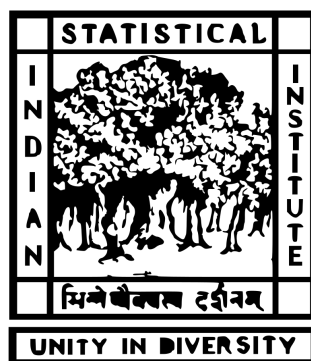


Locally Dependent Natural Image Priors for Non-blind and Blind Image Deconvolution

Kaustav Nandy



Indian Statistical Institute

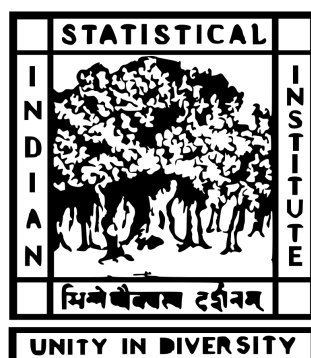
December, 2020

Locally Dependent Natural Image Priors for Non-blind and Blind Image Deconvolution

by

Kaustav Nandy

Thesis submitted to the Indian Statistical Institute
in partial fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy
December, 2020



Indian Statistical Institute

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Deepayan Sarkar for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank every faculty member of Indian Statistical Institute, Delhi Centre for their continuous support and motivation. My sincere thanks also goes to Indian Statistical Institute for the scholarship during my PhD tenure. Last but not the least, I would like to thank cooks, housekeeping staffs and all other workers of Indian Statistical Institute who made my stay at the Institute comfortable.

Finally, I would like to thank the two reviewers whose thoughtful comments led to many improvements in the thesis.

List of Figures

1.1	Some real-life examples of blurred images.	2
1.2	Example of image blurring.	6
1.3	The PSF of the flawed main mirror of the Hubble space telescope.	7
1.4	Eight unblurred test images.	9
1.5	Eight test blur kernels.	9
1.6	Synthetically blurred images without additional noise.	11
1.7	Synthetically blurred images with additional noise.	12
1.8	Distribution of raw image intensities and image gradients	13
2.1	Non-blind deconvolution using direct inversion (without noise).	25
2.2	Non-blind deconvolution using direct inversion (with noise).	26
2.3	Non-blind deconvolution using Richardson-Lucy (without noise).	32
2.4	Non-blind deconvolution using Richardson-Lucy (with noise).	33
3.1	Modulus of Fourier coefficients for synthetically blurred test images (without noise)	49
3.2	Sketch of Gerchberg-Saxton algorithm	50
5.1	Periodograms of synthetically blurred test images (without noise)	70
5.2	Periodograms of synthetically blurred test images (with noise)	71
5.3	Locally smoothed periodograms corresponding to gradients of test images. . . .	73
5.4	Estimated ACF and PACF of image gradients.	74
5.5	Plot of $V_L(\rho, \omega)$ for various values of ρ	78
5.6	Periodograms of synthetically blurred test images (without noise), corrected by variance function.	79
5.7	Estimates of blur kernel from unblurred images assuming symmetry.	82
5.8	Symmetrized blur kernels estimated using the IID prior model from synthetically blurred images (without noise)	85
5.9	Symmetrized blur kernels estimated using the dependent prior model from syn- thetically blurred images (without noise)	86

5.10	Symmetrized blur kernels estimated using the IID prior model from synthetically blurred images (with noise)	87
5.11	Symmetrized blur kernels estimated using the dependent prior model from synthetically blurred images (with noise)	88
6.1	Examples of super-resolution with user-supplied kernel.	99
6.2	Examples of super-resolution with estimated kernel.	100
6.3	Another example of Super-resolution.	101
6.4	Deconvolution of test images using estimated kernel.	103
6.5	Deconvolution of real-life images using estimated kernel.	104
6.6	Performance of non-blind deconvolution for synthetic images.	105
6.7	Performance of super-resolution for synthetic images.	106
7.1	Performance of auto-regressive model prior compared with IID prior.	115
7.2	Role of prior during non-blind deconvolution step.	117
7.3	Estimated kernels in the exact setup using the IID sparse prior	118
7.4	Estimated kernels in the exact setup using the AR sparse prior	119
7.5	Estimated kernels in the less noisy setup using the IID sparse prior	120
7.6	Estimated kernels in the less noisy setup using the AR sparse prior model	121
7.7	Estimated kernels in the more noisy setup using the IID sparse prior	122
7.8	Estimated kernels in the more noisy setup using the AR sparse prior	123
7.9	Estimated kernels under misspecified noise parameter	124
7.10	Sub-portions of real images used to estimate blur kernel.	126
7.11	Original images from Fergus et al. [17] and their deblurred versions.	127
7.12	Original images from personal collection and their deblurred versions.	128
8.1	Estimated distribution, ACF and PACF of image gradients of MRI scans.	136
A.1	Maximum correlation between DFT coefficients	149

Contents

1	Introduction and background	1
1.1	Overview	1
1.2	Mathematical formulation and basic ideas	4
1.3	Test images and kernels	8
1.4	Priors on Natural Images	10
1.5	Convolution and discrete Fourier transform	16
1.5.1	Discrete Fourier transform	19
1.6	Outline of thesis	21
2	Review of non-blind deconvolution methods	23
2.1	Non-blind deconvolution	23
2.1.1	Non-blind deconvolution by direct inversion	23
2.1.2	Wiener Filter	27
2.1.3	Constrained Least Square (CLS) filter	28
2.1.4	Total Variation based image restoration	29
2.1.5	Richardson Lucy Algorithm	29
2.2	Bayesian approaches	31
2.2.1	MAP estimation	34
2.3	Non-blind deconvolution: modern approaches	38
2.3.1	Direct MLE using gradient descent	39
2.3.2	Prior on latent image	40
2.4	Super-resolution as a generalization of deconvolution	41
2.4.1	Previous work on Single Image Super Resolution	42
3	Review of blind deconvolution methods	45
3.1	Overview of previous work	45
3.2	Phase retrieval	48
3.3	MAP estimation	52

3.4	The variational Bayes approach	55
4	MAP_k Blind deconvolution using IID prior	60
4.1	Problem formulation	60
4.2	The EM approach	61
4.3	The Gaussian Case	64
4.4	The Sparse MOG Case	66
4.5	Practicalities	68
5	Dependent prior: motivation and simple applications	69
5.1	Motivation: Fourier domain	69
5.2	Proposed generalization	75
5.3	Application: Estimation of symmetric blur kernel	78
5.4	Application: estimation of kernel size	83
6	Non-blind deconvolution and super-resolution using dependent prior	89
6.1	A unified approach	89
6.2	Mathematical formulation and derivations	92
6.3	Results	97
7	Blind deconvolution using dependent prior	108
7.1	Estimating blur kernel using generalized prior	109
7.2	Performance comparison for synthetic images	113
7.3	Results on real-life examples	117
8	Discussion and future work	129
8.1	Summary	129
8.2	Results	131
8.3	Extensions	132
8.4	Future work	134
8.5	Software	135
A	Details of auto-regressive model	146
A.1	Calculation of g_ω for auto-regressive model	146
A.2	Correlation among DFT coefficients	148

Chapter 1

Introduction and background

1.1 Overview

Degradation of photographic images is a common phenomenon that can occur due to several reasons. Astronomical images may be degraded due to atmospheric factors or telescope optics. Photographs taken using standard digital cameras may be blurred due to lack of focus, due to motion of the subject, due to low resolution, or due to camera shake during relatively long exposures. Often one wishes to correct for the effect of such degradation and recover the original image, and this has been a long-standing research problem in digital imaging. Effective solutions depend critically on the context of the problem and the type of degradation. Depending on the type of degradation, different types of methodologies can be adopted. At a high level, image restoration can be classified as deblurring of image, image upscaling or super resolution, image denoising, etc.

In context of image deblurring, we are specifically interested in motion blur in digital photography caused by camera shake. Blurring caused by camera shake is one of the most common artifacts in amateur digital photography. Long exposures are unavoidable in low light situations, and images taken without a camera stand often result in blurry and disappointing images. Recovering an unblurred image from such a motion-blurred photograph has long been a fundamental research problem. Figure [1.1](#) gives some real-life examples of blurred photographs; we use these images later as representative test cases.

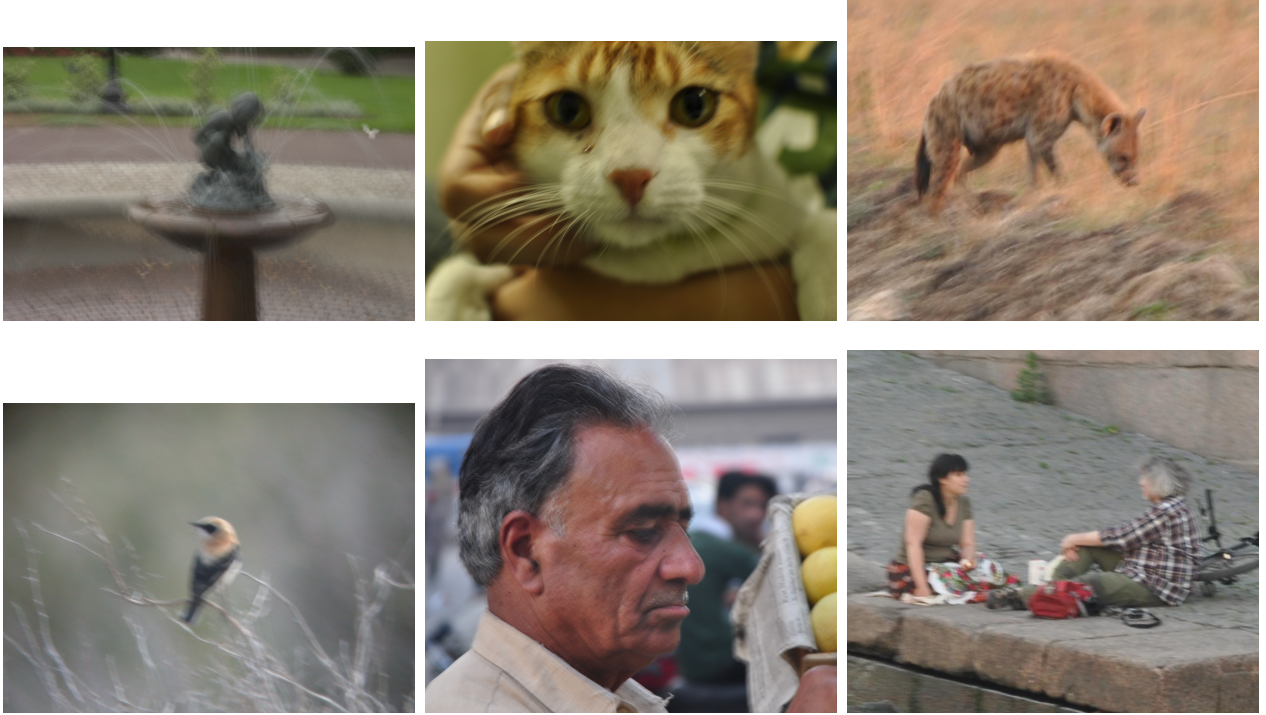


Figure 1.1: Some real-life examples of blurred images. The two images on the left-most column are from Fergus et al. [17]; the remaining four are from our personal collection.

Under-resolution is another common problem in photographs captured in low end cameras. Upsampling such images to a high resolution image is an useful problem as often it is needed to display a low resolution image in a bigger screen.

Camera shake during the course of an exposure essentially means that the light supposed to fall into a single pixel is instead distributed into several neighbouring pixels in a pattern determined by the motion of the camera. This pattern is commonly referred to as the *point spread function* (PSF) or *blur kernel*. If one assumes that the PSF is shift-invariant,¹ that is, the same PSF applies to all pixels of the underlying unblurred image, then the resulting observed image can be viewed as a convolution of the true unblurred image and the PSF. For this reason, this problem is often referred to the *image deconvolution* problem to distinguish it from the problem of recovering from other kinds of image degradation.

The broader class of image deconvolution problems are generally separated into *non-blind* and *blind* deconvolution problems.

¹This is a strong assumption that does not hold, for example, if there is camera rotation, or for long-exposure astronomical images that are affected by the Earth's rotation.

In non-blind deconvolution, the blur kernel is assumed to be known or estimated in advance; the only task is to estimate the unblurred latent image. This is commonly the case for astronomical images, where either systematic lens distortions are known, or can be estimated using the observed image of an isolated star that can be treated as a point source that does not interact with any neighbouring light sources.

In case of blind deconvolution, both image and the blur kernel are assumed to be unknown. This is the situation we are primarily interested in, as in amateur digital photography the camera shake pattern is unpredictable and different in each photograph, and usually not easily estimated using the dispersion pattern of point light source, as might be possible in astronomical images. Naturally, this problem is considerably more difficult, and reasonable progress has been made only in the last two decades or so.

As noted above, camera shake can be modeled through a blur kernel or PSF describing the camera motion during exposure, which is convolved with the underlying true image intensities to produce the observed blurred image. Removing the unknown camera shake is thus a form of blind image deconvolution, which is a problem with a long history in the image and signal processing literature. The main difficulty is that the problem is severely ill-posed when both the blur kernel and the true image are unknown, as there are more unknowns (the original image and the blur kernel) than measurements (the observed image). Hence, all practical solutions must make strong prior assumptions about the underlying image or the blur kernel or both.

Traditional signal processing formulations of the problem usually make only very general assumptions in the form of frequency-domain power laws; the resulting algorithms can typically handle only very small blurs and not the complicated blur kernels often associated with camera shake. Furthermore, algorithms exploiting image priors specified in the frequency domain may not preserve important spatial-domain structures such as edges. More recent approaches, such as those of by Fergus et al. [17] and Levin et al. [45], have been considerably more successful by using more complicated but realistic priors assumptions on the underlying image. Our work is an attempt to generalize the prior they have used and use it to extend their methods and

other methods based on similar priors.

In the next few sections, we give a precise mathematical formulation of the problem we are interested in solving, introduce a set of example images and blur kernels that we use for illustration throughout the thesis, and describe the prior on underlying true images that has been successfully used by Fergus et al. [17] and Levin et al. [45]. We end this chapter with a brief outline of the discrete Fourier transform (DFT) and its properties, as it is used extensively throughout this thesis.

In Chapters 2 and 3, we briefly review previous work on non-blind and blind deconvolution, respectively. Chapter 4 describes in detail the approach of Levin et al. [45], which we use as a basis for our generalization for the blind deconvolution problem, although in principle our methods can be used to generalize any method based on a similar prior. We describe the motivation for our proposed generalization of image priors in Chapter 5, along with simple applications. We discuss non-blind deconvolution and super-resolution (upsampling) using the generalized prior in Chapter 6, and blind deconvolution in Chapter 7. Finally, Chapter 8 reviews our findings and discusses possible future avenues of work.

1.2 Mathematical formulation and basic ideas

In this thesis, we deal exclusively with photographic images of “natural” scenes. Here by “natural” we mean the common everyday scenes that are the usual subjects of amateur digital photography, and not images taken in specialized contexts such as astronomy, satellite imaging, or even professional photography. More precisely, we assume that the scene being photographed satisfies certain statistical properties, to be discussed in due course, that happen to be satisfied by a wide range of scenes that might be informally described as “natural” scenes.

In this work, we mainly consider images degraded due to motion blur caused by camera shake, and try to recover the original image from the degraded one. We optionally allow for further loss of resolution of the image due to subsampling; this is considered only in Chapter 6 where we study the super-resolution problem. We exclude the subsampling operation from the formulation here to avoid notational complexity, and discuss it in the relevant chapter.

If we assume that the scene remains static for the duration of the photograph and there is no significant camera rotation, then the blur caused by camera shake can be described by a convolution or filter. An observed blurred image $\mathbf{b}$ of dimension $M \times N$ can be modeled as

$$\mathbf{b} = \mathbf{k} \otimes \boldsymbol{\ell} + \boldsymbol{\varepsilon}, \quad (1.1)$$

where $\mathbf{k}$ is an $m \times n$ blur kernel or point spread function (PSF), $\boldsymbol{\ell}$ is the $(M + m) \times (N + n)$ “true” latent image we want to estimate, and $\boldsymbol{\varepsilon}$ is an $M \times N$ matrix of noise. Throughout this thesis, we will interchangeably use $\otimes$ to mean either *convolution*, where an element of the resulting blurred image is given by (excluding noise)²

$$(\mathbf{k} \otimes \boldsymbol{\ell})_{kl} = \sum_{i,j} \boldsymbol{\ell}_{(k-i+1)(l-j+1)} \mathbf{k}_{ij},$$

or *filtering*, where

$$(\mathbf{k} \otimes \boldsymbol{\ell})_{kl} = \sum_{i,j} \boldsymbol{\ell}_{(k+i-1)(l+j-1)} \mathbf{k}_{ij}.$$

The difference matters for implementation and will be emphasized when necessary, but is unimportant for the most part. We note here that convolution with $\mathbf{k}$ is same as filtering with $\text{flip}(\mathbf{k})$, and vice versa, where $\text{flip}(\mathbf{k})$ is obtained from $\mathbf{k}$ by reversing the order of both rows and columns. This simple observation, along with the notion of $\text{flip}(\mathbf{k})$, will be useful later.

It is to be noted that we will generally take convolution to mean what is commonly referred to as “valid” convolution, which retains only those pixels for which all relevant neighbors are available. This has the consequence that the dimension of the observed blurred image $\mathbf{b}$ is smaller than that of the latent image $\boldsymbol{\ell}$. It is important to keep in mind that a useful mathematical technique to deal with convolutions, namely the result that Fourier transforms of convolutions are products of Fourier transforms of the components, implicitly assumes that the inputs are periodic, and fills in missing pixels by “wrapping” around the input image, which is different from what we would get with valid convolution.

²In this thesis we use the notation $\mathbf{x}_{ij}$ to indicate the (i, j) -th element of the matrix $\mathbf{x}$, instead of the more standard x_{ij} , to avoid additional notation to represent elements of $\mathbf{k} \otimes \boldsymbol{\ell}$, etc.

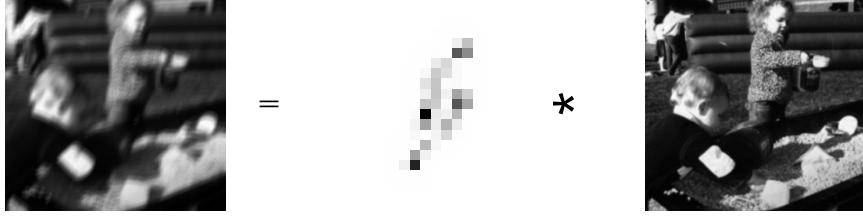


Figure 1.2: Example of image blurring. The Blurred image (left) is the convolution of the blur kernel or PSF (middle) and the true / latent image (right).

The observed blurred image can be a colored image or a grayscale image. Colored images are usually represented as a combination of three channels (red, green, and blue). For such images, we either assume that (1.1) holds for each channel separately, or that it holds after transforming the color image to a grayscale image (e.g., using a standard conversion formula such as $\text{Grey Image} = 0.3 \times \text{Red Channel} + 0.59 \times \text{Green Channel} + 0.11 \times \text{Blue Channel}$). In practice, we adopt the second approach during estimation of an unknown blur kernel (blind deconvolution), and the first approach during image reconstruction (non-blind deconvolution or upsampling) to reconstruct each channel separately before combining them into a color image. Additionally, we scale the input images so that elements of $\mathbf{b}$ between 0 and 1; pixel values usually represented as integers between 0 and 255 are divided by 255.

Figure 1.2 demonstrates the blurring phenomenon, modeled as a convolution, through a synthetically blurred image (without additional noise). The inference problem of recovering the sharper “true” image ℓ from the blurred image $\mathbf{b}$ is known as the image deconvolution problem. Image degradation can occur due to noise too, but we only consider the problem of image deconvolution, potentially in the presence of noise.

For a given blurred image, the underlying point spread function or blur kernel $\mathbf{k}$ may or may not be known. Depending on the availability of $\mathbf{k}$, the deconvolution problem can be *non-blind deconvolution* or *blind deconvolution*. We briefly describe these two problems here and review them in greater detail in the next two chapters.

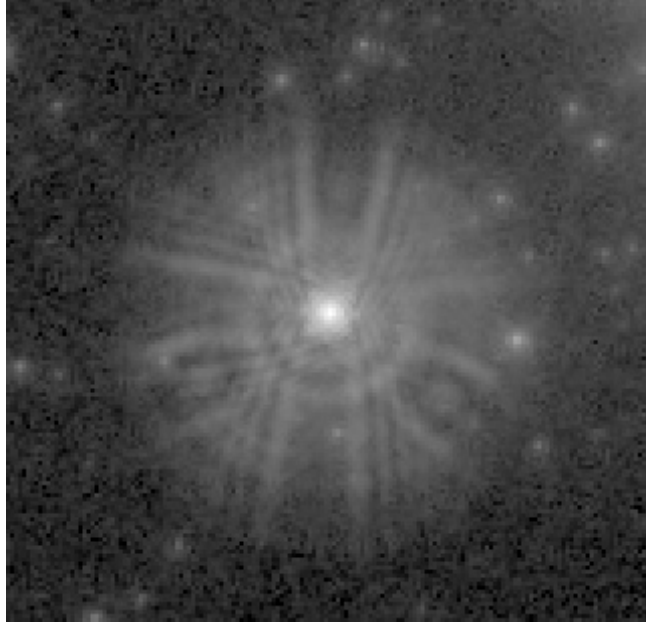


Figure 1.3: The PSF of the flawed main mirror of the Hubble space telescope, as seen in the image of an isolated star before the optics were corrected by a servicing mission. Image courtesy Wikipedia.

Non-blind Deconvolution In non-blind deconvolution, the blur kernel $\mathbf{k}$ is known, and the problem is to get an estimate of ℓ given $\mathbf{b}$ and $\mathbf{k}$. Non-blind deconvolution methods have long been used for astronomical images, and received particular attention after the launch of the Hubble space telescope (HST) in 1990. Within weeks of the launch, the returned images indicated a serious problem with the optical system. It turned out that the main mirror in the HST had been ground imperfectly, creating slightly unfocused images. A subsequent servicing mission in 1993 eventually corrected the distortion by adding another corrective optical component. However, the images taken before this mission were affected by a PSF that was well-characterized and stable, as seen in the image of an isolated star (that should have appeared as a point source) in Figure 1.3. As it was possible for astronomers to calculate the PSF quite accurately, these images were ideal candidates for the non-blind deconvolution problem, and were in fact quite useful after being processed using deconvolution techniques. We review several solutions of the non-blind deconvolution problem in Chapter 2.

Blind Deconvolution Deblurring a blurred image becomes more challenging when the blur kernel is unknown, which is more common in practice. In the case of astronomical images,

it is often possible to guess the PSF using the image of an isolated point source. However, in day-to-day amateur photography, blur patterns can be different for every blurred image. Specialized cameras can sometimes contain additional sensors to determine the blur pattern, but access to such equipment is limited [43].

