sample size*. arXiv preprint arXiv:1809.04129, 2018. (Cited on page 35.)

[Elvira *et al.* 2019a] V. Elvira, L. Martino, M. F. Bugallo and P. M. Djurić. *Elucidating the Auxiliary Particle Filter via Multiple Importance Sampling [Lecture Notes]*. IEEE Signal Processing Magazine, vol. 36, no. 6, pages 145–152, 2019. (Cited on pages 17 and 28.)

[Elvira *et al.* 2019b] V. Elvira, L. Martino, D. Luengo, M. F. Bugallo*et al. Generalized multiple importance sampling*. Statistical Science, vol. 34, no. 1, pages 129–155, 2019. (Cited on pages 14 and 33.)

[Elvira *et al.* 2021] V. Elvira, J. Míguez and P. M. Djurić. *On the performance of particle filters with adaptive number of particles*. Statistics and Computing, vol. 31, no. 6, pages 1–18, 2021. (Cited on page 17.)

[Escande & Weiss 2015] P. Escande and P. Weiss. *Sparse Wavelet Representations of Spatially Varying Blurring Operators*. SIAM Journal on Imaging Sciences, vol. 8, pages 2976–3014, 2015. (Cited on pages 21, 22, 23, 30 and 91.)

[Evans & Swartz 1995] M. Evans and T. Swartz. *Methods for Approximating Integrals in Statistics with Special Emphasis on Bayesian Integration Problems*. Statistical Science, vol. 10, no. 3, pages 254–272, 1995. (Cited on page 52.)

[Fergus *et al.* 2006] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis and W. T. Freeman. *Removing Camera Shake from a Single Photograph*. ACM Transactions on Graphics, vol. 25, no. 3, pages 787–794, Jul. 2006. (Cited on page 9.)

[Filippone 2014] M. Filippone. *Bayesian Inference for Gaussian Process Classifiers with Annealing and Exact-Approximate MCMC*. In Proceedings of the International Conference on Pattern Recognition (ICPR 2014), Stockholm, Sweden, 24-28 Aug. 2014. (Cited on page 30.)

[Fox & Roberts 2012] C. Fox and S. J. Roberts. *A tutorial on variational Bayesian inference*. Artificial Intelligence Review, vol. 38, pages 85–95, 2012. (Cited on page 52.)

[Galinier *et al.* 2020] M. Galinier, M. Prato, E. Chouzenoux and J.-C. Pesquet. *A hybrid interior point - deep learning approach for Poisson image deblurring*. In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2020), pages 1–6, Virtual, 21-24 Sept. 2020. (Cited on page 70.)

[Gall *et al.* 2005] J. Gall, J. Potthoff and C. Schnörr. *Generalised Annealed Particle Filter - Mathematical Framework, Algorithms and Applications*. PhD thesis, 2005. (Cited on pages 17 and 30.)

[Gamerman & Lopes 2006] D. Gamerman and H. Lopes. Markov chain monte carlo: Stochastic simulation for bayesian inference, second edition. May 2006. (Cited on page 12.)

[Ge *et al.* 2011] D. Ge, J. Idier and E. Le Carpentier. *Enhanced sampling schemes for MCMC based blind Bernoulli-Gaussian deconvolution*. Signal Processing, vol. 91, no. 4, pages 759–772, Apr. 2011. (Cited on page 52.)

[Gelman *et al.* 2004] A. Gelman, J. B. Carlin, H. S. Stern and D. R. Rubin. Bayesian Data Analysis. Second edn. Chapman & Hall, 2004. (Cited on page 7.)

[Gharsalli *et al.* 2012] L. Gharsalli, M.-D. Ali, A. Fraysse and T. Rodet. *Variational Bayesian approximation with scale mixture prior: a comparison between three algorithms*. In 32nd International Workshop on Bayesian Inference and Maximun Entropy Methods in Sciences and Engineerin, volume 1353, pages 130–138, 2012. (Cited on page 52.)

[Gilton *et al.* 2021] D. Gilton, G. Ongie and R. Willett. *Model Adaptation for Inverse Problems in Imaging*. IEEE Transactions on Computational Imaging, vol. 7, pages 661–674, 2021. (Cited on page 71.)

[Gordon *et al.* 1993] N. J. Gordon, D. J. Salmond and A. F. M. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian State Estimation*. IEE Proceedings F - Radar and Signal Processing, vol. 140, no. 2, pages 107–113, 1993. (Cited on pages 20, 27 and 30.)

[Guadalupe Sanchez *et al.* 2012] M. Guadalupe Sanchez, M. Guadalupe Sánchez, V. Vidal, G. Verdu, G. Verdú, P. Mayo and F. Rodenas. *Medical image restoration with different types of noise*. In Proceedings of International Engineering in Medicine and Biology Conference (EMBC 2012), pages 4382–4385, San Diego, CA, USA, 28 Aug. - 1 Sept. 2012. (Cited on page 9.)