The main challenge in blind deconvolution is that the problem is ill-posed. As both ℓ and $\mathbf{k}$ are unknown, there are more unknowns to estimate than the number of points in $\mathbf{b}$, and as a result, there are multiple solutions. For example, $\mathbf{k}_{1 \times 1} = [1]$ and $\ell = \mathbf{b}$ is an exact but useless solution.³ To deal with this problem, additional structural assumptions about ℓ and / or $\mathbf{k}$ are typically made. The relevant assumptions depend on the problem context; for example, satellite images used for remote sensing may have different characteristics than medical images or amateur digital photographs. For photographic images, it is natural to take a Bayesian approach which assumes priors $f(\ell)$ on the latent image and possibly $f(\mathbf{k})$ on the kernel. These approaches typically lead to some form of regularization. We review this approach in Chapter 3 and discuss a specific method in further detail in Chapter 4. Our main contribution, described in the subsequent chapters, is to generalize a particular prior on ℓ that was first used successfully in this context by Fergus et al. [17].

1.3 Test images and kernels

We use a collection of eight unblurred test images and eight test kernels throughout this thesis to illustrate various ideas and demonstrate the performance of various methods. The eight test images are shown in Figure 1.4 and the eight blur kernels are shown in Figure 1.5.

The obvious advantage of working with such test images is that we can synthetically create blurred images using any image-kernel combination, with or without additional noise, and evaluate the performance of various deconvolution methods by comparing the recovered image with the original. We have created three sets of synthetically blurred images, each consisting of 64 such combinations.

The first set, shown in Figure 1.6, are exact convolutions of the test image and test kernel,

³This problem also arises in the non-blind deconvolution case, but is not as severe there.



Figure 1.4: Eight unblurred test images. The first four images (top row) are from Levin et al. [44]. The last four images (bottom row) are from our personal collection.

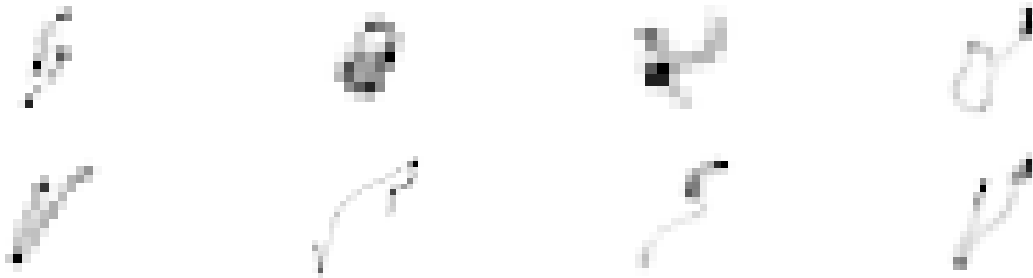


Figure 1.5: Eight test blur kernels, taken from from Levin et al. [44].

without any additional noise other than rounding due to storage as 8-bit digital images. We refer to this set as the “exact” synthetic setup.

To investigate the effect of noise, we also obtain two other sets of 64 test input images where noise is added; if the intensity of a pixel in the corresponding “exact” blurred image is an integer L between 0 and 255, it is replaced by X/S , where X is a Poisson random variable with mean SL , rounded to the nearest integer between 0 and 255. This is a natural noise model for the raw image. We refer to this as the “noisy” setup. Images generated with $S = 100$ are shown in Figure 1.7. Although the noisy images are often visually indistinguishable, we will see that naive methods often exhibit significant performance degradation for these noisy images. Another set of even more noisy images generated with $S = 10$ are not shown here, but used to illustrate the methods described in Chapters 6 and 7.

1.4 Priors on Natural Images

The main focus of our work is on a Bayesian image deconvolution approach that derives its power primarily from the choice of prior it puts on the latent image ℓ . It is thus important to understand the kinds of priors that are used in such methods.

If the goal is to find a representative prior that may be used to model a diverse range of natural scenes, it is not obvious what such a prior might look like, as it is rather difficult to define what a “natural” image is and what commonalities they might have. The prior family used for the latent image is motivated by the observation that when a band-pass filter is applied to an image, the distribution of the output has a consistent and distinctive form across a wide range of scene types, with the distribution sharply peaked at zero and relatively heavy-tailed. In particular, this applies to wavelet coefficients [56, 70, 71] as well as image derivatives or gradients [78].

This phenomenon is illustrated using the eight unblurred test images in Figure 1.8, which shows the frequency distribution of horizontal gradients, that is, lag-1 differences along rows, of the images. The vertical gradients (along columns) are not shown, but exhibit similar behaviour. As noted above, the distribution of image gradients consistently have a sharper peak at zero



Figure 1.6: Synthetically blurred images for all combinations of the eight test images (columns) and eight test kernels (rows). There is no additional noise, except for the integer rounding inherent in storing 8-bit images.



Figure 1.7: Synthetically blurred images for all combinations of the eight test images (columns) and eight test kernels (rows). Poisson noise ($S = 100$) is added to the exact blurred images shown in Figure 1.6 as described in the text.

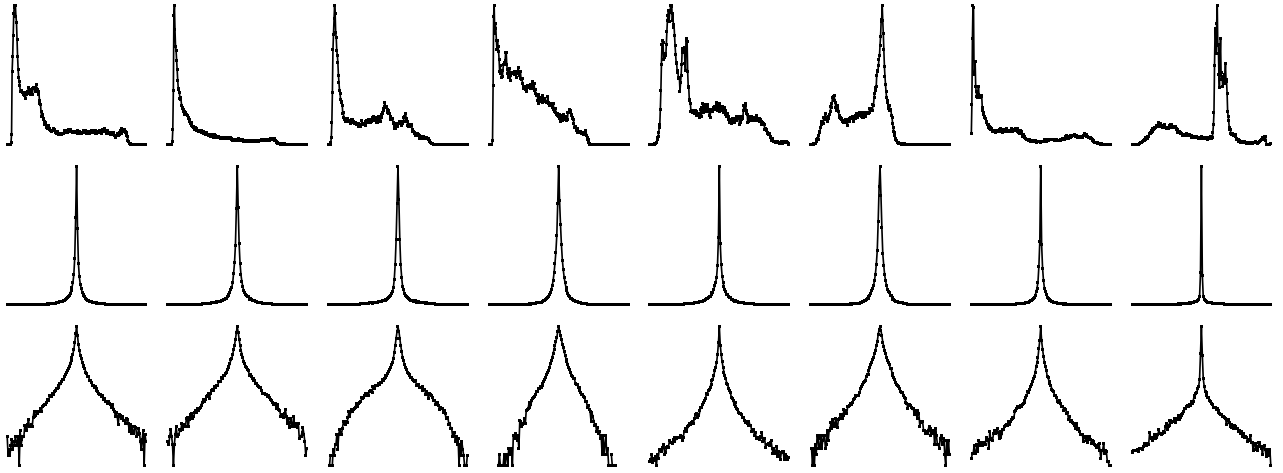


Figure 1.8: Distribution of raw intensities and gradients (derivatives) across the eight test images shown in Figure 1.4. Although the histograms of image intensity values vary considerably in shape across images (top row), the histograms of horizontal gradients (lag-1 differences along rows) are remarkably similar (middle row). The bottom row shows the histograms of the horizontal gradients with log-transformed frequencies.

and heavier tails than a Gaussian distribution. To reflect this behaviour, the prior on ℓ should be such that its gradients follow a similar distribution.

Prior on image gradients We henceforth refer to the distribution of gradients by $\pi(x)$. From a computational standpoint, a convenient assumption on $\pi(x)$ is that it is a zero-mean Gaussian, i.e.,

$$\pi(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (1.2)$$

However, it is clear from the discussion above that this is not a reasonable assumption, and we require better alternatives.

Sparse priors As can be seen from Figure 1.8, the distribution of image gradients is more sparse (i.e., sharper peak at 0) and have heavier tails than a Gaussian distribution, and even a Laplacian (double exponential) distribution. This effectively captures two common aspects of the distribution of gradients of natural images that are not supported by the Gaussian family: the more frequent values near zero captures smooth regions, and the more frequent large values (in absolute magnitude) captures the sharper edges. Priors with these features are often referred to as *sparse priors*, and a useful parametric family to model such priors is the

so-called generalized or hyper-Laplacian distribution given by (upto scale)

$$f_\alpha(z) = \frac{\alpha}{2\Gamma(1/\alpha)} \exp(-|z|^\alpha), z \in \mathbb{R}, \quad (1.3)$$

for $\alpha > 0$. This family reduces to the Gaussian distribution for $\alpha = 2$ and the Laplacian distribution for $\alpha = 1$. Values of $\alpha \in [0.5, 0.8]$ have been empirically found to be appropriate for natural images [43, 37]. We use both $\alpha = 2$ and $\alpha = 0.8$ for the results shown in Chapter 6. Although these values may not be exactly appropriate for all input images, we usually find that the image fidelity constraint (i.e., the requirement that the conditional mean of $\mathbf{b}$ given ℓ should be close to $\mathbf{k} \otimes \ell$) plays a bigger role in determining the result than the choice of α .

In practice, the hyper-Laplacian prior is sometimes computationally difficult to work with, and is often approximated by a mixture of zero-mean Gaussians (MOG), with

$$\pi(x) = \sum_{j=1}^J p_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{x^2}{2\sigma_j^2}\right), \quad \sum_{j=1}^J p_j = 1. \quad (1.4)$$

where the p_j -s are mixture probabilities and the σ_j^2 -s are the variances of the Gaussian components. We will refer to such priors interchangeably as *sparse* or *MOG* priors, and use it for the results shown in Chapter 7. We will also often compare the MOG model with the simple Gaussian prior (1.2) which, although not as effective, is simpler to analyze and computationally more convenient. The parameters of these priors, including the number of components in the case of the MOG prior, are usually estimated from a library of “good” unblurred natural images.

Prior on image It is not immediately clear how a prior on gradients can lead to a prior on the original image ℓ , as the horizontal gradients ($[-1, 1] \otimes \ell$) and vertical gradients ($[-1, 1]^T \otimes \ell$) must integrate to the same image ℓ and are therefore not independent. It would be more correct to formulate the prior on ℓ as a Markov Random Field; however, it is simpler to pretend that the horizontal gradients and vertical gradients are independent, and this approach works well in practice.

As the operation of taking gradients can be expressed as a taking convolutions with $\delta_h = [-1, 1]$ and $\delta_v = [-1, 1]^T$, and convolutions commute, (1.1) gives

$$\delta_h \otimes \mathbf{b} = \delta_h \otimes (\mathbf{k} \otimes \ell + \varepsilon) = \mathbf{k} \otimes (\delta_h \otimes \ell) + (\delta_h \otimes \varepsilon), \quad (1.5)$$

$$\delta_v \otimes \mathbf{b} = \delta_v \otimes (\mathbf{k} \otimes \ell + \varepsilon) = \mathbf{k} \otimes (\delta_v \otimes \ell) + (\delta_v \otimes \varepsilon) \quad (1.6)$$

To keep notation simple, we will henceforth take the model

$$\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n} \quad (1.7)$$

to represent both (1.5) and (1.6), where $\mathbf{y}$ is the *gradient* (either horizontal or vertical) of the blurred image rather than the original blurred image itself, $\mathbf{x}$ is similarly the gradient of the true latent image, and $\mathbf{n}$ is the gradient of the noise. For the blind deconvolution problem, all computations are done separately for the two sets of gradients, and used to estimate a common $\mathbf{k}$ assuming that the two gradients are independent.

With this background, it is now reasonable to assume that the image gradient $\mathbf{x}$ is a stationary process and that the prior *marginal* distribution for individual pixel elements $\mathbf{x}_i$ is given by the common density $\pi(x)$. Existing methods additionally assume that the individual gradients are independent. However, this assumption is not justified *a priori*, and in fact we demonstrate in this thesis that the assumption of independence does not hold for most images. We instead propose a more general dependent prior, and obtain an extension of the algorithm of Levin et al. [45] that is fully adapted for the generalized prior when $\pi(x)$ is Gaussian, and gives an approximate solution when $\pi(x)$ is sparse.

It is important to note that some earlier works on image deconvolution used Markov random field type priors to model the dependency [see, e.g., 6]. However such models are typically more suitable for textured images. More recently, Roth and Black [64] have used Field of Expert models which is an extension of traditional Markov random field model. They use a graphical model and write the probability distribution as Gibbs distribution based on clique potentials. Though this approach is better in terms of capturing dependency among nearby image pixels,

such priors are complicated and difficult to use for practical purposes.

1.5 Convolution and discrete Fourier transform

We finish this chapter with a brief review of convolutions, discrete Fourier transforms, and their properties, which we use throughout this thesis.

Recall that we use the $\otimes$ operator in the expression $\mathbf{y} = \mathbf{k} \otimes \mathbf{x}$ to represent either two-dimensional convolution or two-dimensional filtering. When $\otimes$ is taken to be a convolution, an element of the result is

$$\mathbf{y}_{kl} = \sum_{i,j} \mathbf{x}_{(k-i+1)(l-j+1)} \mathbf{k}_{ij},$$

whereas for filtering it is

$$\mathbf{y}_{kl} = \sum_{i,j} \mathbf{x}_{(k+i-1)(l+j-1)} \mathbf{k}_{ij}.$$

The map $\mathbf{x} \mapsto \mathbf{k} \otimes \mathbf{x}$ can be easily seen to be a linear transformation, and so in principle, it can be expressed as a matrix operation. In other words, if we view $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{k}$ as vectors, and denote the vector form of a matrix $\mathbf{m}$ by $\text{vec}(\mathbf{m})$, we can write

$$\text{vec}(\mathbf{y}) = \text{vec}(\mathbf{k} \otimes \mathbf{x}) = T_{\mathbf{k}} \text{vec}(\mathbf{x})$$

where $T_{\mathbf{k}}$ is known as the the convolution matrix corresponding to $\mathbf{k}$. The structure of $T_{\mathbf{k}}$ is easy to observe in one dimension, in which case it is a Toeplitz matrix with each non-zero element of $\mathbf{k}$ contributing a non-zero diagonal. Specifically, suppose $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ and $\mathbf{k} = (k_1, k_2, \dots, k_m)^T$ are respectively n and m dimensional vectors, with $m < n$. Then the convolution $\mathbf{x} \mapsto \mathbf{k} \otimes \mathbf{x}$ can be written as

$$\mathbf{y} = \mathbf{k} \otimes \mathbf{x} = T_{\mathbf{k}} \mathbf{x}$$

where T is a $n \times n$ matrix whose transpose is given by

$$T^T = \begin{bmatrix} k_m & 0 & \dots & 0 & 0 \\ k_{m-1} & k_m & \dots & \vdots & \vdots \\ k_{m-2} & k_{m-1} & \dots & 0 & 0 \\ \vdots & k_{m-2} & \dots & k_m & 0 \\ k_2 & \vdots & \dots & k_{m-1} & k_m \\ k_1 & k_2 & \vdots & \vdots & k_{m-1} \\ 0 & k_1 & \dots & k_3 & \vdots \\ 0 & 0 & \dots & k_2 & k_3 \\ \vdots & \vdots & \vdots & k_1 & k_2 \\ 0 & 0 & 0 & \dots & k_1 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

Essentially, the i -th row of T is

$$T_{i,*} = (\mathbf{0}_{(i-1)}, k_m, k_{m-1}, \dots, k_2, k_1, \mathbf{0}_{(n-m-i+1)})$$

where $\mathbf{0}_{(i)}$ is a i -tuple with all 0 entries. If $\otimes$ is a filtering instead, then $\mathbf{x} \mapsto \mathbf{k} \otimes \mathbf{x}$ is written as

$$\mathbf{y} = \mathbf{k} \otimes \mathbf{x} = T\mathbf{x}$$

where the transpose of T is

$$T^T = \begin{bmatrix} k_1 & 0 & \dots & 0 & 0 \\ k_2 & k_1 & \dots & \vdots & \vdots \\ k_3 & k_2 & \dots & 0 & 0 \\ \vdots & k_3 & \dots & k_1 & 0 \\ k_{m-1} & \vdots & \vdots & k_2 & k_1 \\ k_m & k_{m-1} & \dots & \vdots & k_2 \\ 0 & k_m & \dots & k_{m-2} & \vdots \\ 0 & 0 & \dots & k_{m-1} & k_{m-2} \\ \vdots & \vdots & \vdots & k_m & k_{m-1} \\ 0 & 0 & 0 & \dots & k_m \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

That is, the i -th row of T is given by

$$T_{i,*} = (\mathbf{0}_{(i-1)}, k_1, k_2, \dots, k_{m-1}, k_m, \mathbf{0}_{(n-m-i+1)})$$

In the case of two-dimensional convolution or filtering, a similar property holds; i.e., the convolution matrix $T_{\mathbf{k}}$ is still a Toeplitz matrix, with diagonals corresponding to elements of $\mathbf{k}$, although they may no longer be contiguous, and the pattern depends on how $\mathbf{x}$ is vectorized.

An important practical issue in defining convolutions is the question of how to define it at the boundaries. Several alternatives are available in typical software implementations, such as padding by zeros or assume that the boundaries “reflect”. One particular choice that has implications in the mathematical modeling is to assume that $\mathbf{x}$ is periodic, as this choice allows us to use the convolution theorem relating convolution or filtering to discrete Fourier transforms (see below). In practice, one often assumes the validity of the convolution theorem regardless of the choice of boundary calculations, and some heuristic techniques are applied to lessen undesirable computational effects.

The treatment of non-blind deconvolution and super-resolution in Chapter 6 handles boundary issues more carefully by avoiding Fourier domain calculations. There the convolution matrix

represents valid convolution, with suitable modifications.

1.5.1 Discrete Fourier transform

It is naturally of interest to solve linear equations of the form

$$T_{\mathbf{k}} \text{vec}(\mathbf{x}) = \text{vec}(\mathbf{y})$$

for the unknown $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{k}$, and other similar related equations. If one can invert $T_{\mathbf{k}}$, $\mathbf{x}$ can be recovered from $\mathbf{y}$ as $\mathbf{x} = T_{\mathbf{k}}^{-1}\mathbf{y}$. For large images, inverting $T_{\mathbf{k}}$ directly can be computationally costly. A popular and computationally efficient way to compute this inverse is done in Fourier domain through the discrete Fourier transform (DFT) and its inverse (IDFT) [13, 57].

In one dimension, given a (time-domain) signal $x_0, x_2, \dots, x_{N-1}$ which is assumed to be periodic, the discrete Fourier transform is defined by

$$X(u) = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi i u n / N}, \quad u \in \mathbb{Z}$$

Because of periodicity, evaluating $\mathbf{X}(\cdot)$ in $\{0, 1, \dots, N-1\}$ is enough; e.g., taking $m \geq N$ such that $m \equiv m' \pmod{N}$, i.e., $m = uN + m'$,

$$\begin{aligned} X(m) &= \sum_{n=1}^N x_n \cdot e^{-2\pi i \omega (uN+m')/N} \\ &= \sum_{n=1}^N x_n \cdot e^{-2\pi i \omega u} e^{-2\pi i \omega m'/N} \\ &= \sum_{n=1}^N x_n \cdot e^{-2\pi i \omega m'/N} = X(m') \end{aligned}$$

This definition can be generalized to higher dimension as follows. Given (spatial domain) data $x_{n_1, n_2, \dots, n_d}$ that is a function of d discrete indices $n_l = 0, 1, \dots, N_l - 1$, where l is in $1, 2, \dots, d$,

the discrete Fourier transform is given by

$$X_{u_1, u_2, \dots, u_d} = \sum_{n_1=0}^{N_1-1} \left(\omega_{N_1}^{u_1 n_1} \sum_{n_2=0}^{N_2-1} \left(\omega_{N_2}^{u_2 n_2} \cdots \sum_{n_d=0}^{N_d-1} \omega_{N_d}^{u_d n_d} \cdot x_{n_1, n_2, \dots, n_d} \right) \right),$$

where $\omega_{N_\ell} = \exp(-2\pi i/N_\ell)$, the d output indices run from $u_l = 0, 1, \dots, N_l - 1$. This is more compactly expressed in vector notation. If we define $\mathbf{n} = (n_1, n_2, \dots, n_d)$ and $\mathbf{u} = (u_1, u_2, \dots, u_d)$ as d -dimensional vectors of indices from 0 to $\mathbf{N} - 1$, which we define as $\mathbf{N} - 1 = (N_1 - 1, N_2 - 1, \dots, N_d - 1)$, we have

$$\mathbf{X}_{\mathbf{u}} = \sum_{\mathbf{n}=\mathbf{0}}^{\mathbf{N}-1} e^{-2\pi i \mathbf{u} \cdot (\mathbf{n}/\mathbf{N})} \mathbf{x}_{\mathbf{n}}.$$

For our purpose, the dimension $d = 2$.

The following are some useful properties of Discrete Fourier Transforms that are easy to verify for $d = 1$ also hold for general d . We state these properties without proof as they are standard results.

1. DFT is an *orthogonal* transformation. The vectors $v_u = \{e^{-\frac{2\pi i}{N}un} : n = 0, 1, \dots, N - 1\}$ is an orthogonal basis over the set of N -dimensional complex vectors. Here,

$$v_u^* v_{u'} = N \delta_{uu'}$$

where $\delta_{uu'}$ is the Kronecker delta function.

2. DFT is *linear* and *invertible*. This leads to defining Inverse Discrete Fourier Transformation (IDFT). Considering the vector notations defined in definition of DFT, IDFT can be described as a transformation as

$$\mathbf{x}_{\mathbf{n}} = \frac{1}{N_1 N_2 \cdots N_d} \sum_{\mathbf{u}=\mathbf{0}}^{\mathbf{N}-1} e^{2\pi i \mathbf{u} \cdot (\mathbf{n}/\mathbf{N})} \mathbf{X}_{\mathbf{u}}.$$

Note that IDFT is essentially the same as DFT apart from a normalizing constant.

3. In case of real input, DFT obeys symmetry. To illustrate, e.g., for $d = 1$, if $x_0, \dots, x_{N-1}$

are real numbers, then for every u , $X_{N-u} = X_{-u} = X_u^*$, where X_u^* is the complex conjugate of X_u . Spatial symmetry also holds in higher dimensions.

4. The convolution theorem for the general Fourier transform also holds for DFT, assuming convolution is computed assuming periodic signals.

Theorem 1.5.1 (Convolution Theorem) *For two discrete arrays $\mathbf{x}$ and $\mathbf{y}$,*

$$DFT(\mathbf{x} \otimes \mathbf{y}) = DFT(\mathbf{x}) \odot DFT(\mathbf{y})$$

where $\otimes$ denotes usual convolution and $\odot$ represents element-wise multiplication.