[Hadj *et al.* 2014] S. Ben Hadj, L. Blanc-Féraud and G. Aubert. *Space Variant Blind Image Restoration*. SIAM Journal on Imaging Sciences, vol. 7, 2014. (Cited on pages 21 and 22.)

[Hansen & Jensen 2008] P. C. Hansen and T. K. Jensen. *Noise propagation in regularizing iterations for image deblurring*. Electronic Transactions on Numerical Analysis, vol. 31, pages 204–220, 2008. (Cited on page 25.)

[Hirsch *et al.* 2010] M. Hirsch, S. Sra, B. Scholkopf and S. Harmeling. *Efficient Filter Flow for Space-Variant Multiframe Blind Deconvolution*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2010), pages 607–614, San Francisco, CA, USA, 13-18 Jun. 2010. (Cited on pages 20, 21 and 22.)

[Holmes *et al.* 2006] T. J. Holmes, D. Biggs and A. Abu-Tarif. Blind deconvolution. Springer US, Boston, MA, 2006. (Cited on pages 9 and 51.)

[Hortholary *et al.* 2021] T. Hortholary, C. Carrion, E. Chouzenoux, J.-C. Pesquet and C. Lefort. *Multiplex-multiphoton microscopy and computational strategy for biomedical imaging*. Microscopy Research and Technique, 2021. (Cited on page 20.)

[Huang *et al.* 2019] Y. Huang, E. Chouzenoux and V. Elvira. *Particle Filtering for Online Space-Varying Blur Identification*. In Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2019), pages 544–548, Le Gosier, Guadeloupe, French West Indies, 15-18 Dec. 2019. (Cited on pages v, 4 and 90.)

[Huang *et al.* 2021a] Y. Huang, E. Chouzenoux and V. Elvira. *Probabilistic Modeling and Inference for Sequential Space-Varying Blur Identification*. IEEE Transactions on Computational Imaging, vol. 7, pages 531–546, 2021. (Cited on pages v, vi, 4, 10, 51, 83 and 90.)

[Huang *et al.* 2021b] Y. Huang, E. Chouzenoux and J.-C. Pesquet. *Unrolled Variational Bayesian Algorithm for Image Blind Deconvolution*. Technical report, 2021. https://arxiv.org/abs/2110.07202. (Cited on pages vii, viii and 4.)

[Iba 2001] Y. Iba. *Population Monte Carlo algorithms*. Transactions of the Japanese Society for Artificial Intelligence, vol. 16, no. 2, pages 279–286, 2001. (Cited on page 14.)

[Jayamanmadharao *et al.* 2010] M. Jayamanmadharao, S. Anuradha and K. Reddy. *Impulse Noise removal in Digital Images*. International Journal on Computer Science and Engineering, vol. 2, Oct. 2010. (Cited on page 9.)

[Jeffreys 1939] H. Jeffreys. Theory of Probability. Oxford University Press, 1939. (Cited on page 7.)

[Jezierska *et al.* 2018] A. Jezierska, H. Talbot and J.-C.Pesquet. *Spatially variant PSF modeling in confocal macroscopy*. In Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI 2018), pages 489–492, Washington, DC, USA, 4-7 Apr. 2018. (Cited on pages 20 and 22.)

[Joshi *et al.* 2008] N. Joshi, R. Szeliski and D. J. Kriegman. *PSF estimation using sharp edge prediction*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2008), pages 1–8, Anchorage, AK, USA, 23-28 Jun. 2008. (Cited on page 52.)

[Jospin *et al.* 2020] L. V. Jospin, W. L. Buntine, F. Boussaid, H. Laga and M. Bennamoun. *Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users*. ArXiv, vol. abs/2007.06823, 2020. (Cited on page 70.)

[Kail *et al.* 2012] G. Kail, J.-Y. Tourneret, F. Hlawatsch and N. Dobigeon. *Blind Deconvolution of Sparse Pulse Sequences Under a Minimum Distance Constraint: A Partially Collapsed Gibbs Sampler Method*. IEEE Transactions on Signal Processing, vol. 60, no. 6, pages 2727–2743, 2012. (Cited on page 52.)

[Kalman 1960] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*. Transaction of the ASME—Journal of Basic Engineering, pages 35–45, 1960. (Cited on page 27.)

[Kenig *et al.* 2010] T. Kenig, Z. Kam and A. Feuer. *Blind Image Deconvolution Using Machine Learning for Three-Dimensional Microscopy*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pages 2191–204, Dec. 2010. (Cited on page 19.)

[Kingma & Welling 2014] D. P. Kingma and M Welling. *Auto-Encoding Variational Bayes*. In Proceedings of the International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14-16 Apr. 2014. (Cited on page 70.)