In the case where $\otimes$ denotes filtering, we have

$$DFT(\mathbf{x} \otimes \mathbf{y}) = DFT(\mathbf{x})^* \odot DFT(\mathbf{y}).$$

1.6 Outline of thesis

We conclude this chapter with an outline of the remainder of this thesis. Chapter 2 reviews methods for solving the non-blind deconvolution problem. This problem is relatively easy, and we summarize several classical methods as well as some modern Bayesian approaches that use sparse natural image priors. Blind deconvolution is a much more difficult problem to solve, and techniques can widely vary based on the underlying assumptions. In Chapter 3, we review various techniques that have been developed over the last few decades, emphasizing the considerable progress enabled by the adoption of sparse priors in conjunction with the use of variational Bayes methods. We proceed to discuss three specific techniques in further detail: we describe some phase retrieval algorithms that attempt to reconstruct a signal from only the magnitude of its Fourier transform; we provide an overview of the Bayesian *maximum a posteriori* (MAP) approach; and briefly introduce the variational Bayes method. Chapter 4 delves into the details of the EM algorithm approach used by Levin et al. [45] which we use as a benchmark.

In Chapter 5, we motivate and describe a proposed generalization of the IID priors described in the previous chapters. We do so by placing additional model assumptions not directly on the image derivatives, but rather on their Fourier transforms. Chapter 5 also includes two related simple applications of the generalized prior we propose: estimation of the unknown blur kernel assuming symmetry, and estimation of the unknown kernel size from an observed blurred image.

Chapter 6 outlines a method to incorporate the generalized prior in a Bayesian approach to non-blind deconvolution and super-resolution. In addition to the usual conjugate-gradient approach, we also implement a direct solution using numerical sparse matrix libraries. This gives more accurate results, and also allows further generalization to allow inpainting.

Chapter 7 outlines a general approach to modify the existing blind deconvolution algorithm of Levin et al. [45] based on IID priors to use the dependent prior. The crux of the method is an imputation step where given a current estimate of the blur kernel $\mathbf{k}$, the observed blurred image gradient $\mathbf{y}$ is used to estimate a transformed variable $\mathbf{u}$ that more closely follows the IID prior model. The usual estimation then proceeds with this imputed gradient matrix. To demonstrate the improvement facilitated by our proposed generalization, we use our approach to modify the estimation procedure of Levin et al. [45], and apply it to synthetically blurred images obtained by pairwise convolution of a set of test images and test kernels, with and without additional noise. We also apply the method to a few real-life blurred images.

Finally, Chapter 8 provides a summary of the work presented in the thesis and discusses future avenues of exploration. In particular, we note that the kernels estimated using the generalized prior are visibly sparser and cleaner compared to the kernels estimated using the IID prior, suggesting that the generalized prior leads to concrete improvement in the estimated kernels. However, we also note that improvement in the recovered *unblurred* images, as measured by comparison with the true latent image in experiments with synthetic images, is usually nominal in absolute terms for a given non-blind deconvolution method, whereas choosing the generalized prior in the non-blind deconvolution step improves the result much more substantially. Overall, it seems useful to consider using such dependent priors in other problem areas and methods that currently make use of IID priors for natural images.

Chapter 2

Review of non-blind deconvolution methods

2.1 Non-blind deconvolution

The non-blind deconvolution problem has a long history in image processing research, with applications particularly in the field of astronomy. The most well-known of the classical solutions is the Richardson-Lucy algorithm [63, 47] which can be viewed as an example of the expectation-maximization (EM) algorithm. For a detailed review of available approaches and methods, we refer the reader to Starck et al. [73].

For our purposes, non-blind deconvolution is important as the second step in blind deconvolution, where the kernel estimated in the first step is used to recover the unblurred image from the observed blurred image. With this application in mind, we briefly review some basic approaches to non-blind deconvolution without going into too much detail.

2.1.1 Non-blind deconvolution by direct inversion

A direct and crude method to carry out the non-blind deconvolution problem is to invert the convolution operation. Consider the convolution model (with $\otimes$ denoting filtering)

$$\mathbf{b} = \mathbf{k} \otimes \ell \tag{2.1}$$

where $\mathbf{b}$ is the observed blurred image, $\mathbf{k}$ is the known blur kernel, and ℓ is the latent image to be recovered. Recall from Section 1.5 that we can rewrite this as a matrix equation of the form

$$T_{\mathbf{k}} \text{vec}(\ell) = \text{vec}(\mathbf{b}).$$

Since $T_{\mathbf{k}}$ is known in the case of non-blind deconvolution, we can in principle solve for $\text{vec}(\ell)$ provided the system is not under-constrained. A direct solution maybe computationally expensive in practice, and it is more common to invoke the convolution theorem (Theorem 1.5.1) and take advantage of efficient algorithms to compute DFT (this assumed a specific form of the convolution, as discussed earlier). Specifically, if capitalized versions represent Fourier transforms for the matrices in (2.1), we have

$$\mathbf{B} = \mathbf{K}^* \odot \mathbf{L} \implies \mathbf{L} = \mathbf{B} \oslash \mathbf{K}^* \implies \ell = \text{IDFT}(\mathbf{B} \oslash \mathbf{K}^*)$$

where, $\odot$ and $\oslash$ denote elementwise multiplication and division respectively, IDFT stands for inverse discrete Fourier transformation, and $\mathbf{K}^*$ is complex conjugate of $\mathbf{K}$. This is computationally efficient as DFT (and IDFT) can be computed in $\mathcal{O}(n \log n)$ time, where n is the number of elements in the matrix.

Problems with this approach This method completely ignores possible noise in the observed image. Not surprisingly, it is quite unstable in the presence of noise, in which case the estimated unblurred image has the form

$$\hat{\ell} = \text{IDFT}(\mathbf{B} \oslash \mathbf{K}^* - \mathbf{N} \oslash \mathbf{K}^*).$$

where $\mathbf{N}$ is DFT of noise $\mathbf{n}$. If $|\mathbf{K}|$ is small compared to $|\mathbf{N}|$, the effect of noise is amplified making the solution unstable. We demonstrate this effect using the synthetically blurred combinations of test images and test kernels described in Section 1.3. In Figure 2.1 we explore the performance of direct deconvolution algorithm when no noise is present. As we can see the performance of this algorithm is not good even when there is no noise in the image. In pres-



Figure 2.1: Results of non-blind deconvolution using the direct inversion method for all image-kernel combinations shown in Figure 1.6 without additional noise.



Figure 2.2: Results of non-blind deconvolution using the direct inversion method for all image-kernel combinations shown in Figure 1.7 with additional noise ($S = 100$).

ence of noise, the performance deteriorates even further, as seen in Figure 2.2 where Poisson noise is added to the blurred image, although the deterioration is not significant as the noise is relatively low.

2.1.2 Wiener Filter

To counter the problem encountered in presence of noise, a more robust model is proposed by Wiener [84]. Consider the filtering model including noise given by

$$\mathbf{b} = \mathbf{k} \otimes \ell + \mathbf{n}$$

Using Theorem 1.5.1, this is equivalent to (for every DFT frequency ω)

$$B_\omega = \bar{K}_\omega L_\omega + N_\omega$$

or equivalently,

$$\mathbf{B} = \mathbf{K}^* \odot \mathbf{Y} + \mathbf{N}$$

To estimate $\mathbf{L}$ from $\mathbf{B}$, we try to find an inverse filter G such that

$$\hat{\mathbf{L}}_\omega = \mathbf{G}_\omega B_\omega,$$

where $\mathbf{G}_\omega$ minimizes $E(\|\mathbf{L}_\omega - \hat{\mathbf{L}}_\omega\|^2)$. Assuming $S_\omega = E(|\mathbf{L}_\omega|^2)$ and $V_\omega = E(|\mathbf{N}_\omega|^2)$ can be estimated for every ω , we can find an optimal $\mathbf{G}$, known as *Wiener Filter*, which is given by

$$\mathbf{G}_\omega = \frac{\mathbf{K}_\omega}{|\mathbf{K}_\omega|^2 + \frac{V_\omega}{S_\omega}}.$$

S_ω and V_ω should be estimated from natural images and preferable noise models; see, e.g., Russ [66] for details.

An immediate problem of the Wiener filter is that when the signal to noise ratio $\frac{S_\omega}{V_\omega}$ is very large, that is the signal is very strong compared to the noise, the Wiener filter reduces to the

direct DFT filter. Correctly estimating S_ω is critical for good results, and there is no ‘ideal’ choice of S_ω which can be used for all images. Overall, the Wiener filter tries to trade off the restoration by inverse filtering and suppression of noise for those frequencies for which $|\mathbf{K}_\omega|$ is close to zero. The scope of the Wiener filter is limited in the problem of image deconvolution due to poor performance. For the most part, it is used as a basic tool in image denoising, but technically this can be used as a deblurring tool as well.

2.1.3 Constrained Least Square (CLS) filter

While noise amplification is a problem with inverse filter, estimation of power spectrum of “ideal” image is problematic for Wiener filter. As a possible solution, constrained least square estimation is proposed. The idea was first proposed by Hunt [31]. Note that, if a restoration of unknown image ℓ , say $\hat{\ell}$, is good, ideally, $\hat{\ell} \otimes \mathbf{k}$ should be close to $\mathbf{b}$. In case of inverse filter they are made to be exact which leads to the problem of noisy output. It is more reasonable to require $|\mathbf{b}_{ij} - (\mathbf{k} \otimes \ell)_{ij}|^2 = \sigma_{ij}^2$ for some “errors” σ_{ij}^2 , which often assumed to be the same for every (i, j) . There may exist many solution of this optimization problem, hence some regularization is required. Keeping in mind that the inverse filter tends to amplify the knowledge, our aim should be to prefer “smooth” solutions. Consider any high pass filter c , e.g. c can be the Laplacian filter

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

In constrained least square method we try to minimize

$$\|c \otimes \hat{\ell}\|^2$$

subject to the constraint $|\mathbf{b}_{ij} - (\mathbf{k} \otimes \ell)_{ij}|^2 \approx \sigma_{ij}^2$ for every (i, j) . This optimization problem can be written as

$$\min_{\ell} \|\mathbf{b} - \mathbf{k} \otimes \ell\|^2 + \alpha \|c \otimes \ell\|^2. \quad (2.2)$$

It can be directly solved in Fourier domain, giving the deconvolution filter

$$\mathbf{G}_\omega^{CLS} = \frac{\mathbf{K}_\omega}{\bar{\mathbf{K}}_\omega \mathbf{K}_\omega + \alpha \bar{C}_\omega C_\omega}$$

where α is some tuning parameter. One can theoretically calculate the tuning parameter from the constraint of the above optimization problem, but in general it is left as a user defined tuning parameter.

2.1.4 Total Variation based image restoration

There are several other ways to put a constraint on the optimization problem, a popular one is based on *total variation*. In this method we try to optimize

$$\hat{\ell} = \arg \min_{\ell} L(\ell)$$

where $L(\ell) = \|\mathbf{b} - \mathbf{k} \otimes \ell\|^2 + \alpha TV(\ell)$. TV or Total Variation is defined as

$$TV(\ell) = \sum_{i,j} \sqrt{(\delta_h \otimes \ell)_{i,j}^2 + (\delta_v \otimes \ell)_{i,j}^2} \quad (2.3)$$

where, δ_h and δ_v are the operators corresponding to horizontal and vertical first order differences, i.e., $(\delta_h \otimes \ell)_{i,j} = \ell_{i,j+1} - \ell_{i,j}$ and $(\delta_v \otimes \ell)_{i,j} = \ell_{i+1,j} - \ell_{i,j}$. α is some user defined tuning parameter. The optimization is performed using iterative methods.

2.1.5 Richardson Lucy Algorithm

Bayesian methods are widely used in deconvolution problems. The *Richardson-Lucy Algorithm* is one early example of a Bayesian iterative method, although it can be viewed as a likelihood maximization approach as well. The Richardson-Lucy algorithm formulates the deconvolution problem as a missing data problem and solves it using the expectation-maximization (EM)

algorithm [50]. Photon noise is often modeled using Poisson model. Consider the blur model

$$\mathbf{b} = \mathbf{k} \otimes \boldsymbol{\ell} + \mathbf{n}$$

with $\mathbf{k}$ known. We try to frame this as a missing data model and successively apply EM-algorithm to get some solution. Define the latent data

$$\mathbf{z}_{(kl)(ij)} = \text{Contribution of } \boldsymbol{\ell}_{kl} \text{ in } \mathbf{b}_{ij} .$$

The complete data model is given by

$$\mathbf{z}_{(kl)(ij)} \sim \text{Poi}(\boldsymbol{\ell}_{kl} \mathbf{k}_{(k-i+1)(l-j+1)})$$

The goal is to maximize the likelihood based on an IID Poisson model which is given by

$$\ell(\boldsymbol{\ell}|\mathbf{b}, \mathbf{k}) = \prod_{i,j} e^{-(\boldsymbol{\ell} \otimes \mathbf{k})_{ij}} \frac{(\boldsymbol{\ell} \otimes \mathbf{k})_{ij}^{\ell_{ij}}}{\ell_{ij}!} \quad (2.4)$$

In the E-step we compute the expectation of the log likelihood given the current estimates $\boldsymbol{\ell}^{(t)}$ of the parameters, which reduces to computing

$$\hat{\mathbf{z}}_{(kl)(ij)} = \mathbf{b}_{ij} \left(\frac{\boldsymbol{\ell}_{kl}^{(t)} \mathbf{k}_{(k-i+1)(l-j+1)}}{\sum_{k',l'} \boldsymbol{\ell}_{k'l'}^{(t)} \mathbf{k}_{(k'-i+1)(l'-j+1)}} \right)$$

The M-Step then yields

$$\hat{\boldsymbol{\ell}}_{kl}^{(t+1)} = \sum_{i,j} \hat{\mathbf{z}}_{(kl)(ij)}$$

Combining we get,

$$\begin{aligned}
\ell_{kl}^{(t+1)} &= \ell_{kl}^{(t)} \sum_{i,j} \left[b_{ij} \frac{\mathbf{k}_{(k-i+1)(l-j+1)}}{\sum_{k',l'} \ell_{k'l'}^{(t)} \mathbf{k}_{(k'-i+1)(l'-j+1)}} \right] \\
&= \ell_{kl}^{(t)} \sum_{i,j} \left[\frac{b_{ij}}{(\ell^{(t)} \otimes \mathbf{k})_{ij}} \mathbf{k}_{k-i+1, l-j+1} \right] \\
&= \ell_{kl}^{(t)} \cdot \left(\frac{\mathbf{b}}{\ell^{(t)} \otimes \mathbf{k}} \otimes \mathbf{k} \right)_{kl}
\end{aligned}$$

where the division is interpreted as being element-wise. Although a direct computation of these updates are computationally costly, convolutions can be computed efficiently using DFT, making this a very attractive approach. [Figure 2.3](#) demonstrates performance of Richardson Lucy algorithm on blurred images when no noise is present and [Figure 2.4](#) when noise is present.

While the Richardson Lucy algorithm performs quite well in many situations, it is susceptible to boundary artifacts without further modifications, and also tends to result in ringing artifacts in many cases, especially if the kernel is misspecified.

2.2 Bayesian approaches

Given an independent and identically distributed sample $\mathcal{X} = (x_1, x_2, \dots, x_n)$ from a density f_θ , with an unknown parameter $\theta \in \Theta$, the associated likelihood function is

$$\ell(\theta|\mathcal{X}) = \prod_{i=1}^n f_\theta(x_i) \quad (2.5)$$

This likelihood is the principal quantity of the information provided about θ by the sample $\mathcal{X}$. Instead of optimizing the likelihood function directly, in Bayesian approach, the likelihood is modified into a *posterior* distribution on Θ defined by

$$\pi(\theta|\mathcal{X}) = \frac{\ell(\theta|\mathcal{X})\pi(\theta)}{\int_{\phi \in \Theta} \ell(\phi|\mathcal{X})\pi(\phi)d\phi} \quad (2.6)$$

The factor $\pi(\theta)$ is the *prior*, a concept we already discussed. The posterior probability can



Figure 2.3: Results of non-blind deconvolution using the Richardson-Lucy algorithm for all image-kernel combinations shown in Figure 1.6 without additional noise.



Figure 2.4: Results of non-blind deconvolution using the Richardson-Lucy algorithm for all image-kernel combinations shown in Figure 1.6 with additional noise ($S = 100$).

also be thought of as the following

$$\text{Posterior probability} \propto \text{Likelihood} \times \text{Prior probability} \quad (2.7)$$

2.2.1 MAP estimation

In the context of Bayesian statistics maximum a posteriori estimation involves optimizing posterior density with respect to unknown population parameter(s). By optimizing, we refer to finding mode of the posterior density. Basically try to optimize (2.6) with respect to θ as

$$\hat{\theta}_{MAP} = \arg \max_{\theta} \pi(\theta|\mathcal{X}) \quad (2.8)$$

Solving such optimization problem can be hard, as finding the posterior distribution in closed form is often difficult, if at all possible. In the context of image deconvolution, we shall face the problem where the posterior cannot be evaluated explicitly. There are several way to tackle this, some popular ones being *Conjugate Gradient* method, *Expectation Maximization* algorithm (EM Algorithm), and *simulated annealing*. We briefly describe these methods in the remainder of this section.

Conjugate Gradient Method

Suppose we want to solve the syetem of linear equation

$$A\mathbf{x} = \mathbf{b} \quad (2.9)$$

Here A is a known symmetric, positive definite matrix, $\mathbf{b}$ is known and $\mathbf{x}$ is the unknown quantity we want to find. There are two general ways of solving this problem.

Direct solution: Two vectors $\mathbf{u}$ and $\mathbf{v}$ are conjugate (with respect to A) if

$$\mathbf{u}^T A \mathbf{v} = 0$$

Using the fact that A is symmetric and positive definite, write the above in terms of inner product as,

$$\langle \mathbf{u}, \mathbf{v} \rangle_A := \langle A\mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, A^T \mathbf{v} \rangle = \langle \mathbf{u}, A\mathbf{v} \rangle = \mathbf{u}^T A\mathbf{v}$$

Consider a set of mutually conjugate vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$. They form a basis for $\mathbb{R}^n$, so a solution of the linear equation (2.9) can be written as linear combination of these basis vectors.

If $\mathbf{x}_0$ is a solution, write,

$$\mathbf{x}_0 = \sum_{i=1}^n \alpha_i \mathbf{v}_i$$

For any $k \in \{1, 2, \dots, n\}$, write,

$$\begin{aligned} A\mathbf{x}_0 &= \sum_{i=1}^n \alpha_i A\mathbf{v}_i \\ \mathbf{v}_k A\mathbf{x}_0 &= \sum_{i=1}^n \alpha_i \mathbf{v}_k A\mathbf{v}_i \\ \mathbf{v}_k^T \mathbf{b} &= \sum_{i=1}^n \alpha_i \langle \mathbf{v}_k, \mathbf{v}_i \rangle_A \end{aligned}$$

Using, the fact $\langle \mathbf{v}_k, \mathbf{v}_i \rangle_A = 0$ for $k \neq i$, we get

$$\alpha_k = \frac{\langle \mathbf{v}_k, \mathbf{b} \rangle}{\langle \mathbf{v}_k, \mathbf{v}_k \rangle_A}$$

for every $k = 1, 2, \dots, n$. While this method is straightforward, for large n , finding all the conjugate vectors $\mathbf{v}_k$ can be a costly operation. So approximate iterative methods are more popular in such scenario.

Iterative solution Suppose $\mathbf{x}_0$ is some initial guess, without loss of generality $\mathbf{x}_0 = 0$, otherwise, consider the system $A\mathbf{z} = \mathbf{b} - A\mathbf{x}_0$ instead. Starting from $\mathbf{x}_0$ we successively improve the estimate to get the final estimate. If $\mathbf{x}_*$ is the final solution of the system of linear

equation, it can be shown that $\mathbf{x}_*$ is the unique minimizer of the quadratic function

$$\psi(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b}, \quad \mathbf{x} \in \mathbb{R}^n \quad (2.10)$$

Next a gradient descent method is used to find the basis vectors. First, compute the gradient value of (2.10) at starting point $\mathbf{x}_0$ which is $A\mathbf{x}_0 - \mathbf{b}$ and the first basis $\mathbf{v}_0$ is taken to be the negative of the gradient value as $\mathbf{b} - A\mathbf{x}_0$. Now an algorithm similar to Gram-Schmidt orthogonalization is used to find out rest of the orthogonal basis iteratively as follows:

Given $\mathbf{v}_k$ and the solution at k -th stage $\mathbf{x}_k$, define residual

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$$

Basically, $\mathbf{r}_k$ is the negative gradient of ψ at $\mathbf{x} = \mathbf{x}_k$. Moving in the direction of gradient under the orthogonality condition, the solution at $k + 1$ -th stage is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{v}_k$$

where,

$$\mathbf{v}_k = \mathbf{r}_k - \sum_{i < k} \frac{\langle \mathbf{v}_i, \mathbf{r}_k \rangle_A}{\langle \mathbf{v}_i, \mathbf{v}_i \rangle_A} \mathbf{v}_i$$

It is generally not required to go upto n -step. After certain stage norm of $\mathbf{r}_k$ becomes negligible. This is used as a stopping criterion of the algorithm.

Expectation Maximization algorithm

The Expectation Maximization (EM) algorithm is due to Dempster et al. [14] who proposed an algorithm to find out MAP estimate framing the problem as a missing data problem. Suppose, we are given data $\mathcal{X} = (x_1, x_2, \dots, x_n)$ coming from some distribution f_θ where θ is the parameter of interest. Suppose Z is some set of unobserved (latent) data in the model (see

subsection 2.1.5 for an example). The marginal likelihood of $\mathcal{X}$ is given by:

$$\ell(X|\theta) = f_\theta(\mathcal{X}) = \sum_Z f_\theta(\mathcal{X}, Z)$$

Here, $\ell(\theta; \mathcal{X}, Z) = f_\theta(\mathcal{X}, Z)$ is called complete data likelihood. Aim is to optimize this likelihood with respect to θ . In practice, often finding the marginal likelihood becomes tricky as values of Z might become very large. So an iterative approach, consisting of two steps, is proposed.

E Step: At t -th stage, given some current estimate of θ say $\theta^{(t)}$, calculate the expected (with respect to the conditional density of Z given $\mathcal{X}$) value of the (complete data) log likelihood function as

$$Q(\theta|\theta^{(t)}) = E_{Z|\mathcal{X}, \theta^{(t)}}(\log \ell(\theta; \mathcal{X}, Z))$$

M Step: Find the parameter that maximizes Q .

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta|\theta^{(t)})$$

Iterate between these two steps until the change in the likelihood is insignificant. In their work Dempster et al. [14] showed that in each step of the algorithm, the likelihood function should increase. However there is always a chance that the algorithm converge to a local maxima which is a serious problem specially at higher dimension. So the choice of starting point plays an important role in the success of EM algorithm.

In context of blind deconvolution, to find MAP estimate, EM algorithm is one of the most popular methods. We shall discuss some of those methods in the latter sections.

Simulated Annealing

Suppose we have data $\mathcal{X} \subset \Omega$ from some coming density f_θ . Our aim is to find some $\mathcal{Y} \subset \Omega$ which maximize the conditional density $\pi(\mathcal{Y}|\mathcal{X})$. Using Bayes' theorem, we can write,

$$\pi(\mathcal{Y}|\mathcal{X}) \propto \ell(\mathcal{Y}|\mathcal{X}) \cdot \pi(\theta) \tag{2.11}$$

where, $\ell(\mathcal{Y}|\mathcal{X})$ is the likelihood function. In simulated annealing method, define a density

$$\pi_T(\mathcal{Y}|\mathcal{X}) \propto (\ell(\mathcal{Y}|\mathcal{X}) \cdot \pi(\theta))^{1/T} \quad (2.12)$$

In (2.12) $T > 0$, is often referred to as “temperature” of the annealing process. Note that, if $T \rightarrow \infty$, (2.12) is an uniform distribution on the sample space Ω , if $T = 1$, (2.12) corresponds to (2.11), if $T \rightarrow 0$, (2.12) becomes concentrated around the mode of the conditional density (assuming the distribution is unimodal), which is basically the MAP estimate.

Provided we can simulate from $\mathcal{Y}|\mathcal{X}$, letting $T \rightarrow 0$ slowly, we can find some MAP estimate. Unfortunately, such simulation are rarely possible as it is not obvious to simulate from such densities. In general, it is relatively easy to construct the transition matrix Q_T of a discrete time Markov chain, with state space Ω and limit distribution (2.12). In simulated annealing, a time inhomogeneous Markov chain with transition matrix Q_T is run where T is extremely slowly decreased to 0 following a “temperature” schedule.

However, the simulated annealing method, and in general methods based on Markov random fields, are more appropriate in the case of textured images (e.g. satellite images). See e.g. Besag [6] for application of simulated annealing method in finding MAP estimate in context of satellite (textured) images.

2.3 Non-blind deconvolution: modern approaches

Recent research on image restoration has focused on Bayesian based approaches. While we already discussed Richardson Lucy algorithm which is one of the oldest algorithm in that direction, here we shall discuss a few algorithms due to Levin et al. [42] which are mostly based on concepts we have discussed in the last section.

2.3.1 Direct MLE using gradient descent

Consider the standard deconvolution model

$$\mathbf{b} = \mathbf{k} \otimes \boldsymbol{\ell} + \mathbf{n}$$

Assume $\mathbf{n} \sim N(0, \eta^2 I)$, where η^2 is known. Then the negative log likelihood (ignoring constants) is given by

$$-2 \log \ell(\mathbf{b} | \boldsymbol{\ell}, \mathbf{k}) = \|\mathbf{b} - \mathbf{k} \otimes \boldsymbol{\ell}\|^2$$

Aim is to minimize the negative log likelihood as

$$\begin{aligned} \hat{\boldsymbol{\ell}} &= \arg \min_{\boldsymbol{\ell}} \|\mathbf{b} - \mathbf{k} \otimes \boldsymbol{\ell}\|^2 \\ &= \arg \min_{\boldsymbol{\ell}} \|\mathbf{b} - T_k \boldsymbol{\ell}\|^2 \end{aligned}$$

where, T_k is the convolution matrix corresponding to $\mathbf{k}$. Again ignoring terms independent of $\boldsymbol{\ell}$, this minimization problem can alternatively written as

$$\min_{\boldsymbol{\ell}} \boldsymbol{\ell}^T T_k^T T_k \boldsymbol{\ell} - 2 \mathbf{b}^T T_k \boldsymbol{\ell}$$

Note that this has the same form as (2.10) which can be iteratively solved using conjugate gradient method.

Solving the quadratic equation above is equivalent to solving a linear equation

$$A \boldsymbol{\ell} = \mathbf{b}$$

where, $A = T_k^T T_k$ and $\mathbf{b} = T_k^T \mathbf{b}$. If the convolution is taken to be periodic, then T_k is circularly symmetric, and in Fourier domain, this can be written as (for each ω)

$$|K_\omega|^2 \cdot L_\omega = \bar{K}_\omega \cdot B_\omega \implies L_\omega = \frac{B_\omega}{K_\omega}.$$

So this method is basically equivalent to the inverse DFT method, and hence has several drawbacks as discussed earlier, the major ones being amplified noise in the output image. To handle those, we introduce some priors on the image domain.

2.3.2 Prior on latent image

Assume an IID prior on image gradient given by

$$\pi(\boldsymbol{\ell}) \propto \exp \left(-\alpha \sum_i \sum_{k \in \{v, h\}} \rho(\delta_k \otimes \boldsymbol{\ell})_i \right)$$

where, i runs over pixels, δ_k 's are (first order) derivative kernels, $\alpha > 0$ is some constant, $\rho(\cdot)$ is some function, (e.g. $\rho(z) = |z|^2$ corresponds to Gaussian prior). To find MAP estimate, with IID Gaussian assumption of noise, write the posterior density as

$$\pi(\boldsymbol{\ell} \mid \mathbf{b}) \propto \ell(\mathbf{b} \mid \boldsymbol{\ell}) \pi(\boldsymbol{\ell})$$

where, $\ell(\mathbf{b} \mid \boldsymbol{\ell})$ is the likelihood of $\mathbf{b}$ given $\boldsymbol{\ell}$. In a similar fashion, this problem can be written as an optimization problem

$$\max_{\boldsymbol{\ell}} \ell(\mathbf{b} \mid \boldsymbol{\ell}) \pi(\boldsymbol{\ell}) = \arg \min_{\boldsymbol{\ell}} \|\mathbf{b} - T_k \boldsymbol{\ell}\|^2 + \lambda \sum_i \sum_k \rho(T_{\delta_k} \boldsymbol{\ell})_i$$

where, T_{δ_k} is convolution matrix corresponding to δ_k and λ is some (known) constant. If $\rho(z) = |z|^2$, i.e. a Gaussian type likelihood on latent images is assumed, this problem reduces to solving a system of linear equations

$$A\boldsymbol{\ell} = \mathbf{b}$$

where, $A = T_k^T T_k + \lambda \sum_k T_{\delta_k}^T T_{\delta_k}$ and $\mathbf{b} = T_k^T \mathbf{b}$. Gradient descent methods can be used to solve this problem. When $\rho(z) = |z|^\alpha$ where $\alpha \neq 2$, the problem is no longer convex and it can not be optimized in a closed form. Levin et al. [42] have proposed an IRLS method to solve such problems. We postpone a discussion of this method to Chapter 6 where we propose a generalization of this method.

2.4 Super-resolution as a generalization of deconvolution

The goal of super-resolution (SR) is to recover a high resolution (HR) image from one or more low resolution (LR) images. The higher the resolution of an image, the better the details of an image is captured. The image resolution is determined by the image sensors (CCD/CMOS/AMS) and optics. However, capturing high resolution images can be costly in many situations, so it remains an important problem to reconstruct high resolution image from low resolution image(s). In medical images, often several LR images of an organ are captured. Application of SR techniques is quite popular there. Additionally, in remote sensing, reconstruction of HR images from several LR images is common. Another example is images / videos captured in a surveillance camera. In this case, often a single frame of some event is captured, and it is an important problem to recover a HR image from this LR frame. In this work, we shall focus on the problem of reconstructing HR image from a single available image. This problem is known as *Single Image Super Resolution* or simply *Super Resolution*.

Recovering a high resolution image from a single low resolution image is a severely ill-posed problem. Suppose we have a latent image ℓ , from which we obtain an observed image $\mathbf{b}$. In the case of super-resolution, the resulting image is downsampled by an (integral) factor of f by retaining every f th row and column and dropping the rest; the resulting image is denoted by $\mathbf{d}_f^\downarrow(\ell)$. Mathematically,

$$\mathbf{b} = \mathbf{d}_f^\downarrow(\ell) + \epsilon$$

where ϵ represents independent noise that is usually assumed to be Gaussian. However, it is common to include a (small) blur kernel $\mathbf{k}$ to model camera optics before the downsampling operation, and thus the observed $\mathbf{b}$ is modeled as

$$\mathbf{b} = \mathbf{d}_f^\downarrow(\mathbf{k} \otimes \ell) + \epsilon.$$

This process can thus be viewed as a generalization of image convolution.

Although not emphasized in the description above, this general model (incorporating both

convolution and super resolution) can be viewed as a linear process

$$\text{vec}(\mathbf{b}) = A \text{vec}(\boldsymbol{\ell}) + \text{vec}(\boldsymbol{\varepsilon}),$$

where A depends on $\mathbf{k}$ and f . In the case of super resolution, the size of $\mathbf{b}$ is smaller than that of $\boldsymbol{\ell}$ by a factor of f in each dimension, and thus A has many more columns than rows, making the corresponding least squares problem under-constrained. A somewhat subtle but important issue is the effect of $\mathbf{k}$ on the size of $\mathbf{b}$ relative to that of $\boldsymbol{\ell}$. Suppose that each element (pixel) of $\mathbf{k} \otimes \boldsymbol{\ell}$ is a $\mathbf{k}$ -weighted average of a patch of $\boldsymbol{\ell}$ centered around that pixel. For pixels near the boundary of $\mathbf{k} \otimes \boldsymbol{\ell}$ where this patch exceeds the limits of $\boldsymbol{\ell}$, the result cannot be computed because it involves unknown values. If we assume that $\mathbf{k} \otimes \boldsymbol{\ell}$ contains only the elements that can be computed from $\boldsymbol{\ell}$, without assuming any values outside the boundary, then $\mathbf{b}$ will have fewer rows and columns than $\boldsymbol{\ell}$ because some boundary pixels will be dropped.¹ This also leads to A having fewer rows than columns, although the disparity is not as severe as for super resolution. The Bayesian approach employed in this thesis is a common and natural approach to tackle this problem.

2.4.1 Previous work on Single Image Super Resolution

The problem of single image super resolution has been well-studied over the past two decades. One of the easiest and most commonly used approach is to apply interpolation types techniques to low resolution images to increase the resolution. In this regard, bicubic interpolation proposed by Keys [35] is one of the most useful methods. The main advantage of this method was its low computational cost. Further development on cubic interpolations are summarized in Maeland [49] and Parker et al. [58]. A B-spline based interpolation was proposed by Unser et al. [82]. However, such methods often result in blurring of image edges and image details. Schultz and Stevenson [67] proposed a method for nonlinear image expansion which preserves the discontinuities of the original image, producing an expanded image with improved definition.

¹As noted earlier, this is commonly referred to as valid convolution.

One way to improve the image edges and details is to introduce priors on the latent image and solve the problem in a Bayesian setup. An early work using such an approach is by Sun et al. [74] who propose a Bayesian framework to hallucinate a HR image from a LR image based on primal sketch prior. In their approach, they first interpolate a LR image to a low frequency image using bicubic interpolation. Next they use a primal sketch prior which is *learned* from a set of training data. This prior tries to capture the statistical relationship between a HR and a LR image. With this prior they find the MAP estimate of the HR image, which they refer to as the hallucinated HR image. Finally they correct the hallucinated image with a reconstruction correction. The approach of mapping input low resolution image to a target high resolution image has been popular since then. Kim and Kwon [36] used kernel ridge regression. To reduce computational complexity, they proposed a sparse solution combining the ideas of kernel matching pursuit and gradient descent. Sun et al. [75] tried to learn the relation between image gradient profile of high and low resolution images and motivated gradient profile prior which reduced ringing artifacts in image reconstruction. In learning based approaches, Yang et al. [87] proposed a method based on sparse signal recovery. Freeman and Liu [20] used a Markov random field based approach to understand the relation between high and low resolution images. Data driven or dictionary based approaches have become popular in the last few years [29, 15]. With advancement of computing technologies, graphical models have also become increasingly popular.

An important aspect of single image super resolution is the reconstruction constraint, which refers to the idea that downsampling a reconstructed high resolution image should closely match the input low resolution image. Efrat et al. [16] note that the image reconstruction constraint has received relatively little attention in many super resolution methods, and argue that the reconstruction constraint is at least as important as the image prior. They explored the effect of incorrect blur model on SR algorithms and the comparative effect of blur model with image prior, and showed that a combination of a simple image prior, namely IID Gaussian or sparse image gradients, with an accurate blur model gives results as good as state of the art SR algorithms. They empirically examined the sensitivity and specificity of several SR algorithms

to the accuracy of the estimated blur kernel, and demonstrated that the success of SR algorithms heavily depend on the correctness of the estimation of blur kernel; incorrect estimation of the blur kernel results in over-smoothed images or images with artifacts.

Chapter 3

Review of blind deconvolution methods

The problem of image deconvolution is more difficult when we do not have any information on the blur kernel or point spread function. Recall that we generally model image convolution as $\mathbf{b} = \mathbf{k} \otimes \ell + \varepsilon$, where both the blur kernel $\mathbf{k}$ and the latent image ℓ are unknown, and we only observe the blurred image $\mathbf{b}$. Here ε is a noise term whose distribution is generally assumed to be known. Suppose the size of ℓ is $M \times N$ and that of $\mathbf{k}$ is $m \times n$. Then the total number of unknown parameters is $MN + mn$, which is larger than the size of $\mathbf{b}$, generally taken to be the same as that of ℓ . Unless we use some kind of regularization, usually in the form of a Bayesian prior, we cannot hope to get well-behaved estimates of ℓ .

3.1 Overview of previous work

The problem of blind image deconvolution has been of long-standing research interest. Several early approaches attempted to solve the problem by putting constraints (e.g., non-negativity) on the latent image and blur kernel [1, 88, 80]. Katsaggelos and Lay [34] took an expectation-maximization (EM) approach, treating the latent image as missing data with a Gaussian prior. Molina et al. [53] used a Dirichlet distribution to model prior knowledge about the blur kernel, together with smoothness constraints on the latent image. For a review of then current blind deconvolution approaches, see Kundur and Hatzinakos [38] and Kundur and Hatzinakos [39].

Another approach is to assume specific parametric forms on the blur kernel, but make

very mild assumptions about the latent image. For example, a Gaussian blur model may be appropriate for atmospheric blur in satellite images used for remote sensing, or linear motion blur models may be appropriate when the camera is in uniform motion. An interesting approach is that of Hall and Qiu [26], where a parametric model is assumed for the blur kernel, but not for the latent image. The method assumes that the image includes sharp edges (lines of discontinuity in image brightness), and tries to find the kernel that results in edges that are as sharp as possible. Qiu [61] assumes milder conditions on the blur kernel, but still requires it to be circularly symmetric or confined to one specific direction. For the latent image, the method only assumes that it has one or more regions in which line edges are surrounded by uniform backgrounds, which holds, for example, for handwritten text or test patterns used for camera calibration. Hall and Qiu [27] propose a non-iterative estimate of the blur kernel with minimal assumptions on the kernel, but requires that the true image be known, which is again useful for camera calibration using a test image. While significant from a theoretical perspective, none of these methods are applicable to the kind of complex non-symmetric blur patterns seen in amateur photography where the latent image is also completely unknown.

An important landmark in the development of blind deconvolution algorithms for natural images is the adoption of variational Bayes approximation methods, such as that of Likas and Galatsanos [46], who view the variational approximation approach as a generalization of the EM algorithm, using it with Gaussian priors. See Tzikas et al. [81] for an introduction to the variational Bayes approach, and Ruiz et al. [65] for a review of applications in the domain of blind image deconvolution.

Another important ingredient in developing successful blind deconvolution algorithms for photographic images is the use of natural sparse image priors [56, 71, 78] for blind image deconvolution [8, 41]. Fergus et al. [17] achieved particularly remarkable success by combining the variational Bayes approach with a sparse image prior, approximating the sparse prior by a mixture of Gaussians to make the problem computationally tractable. Natural image priors have also played an important role in a related class of problems known as super-resolution; see Efrat et al. [16] and references therein.

Several subsequent approaches, some of which have considered non-uniform motion blur as well, have adopted similar Bayesian approaches [32, 68, 12, 69, 33, 11, 85, 2]. Many of these are examples of what Levin et al. [44] refer to as the $MAP_{\mathbf{x},\mathbf{k}}$ approach, where given priors on parameters $\mathbf{x}$ and $\mathbf{k}$ and a conditional distribution $f(\mathbf{y} \mid \mathbf{x}, \mathbf{k})$ for the observed data $\mathbf{y}$, one tries to maximize the posterior $f(\mathbf{x}, \mathbf{k} \mid \mathbf{y})$ jointly with respect to $\mathbf{x}$ and $\mathbf{k}$ to obtain the joint estimate of $(\mathbf{x}, \mathbf{k})$. However, Levin et al. [44] show that the $MAP_{\mathbf{x},\mathbf{k}}$ approach tends to favour the trivial solution $\mathbf{x} = \mathbf{y}, \mathbf{k} = 1$, and instead suggest the $MAP_{\mathbf{k}}$ approach where the one maximizes $f(\mathbf{k} \mid \mathbf{x}, \mathbf{y})$ to estimate only $\mathbf{k}$, subsequently using a non-blind deconvolution algorithm to estimate $\mathbf{x}$ from $\mathbf{y}$ and the estimated $\mathbf{k}$. In fact, Fergus et al. [17] is one example of this $MAP_{\mathbf{k}}$ approach, and their method has been subsequently simplified by Levin et al. [45].

In this thesis, we focus on the natural image prior used in these methods, and our main contribution is to motivate a generalization and adapt it to the method of Levin et al. [45]. As such, no further work on blind image deconvolution is needed to understand this work. However, blind image deconvolution is an active research area and many subsequent advances have been made, particularly for blind deconvolution of images with non-uniform blur [25, 83, 86]. Interesting alternative approaches have been proposed by Sun et al. [77] based on an edge-based “patch prior” and Michaeli and Irani [51] based on patch recurrence. More recent work has focused on the use of deep learning techniques [76, 9, 24] and have included some kernel-free methods [55, 62]. Lee et al. [40] have used robust statistical techniques for blind deconvolution of astronomical images.

In the remainder of this chapter, we briefly discuss some particular topics that will be relevant for our work. Section 3.2 discusses some methods that attempt to recover $\mathbf{k}$ from an estimate of the magnitude of its Fourier transform $|\mathbf{K}_\omega|$ (which has an obvious estimate). Section 3.3 gives an outline of the general Bayesian *maximum a posteriori* approach adopted by several image deconvolution methods. Finally, Section 3.4 introduces the basic ideas of the variational Bayes approach which has also been critical in the implementation of many image deconvolution methods. We postpone a fuller discussion of the method of Levin et al. [45], which we use as a basis for our implementation, to the next Chapter.

3.2 Phase retrieval

The blind image deconvolution problem and the usefulness of an image prior can be easily demonstrated in the Fourier domain. Recall that by the Convolution theorem, the image domain model (where $\otimes$ denotes filtering, not convolution)

$$\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n}$$

can be rewritten in the Fourier domain as

$$\mathbf{Y}_\omega = \bar{\mathbf{K}}_\omega \cdot \mathbf{X}_\omega + \mathbf{N}_\omega, \quad (3.1)$$

where the model holds pointwise for each frequency pair ω . Figure 3.1 shows log-transformed modulus of the Fourier coefficients obtained from the 64 synthetically blurred test images described in Chapter 1. It is obvious from these plots that the modulus of the DFT coefficients carry information about $|\mathbf{K}_\omega|$ and hence $\mathbf{k}$, although unfortunately information about the phase (argument) of $\mathbf{K}_\omega$ is not contained in $\mathbf{Y}_\omega$. We discuss the implications of this formulation in further detail in Chapter 5, but note here that the ω -wise maximum likelihood estimate of $|\mathbf{K}_\omega|$ is easily seen to be given by

$$|\hat{\mathbf{K}}_\omega| = \frac{\sqrt{\max(0, |\mathbf{Y}_\omega|^2 - \eta^2)}}{\sigma}, \quad (3.2)$$

where σ^2 is the variance of $\mathbf{X}_\omega$ and η^2 is the variance of $\mathbf{N}_\omega$. One possible avenue to estimate $\mathbf{k}$ is thus to find a solution for which the Fourier magnitude matches this maximum likelihood estimate.

In principle, it is possible to recover $\mathbf{k}$ from only the magnitude of its Fourier transform since $\mathbf{k}$ is positive valued [28]. However, this relationship is highly non-linear, making the reconstruction difficult. One of the first attempts in recovering phase from a given magnitude of a Fourier transform is by Gerchberg and Saxton [21] where through an iterative algorithm phase is retrieved from a pair of magnitudes. Their algorithm can be rewritten to use in case of recovering phase from a single set of amplitudes. The modified algorithm consists of the

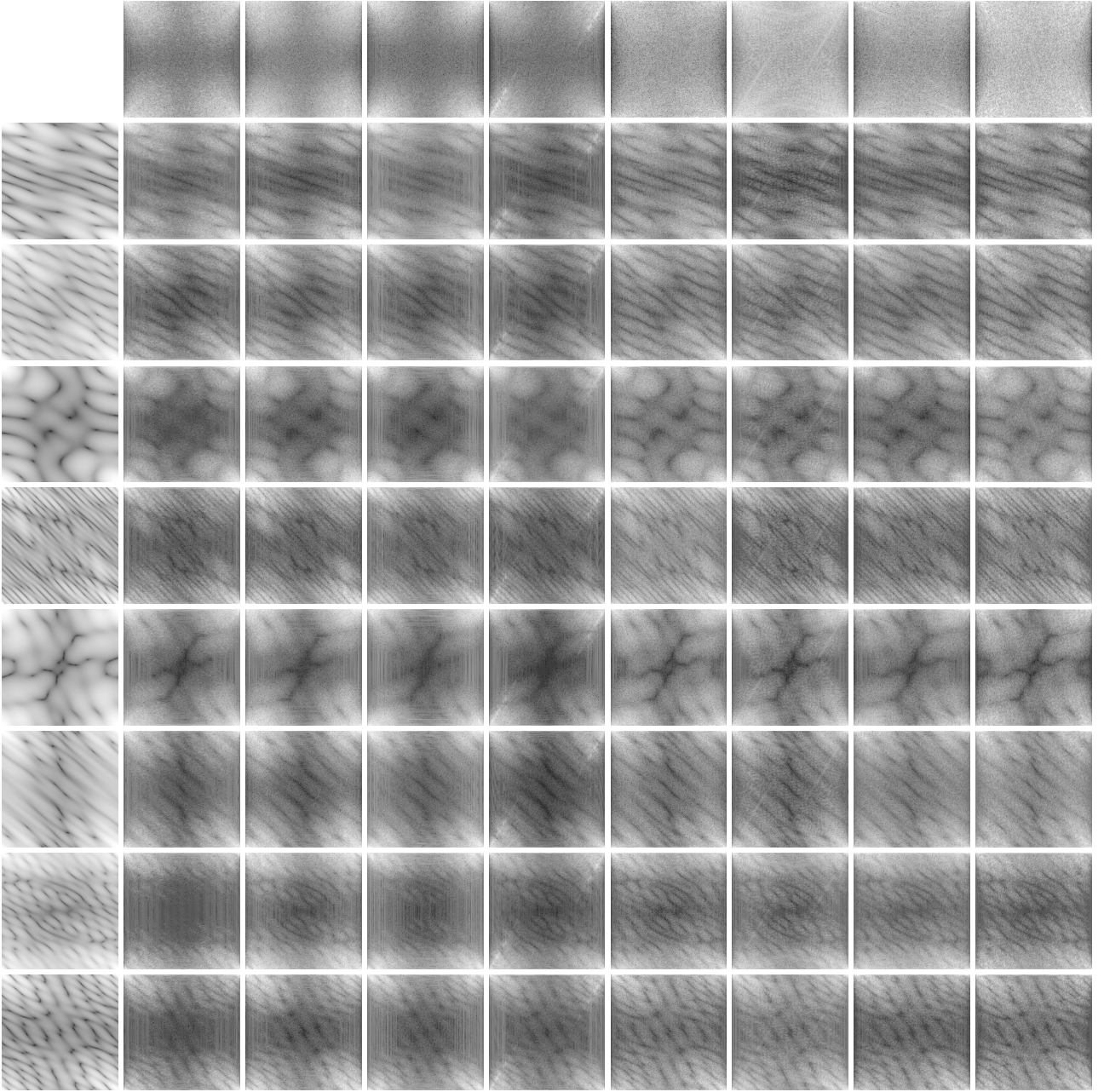


Figure 3.1: Plot of modulus of Fourier coefficients corresponding to synthetically blurred exact test images in Figure 1.6. The x and y coordinates represent frequency (from 0 to 2π); intensity (black = low, white = high) represents log-transformed modulus of the corresponding Fourier coefficient. The top row corresponds to DFT of the horizontal gradients of the eight unblurred test images. The left row corresponds to DFT of the eight test kernels, after padding with zeros. The remaining plots correspond to DFT of the horizontal gradients of the blurred images obtained by convolving the corresponding test image and the test kernel.

following steps:

1. Initialization: Begin with an initial estimate of the image domain function to be estimated.
2. Get Fourier transformation of the current (image domain) estimate of the function to be estimated.
3. Replace the modulus of the Fourier transform obtained in the previous step with the available magnitude of the object. Use this as a new estimate of the Fourier transform of the object.
4. Obtain the inverse Fourier transform of the object to get an image domain estimate of the object.
5. Modify the image domain estimate to satisfy image domain constraints such as positivity.
6. Use this estimate as the new current estimate in step 2.
7. The iteration continues until the computed Fourier transform satisfies all image domain constraints, or the mismatch error is below some tolerance level.