[Kirshner *et al.* 2012a] H. Kirshner, F. Aguet, D. Sage and M. Unser. *3-D PSF fitting for fluorescence microscopy: implementation and localization application*. Journal of Microscopy, vol. 249, no. 1, pages 13–25, 2012. (Cited on pages 20 and 22.)

[Kirshner *et al.* 2012b] H. Kirshner, T. Pengo, N. Olivier, D. Sage, S. Manley and M. Unser. *A PSF-based approach to Biplane calibration in 3D super-resolution microscopy*. In Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI 2012), pages 1232–1235, Barcelona, Spain, 2-5 May 2012. (Cited on page 20.)

[Klaas *et al.* 2005] M. Klaas, N. de Freitas and A. Doucet. *Toward Practical N2 Monte Carlo: the Marginal Particle Filter*. In Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI 2005), pages 308–315, Arlington, Virginia, 26-29 Jul. 2005. AUAI Press. (Cited on page 17.)

[Komodakis & Paragios 2012] N. Komodakis and N. Paragios. *MRF-Based Blind Image Deconvolution*. In Proceedings of the Asian Conference on Computer Vision (ACCV 2012), pages 361–374, Daejeon, Korea, 5-9 Nov. 2012. (Cited on page 52.)

[Kong *et al.* 1994] A. Kong, J. S. Liu and W. H. Wong. *Sequential imputations and Bayesian missing data problems*. Journal of the American Statistical Association, vol. 9, pages 278–288, 1994. (Cited on page 35.)

[Kong 1992] A. Kong. *A note on importance sampling using standardized weights*. University of Chicago, Dept. of Statistics, Tech. Rep, vol. 348, 1992. (Cited on page 35.)

[Kotecha & Djurić 2003] J. H. Kotecha and P. M. Djurić. *Gaussian sum particle filtering*. IEEE Transactions on signal processing, vol. 51, no. 10, pages 2602–2612, 2003. (Cited on page 17.)

[Kotera *et al.* 2013] J. Kotera, F. Šroubek and P. Milanfar. *Blind Deconvolution Using Alternating Maximum a Posteriori Estimation with Heavy-Tailed Priors*. Computer Analysis of Images and Patterns, vol. 8048, pages 59–66, 2013. (Cited on pages 52 and 76.)

[Krishnan *et al.* 2011] D. Krishnan, T. Tay and R. Fergus. *Blind deconvolution using a normalized sparsity measure*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2011), pages 233–240, Colorado Springs, CO, USA, 20-25 Jun. 2011. (Cited on pages 52 and 78.)

[Kundur & Hatzinakos 1996] D. Kundur and D. Hatzinakos. *Blind image deconvolution*. IEEE Signal Processing Magazine, vol. 13, no. 3, pages 43–64, 1996. (Cited on page 10.)

[Kupyn *et al.* 2018] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin and J. Matas. *DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018), pages 8183–8192, Salt Lake City, UT, USA, 18-22 Jun. 2018. (Cited on pages 69 and 78.)

[Lebrun 2012] M. Lebrun. *An Analysis and Implementation of the BM3D Image Denoising Method*. Image Processing On Line, vol. 2, pages 175–213, Aug. 2012. (Cited on page 78.)

[Lefort *et al.* 2020] C. Lefort, E. Chouzenoux, L. Magnol, H. Massias and J.-C. Pesquet. *Multi-parametric 3D-point-spread function estimation in deep multiphoton microscopy with an original computational strategy dedicated to the reconstruction of muscle images*. In SPIE Photonics Europe, volume 11354 of *Proceedings of SPIE*, Online Only, United States, Apr. 2020. SPIE. (Cited on page 22.)

[Levin *et al.* 2009] A. Levin, Y. Weiss, F. Durand and W. T. Freeman. *Understanding and evaluating blind deconvolution algorithms*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2009), pages 1964–1971, Miami, FL, USA, 20-25 Jun. 2009. (Cited on pages 51, 74 and 75.)

[Levin *et al.* 2011] A. Levin, Y. Weiss, F. Durand and W. T. Freeman. *Understanding Blind Deconvolution Algorithms*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pages 2354–2367, 2011. (Cited on page 19.)

[Levin 2006] A. Levin. *Blind Motion Deblurring Using Image Statistics*. In Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS 2006), volume 19, pages 841–848, Vancouver, British Columbia, Canada, 4-7 Dec. 2006. (Cited on page 52.)

[Li *et al.* 2007] D. Li, R. M. Mersereau and S. Simske. *Atmospheric Turbulence-Degraded Image Restoration Using Principal Components Analysis*. IEEE Geoscience and Remote Sensing Letters, vol. 4, no. 3, pages 340–344, 2007. (Cited on page 20.)

[Li *et al.* 2019] Y. Li, M. Tofighi, V. Monga and Y. C. Eldar. *An Algorithm Unrolling Approach to Deep Image Deblurring*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), pages 7675–7679, Brighton, UK, 12-17 May 2019. (Cited on page 71.)