If $|F|$ is the magnitude of the Fourier transform under consideration and g is some initial estimate of it, then Figure 3.2 gives a block diagram explaining the modified Gerchberg-Saxton algorithm.

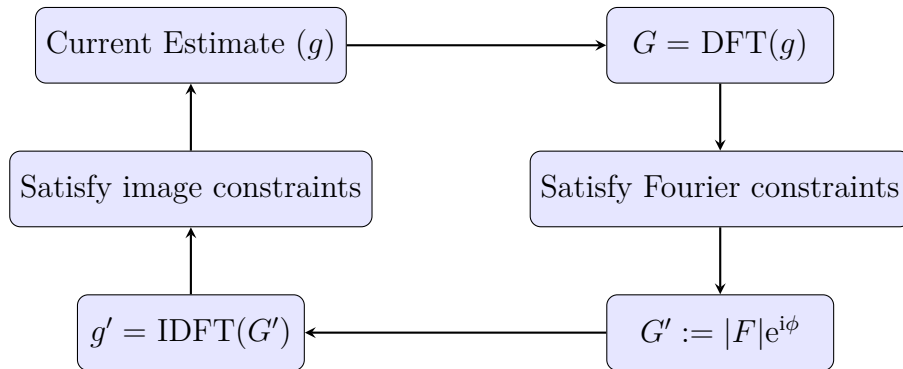


Figure 3.2: Sketch of Gerchberg-Saxton algorithm

A major problem of this algorithm is its convergence rate. It is generally seen that the convergence rate of the algorithm is very fast in the first few steps, but then slow to get to a

good final result. Some modifications have been proposed to get better convergence rates; one such well-known modification is due to Fienup [18] who proposes three different approaches to tackle the problem of slow convergence.

The Gerchberg-Saxton error-reduction algorithm described above is a greedy algorithm, as at each step, the current estimate is modified to satisfy all the image domain constraints of the problem. Enforcing such a strict condition on the estimate in each step is one of the main reasons for slow convergence. Instead of enforcing the constraints in each step, Fienup [18] suggests moving towards the direction of the strict solution at a certain rate. If g_k is the current estimate of f , it is seen that a small change of the input results in a change of the output in the same general direction as the change in the input. More precisely, for a small change of the input the expected value of the corresponding change of the output is a constant α times the change of the input. However additional non-linear terms which will appear in the output due to some linear change in the input are difficult to correctly measure. Nevertheless, we can hope to push the output in the general desired direction. So if a change of $\Delta g(x)$ is desired in the output, a change in the input should be $\beta \Delta g(x)$. In our case,

$$\Delta g_k(x) = \begin{cases} 0, & \text{if } x \in \Gamma, \\ -g'_k(x) & \text{otherwise,} \end{cases}$$

where Γ is the set of values for which the object constraints are satisfied. Therefore, given k -th stage input g_k , a possible $(k + 1)$ -th stage input can be

$$\begin{aligned} g_{k+1}(x) &= g_k(x) + \beta \Delta g_k(x) \\ &= \begin{cases} g_k(x), & \text{if } x \in \Gamma \\ g_k(x) - \beta g'_k(x) & \text{otherwise.} \end{cases} \end{aligned}$$

So in this method, while the first three steps are exactly the same as the error reduction algorithm, the updation step is modified. Fienup [18] calls this as *input-output algorithm*. Note that in this algorithm if g' satisfies the Fourier domain constraints, and if we use g' as an

input, it remains unaffected as it passes through the first three steps and it becomes the output itself. From this angle, one can modify step 5 as follows:

$$\begin{aligned} g_{k+1}(x) &= g'_k(x) + \beta \Delta g_k(x) \\ &= \begin{cases} g'_k(x), & \text{if } x \in \Gamma \\ g'_k(x)(1 - \beta) & \text{otherwise} \end{cases} \end{aligned}$$

This algorithm is often referred to as *output-output algorithm*. Note that, if $\beta = 1$, this reduces to error reduction algorithm. Another method of choosing the next input is somewhat a hybrid of the input-output and output-output algorithm which is called *hybrid algorithm* where the next input is chosen as:

$$g_{k+1}(x) = \begin{cases} g'_k(x), & \text{if } x \in \Gamma \\ g_k(x) - \beta g'_k(x), & \text{otherwise} \end{cases}$$

It is to be noted that while we attempt to recover phase of Fourier transform of the blur kernel $\mathbf{k}$, we must keep in mind that $\mathbf{k}$ is not uniquely defined by $|\mathbf{K}|$ as it is invariant to location shifts of $\mathbf{k}$ as well as “flipping”. While location shift is not an important issue, flipping is more important. In practice, one simply tries both flipped and non-flipped version to choose manually.

However performance of algorithms that attempt to recover phase from Fourier magnitude are not particularly good, and more successful approaches instead take a more formal Bayesian approach leading to MAP-type estimators, which we briefly describe in the next section.

3.3 MAP estimation

Recall the convolution / filtering model

$$\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n}$$

where $\mathbf{y}$ is observed blurred image (gradient), $\mathbf{k}$ is unknown blur kernel or point spread function, $\mathbf{x}$ is unknown “true” image (gradient), and $\mathbf{n}$ is an error term. In *maximum a posteriori* (MAP) estimation, we try to find the mode of the posterior distribution with respect to unknown quantities of the model. For our specific model, $\mathbf{x}$ and $\mathbf{k}$ are unknown, so the MAP estimation problem is to find

$$\arg \max_{\mathbf{x}, \mathbf{k}} \log f(\mathbf{x}, \mathbf{k} \mid \mathbf{y})$$

where $f(\mathbf{x}, \mathbf{k} \mid \mathbf{y})$ is the posterior distribution of $\mathbf{x}$ and $\mathbf{k}$ given $\mathbf{y}$. Ignoring the prior on $\mathbf{k}$, i.e., treating it as a fixed parameter, the posterior can be written as

$$f(\mathbf{x}, \mathbf{k} \mid \mathbf{y}) \propto \ell(\mathbf{y} \mid \mathbf{x}, \mathbf{k}) \pi(\mathbf{x})$$

Here, $\ell(\mathbf{y} \mid \mathbf{x}, \mathbf{k})$ is the conditional likelihood of the observed image gradient $\mathbf{y}$ and $\pi(\cdot)$ is the prior on latent image gradient $\mathbf{x}$. It is standard to assume that the errors $\mathbf{n}$ are independent Gaussian white noise, i.e.,

$$\mathbf{n} \sim N(0, \eta^2 \mathbf{I}),$$

where the variance η^2 is assumed to be known. The likelihood term $\ell(\mathbf{y} \mid \mathbf{x}, \mathbf{k})$, which is the data fitting term, can thus be written as,

$$\ell(\mathbf{y} \mid \mathbf{x}, \mathbf{k}) \propto \exp \left\{ -\frac{1}{2\eta^2} \|\mathbf{y} - \mathbf{k} \otimes \mathbf{x}\|^2 \right\}$$

Ignoring constants,

$$\log \ell(\mathbf{y} \mid \mathbf{x}, \mathbf{k}) = -\lambda \|\mathbf{y} - \mathbf{k} \otimes \mathbf{x}\|^2$$

with $\lambda = -\frac{1}{2\eta^2} > 0$. Clearly, ℓ is maximized when the distance between data $\mathbf{y}$ and $\mathbf{k} \otimes \mathbf{x}$ is minimized. The prior $\pi(\mathbf{x})$ favours natural images through a sparse (e.g., hyper-Laplacian) structure of the gradient distribution. Direct MAP estimation, often called the MAP _{$\mathbf{x}, \mathbf{k}$} approach, seeks $(\hat{\mathbf{x}}, \hat{\mathbf{k}})$ minimizing $f(\mathbf{x}, \mathbf{k} \mid \mathbf{y})$ where

$$f(\mathbf{x}, \mathbf{k} \mid \mathbf{y}) \propto \ell(\mathbf{y} \mid \mathbf{x}, \mathbf{k}) \pi(\mathbf{x}).$$

Ignoring constants and taking logarithm, the goal is thus to solve the optimization problem

$$\min_{\mathbf{x}, \mathbf{k}} \lambda \|\mathbf{y} - \mathbf{k} \otimes \mathbf{x}\|^2 + \log \pi(\mathbf{x})$$

Of course, the feasibility of solving the problem depends on the nature of $\pi(\mathbf{x})$.

Levin et al. [44] point out that this $\text{MAP}_{\mathbf{x}, \mathbf{k}}$ approach tends to favour the no-blur explanation where $\mathbf{x}$ is estimated as the observed blurred image $\mathbf{y}$. The main issue with this approach is that the number of unknown parameters is too large. Intuitively, if the size of $\mathbf{y}$ (and thus $\mathbf{x}$) is increased while the size of the kernel $\mathbf{k}$ remains constant, we expect to get better results. However, the $\text{MAP}_{\mathbf{x}, \mathbf{k}}$ approach tries to simultaneously estimate both $\mathbf{x}$ and $\mathbf{k}$, and thus loses the advantage of having more data. They instead suggest a two-step approach called $\text{MAP}_{\mathbf{k}}$. In the first step, we try to estimate $\mathbf{k}$, optimizing the conditional distribution $f(\mathbf{k} \mid \mathbf{y})$ which is obtained by marginalizing $f(\mathbf{k}, \mathbf{x} \mid \mathbf{y})$ over $\mathbf{x}$, i.e.,

$$f(\mathbf{k} \mid \mathbf{y}) = \int f(\mathbf{k}, \mathbf{x} \mid \mathbf{y}) d\mathbf{x} \quad (3.3)$$

In the second step, armed with an estimate of $\mathbf{k}$, a non-blind deconvolution algorithm is applied with the $\mathbf{k}$ found in the first step.

The main difficulty in this method is computing the integral in (3.3), which is often computationally intractable. For a mixture of Gaussian approximation to the hyper-Laplacian prior they chose for $\pi(\mathbf{x})$, Fergus et al. [17] proposed a method based on a variational Bayes approximation. This approach has been used successfully by others as well. In the next section, we briefly introduce the basic ideas of the variational Bayes method following Miskin and MacKay [52], and refer the reader to Fergus et al. [17] and Levin et al. [44] for details specific to the sparse prior used in blind image deconvolution.

3.4 The variational Bayes approach

Consider a missing data problem where we observe data $\mathcal{X} = \{x_1, \dots, x_n\}$. The set of unobserved latent variable is denoted by $Z = \{Z_1, \dots, Z_m\}$. As in the EM algorithm setup, the form of the joint distribution of $\mathcal{X}$ and Z , denoted by $P(\mathcal{X}, Z)$, is available, but we are interested in the posterior distribution $P(\mathcal{X} | Z)$. The eventual aim is to optimize the posterior distribution. However, in reality, often finding a closed form of this posterior distribution can be difficult, and optimizing it can be even more challenging. So finding a “good” approximation of the posterior distribution using another distribution with a simpler form is useful. In the variational Bayes method, we aim to approximate this posterior distribution $P(\mathcal{X} | Z)$ by a *variational distribution* $Q_\theta(Z)$, where $Q_\theta(Z)$ (or simply $Q(Z)$) belongs to a simpler family of distributions, usually completely determined by θ .

To estimate the parameter θ , or the distribution Q in general, so that $Q_\theta(Z)$ can approximate $P(\mathcal{X} | Z)$ well, we use the Kullback-Leibler divergence to measure the distance between P and Q as

$$\begin{aligned} D_{KL}(Q||P) &= \int_Z Q(Z) \log \frac{Q(Z)}{P(Z|\mathcal{X})} dZ \\ &= \log P(\mathcal{X}) + \int_Z Q(Z) \log \frac{Q(Z)}{P(Z, \mathcal{X})} dZ. \end{aligned} \quad (3.4)$$

Note that the first term in the RHS of (3.4) is independent of the choice of Q . So if we maximize

$$\mathcal{L}(Q) = \int_Z Q(Z) \log \frac{P(Z, \mathcal{X})}{Q(Z)} dZ$$

with appropriate choice of Q , the KL divergence $D_{KL}(Q||P)$ will be minimized.

If we allow any possible form of Q , minimum value will be achieved when Q equals the posterior distribution $P(Z|\mathcal{X})$. However, in practice the model is such that working with the true posterior distribution is intractable, so we consider a restricted family of distributions Q and seek the member in it for which KL divergence is minimized. The restricted family should be such that it consists of distributions that are tractable, while at the same time good enough

to provide a decent approximation of the true posterior distribution.

A common approach is to use factorized distributions to approximate the posterior. By factorization, we mean the following: Suppose we partition the elements of Z as Z_i for $i = 1, 2, \dots, S$ and assume Q factorizes with respect to these groups as

$$Q(Z|\mathcal{X}) = \prod_{i=1}^S q_i(Z_i|\mathcal{X}). \quad (3.5)$$

No further assumption on the functional form of the distributions q_i is needed. Now among all the distributions having the form (3.5), we seek the one for which $\mathcal{L}(Q)$ is largest. Note that

$$\begin{aligned} \mathcal{L}(Q) &= \int_Z \left(\prod_i q_i \right) \left\{ \log P(\mathcal{X}, Z) - \sum_i \log q_i \right\} dZ \\ &= \int_{Z_j} q_j \left(\int_{Z_{i \neq j}} \log P(\mathcal{X}, Z) \prod_{i \neq j} q_i dZ_i \right) dZ_j - \int_{Z_j} q_j \log q_j dZ_j + \text{const.} \\ &= \int q_j \log \tilde{P}(\mathcal{X}, Z_j) dZ_j - \int q_j \log q_j dZ_j + \text{const.} \end{aligned} \quad (3.6)$$

where $\tilde{P}$ is a probability distribution that satisfies

$$\log \tilde{P}(\mathcal{X}, Z_j) = \mathbb{E}_{i \neq j}(\log P(\mathcal{X}, Z)) + \text{const.}$$

Here,

$$\mathbb{E}_{i \neq j}(\log P(\mathcal{X}, Z)) = \int \log P(\mathcal{X}, Z) \prod_{i \neq j} q_i dZ_i.$$

For a given j , our aim is to maximize $\mathcal{L}(q)$ in (3.6). Note that (3.6) is nothing but the negative Kullback-Leibler divergence between $q_j(Z_j)$ and $\tilde{P}(X, Z_j)$. Thus, maximizing $\mathcal{L}(Q)$ is equivalent of minimizing the KL divergence. As argued earlier, the KL divergence is minimized when $q^{(opt)} = \tilde{P}$, that is, when $q^{(opt)}$ is given by

$$q_j^{(opt)}(Z_j|\mathcal{X}) \propto e^{E_{i \neq j}[\log p(Z, \mathcal{X})]} \quad (3.7)$$

In the remainder of this section, we illustrate the variational Bayes method using some common

examples [48, 7].

Univariate Gaussian: In this example we illustrate the factorized variational approximation using a Gaussian distribution over a single variable x . Our goal is to infer the posterior distribution for the mean μ and precision (inverse variance) τ , given a data set $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ which are assumed to be drawn independently from a Gaussian density with mean μ and variance $\frac{1}{\tau}$. The likelihood function is given by

$$p(\mathcal{X}|\mu, \tau) = \left(\frac{\tau}{2\pi}\right)^{N/2} \exp\left\{-\frac{\tau}{2} \sum_{n=1}^N (x_n - \mu)^2\right\}$$

For computational simplicity we consider conjugate priors on μ and τ which are given by

$$\begin{aligned} p(\mu|\tau) &= N(\mu|\mu_0, (\lambda_0\tau)^{-1}) \\ p(\tau) &= \text{Gamma}(\tau|a_0, b_0) \end{aligned}$$

The Gamma density with parameters a and b is given by

$$\text{Gamma}(\tau|a, b) = \frac{1}{\Gamma(a)} b^a \tau^{a-1} \exp(-b\tau)$$

In this case, computation of the posterior distribution directly is straightforward and the exact structure of the posterior distribution can be easily found. However for the sake of illustration, we explore the variational approximation to the posterior distribution as

$$q(\mu, \tau) = q_\mu(\mu)q_\tau(\tau)$$

Using (3.7), the optimal q_μ and q_τ can be obtained as

$$\begin{aligned} \log q_\mu^*(\mu) &= \mathbb{E}_\tau(\log p(\mathcal{X}|\mu, \tau) + \log p(\mu|\tau)) + \text{const.} \\ &= -\frac{\mathbb{E}(\tau)}{2} \left(\lambda_0(\mu - \mu_0)^2 + \sum_{n=1}^N (x_n - \mu)^2 \right) + \text{const.} \end{aligned}$$

Simplifying, it can be seen that $q_\mu(\mu)$ is a Gaussian with mean μ^* and precision τ^* given by

$$\begin{aligned}\mu^* &= \frac{\lambda_0 \mu_0 + N \bar{x}}{\lambda_0 + N} \\ \tau^* &= (\lambda_0 + N) \mathbb{E}_\tau(\tau)\end{aligned}$$

Similarly, the optimal solution for the factor $q_\tau(\tau)$ is given by

$$\begin{aligned}\log q_\tau^*(\tau) &= \mathbb{E}_\mu(\log p(\mathcal{X}|\mu, \tau) + \log p(\mu|\tau)) + \log p(\tau) + \text{const.} \\ &= (a_0 - 1) \log \tau - b_0 \tau + \frac{1}{2} \log \tau + \frac{N}{2} \log \tau \\ &\quad - \frac{\tau}{2} \mathbb{E}_\mu \left(\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0 (\mu - \mu_0)^2 \right) + \text{const.}\end{aligned}$$

Therefore, q_τ^* is a Gamma distribution with parameters

$$\begin{aligned}a^* &= a_0 + \frac{N + 1}{2} \\ b^* &= b_0 + \frac{1}{2} \mathbb{E} \left(\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0 (\mu - \mu_0)^2 \right)\end{aligned}$$

Univariate Gaussian with sub-optimal choices Suppose we restrict our choice of approximate posterior q (Miskin and MacKay [52]) as

$$Q(\mu, \xi) = Q(\mu)Q(\xi) = N(\mu|\hat{\mu}, \tilde{\mu})N(\xi|\hat{\xi}, \tilde{\xi})$$

where $\xi = \log \tau$ as $\xi > 0$ and a Gaussian assumption on it is not legitimate. Consider priors on μ and τ as

$$p_\mu(\mu) = N(\mu|0, s_0) \tag{3.8}$$

$$p_\tau(\tau) = \text{Gamma}(\tau|a_0, b_0) \tag{3.9}$$

A transformation $\xi = \log \tau$ gives the prior on ξ as

$$p_\xi(\xi) = p_\tau(\exp(\xi)) \exp(\xi) I_{(-\infty, +\infty)}(\xi) \quad (3.10)$$

So we try to optimize,

$$\begin{aligned} D_{KL}(Q||P) &= \mathbb{E}_Q \left(\log \left(\frac{Q(\mu)}{p_\mu(\mu)} \right) + \log \left(\frac{Q(\xi)}{p_\xi(\xi)} \right) - \log P(\mathcal{X}|\mu, \xi) \right) \\ &= \frac{1}{2} \left(\log \tilde{\mu} - 1 - \log s_0 + s_0 \left(\hat{\mu} + \frac{1}{\tilde{\mu}} \right) + \right. \\ &\quad \frac{1}{2} (\log \tilde{\xi} - 1) + \log \Gamma(b_0) - b_0 \log a_0 - b_0 \hat{\xi} + a_0 \exp \left(\hat{\xi} + \frac{1}{2\tilde{\xi}} \right) + \\ &\quad \left. - \sum_{i=1}^N \left(-\frac{1}{2} \log(2\pi) + \frac{\hat{\xi}}{2} - \frac{1}{2} \exp \left(\hat{\xi} + \frac{1}{2\tilde{\xi}} \right) \left((x_i - \hat{\mu})^2 + \frac{1}{\tilde{\mu}} \right) \right) \right) \end{aligned} \quad (3.11)$$

Minimizing with respect to the parameters, we get

$$\tilde{\mu} = s_0 + N \exp \left(\hat{\xi} + \frac{1}{2\tilde{\xi}} \right) \quad (3.12)$$

$$\hat{\mu} = \frac{1}{\tilde{\mu}} \exp \left(\hat{\xi} + \frac{1}{2\tilde{\xi}} \right) \sum_{i=1}^N x_i \quad (3.13)$$

$$\tilde{\xi} = b_0 + \frac{N}{2} \quad (3.14)$$

$$\hat{\xi} = \log \tilde{\xi} - \frac{1}{2\tilde{\xi}} - \log \left(a_0 + \frac{1}{2} \left((x_i - \hat{\mu})^2 + \frac{1}{\tilde{\mu}} \right) \right). \quad (3.15)$$

In the next chapter, we describe in detail the specific estimation method described by Levin et al. [45], which we use as a benchmark for our generalization.