[Li *et al.* 2020] Y. Li, M. Tofighi, J. Geng, V. Monga and Y. C. Eldar. *Efficient and Interpretable Deep Blind Image Deblurring Via Algorithm Unrolling*. IEEE Transactions on Computational Imaging, vol. 6, pages 666–681, 2020. (Cited on pages 70 and 71.)

[Li *et al.* 2021] Y. Li, O. Bar-Shira, V. Monga and Y. C. Eldar. *Deep Algorithm Unrolling for Biomedical Imaging*. Computing Research Repository, vol. abs/2108.06637, 2021. (Cited on page 71.)

[Likas & Galatsanos 2004] A. C. Likas and N. P. Galatsanos. *A variational approach for Bayesian blind image deconvolution*. IEEE Transactions on Signal Processing, vol. 52, no. 8, pages 2222–2233, 2004. (Cited on pages 10 and 12.)

[Lindsten & Schön 2013] F. Lindsten and T. B. Schön. *Backward Simulation Methods for Monte Carlo Statistical Inference*. In Foundations and Trends in Machine Learning, volume 6, 2013. (Cited on page 91.)

[Liu *et al.* 2021a] J. Liu, M. Yan and T. Zeng. *Surface-Aware Blind Image Deblurring*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 3, pages 1041–1055, 2021. (Cited on page 52.)

[Liu *et al.* 2021b] Y.-Q. Liu, X. Du, H.-L. Shen and S.-J. Chen. *Estimating Generalized Gaussian Blur Kernels for Out-of-Focus Image Deblurring*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 3, pages 829–843, 2021. (Cited on pages 23 and 30.)

[Liu 2004] J. S. Liu. Monte Carlo strategies in scientific computing. Springer, 2004. (Cited on page 13.)

[Luengo *et al.* 2020] D. Luengo, L. Martino, M. Bugallo, V. Elvira and S. Särkkä. *A survey of Monte Carlo methods for parameter estimation*. EURASIP Journal on Advances in Signal Processing, vol. 2020, pages 1–62, 2020. (Cited on page 52.)

[Madapura & Li 2007] J. Madapura and B. Li. *3D Articulated Human Body Tracking using KLD-Annealed Rao-Blackwellised Particle Filter*. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2007), pages 1950–1953, Beijing, China, 2-5 Jul. 2007. (Cited on page 30.)

[Mannan & Langer 2016] F. Mannan and M. S. Langer. *Blur Calibration for Depth from Defocus*. In Proceedings of the Conference on Computer and Robot Vision (CRV 2016), pages 281–288, Victoria, BC, Canada, 1-3 Jun. 2016. (Cited on page 20.)

[Marnissi *et al.* 2017] Y. Marnissi, Y. Zheng, E. Chouzenoux and J.-C. Pesquet. *A Variational Bayesian Approach for Image Restoration—Application to Image Deblurring With Poisson–Gaussian Noise*. IEEE Transactions on Computational Imaging, vol. 3, no. 4, pages 722–737, 2017. (Cited on pages 10, 12, 52, 54, 55, 61, 62, 76, 90 and 91.)

[Marnissi *et al.* 2018] Y. Marnissi, E. Chouzenoux, A. Benazza-Benyahia and J.-C. Pesquet. *An Auxiliary Variable Method for Markov Chain Monte Carlo Algorithms in High Dimension*. Entropy, vol. 20, no. 2, page 110, 2018. (Cited on page 12.)

[Marnissi *et al.* 2020] Y. Marnissi, E. Chouzenoux, A. Benazza-Benyahia and J.-C. Pesquet. *Majorize-Minimize Adapted Metropolis-Hastings Algorithm*. IEEE Transactions on Signal Processing, vol. 68, pages 2356–2369, Mar. 2020. (Cited on pages 12 and 52.)

[Martino *et al.* 2017a] L. Martino, V. Elvira and D. Luengo. *Anti-tempered layered adaptive importance sampling*. In 22nd International Conference on Digital Signal Processing (DSP), pages 1–5. IEEE, 2017. (Cited on page 14.)

[Martino *et al.* 2017b] L. Martino, V. Elvira, D. Luengo and J. Corander. *Layered adaptive importance sampling*. Statistics and Computing, vol. 27, no. 3, pages 599–623, 2017. (Cited on page 14.)

[Mateos *et al.* 2016] J. Mateos, A. López, M. Vega, R. Molina and A. K. Katsaggelos. *Multiframe blind deconvolution of passive millimeter wave images using variational dirichlet blur kernel estimation*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2016), pages 2678–2682, Phoenix, Arizona, 25-28 Sept. 2016. (Cited on page 12.)