Chapter 4

MAP_k Blind deconvolution using IID prior

In the previous chapter, we have outlined the overall Bayesian approach that has been used with considerable success in many blind image deconvolution methods. Needless to say, obtaining good results depends on formulating suitable priors and efficiently solving the resulting maximization problem. For our work, we focus on the approach used by Levin et al. [45], which we describe in this chapter. Much of it follows and repeats their presentation, but is included here for ease of exposition. Our own contribution is to propose a generalization of the prior, which we motivate in Chapter 5 and incorporate in the estimation process for blind deconvolution in Chapter 7.

4.1 Problem formulation

Consider the previously introduced convolution / filtering model

$$\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n} \tag{4.1}$$

Here $\mathbf{x}$ and $\mathbf{y}$ are taken to be image gradients (horizontal or vertical) of the true latent image and the observed blurred image respectively. As noted in Chapter 1, image gradients have

remarkable structural similarity across a variety of natural images, which leads us to assuming a prior marginal distribution on components of $\mathbf{x}$ given by either the Gaussian model

$$\psi(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (4.2)$$

or the sparse mixture of Gaussian model

$$\psi(x) = \sum_{j=1}^J p_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{x^2}{2\sigma_j^2}\right), \quad \sum_{j=1}^J p_j = 1. \quad (4.3)$$

The individual components $\mathbf{x}_i$ are assumed to be independent, and so the prior on the full matrix of image gradients $\mathbf{x}$ is

$$\pi(\mathbf{x}) = \prod_i \psi(\mathbf{x}_i).$$

The other component required to model the conditional distribution of $\mathbf{y}$ given $\mathbf{x}$ and $\mathbf{k}$ is the distribution of the noises $\mathbf{n}$. The components $\mathbf{n}_i$ are assumed to be independent Gaussians with mean 0 and variance η^2 . In other words, the model is specified as

$$\begin{aligned} \mathbf{x} &\sim \pi(\mathbf{x}) \\ \mathbf{y} | \mathbf{x}, \mathbf{k} &\sim p(\mathbf{y} | \mathbf{x}, \mathbf{k}) \end{aligned}$$

where $p(\mathbf{y} | \mathbf{x}, \mathbf{k})$ is the density of $N(\mathbf{k} \otimes \mathbf{x}, \eta^2 \mathbf{I})$. Here, and in what follows, we abuse notation somewhat by treating the $M \times N$ matrices $\mathbf{y}$ and $\mathbf{k} \otimes \mathbf{x}$ as vectors of length MN whenever convenient.

4.2 The EM approach

Levin et al. [45] adopt an iterative expectation-maximization (EM) approach [14, 50] to estimate $\mathbf{k}$, viewing $\mathbf{x}$ as missing data. The complete data likelihood is given by

$$p(\mathbf{x}, \mathbf{y} | \mathbf{k}) = p(\mathbf{y} | \mathbf{x}, \mathbf{k}) \pi(\mathbf{x}).$$

E-step Following the general prescription of the EM algorithm, the E-step requires computation of

$$Q(\mathbf{k} | \mathbf{k}^{(t)}) = \mathbb{E}_{\mathbf{k}^{(t)}} (\log p(\mathbf{x}, \mathbf{y} | \mathbf{k}) | \mathbf{y}) \quad (4.4)$$

$$= \mathbb{E}_{\mathbf{k}^{(t)}} (\log p(\mathbf{y} | \mathbf{x}, \mathbf{k}) + \log \pi(\mathbf{x}) | \mathbf{y}), \quad (4.5)$$

where $\mathbf{k}^{(t)}$ is the current estimate of $\mathbf{k}$. For this specific problem, ignoring constants, we have

$$\log p(\mathbf{y} | \mathbf{x}, \mathbf{k}) + \log \pi(\mathbf{x}) \approx - \sum_i \left(\frac{(\mathbf{y} - \mathbf{k} \otimes \mathbf{x})_i^2}{\eta^2} + \log \pi(\mathbf{x}_i) \right),$$

where i runs over pixels. It is useful to recall note here that the convolution operation $\mathbf{k} \otimes \mathbf{x}$ is linear (in both $\mathbf{k}$ and $\mathbf{x}$) and can be expressed as a matrix operation $A_{\mathbf{x}} \mathbf{k}$, where $\mathbf{k}$ is viewed as an $mn \times 1$ vector and $A_{\mathbf{x}}$ is an $MN \times mn$ matrix whose entries are elements of $\mathbf{x}$. This lets us write

$$\sum_i \left(\frac{(\mathbf{y} - \mathbf{k} \otimes \mathbf{x})_i^2}{\eta^2} + \log \pi(\mathbf{x}_i) \right) = \frac{\mathbf{y}^T \mathbf{y} + \mathbf{k}^T A_{\mathbf{x}}^T A_{\mathbf{x}} \mathbf{k} - 2 \mathbf{y}^T A_{\mathbf{x}} \mathbf{k}}{\eta^2} + \sum_i \log \pi(\mathbf{x}_i).$$

Ignoring terms independent of $\mathbf{k}$, which are irrelevant in the subsequent maximization step with respect to $\mathbf{k}$, (4.4) reduces to

$$\eta^2 Q(\mathbf{k} | \mathbf{k}^{(t)}) \approx - \mathbb{E}_{\mathbf{k}^{(t)}} (\mathbf{k}^T A_{\mathbf{x}}^T A_{\mathbf{x}} \mathbf{k} - 2 \mathbf{y}^T A_{\mathbf{x}} \mathbf{k} | \mathbf{y}) \quad (4.6)$$

$$= - \mathbf{k}^T \mathbb{E}_{\mathbf{k}^{(t)}} (A_{\mathbf{x}}^T A_{\mathbf{x}} | \mathbf{y}) \mathbf{k} + 2 \mathbf{y}^T \mathbb{E}_{\mathbf{k}^{(t)}} (A_{\mathbf{x}} | \mathbf{y}) \mathbf{k}. \quad (4.7)$$

$\mathbb{E}_{\mathbf{k}^{(t)}} (A_{\mathbf{x}}^T A_{\mathbf{x}} | \mathbf{y})$ and $\mathbb{E}_{\mathbf{k}^{(t)}} (A_{\mathbf{x}} | \mathbf{y})$ can be evaluated (see below) in terms of the mean and variance of $\mathbf{x}$ given $\mathbf{y}$ under the current estimate of $\mathbf{k}$,

$$\boldsymbol{\mu} = \mathbb{E}_{\mathbf{k}^{(t)}} (\mathbf{x} | \mathbf{y}), \quad \boldsymbol{\Sigma} = \text{Var}_{\mathbf{k}^{(t)}} (\mathbf{x} | \mathbf{y}). \quad (4.8)$$

Computing these quantities is relatively easy for the Gaussian model (4.2), and difficult for the MOG model (4.3) where an approximate solution is obtained using a variational Bayes

approach.

M-step Once $Q(\mathbf{k} | \mathbf{k}^{(t)})$ is obtained, the M-step requires maximizing (4.7) with respect to $\mathbf{k}$, subject to the constraint that elements of $\mathbf{k}$ are non-negative and add up to 1. Specifically, the problem to be solved can be written as a *minimization* problem in standard form as

$$\min_{\mathbf{k}} \frac{1}{2} \mathbf{k}^T B \mathbf{k} - \mathbf{d}^T \mathbf{k} \quad \text{such that } \mathbf{k}_i \geq 0 \text{ and } \sum_i \mathbf{k}_i = 1$$

where, $\mathbf{d} = 2\mathbf{y}^T \mathbb{E}_{\mathbf{k}^{(t)}}(A_{\mathbf{x}} | \mathbf{y})$ and $B = 2 \mathbb{E}_{\mathbf{k}^{(t)}}(A_{\mathbf{x}}^T A_{\mathbf{x}} | \mathbf{y})$. This is a standard quadratic programming problem with linear constraints, and can be solved using standard methods such as those proposed by Goldfarb and Idnani [22, 23]. Although we have not included it in the computation above, an additional regularization penalty on $\mathbf{k}$, either quadratic or linear, can be easily incorporated in this step.

Note that computing the vector $\mathbf{d}$ and the matrix B involves computing conditional expectations (under $\mathbf{k}^{(t)}$) of $A_{\mathbf{x}}$ and $A_{\mathbf{x}}^T A_{\mathbf{x}}$ respectively. Entries of $A_{\mathbf{x}}$ are just entries of $\mathbf{x}$ arranged in a specific form, because $A_{\mathbf{x}} \mathbf{k}$ is simply the linear transformation $\mathbf{k} \mapsto \mathbf{k} \otimes \mathbf{x}$ written as a matrix operation. Thus, $\mathbf{d}$ can be computed using (4.8) as

$$\mathbf{d} = 2\mathbf{y}^T A_{\mu}$$

(i.e., replacing entries of $\mathbf{x}$ by corresponding entries of $\boldsymbol{\mu}$). Computation of B involves cross-product terms, and thus requires the conditional variance $\boldsymbol{\Sigma}$ in (4.8).

Computing $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is in practice the most difficult step in EM approach, and in fact computing these quantities for a general distribution may not be feasible. In the remainder of this chapter, we briefly discuss the estimation of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the Gaussian and sparse MOG prior models for $\mathbf{x}$.

4.3 The Gaussian Case

Frequency domain calculation In the Gaussian case, $\boldsymbol{\mu}$ can be computed efficiently in the Fourier domain. To see this, note that by taking discrete Fourier transform on both sides of the convolution model $\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n}$, we can rewrite it as

$$\mathbf{Y}_\omega = \bar{\mathbf{K}}_\omega \cdot \mathbf{X}_\omega + \mathbf{N}_\omega, \quad (4.9)$$

where the uppercase variables denote Fourier transforms of the respective lowercase variables. This follows from the Convolution Theorem, whereby the Fourier transform of a convolution (in the image domain) is the pointwise product of Fourier transforms (in the frequency domain). The pointwise nature of this relationship is emphasized by the frequency pair index ω . This is one situation where the difference between convolution and filtering becomes important; (4.9) holds when the $\otimes$ operator represents filtering, and $\bar{\mathbf{K}}_\omega$ would change to $\mathbf{K}_\omega$ if it were to mean convolution.

It is implicit in (4.9) that $\mathbf{x}, \mathbf{y}$ and $\mathbf{k}$ have the same dimensions. Conceptually, this is achieved by padding smaller matrices; the details are important for implementation, but not relevant in the current discussion.

Being Fourier transforms of real matrices, $\mathbf{X}$ and $\mathbf{N}$ must satisfy certain constraints (some elements must be real, and others must occur as conjugate pairs). It is easy to show that subject to these constraints, the assumption that elements of $\mathbf{x}$ are IID Gaussian implies that $\mathbf{X}_\omega$ are IID complex Gaussian. The same holds for $\mathbf{N}_\omega$. In other words, we have

$$\mathbf{X}_\omega \sim \mathbb{CN}(0, \sigma^2) \text{ and} \quad (4.10)$$

$$\mathbf{N}_\omega \sim \mathbb{CN}(0, \eta^2), \text{ and so} \quad (4.11)$$

$$\mathbf{Y}_\omega \mid \mathbf{X}_\omega, \mathbf{K}_\omega \sim \mathbb{CN}(\bar{\mathbf{K}}_\omega \cdot \mathbf{X}_\omega, \eta^2) \quad (4.12)$$

where σ^2 and η^2 are the same as in the image domain provided $\mathbf{X}$ and $\mathbf{Y}$ are suitably normalized.

It easily follows that

$$\mathbf{X}_\omega | \mathbf{Y}_\omega, \mathbf{K}_\omega \sim \mathbb{CN} \left(\frac{\sigma^2 \mathbf{K}_\omega \mathbf{Y}_\omega}{\eta^2 + \sigma^2 |\mathbf{K}_\omega|^2}, \frac{\eta^2 \sigma^2}{\eta^2 + \sigma^2 |\mathbf{K}_\omega|^2} \right). \quad (4.13)$$

As DFT is a linear operator, $\boldsymbol{\mu} = \mathbb{E}(\mathbf{x} | \mathbf{y}, \mathbf{k})$ is obtained by simply taking the inverse DFT of $\mathbb{E}(\mathbf{X} | \mathbf{Y}, \mathbf{K})$. The constraints on $\mathbf{K}$ and $\mathbf{Y}$ ensure that all entries of $\boldsymbol{\mu}$ are real.

The calculation of $\boldsymbol{\Sigma}$ is slightly more complicated. Even though $\mathbf{X}_\omega$ are conditionally independent given $\mathbf{Y}_\omega$ and $\mathbf{K}_\omega$, their variances are not the same across ω . Consequently, the elements of $\mathbf{x}$ are not conditionally independent. It is not practical to compute the full conditional variance $\text{Var}(\mathbf{x} | \mathbf{y}, \mathbf{k})$, which is an $L \times L$ matrix where L is the number of pixels in $\mathbf{x}$. Instead, the off-diagonal elements are assumed to be zero; this is a reasonable approximation in practice. It is easy to see that the diagonal entries are all same, and given by

$$\frac{1}{\Sigma_{ii}} = \frac{1}{\sigma^2} + \frac{\sum \mathbf{k}_j^2}{\eta^2}.$$

Image domain calculation As we see below, the Fourier domain calculation does not translate to the case of the MOG prior, where computations must be done in the image domain. Although not directly useful for computational purposes in the case of the Gaussian prior, it is instructive for later analogy to obtain expressions for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ directly in the image domain in this case as well.

Identify $\mathbf{x}$ and $\mathbf{y}$ as vectors rather than matrices, and write $\mathbf{k} \otimes \mathbf{x}$ (with $\mathbf{x}$ and $\mathbf{k}$ as matrices) as the linear transformation $T_{\mathbf{k}} \mathbf{x}$. Here $T_{\mathbf{k}}$ is a Toeplitz matrix determined by $\mathbf{k}$, with the useful property that if $T_{\mathbf{k}} \mathbf{x}$ denotes convolution then $T_{\mathbf{k}}^T \mathbf{x}$ denotes filtering; in other words, $T_{\mathbf{k}}^T = T_{\text{flip}(\mathbf{k})}$, recalling that $\text{flip}(\mathbf{k})$ is obtained from $\mathbf{k}$ by reversing the order of both rows and columns. With this setup, the problem of interest can be written as

$$\mathbf{x} \sim N(0, \sigma^2 \mathbf{I}), \mathbf{y} | \mathbf{x}, \mathbf{k} \sim N(T_{\mathbf{k}} \mathbf{x}, \eta^2 \mathbf{I}), \quad (4.14)$$

where we are interested in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, the mean and variance of $\mathbf{x} | \mathbf{y}$. It is immediate that

(4.14) represents a Ridge regression problem, albeit with a highly structured and sparse design matrix, and the parameters of the conditional distribution are given by

$$\left(\frac{1}{\eta^2} T_{\mathbf{k}}^T T_{\mathbf{k}} + \frac{1}{\sigma^2} \mathbf{I} \right) \boldsymbol{\mu} = \frac{1}{\eta^2} T_{\mathbf{k}}^T \mathbf{y} \text{ and } \boldsymbol{\Sigma}^{-1} = \frac{1}{\eta^2} T_{\mathbf{k}}^T T_{\mathbf{k}} + \frac{1}{\sigma^2} \mathbf{I}. \quad (4.15)$$

As noted above, (4.15) can be solved efficiently for $\boldsymbol{\mu}$ by rewriting the first equation as

$$\frac{1}{\eta^2} \text{flip}(\mathbf{k}) \otimes \mathbf{k} \otimes \boldsymbol{\mu} + \frac{1}{\sigma^2} \boldsymbol{\mu} = \frac{1}{\eta^2} \text{flip}(\mathbf{k}) \otimes \mathbf{y}$$

which in the Fourier domain gives

$$\mathbb{E}(\mathbf{X}_{\omega} | \mathbf{Y}_{\omega}) = \frac{\frac{1}{\eta^2} \mathbf{K}_{\omega} \mathbf{Y}_{\omega}}{\frac{1}{\eta^2} |\mathbf{K}_{\omega}|^2 + \frac{1}{\sigma^2}},$$

which is equivalent to (4.13). This follows because the Fourier coefficient of $\text{flip}(\mathbf{k})$ for frequency ω is the complex conjugate of the corresponding Fourier coefficient of $\mathbf{k}$. In fact, $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$, with DFT at ω given by $|\mathbf{K}_{\omega}|^2$, represents the probability distribution obtained by “symmetrizing” $\mathbf{k}$, corresponding to the random variable $\mathbf{U} - \mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are independent copies of an integer-valued bivariate random variable with distribution $\mathbf{k}$.

We will see below that in the sparse prior case, $\boldsymbol{\mu}$ is approximately determined by an equation very similar to (4.15) which, unfortunately, cannot be solved efficiently in the Fourier domain.

4.4 The Sparse MOG Case

The computation of $\boldsymbol{\mu}$ is considerably more difficult in the case of the sparse MOG prior, and in fact a direct computation is intractable. The success of the methods of Fergus et al. [17] and Levin et al. [45] is largely to their use of a variational Bayes approximation of the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{k}$. Recall the MOG prior model in (4.3) given by

$$\pi(x) = \sum_{j=1}^J p_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{x^2}{2\sigma_j^2}\right), \quad \sum_{j=1}^J p_j = 1.$$

where the p_j -s are mixture probabilities and the σ_j^2 -s are the variances of the Gaussian components. In the convolution model

$$\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n},$$

we assume that the image gradients $\mathbf{x}$ follow the IID mixture of Gaussian given by $\pi(\cdot)$. The E-step in this case cannot be computed in the Fourier domain, and instead follows steps analogous to the image domain calculations outlined above for the Gaussian case. Recall that the conditional expectation $\boldsymbol{\mu}$ in the Gaussian case can be expressed as the solution to the linear equations (4.15) given by

$$\left(\frac{1}{\eta^2} T_{\mathbf{k}}^T T_{\mathbf{k}} + \frac{1}{\sigma^2} \mathbf{I} \right) \boldsymbol{\mu} = \frac{1}{\eta^2} T_{\mathbf{k}}^T \mathbf{y}.$$

In the MOG case, the approximate solution is instead given by the linear equations

$$\left(\frac{1}{\eta^2} T_{\mathbf{k}}^T T_{\mathbf{k}} + \mathbf{W} \right) \boldsymbol{\mu} = \frac{1}{\eta^2} T_{\mathbf{k}}^T \mathbf{y}. \quad (4.16)$$

where $\mathbf{W}$ is still a diagonal matrix, but with unequal diagonal entries W_{ii} given by

$$W_{ii} = \sum_{j=1}^J \frac{w_{i,j}}{\sigma_j^2},$$

where

$$w_{i,j_0} = \frac{\frac{p_{j_0}}{\sigma_{j_0}} e^{-\mathbb{E}(\mathbf{x}_i^2)/2\sigma_{j_0}^2}}{\sum_{j=1}^J \frac{p_j}{\sigma_j} e^{-\mathbb{E}(\mathbf{x}_i^2)/2\sigma_j^2}}$$

with i running over image pixels and j running over mixture components. Here $\mathbb{E}(\mathbf{x}_i^2) = \boldsymbol{\mu}_i^2 + \boldsymbol{\Sigma}_{ii}$ is calculated using the conditional mean and variance $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ estimated in the previous iteration. The updated $\boldsymbol{\Sigma}$ is set to the inverse of $\frac{1}{\eta^2} T_{\mathbf{k}}^T T_{\mathbf{k}} + \mathbf{W}$. The M-step remains unchanged. We refer the reader to Levin et al. [45] for detailed derivation.

4.5 Practicalities

With the computations derived above, the EM algorithm iterates between the E and M-steps, either until it converges according to some numerical criteria, or after running for a fixed number of iterations. Depending on the initial estimate, the algorithm may converge to a local minima of the likelihood. As this is a high dimensional non-convex problem, this approach often often converges to undesirable local minima. Both Fergus et al. [17] and Levin et al. [45] have found a multi-scale approach to be useful, where they perform the estimation by varying the image resolution in a coarse to fine manner, at the same time starting with a high value of the noise variance η^2 and gradually decreasing it to the user-set target. The initial estimate of the blur kernel is a simple pattern, typically $[0.5, 0.5]$ or $[0.5, 0.5]^T$. When stepping up the pyramid to work with the next higher resolution image, the estimated $\mathbf{k}$ is upsampled to act as the initial estimate for inference at that scale. The authors note that the choice of prior parameters is often crucial in the success of the algorithm, and we also find that minor changes in the parameters can lead to very different results.

Chapter 5

Dependent prior: motivation and simple applications

In this chapter, we describe a proposed generalization of the IID priors described in the previous chapter. We first motivate this generalization before discussing some simple applications in subsequent sections and chapters.

5.1 Motivation: Fourier domain

As seen earlier in Section 4.3, calculations are substantially simplified in the Fourier domain in the case of the Gaussian prior. Another advantage of working in the Fourier domain is that it is easier to visualize the underlying problem and motivate the prior.