[Mattei 2020] P.-A. Mattei. A Parsimonious Tour of Bayesian Model Uncertainty. working paper or preprint, Dec. 2020. (Cited on page 7.)

[Michailovich & Tannenbaum 2007] O. Michailovich and A. Tannenbaum. *Blind Deconvolution of Medical Ultrasound Images: A Parametric Inverse Filtering Approach*. IEEE Transactions on Image Processing, vol. 16, no. 12, pages 3005–3019, 2007. (Cited on pages 9 and 51.)

[Mohan *et al.* 2021] M. R. Mahesh Mohan, G. K. Nithin and A. N. Rajagopalan. *Deep Dynamic Scene Deblurring for Unconstrained Dual-Lens Cameras*. IEEE Transactions on Image Processing, vol. 30, pages 4479–4491, 2021. (Cited on page 70.)

[Monga *et al.* 2021] V. Monga, Y. Li and Y. C. Eldar. *Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing*. IEEE Signal Processing Magazine, vol. 38, no. 2, pages 18–44, 2021. (Cited on pages 70 and 71.)

[Mourya 2016] R. K. Mourya. *Contributions to image restoration : from numerical optimization strategies to blind deconvolution and shift-variant deblurring*. PhD thesis, 2016. (Cited on pages 20, 21, 22, 27 and 34.)

[Mousavi *et al.* 2021] A. Mousavi, R. Monsefi and V. Elvira. *Hamiltonian adaptive importance sampling*. IEEE Signal Processing Letters, vol. 28, pages 713–717, 2021. (Cited on page 14.)

[Murtagh *et al.* 2007] F. Murtagh, E. Pantin and J.-L. Starck. Deconvolution and blind deconvolution in astronomy, pages 277–316. Taylor and Francis, 2007. (Cited on pages 9 and 51.)

[Nagy *et al.* 2004] J. G. Nagy, K. Palmer and L. Perrone. *Iterative Methods for Image Deblurring: A Matlab Object-Oriented Approach*. Numerical Algorithms, vol. 36, pages 73–93, 2004. (Cited on page 21.)

[Nan *et al.* 2020] Y. Nan, Y. Quan and H. Ji. *Variational-EM-Based Deep Learning for Noise-Blind Image Deblurring*. In Proceedings of the IEEE/CVF Conference on

Computer Vision and Pattern Recognition (CVPR 2020), Seattle, WA, USA, 16-18 Jun. 2020. (Cited on page 71.)

[Neal 2001] R. M. Neal. *Annealed importance sampling*. Statistics and Computing, vol. 11, no. 2, pages 125–139, 2001. (Cited on pages 28 and 30.)

[Ngolé Mboula & Starck 2017] F. M. Ngolé Mboula and J.-L. Starck. *PSF field learning based on Optimal Transport distances*. SIAM Journal on Imaging Sciences, vol. 10, pages 1549–1578, 2017. (Cited on page 23.)

[Nguyen *et al.* 2015] T. L. T. Nguyen, F. Septier, G. W. Peters and Y. Delignon. *Efficient sequential Monte-Carlo samplers for Bayesian inference*. IEEE Transactions on Signal Processing, vol. 64, no. 5, pages 1305–1319, 2015. (Cited on page 14.)

[Pankajakshan *et al.* 2012] P. Pankajakshan, Z. Kam, A. Dieterlen and J.-C. Olivo-Marin. *Characterizing the 3-D field distortions in low numerical aperture fluorescence zooming microscope*. Optics Express, vol. 20, no. 9, Apr. 2012. (Cited on page 22.)

[Pereyra *et al.* 2016] M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. O. Hero and S. McLaughlin. *A Survey of Stochastic Simulation and Optimization Methods in Signal Processing*. IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 2, pages 224–241, Mar. 2016. (Cited on page 52.)

[Pesquet *et al.* 2021] J.-C. Pesquet, A. Repetti, M. Terris and Y. Wiaux. *Learning Maximally Monotone Operators for Image Recovery*. SIAM Journal on Imaging Sciences, vol. 14, no. 3, pages 1206–1237, 2021. (Cited on page 70.)

[Pitt & Shephard 1999] M. K. Pitt and N. Shephard. *Filtering via Simulation: Auxiliary Particle Filters*. Journal of the American Statistical Association, vol. 94, no. 446, pages 590–599, 1999. (Cited on page 17.)

[Pitt & Shephard 2001] M. K. Pitt and N. Shephard. *Auxiliary variable based particle filters*. In A. Doucet, N. de Freitas and N. Gordon, editors, Sequential Monte Carlo Methods in Practice, chapter 13, pages 273–293. Springer, 2001. (Cited on page 28.)

[Prashnani *et al.* 2018] E. Prashnani, H. Cai, Y. Mostofi and P. Sen. *PieAPP: Perceptual Image-Error Assessment Through Pairwise Preference*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018), pages 1808–1817, Salt Lake City, Utah, 18-22 Jun. 2018. (Cited on page 78.)

[Pustelnik *et al.* 2016] N. Pustelnik, A. Benazza-Benhayia, Y. Zheng and J.-C. Pesquet. *Wavelet-based Image Deconvolution and Reconstruction*. Wiley Encyclopedia of Electrical and Electronics Engineering, 2016. (Cited on pages 12 and 19.)

[Rabbani & Jones 1991] M. Rabbani and P. W Jones. Digital image compression techniques. SPIE Optical Engineering Press, Bellingham, 1991. (Cited on page 27.)

[Rajagopalan & Chaudhuri 1999] A. N. Rajagopalan and S. Chaudhuri. *MRF model-based identification of shift-variant point spread function for a class of imaging systems*. Signal Processing, vol. 76, no. 3, pages 285–299, 1999. (Cited on page 22.)

[Reeves & Mersereau 1992] S. J. Reeves and R. M. Mersereau. *Blur Identification by the method of Generalized Cross-Validation*. IEEE Transactions of Image Processing, vol. 1, no. 3, 1992. (Cited on page 19.)

[Ren *et al.* 2020] D. Ren, K. Zhang, Q. Wang, Q. Hu and W. Zuo. *Neural Blind Deconvolution Using Deep Priors*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), pages 3338–3347, Seattle, WA, USA, 16-18 Jun. 2020. (Cited on pages 10, 69, 74 and 78.)

[Repetti *et al.* 2019] A. Repetti, M. Pereyra and Y. Wiaux. *Scalable Bayesian Uncertainty Quantification in Imaging Inverse Problems via Convex Optimization*. SIAM Journal on Imaging Sciences, vol. 12, no. 1, pages 87–118, 2019. (Cited on page 83.)

[Robert & Casella 2004] C. P. Robert and G. Casella. Monte Carlo statistical methods. Springer, 2004. (Cited on pages 12, 13, 29 and 52.)

[Robert *et al.* 2009] C. Robert, N. Chopin and J. Rousseau. *Harold Jeffreys's Theory of Probability Revisited*. Statistical Science, vol. 24, no. 2, pages 141–172, May 2009. (Cited on page 7.)

[Robert *et al.* 2018] C. P. Robert, V. Elvira, N. Tawn and C. Wu. *Accelerating MCMC algorithms*. Wiley Interdisciplinary Reviews: Computational Statistics, vol. 10, no. 5, page e1435, 2018. (Cited on page 12.)

[Robert 2007] C. P. Robert. The Bayesian Choice: From decision theoretic foundations to computational implementation. New York : Springer, Jan. 2007. (Cited on page 7.)

[Ronneberger *et al.* 2015] O. Ronneberger, P. Fischer and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015), volume 9351, pages 234–241, Munich, Germany, 5-9 Oct. 2015. (Cited on page 77.)

[Rosec *et al.* 2003] O. Rosec, J.-M. Boucher, B. Nsiri and T. Chonavel. *Blind marine seismic deconvolution using statistical MCMC methods*. IEEE Journal of Oceanic Engineering, vol. 28, no. 3, pages 502–512, 2003. (Cited on page 52.)

[Rozumnyi *et al.* 2021] D. Rozumnyi, M. R. Oswald, V. Ferrari, J. Matas and M. Pollefeys. *DeFMO: Deblurring and Shape Recovery of Fast Moving Objects*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021), 19-25 Jun. 2021. (Cited on page 9.)

[Russo 2003] F. Russo. *A method for estimation and filtering of Gaussian noise in images*. IEEE Transactions on Instrumentation and Measurement, vol. 52, no. 4, pages 1148–1154, 2003. (Cited on page 9.)

[Särkkä *et al.* 2007] S. Särkkä, A. Vehtari and J. Lampinen. *Rao-Blackwellized particle filter for multiple target tracking*. Information Fusion, vol. 8, no. 1, pages 2–15, 2007. (Cited on page 17.)

[Särkkä 2013] S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge, UK:Cambridge University Press, 2013. (Cited on pages 26 and 28.)

[Sbert *et al.* 2018] M. Sbert, V. Havran, L. Szirmay-Kalos and V. Elvira. *Multiple importance sampling characterization by weighted mean invariance*. The Visual Computer, vol. 34, no. 6, pages 843–852, 2018. (Cited on page 14.)

[Schmitz *et al.* 2020] M. A. Schmitz, J.-L. Starck, F. M. Ngolé Mboula*et al. Euclid: Non-parametric point spread function field recovery through interpolation on a graph Laplacian*. Instrumentation and Methods for Astrophysics, vol. 636, 2020. (Cited on pages 20, 22 and 23.)

[Schuler *et al.* 2012] C. J. Schuler, M. Hirsh and S. Harmeling. *Blind Correction of Optical Aberrations*. In Proceedings of the European Conference on Computer Vision (ECCV 2012), pages 187–200, Florence, Italy, 7-13 Oct. 2012. (Cited on pages 21 and 27.)