To illustrate this, Figure 5.1 plots the log-transformed modulus of Fourier coefficients obtained from the 64 synthetically blurred test images shown in Figure 1.6. Recall that these 64 images are obtained by blurring eight test images (Figure 1.4) with eight test kernels (Figure 1.5) without incorporating any additional noise. Figure 5.1 can be considered a visualization of (4.9). The top row plots $|\mathbf{X}_\omega|$, and can be viewed as periodograms of the (horizontal) gradients for the test images, and the left column shows $|\mathbf{K}_\omega|$, i.e., the periodograms for the test kernels after suitable padding. The remaining plots visualize $|\mathbf{Y}_\omega|$ for all image-kernel combinations, and can be viewed as periodograms of the gradients of the observed blurred images. For

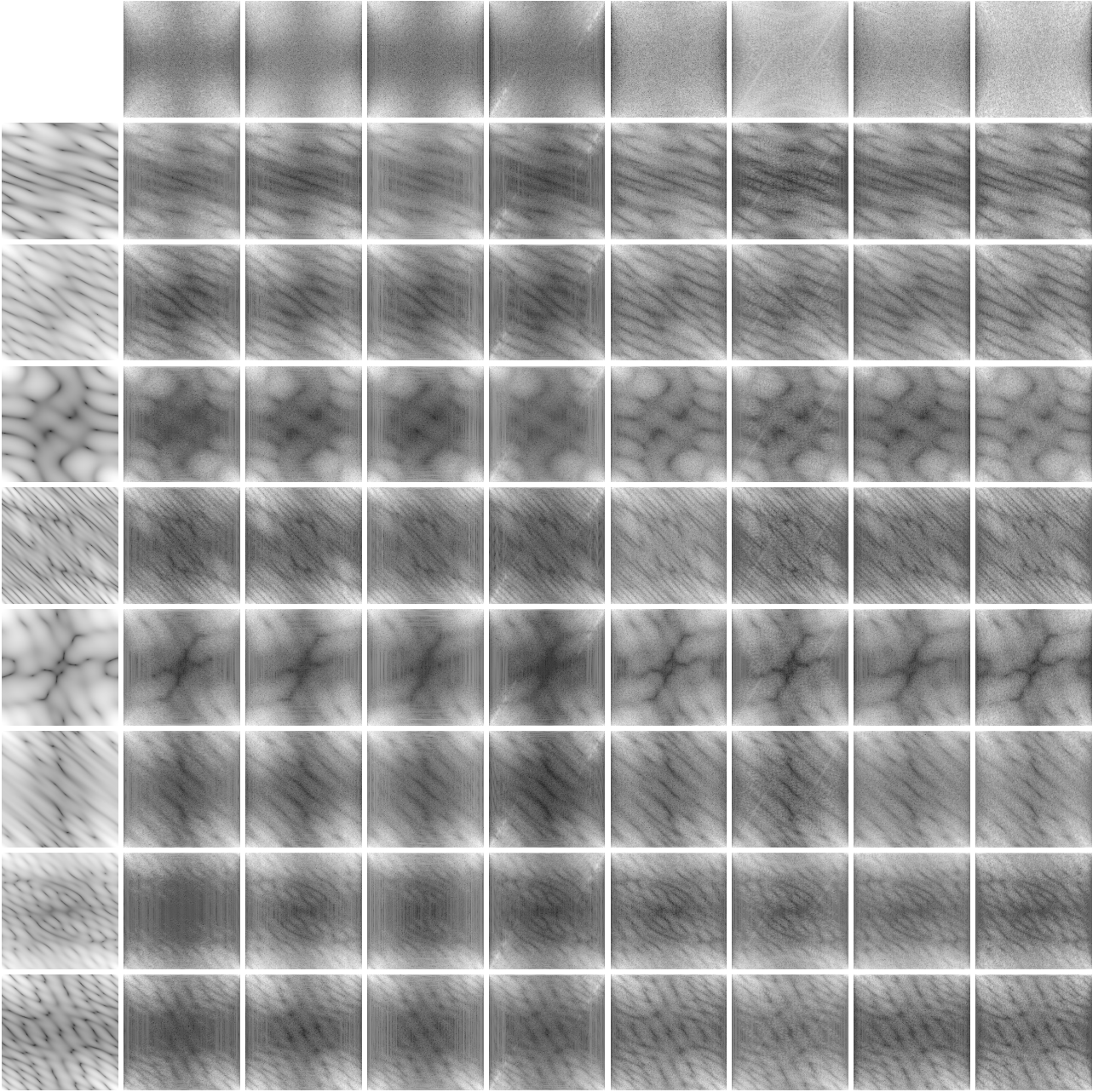


Figure 5.1: “Periodograms” corresponding to synthetically blurred exact test images in Figure 1.6, excluding even the error due to rounding. The x and y coordinates represent frequency (from 0 to 2π); intensity (black = low, white = high) represents log-transformed modulus of the corresponding Fourier coefficient. The top row corresponds to DFT of the horizontal gradients of the eight unblurred test images. The left row corresponds to DFT of the eight test kernels, after padding with zeros. The remaining plots correspond to DFT of the horizontal gradients of the blurred images obtained by convolving the corresponding test image and the test kernel.

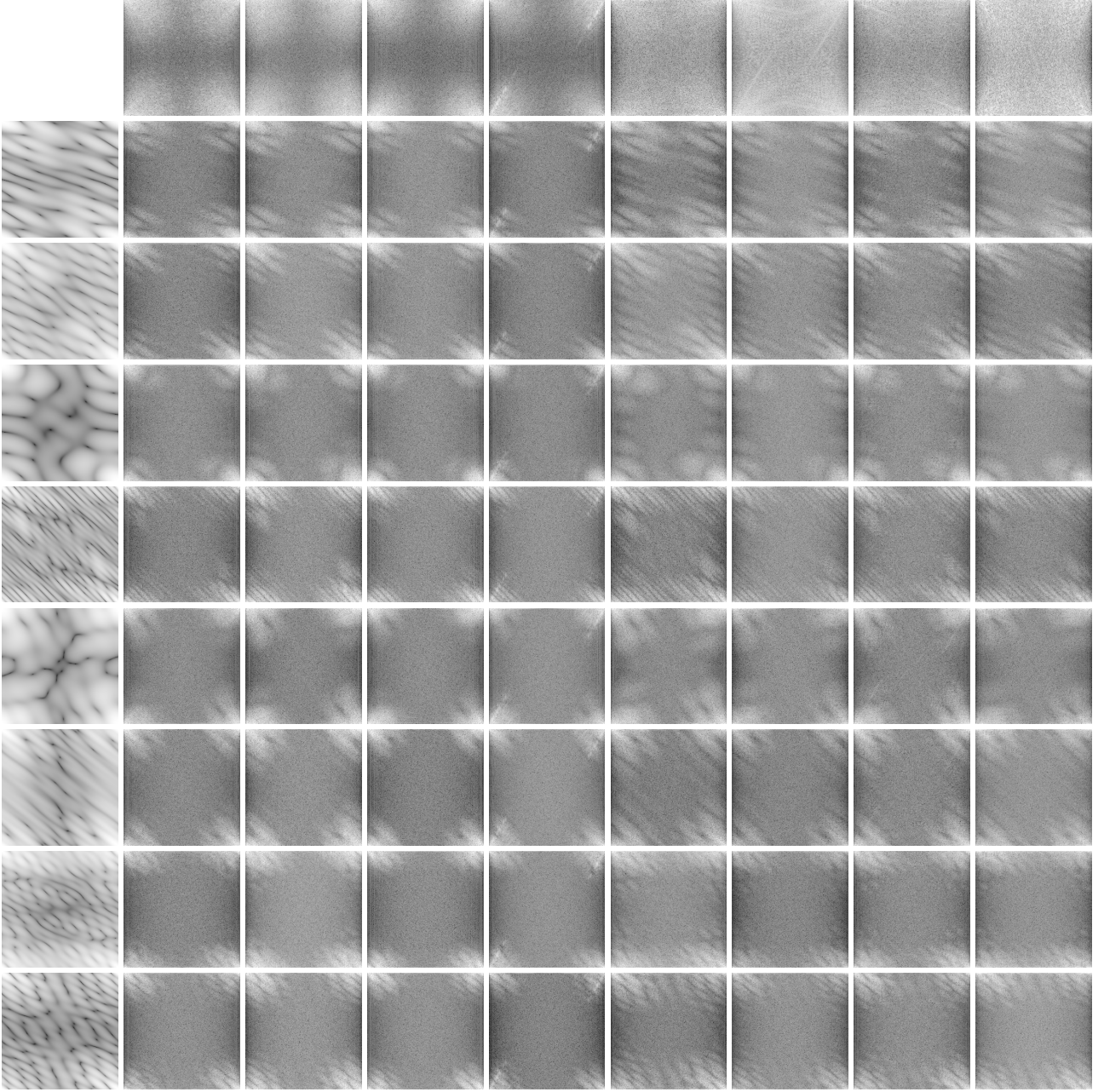


Figure 5.2: “Periodograms” corresponding to synthetically blurred noisy test images in Figure 1.7. As in Figure 5.1, the x and y coordinates represent frequency and intensity represents log-transformed modulus of the corresponding Fourier coefficient, and the top row and left column correspond test images and test kernels respectively. The blurred images whose DFTs are represented in the remaining plots are obtained by convolving the corresponding test image and the test kernel with additional noise ($S = 100$).

the purpose of illustration, these plots are obtained from exact synthetically blurred images, without additional noise and without even the error due to integer rounding.

The top-most row may be viewed as realizations from the prior on $\mathbf{x}$, or equivalently, $\mathbf{X}$, which in the context of the problem is the unobserved nuisance parameter. In the Gaussian case, $\mathbf{X}_\omega$ are IID complex Normal. Although Normality cannot be assessed using this plot, there does not appear to be any systematic patterns, suggesting that the independence assumption may be reasonable. Additionally, there appears to be some local variation, especially in the first four images, in that the coefficient magnitudes tend to be smaller (darker) around the middle frequencies ($\omega \approx \pi$) and larger when ω is close to 0 or 2π . We will revisit this pattern shortly.

The left-most column, representing $|\mathbf{K}_\omega|$, is highly patterned. This is the parameter of interest in our problem. More importantly, the plots for the 64 blurred images, representing the observed data $|\mathbf{Y}_\omega|$, clearly inherits this pattern, albeit corrupted by the nuisance $|\mathbf{X}_\omega|$.

Figure 5.2 shows similar periodograms for the 64 test images with additional noise, shown in Figure 1.7. Essentially the same observations as above hold for this set of plots as well, except that we observe substantial degradation of the systematic pattern of $|\mathbf{K}_\omega|$ in the observed $|\mathbf{Y}_\omega|$. This degradation is present to some extent even in the case of the integer rounding that occurs when the images are stored as 8-bit image files.

The statistical problem is to estimate $\mathbf{K}$ from $\mathbf{Y}$; that is, to find $\mathbf{k}$ of suitable size satisfying non-negativity constraints such that the corresponding $|\mathbf{K}_\omega|$ best matches the observed $|\mathbf{Y}_\omega|$ in some optimum way. It is in principle possible to estimate $\mathbf{k}$ using this approach, as described in Section 3.2. However, this does not lead to particularly good results. We will briefly digress to recall why that is so, as this insight will be useful later.

The first point to note is that we can only use the modulus of the Fourier coefficients, and not their arguments, to estimate $\mathbf{k}$. This is because $\text{Arg } \mathbf{X}_\omega$ behaves like IID $U(0, 2\pi)$ and completely corrupts $\text{Arg } \mathbf{K}_\omega$, and as a result $\text{Arg } \mathbf{Y}_\omega$ contains no information about $\mathbf{K}$. Consequently, we can at best estimate $|\mathbf{K}_\omega|$. Under the Gaussian model, the obvious pointwise estimate of $|\mathbf{K}_\omega|$ is the maximum likelihood estimate given by

$$|\hat{\mathbf{K}}_\omega|^2 = \frac{\max(0, |\mathbf{Y}_\omega|^2 - \eta^2)}{\sigma^2}. \quad (5.1)$$

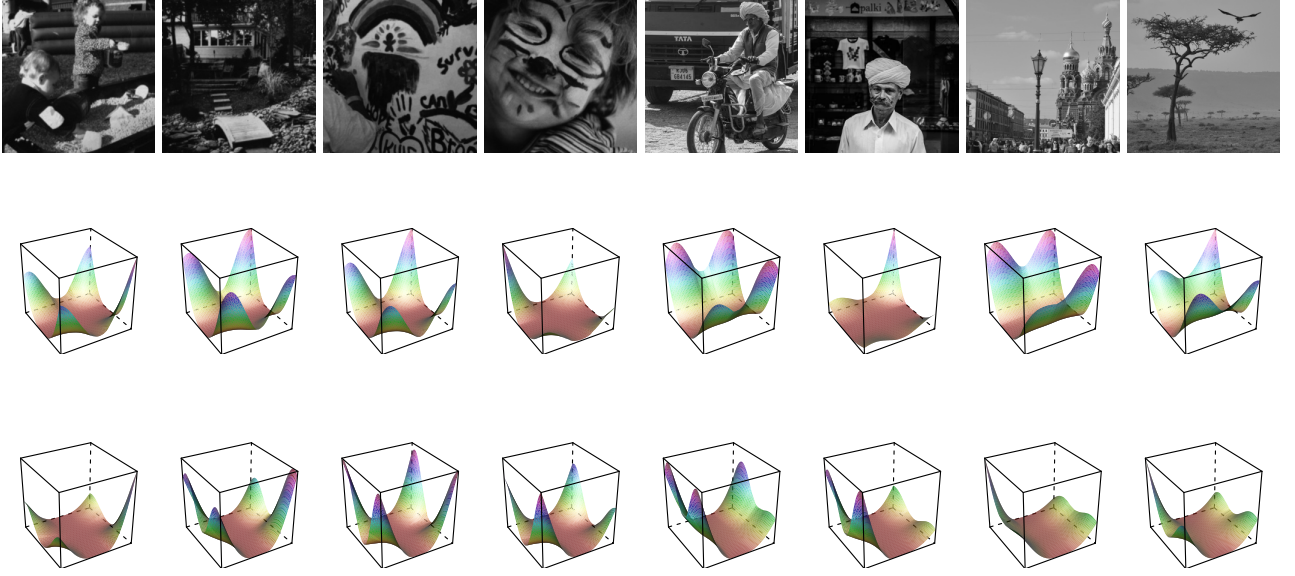


Figure 5.3: Locally smoothed periodograms corresponding to horizontal (middle row) and vertical (bottom row) gradients of the eight test images (top row). The x and y coordinates represent frequency in the bottom two rows, and the z coordinate represents $E(|\mathbf{X}_\omega|^2)$, which is estimated by applying a local Gaussian blur filter on $|\mathbf{X}_\omega|$ (roughly equivalent to smoothing using LOESS) and squaring the result.

The problem then reduces to finding a $\hat{\mathbf{k}}$ whose DFT magnitudes match $|\hat{\mathbf{K}}_\omega|$. The mapping of $\mathbf{k}$ to magnitudes $|\mathbf{K}|$ is technically invertible (up to location shift and flip) if $\mathbf{k}$ is non-negative valued, but the inverse map is computationally intractable. The iterative algorithms described in Section 3.2 need good initial estimates, which are not easy to obtain in this context.

Another fundamental limitation of this approach is that while reasonable for the Gaussian prior, it does not translate well to the sparse (mixture of Gaussian) prior case. Even if the gradients follow a sparse prior, the DFT coefficients are asymptotically Gaussian [59]. This means that, any approach based solely on the distribution of $\mathbf{Y}_\omega$ is unable to distinguish between the Gaussian and the sparse case, even though this distinction clearly has a significant effect in the EM approach described earlier.

Nonetheless, the periodograms of the gradients of the unblurred images in Figure 5.1 do provide additional insight. In particular, the fact that the coefficients of the middle frequencies tend to be lower in magnitude suggests that the variances of the DFT coefficients are not constant across frequencies. To understand the nature of the variation, Figure 5.3 shows the

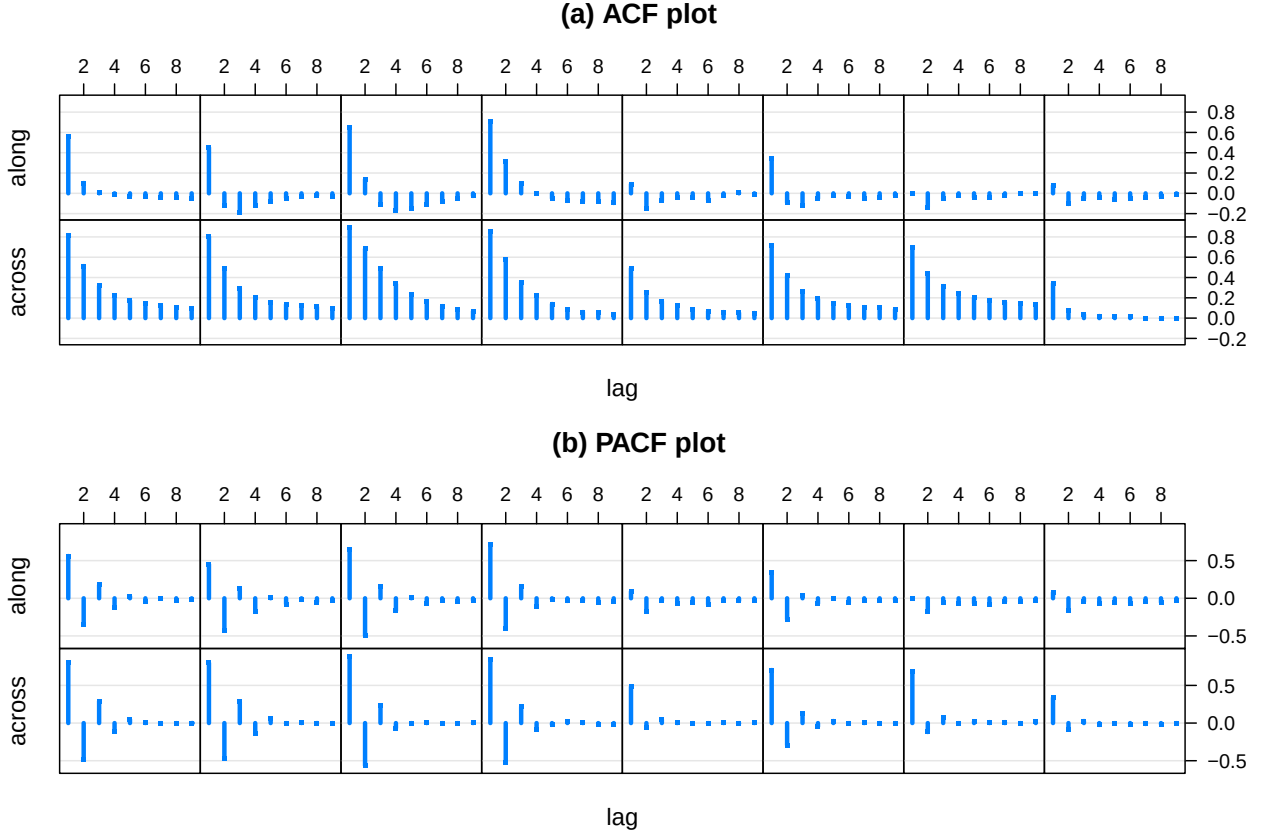


Figure 5.4: Estimated auto-correlation function (ACF) of image gradients along (top row) and across (bottom row) the direction of the gradient. Columns represent the eight unblurred test images. The ACF (a) and PACF (b) are estimated by taking the average of row-wise (top row) and column-wise (bottom row) sample ACF / PACFs from the horizontal gradients. The results for vertical gradients, which are not shown, are similar.

result of locally smoothing $\log |\mathbf{X}_\omega|$ by applying a Gaussian blur. The resulting profile is not flat or constant as one would expect for IID $|\mathbf{X}_\omega|$, but is instead smaller at the center ($\omega \approx \pi$) and increasing towards the corners (ω close to 0 and 2π).

This kind of pattern is typically associated with non-IID stationary processes. In fact, it is not unreasonable to expect that the image gradients are in fact locally dependent and not IID. To understand the nature of this local dependence, Figure 5.4 plots the estimated auto-correlation and partial auto-correlation functions of the image gradients, both along the direction of the gradient as well as across it. These suggest that although not all images have same dependence structure, in some cases the correlation can be quite strong, and is typically stronger across the gradient direction.

5.2 Proposed generalization

The preceding discussion motivates us to suggest a more general model for the image gradient matrix $\mathbf{x}$; namely, that rather than being an IID process, it is a stationary process with possibly dependent entries. More specifically, we postulate the following prior model for $\mathbf{x}$.

Model 1 *The marginal distribution of each pixel element $\mathbf{x}_{ij}$ remains unchanged, and is given by either the Gaussian model (4.2) or the sparse MOG model (4.3); additionally, the prior model is completed by specifying a correlation function $\rho(i, j, k, l) = \text{Cor}(\mathbf{x}_{ij}, \mathbf{x}_{kl})$.*

If Figure 5.4 is viewed as ACF / PACF functions obtained from time-series data, we would possibly consider modeling it as an ARMA(p, q) process with p, q estimated from the data. However, this approach is not feasible in the current context because we do not actually observe the unblurred image, and the model parameters may vary from image to image. Instead, in this thesis we only consider a simple first-order auto-regressive model, given as a product of AR(1) structures in both dimensions, with correlation function given by

$$\rho(i, j, k, l) = \rho_1^{|i-k|} \rho_2^{|j-l|}. \quad (5.2)$$

Here ρ_1 and ρ_2 are additional parameters representing lag-1 correlation along and across the gradient direction.

As $\mathbf{x}$ has an independent product structure, it follows from Theorems 9.13 and 9.14 of Spodarev [72] that the resulting correlation matrix is positive-definite. It follows that such a process exists with Gaussian marginals, although it is less clear whether a similar process with the MOG prior will also exist. We can bypass this technical difficulty by noting that a process $\mathbf{x}$ with correlation function (5.2) can be decorrelated using the filter

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_1 \rho_2 & -\rho_1 \\ -\rho_2 & 1 \end{bmatrix},$$

and model $\boldsymbol{\rho} \otimes \mathbf{x}$ as an IID process with a sparse marginal distribution. However, this argument would not apply to a general dependence structure, which we would like to allow in principle.

In addition to this technicality, Model 1 is also difficult to work with in practice, especially in the Fourier domain, even though it is the natural generalization of the IID prior. An alternative is to specify the generalization in the frequency domain. Recall that if $\mathbf{x}$ is IID with variance σ^2 , the matrix of DFT coefficients $\mathbf{X}$, suitably normalized by image size, also has uncorrelated entries with constant variance σ^2 because DFT is an orthogonal transformation. When elements of $\mathbf{x}$ are not IID, then the DFT coefficients no longer have constant variance, but have variance function

$$\text{Var}(\mathbf{X}_\omega) = g_\omega \sigma^2$$

for some function g_ω that depends only on the correlation function $\rho(\cdot)$. This motivates us to postulate the following alternative prior model for $\mathbf{x}$.

Model 2 *The process $\mathbf{x}$ can be transformed to another process $\mathbf{z}$ with IID components with distribution $\pi(x)$ given by (4.2) or (4.3), by the following relation: $\mathbf{X}_\omega = \sqrt{g_\omega} \mathbf{Z}_\omega$, where $\mathbf{Z}$ is the DFT of $\mathbf{z}$.*

The two models are not exactly equivalent, because the prior $\pi(\mathbf{x})$ applies to $\mathbf{x}$ in Model 1 and $\mathbf{z}$ in Model 2. More importantly, although the variances of $\mathbf{X}_\omega$ match in the two models, the components of $\mathbf{Z}$ are uncorrelated in Model 2 (since $\mathbf{z}$ has IID components and DFT is an orthogonal transformation) and therefore so are the components of $\mathbf{X}$, whereas in Model 1, $\mathbf{X}_\omega$ -s are possibly correlated.

In practice, this is a minor issue because the correlations are low for even moderately sized images (see Section A.2 in the Appendix for explicit computations), and they vanish asymptotically [59]. Although Model 1 is the more natural generalization, it is difficult to incorporate into the estimation procedure, and we focus exclusively on Model 2, which we view as a technical approximation to Model 1.

Choice of g_ω For the remainder of this work, we use the variance function corresponding to the first-order auto-regressive model with correlation function (5.2). This is not strictly necessary, as our method only requires g_ω to be specified and does not use the image-domain model that gave rise to it. In fact, it is potentially useful to estimate g_ω from a library of

unblurred images or perhaps specific unblurred images similar to the blurred image being deblurred, as is usually done for other prior parameters. However, we restrict our attention to the g_ω generated by (5.2) for ease of interpretation, and because it only adds two parameters (ρ_1 and ρ_2) to the overall prior model.

The variance function g_ω corresponding to (5.2) can be easily obtained by direct computation, and is given by the following Lemma.