[Schuler *et al.* 2013] C. J. Schuler, H. Christopher Burger, S. Harmeling and B. Scholkopf. *A Machine Learning Approach for Non-blind Image Deconvolution*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2013), Portland, Oregon, USA, 23-28 Jun. 2013. (Cited on page 10.)

[Shan *et al.* 2008] Q. Shan, J. Jia and A. Agarwala. *High-quality Motion Deblurring from a Single Image*. ACM Transactions on Graphics, vol. 27, pages 1–10, Aug. 2008. (Cited on page 52.)

[Shi *et al.* 2014] J. Shi, L. Xu and J. Jia. *Discriminative Blur Detection Features*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2014), pages 2965–2972, Columbus, OH, USA, 23-28 Jun. 2014. (Cited on page 9.)

[Shi *et al.* 2017] X. Shi, R. Guo, Y. Zhu and Z. Wang. *Astronomical image restoration using variational Bayesian blind deconvolution*. Journal of Systems Engineering and Electronics, vol. 28, no. 6, pages 1236–1247, 2017. (Cited on page 12.)

[Simpkins & Stevenson 2014] J. D. Simpkins and R. L. Stevenson. *Parameterized modeling of spatially varying optical blur*. Journal of Electronic Imaging, vol. 23, no. 1, pages 1–13, 2014. (Cited on pages 25, 27, 30 and 32.)

[Sroubek *et al.* 2016] F. Sroubek, J. Kamenicky and Y. M. Lu. *Decomposition of Space-Variant Blur in Image Deconvolution*. IEEE Signal Processing Letters, vol. 23, no. 3, pages 346–350, 2016. (Cited on page 21.)

[Sun *et al.* 2014] L. Sun, S. Cho, J. Wang and J. Hays. *Good Image Priors for Non-blind Deconvolution*. In Proceedings of the European Conference on Computer Vision (ECCV 2014), pages 231–246, Zurich, Switzerland, 6-12 Sept. 2014. Springer International Publishing. (Cited on page 10.)

[Sun *et al.* 2021] Y. Sun, Z. Wu, X. Xu, B. Wohlberg and U. S. Kamilov. *Scalable Plug-and-Play ADMM With Convergence Guarantees*. IEEE Transactions on Computational Imaging, vol. 7, pages 849–863, 2021. (Cited on page 70.)

[Tezaur *et al.* 2015] R. Tezaur, T. Kamata, H.Li and S.S. Slonaker. *A system for estimating optics blur PSFs from test chart images*. In Nitin Sampat, Radka Tezaur and Dietmar Wüller, editors, Digital Photography XI, volume 9404. International Society for Optics and Photonics, SPIE, 2015. (Cited on pages 20, 21 and 22.)

[Tiancheng *et al.* 2015] L. Tiancheng, B. Miodrag and D. Petar. *Resampling Methods for Particle Filtering: Classification, implementation, and strategies*. IEEE Signal Processing Magazine, vol. 32, pages 70–86, 2015. (Cited on page 28.)

[Tico & Pulli 2009] M. Tico and K. Pulli. *Image enhancement method via blur and noisy image fusion*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2009), pages 1521–1524, Cairo, Egypt, 7-11 Nov. 2009. (Cited on pages 22 and 91.)

[Tico *et al.* 2010] M. Tico, N. Gelfand and K. Pulli. *Motion-blur-free exposure fusion*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2010), pages 3321–3324, Hong Kong, China, 12-15 Sept. 2010. (Cited on page 22.)

[Tolooshams *et al.* 2021] B. Tolooshams, S. Mulleti, D. Ba and Y. C. Eldar. *Unfolding Neural Networks for Compressive Multichannel Blind Deconvolution*. arXiv:2010.1139 [Online], 2021. (Cited on page 71.)

[Trouvé-Peloux *et al.* 2018] P. Trouvé-Peloux, J. Sabater, A. Bernard-Brunel, F. Champagnat, G. Le Besnerais and T. Avignon. *Turning a conventional camera into a 3D camera with an add-on*. Applied optics, vol. 57, no. 10, pages 2553–2563, 2018. (Cited on page 24.)

[Trouvé *et al.* 2011] P. Trouvé, F. Champagnat, G. Le Besnerais and J. Idier. *Single image local blur identification*. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2011), pages 613–616, Bruxelles, Belgium, 11-14 Sept. 2011. (Cited on pages 20 and 22.)

[Tzikas *et al.* 2009] D. G. Tzikas, A. C. Likas and N. P. Galatsanos. *Variational Bayesian Sparse Kernel-Based Blind Image Deconvolution With Student's-t Priors*. IEEE Transactions on Image Processing, vol. 18, no. 4, pages 753–764, 2009. (Cited on page 12.)

[Ulker *et al.* 2011] Y. Ulker, B. Gunsel and A. T. Cemgil. *Annealed SMC Samplers for Nonparametric Bayesian Mixture Models*. IEEE Signal Processing Letters, vol. 18, no. 1, pages 3–6, 2011. (Cited on page 30.)

[Šorel & Flusser 2008] M. Šorel and J. Flusser. *Space-Variant Restoration of Images Degraded by Camera Motion Blur*. IEEE Transactions on Image Processing, vol. 17, no. 2, pages 105–116, 2008. (Cited on page 21.)

[Wang *et al.* 2004] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli. *Image quality assessment: from error visibility to structural similarity*. IEEE Transactions on Image Processing, vol. 13, no. 4, pages 600–612, 2004. (Cited on pages 73 and 77.)

[Wen *et al.* 2021] Y. Wen, J. Chen, B. Sheng, Z. Chen, P. Li, P. Tan and T.-Y. Lee. *Structure-Aware Motion Deblurring Using Multi-Adversarial Optimized CycleGAN*. IEEE Transactions on Image Processing, vol. 30, pages 6142–6155, 2021. (Cited on page 69.)

[Xue & Blu 2014] F. Xue and T. Blu. *A novel SURE-based criterion for parametric PSF estimation*. IEEE Transactions on Image Processing, vol. 24, no. 2, pages 595–607, 2014. (Cited on page 20.)

[Zhang & Hirakawa 2016] Y. Zhang and K. Hirakawa. *Blind Deblurring and Denoising of Images Corrupted by Unidirectional Object Motion Blur and Sensor Noise*. IEEE Transactions on Image Processing, vol. 25, no. 9, pages 4129–4144, 2016. (Cited on pages 20 and 22.)

[Zhang *et al.* 2007] B. Zhang, J. Zerubia and J.-C. Olivo-Marin. *Gaussian approximations of fluorescence microscope point-spread function models*. Applied Optics, vol. 46, no. 10, pages 1819–1829, 2007. (Cited on page 24.)

[Zhang *et al.* 2016] X. Zhang, R. Wang, X. Jiang, W. Wang and W. Gao. *Spatially Variant Defocus Blur Map Estimation and Deblurring from a Single Image*. Journal of Visual Communication and Image Representation, vol. 35, 2016. (Cited on pages 20 and 22.)

[Zhang *et al.* 2020] K. Zhang, L. Van Gool and R. Timofte. *Deep unfolding network for image super-resolution*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020), pages 3217–3226, Virtual, 14-19 Jun. 2020. (Cited on page 70.)

[Zhang *et al.* 2021] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool and R. Timofte. *Plug-and-Play Image Restoration with Deep Denoiser Prior*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021. (Cited on page 70.)

[Zhao *et al.* 2016] N. Zhao, Q. Wei, A. Basarab, D. Kouamé and J.-Y. Tourneret. *Blind Deconvolution of Medical Ultrasound Images Using a Parametric Model for the Point Spread Function*. In Proceedings of the IEEE International Ultrasonics Symposium (IUS 2016), pages 1–4, Tours, France, 18-21 Sept. 2016. (Cited on page 12.)

[Zhe *et al.* 2014] H. Zhe, X. Li and M.H. Yang. *Joint Depth Estimation and Camera Shake Removal from Single Blurry Image*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2014), pages 2893–2900, Columbus, OH, USA, 23-28 Jun. 2014. (Cited on pages 20 and 22.)

[Zheng *et al.* 2015a] Y. Zheng, A. Fraysse and T. Rodet. *Efficient Variational Bayesian Approximation Method Based on Subspace Optimization*. IEEE Transactions on Image Processing, vol. 24, no. 2, pages 681–693, 2015. (Cited on page 52.)

[Zheng *et al.* 2015b] Y. Zheng, A. Fraysse and T. Rodet. *Efficient Variational Bayesian Approximation Method Based on Subspace Optimization*. IEEE Transactions on Image Processing, vol. 24, no. 2, pages 681–693, 2015. (Cited on page 52.)

[Zheng *et al.* 2015c] Y. Zheng, A. Fraysse and T. Rodet. *Wavelet based unsupervised variational Bayesian image reconstruction approach*. In 2015 23rd European Signal Processing Conference (EUSIPCO), pages 2167–2171, 2015. (Cited on page 12.)

[Zhou *et al.* 2015] X. Zhou, J. Mateos, F. Zhou, R. Molina and A. K. Katsaggelos. *Variational Dirichlet Blur Kernel Estimation*. IEEE Transactions on Image Processing, vol. 24, no. 12, pages 5127–5139, 2015. (Cited on page 12.)

[Zhu *et al.* 2017] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. In Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017), pages 2242–2251, Venice, Italy, 22-29 Oct. 2017. (Cited on page 70.)