Lemma 5.2.1 *Suppose the $M \times N$ matrix $\mathbf{x}$ has entries with mean zero, variance σ^2 , and correlation given by*

$$\rho(i, j, k, l) = \text{Cor}(x_{ij}, x_{kl}) = \rho_1^{|i-k|} \rho_2^{|j-l|}.$$

Let $\mathbf{X}_\omega$ be the (normalized) discrete Fourier transform of $\mathbf{x}$ evaluated at the frequency pair $\omega = (\omega_1, \omega_2)$. Then, the variance of $\mathbf{X}_\omega$ is $\sigma^2 g_\omega$, with g_ω given by

$$g_\omega = V_M(\rho_1, \omega_1) V_N(\rho_2, \omega_2),$$

where

$$V_L(\rho, \omega) = \sum_{t_1=1}^N \sum_{t_2=1}^N \rho^{|t_1-t_2|} e^{-i\omega(t_1-t_2)} \quad \text{for } \omega = \frac{2\pi k}{L}, k = 0, 1, 2, \dots, L-1.$$

Here $V_L(\rho, \omega)$ is the variance of the ω -th DFT coefficient for a one-dimensional auto-regressive process with lag-1 correlation ρ , and can be simplified to a closed-form solution suitable for computation. Relevant details are provided in Section A.1 in the Appendix. Plots of $V_L(\rho, \omega)$ for various values of ρ are shown in Figure 5.5.

Choice of ρ_1 and ρ_2 The parameters ρ_1 and ρ_2 are hyperparameters of the model, similar to the variance parameters of the prior and noise, that need to be specified by the user. Recall that we assume horizontal and vertical gradients to be independent in likelihood calculations, and although we have mostly used the horizontal gradients for illustration, we use both horizontal and vertical gradients during estimation. The parameter ρ_1 here represents the correlation component *along* the direction of the gradient, and ρ_2 represents the correlation component

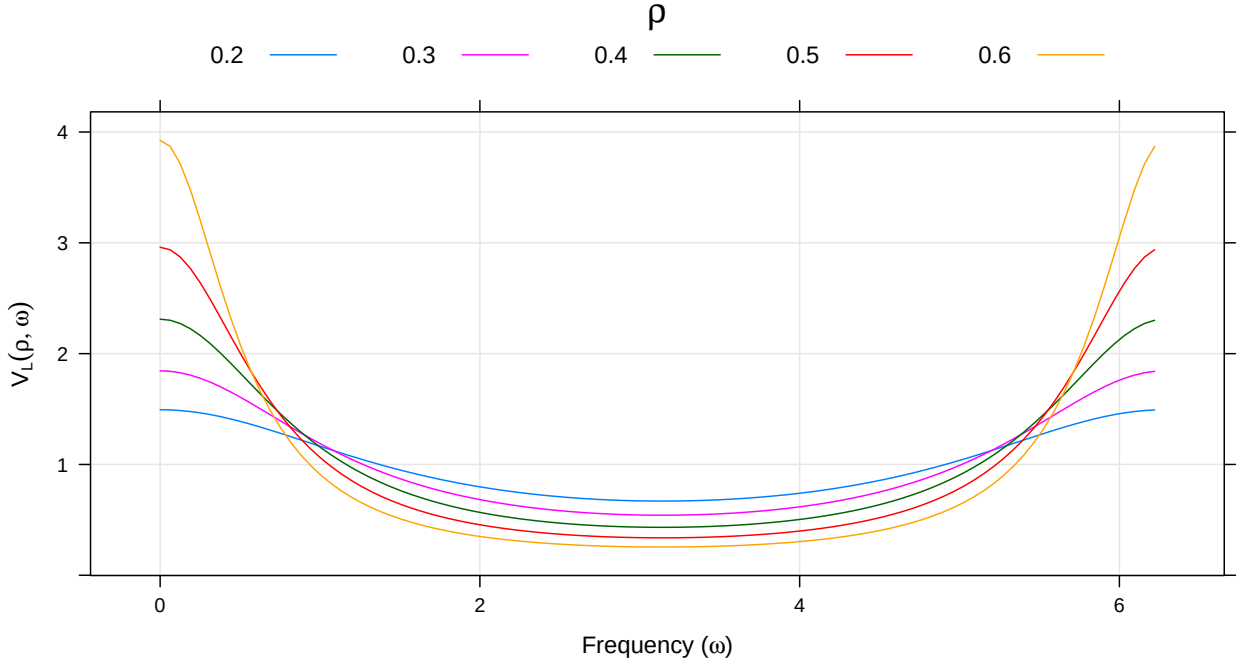


Figure 5.5: Plot of $V_L(\rho, \omega)$ for various values of ρ .

across the direction of the gradient. In other words, ρ_1 represents the horizontal or vertical components for horizontal and vertical gradients respectively, and ρ_2 represents the vertical or horizontal components for horizontal and vertical gradients respectively.

In practice, the “optimum” choice of ρ_1 and ρ_2 will depend on the particular image being deblurred, as indicated by the substantial variation in the ACF plots in Figure 5.4. For the purpose of testing, we have used $\rho_1 = 0.3$ and $\rho_2 = 0.6$ in this thesis. These choices are motivated by the observation that the lag-1 correlation in the ACF plots is generally higher across the direction of the gradient than along. Figure 5.6 shows an updated version of Figure 5.1 that corrects for the non-constant variance by modeling it via g_ω . The spatial pattern in $|\mathbf{X}_\omega|$ seen in Figure 5.1 is substantially reduced in the corrected modulus $|\mathbf{X}_\omega|/\sqrt{g_\omega}$ shown in Figure 5.6.

5.3 Application: Estimation of symmetric blur kernel

For deconvolution problems, the blur kernel is usually unknown in practice, and estimating it is a difficult inference problem in general. In Chapter 7, we outline a procedure to incorporate the dependent prior described above into the blind deconvolution approach of Levin et al. [45]

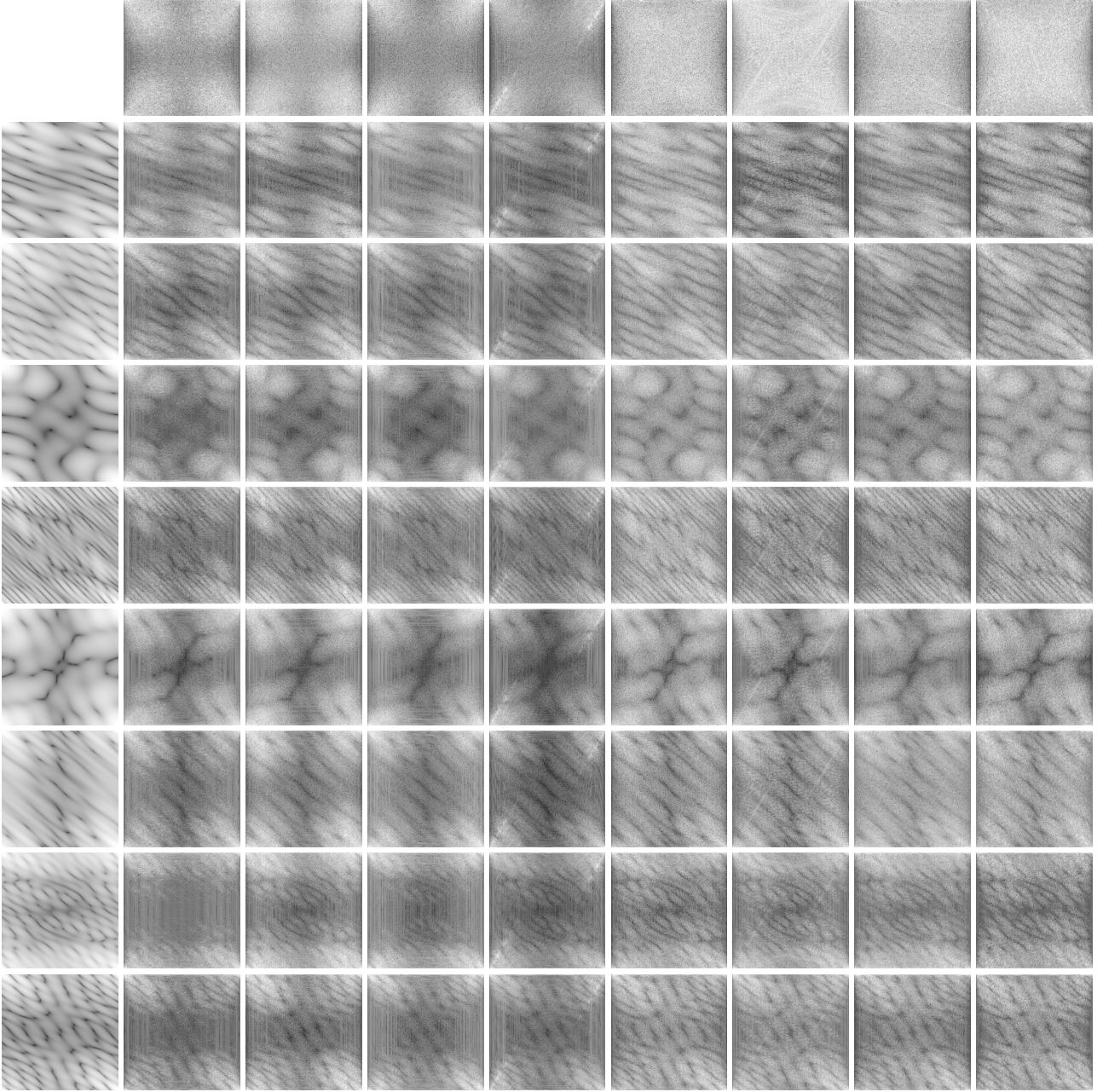


Figure 5.6: Periodograms shown earlier in Figure 5.1, obtained from synthetically blurred exact test images shown in Figure 1.6, corrected by the variance function g_ω corresponding to the first-order auto-regressive model with $\rho_1 = 0.3$ and $\rho_2 = 0.6$. Compared to Figure 5.1, there is less systematic spatial variation.

described in Chapter 4.

In many situations, however, it is reasonable to assume that the blur kernel is relatively small, and more importantly, symmetric. This is usually the case when the blur is due to camera optics, and not camera shake, such as blurring due to mild lack of focus. This is also usually the case for super-resolution, where the choice of blur kernel is important, but usually unknown in advance. Many methods assume a fixed blur kernel, often corresponding to the cubic interpolation method commonly used to resize images in image processing software. Efrat et al. [16] show that this is not a good model for the blur kernel implicitly used by real cameras, and that accurate specification of the blur kernel is in fact often more important than the image prior in determining the quality of the result. They outline a method of estimating the true blur, which they call the ‘camera blur’, when one has access to the camera used to capture the input image, but leave open the problem of estimating the blur from the input image itself.

Motivated by the observation that image gradients appear to be correlated, we propose a fast and simple nonparametric Fourier domain approach to estimate the blur kernel that appears to work quite well for simple blur kernels. The method explicitly assumes that the blur kernel $\mathbf{k}$ is symmetric, in the sense that that it gives the same results whether it is used as a filter or convolution, or equivalently, that $\mathbf{k} = \text{flip}(\mathbf{k})$, where $\text{flip}(\mathbf{k})$ is obtained from $\mathbf{k}$ by reversing the order of both rows and columns.

As noted in Section 5.1, the symmetrized version of $\mathbf{k}$, $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$, has real-valued DFT, with the coefficient at frequency pair ω given by $|\mathbf{K}_\omega|^2$. In general, it is difficult to estimate $\mathbf{K}$ from $|\mathbf{K}_\omega|$, but the assumption of symmetry simplifies the problem substantially, as in this case the Fourier transform $\mathbf{K}$ of $\mathbf{k}$ also has non-negative real-valued coefficients. By the convolution theorem, the Fourier transform of $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ is then the pointwise (Hadamard) product of $\mathbf{K}$ with itself, i.e., the coefficient for a frequency pair ω is $\mathbf{K}_\omega^2$. Our method works by computing a novel pointwise estimate of $\mathbf{K}_\omega$, which is then inverted to obtain an estimate of $\mathbf{k}$.

To motivate this estimate of $\mathbf{k}$, first consider the model $\mathbf{y} = \mathbf{k} \otimes \mathbf{z} + \mathbf{n}$ for an observed matrix $\mathbf{y}$, where elements of the matrix $\mathbf{z}$ are zero-mean IID random variables with the (possibly scaled) hyper-Lapacian distribution given by (1.3), and $\mathbf{n}$ is a matrix of zero-mean IID Gaussian

noise. Taking (two-dimensional) discrete Fourier transform (DFT) on both sides, after padding $\mathbf{k}$ with zeros to get the same size as $\mathbf{z}$, it follows from the convolution theorem that

$$\mathbf{Y}_\omega = \mathbf{K}_\omega \cdot \mathbf{Z}_\omega + \mathbf{N}_\omega,$$

where the uppercase variables denote the DFTs of the respective lowercase variables, and the pointwise nature of this relationship is emphasized by the frequency pair index ω . By orthogonality of DFT, the $\mathbf{N}_\omega$ -s are (almost) IID complex Normal.¹ The $\mathbf{Z}_\omega$ -s are similarly IID complex Normal when $\alpha = 2$, and uncorrelated and asymptotically complex Normal when $\alpha \neq 2$ by the CLT of Peligrad and Wu [59]. Assuming that $\mathbf{Z}_\omega \sim \mathbb{CN}(0, \sigma^2)$ and $\mathbf{N}_\omega \sim \mathbb{CN}(0, \eta^2)$, it is easy to see that the marginal distribution of $\mathbf{Y}_\omega$ is $\mathbb{CN}(0, \eta^2 + \mathbf{K}_\omega^2 \sigma^2)$, and thus the maximum likelihood estimate of $\mathbf{K}_\omega$, assuming that σ and η are known, is given by

$$\hat{\mathbf{K}}_\omega = \frac{1}{\sigma} \sqrt{\max(0, |\mathbf{Y}_\omega|^2 - \eta^2)}. \quad (5.3)$$

The problem then reduces to finding a $\hat{\mathbf{k}}$ whose DFT matches $\hat{\mathbf{K}}$; this is obtained by applying inverse DFT. The resulting $\hat{\mathbf{k}}$ is of the same size as $\mathbf{y}$, whereas the true $\mathbf{k}$ is expected to be much smaller; we usually apply some heuristics to set small elements to zero and crop the estimate to a reasonable size. By requiring $\mathbf{k}$ to be a probability with elements adding up to 1, the estimate becomes independent of σ . The noise variance parameter η controls the sharpness of the estimate, and $\eta^2 = 0.005^2$ is empirically found to be reasonable when $\mathbf{y}$ is the gradient of images scaled to lie between 0 and 1, provided the DFT is normalized for the size of $\mathbf{y}$.

Applying the above method to horizontal and vertical gradients gives two different estimates of $\mathbf{k}$, and we take the average of the two as the final estimate $\hat{\mathbf{k}}$. These estimated kernels derived from the eight test images shown in Figure 1.4 are given in the top row of Figure 5.7. As the input images are not blurred, we expect the true $\mathbf{k}$ to be very small (ideally 1×1), but the estimated kernels appear to show cross-shaped patterns that are clearly artifacts.

To correct for these artifacts, we refine this estimate as follows. If $\mathbf{z}$ is not in fact IID but

¹With some restrictions imposed by the real-valued nature of $\mathbf{n}$, that are not particularly relevant in this context.

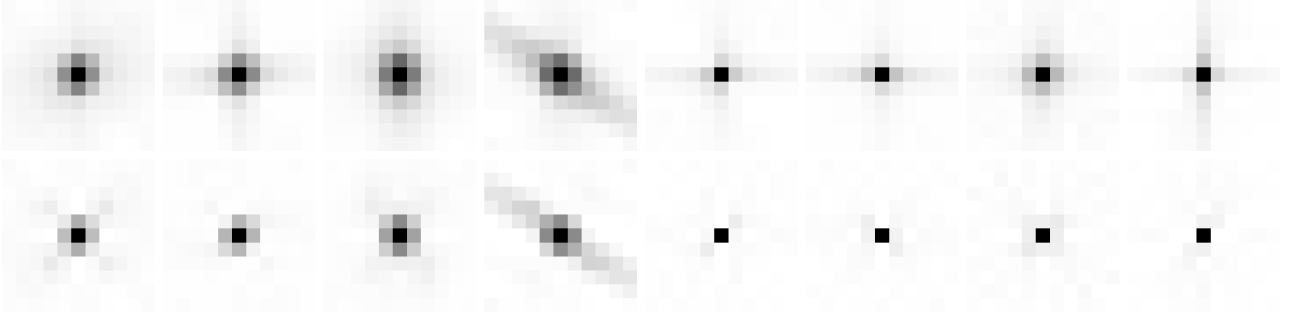


Figure 5.7: Estimates of the blur kernel obtained by inverting the asymptotic MLE of the kernel DFT, assuming symmetry. The top row assumes that DFT coefficients of image gradients have equal variance. The bottom row shows estimates of the blur kernel after adjusting for correlated gradient noise, and more importantly, for unequal variance of the DFT coefficients arising from correlated gradients. As the images are not actually supposed to be blurred, we expect the true kernel to be small; here we crop them to be 11×11 pixels in size. The estimates assuming IID gradients exhibit cross-shaped patterns that are clearly artifacts, which are not seen in our proposed estimates that assume correlated gradients and correct for the resulting non-constant variance of DFT coefficients. Both estimates for the fourth image are affected by the strong patterns in the original image. Interestingly, the last four images have minimal blurring according to our proposed estimate, but the first three images do seem to have a small degree of blurring. In fact, we do obtain visually sharper versions of these images if we perform deconvolution with the estimated kernels.

follows a stationary ARMA process, then the variance of $\mathbf{Z}_\omega$ will be non-constant, given by some function g_ω such that $\text{Var}(\mathbf{Z}_\omega) = \sigma^2 g_\omega$, where g_ω is determined by the correlation structure of $\mathbf{z}$. In addition, if the errors in the original image, given by $\boldsymbol{\varepsilon}$ in (6.1), are assumed to be IID, then its gradient $\mathbf{n}$ will have correlated elements. In particular, successive elements in the direction of the gradient will have correlation 0.5, while all other pairs will be uncorrelated. This will induce a non-constant variance for $\mathbf{N}_\omega$ as well, given by a function h_ω such that $\text{Var}(\mathbf{N}_\omega) = \eta^2 h_\omega$. Assuming that $\mathbf{Z}_\omega \sim \mathbb{CN}(0, \sigma^2 g_\omega)$ and $\mathbf{N}_\omega \sim \mathbb{CN}(0, \eta^2 h_\omega)$, the maximum likelihood estimate of $\mathbf{K}_\omega$ is then given by

$$\hat{\mathbf{K}}_\omega = \frac{1}{\sigma} \sqrt{\max(0, |\mathbf{Y}_\omega|^2 - \eta^2 h_\omega) / g_\omega}, \quad (5.4)$$

which can be similarly inverted to give an estimate of $\mathbf{k}$, provided we know g_ω and h_ω . h_ω can be calculated explicitly, and is given by

$$h_{(\omega_1, \omega_2)} = |e^{-i\omega_1} - 1|^2 \cdot |e^{-i\omega_2} - 1|^2 \text{ for } \omega_1, \omega_2 \in (0, 2\pi),$$

although in practice it has little effect. g_ω can also be computed explicitly from a known correlation structure. The blur kernels shown in the second row of Figure 5.7 are estimated using (5.4) from the eight test images using the g_ω given in Lemma 5.2.1, with correlation parameters 0.3 and 0.6 along and across the direction of the gradient. These kernels do not feature the cross-shaped artifacts in the naive estimate, suggesting that the effect of gradient correlation has been largely accounted for. The first three images seem to have a small degree of blurring, and this is confirmed visually by performing deconvolution with the estimated kernels, which produces visually sharper images (see Figure 6.4).

5.4 Application: estimation of kernel size

In blind deconvolution methods, although the kernel is assumed to be unknown, the kernel size is usually pre-specified by the user. In practice, the kernel size varies from image to image and is not known in advance. It is possible in principle to specify a larger kernel size than necessary, but as the main parameter of interest is the unknown blur kernel, this increases the dimension of the parameter space, which in turn increases variability in estimation as well as computational time. Thus, it is useful to be able to obtain reasonable estimates of the kernel size from the blurred image itself.

Recall that the symmetrized version of $\mathbf{k}$, $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$, has real-valued DFT, with the coefficient at frequency pair ω given by $|\mathbf{K}_\omega|^2$. The estimate of $\mathbf{K}_\omega$ given by (5.4) readily generalizes to

$$\hat{L}_\omega = |\hat{\mathbf{K}}_\omega|^2 = \frac{\max(0, |\mathbf{Y}_\omega|^2 - \eta^2 h_\omega)}{\sigma^2 g_\omega}, \quad (5.5)$$

the pointwise maximum likelihood estimate when $\mathbf{X}_\omega$ are independent Gaussian, an assumption that holds asymptotically even in the sparse MOG prior case. This allows us to obtain a crude estimate of $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ from an observed blurred image by applying inverse DFT on the matrix $\hat{L}$. Figures 5.8–5.11 plot the estimates so obtained for synthetically generated images, for both the IID and the generalized prior, and compares them to the true values of $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$. Note that although the matrix $\hat{L}$ and hence its inverse DFT is as large as the image, the plots are

zoomed to display only the part that is known to be non-zero (which is possible here only because the sizes of the true blur kernels are known), with negative entries set to 0.

Figure 5.8 shows estimates obtained using the IID prior ($g_\omega \equiv 1$) for exact convolutions of test images and test kernels. Figure 5.9 shows corresponding estimates using the g_ω given in Lemma 5.2.1 corresponding to the auto-regressive model with $\rho_1 = 0.3$ and $\rho_2 = 0.8$, for the same set of images. As there is no noise in the blurred image, η^2 is taken to be zero in (5.5).

Figure 5.10 similarly shows estimates obtained using the IID prior for noisy convolutions of test images and test kernels (see Figure 1.7), and Figure 5.11 shows corresponding estimates using the g_ω corresponding to the auto-regressive model with $\rho_1 = 0.3$ and $\rho_2 = 0.8$. Here we take $\eta^2 = 0.01^2$ in (5.5).

As can be seen from these plots, the estimates obtained assuming the IID prior are diffused and blurry with cross-shaped artifacts similar to those seen in the previous section, whereas those using the generalized prior are sharper and more sparse, and relatively free from artifacts. As evident from the ACF and PACF plots shown earlier in Figure 5.4, the same values of ρ_1 and ρ_2 may not be appropriate for all images. We would usually have no prior information about appropriate values of ρ_1 and ρ_2 , but looking at plots of $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ for various values of ρ_1 and ρ_2 can provide valuable visual guidance in making a suitable choice using the blurred image.

Such plots can also provide guidance regarding the *size* of the blur kernel, which is also unknown in practice but assumed to be known during estimation. As can be seen in Figures 5.8 and 5.10, the diffuse estimates obtained using the IID prior can overflow the true support of $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$, whereas those obtained using the generalized prior are relatively more compact. Plotting estimates of $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ for various values of ρ_1 and ρ_2 can thus also help in choosing the kernel size, noting that the size of $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ is approximately double the size of $\mathbf{k}$. For examples of such choices made for real blurred images, see Figure 7.10 in Section 7.3.

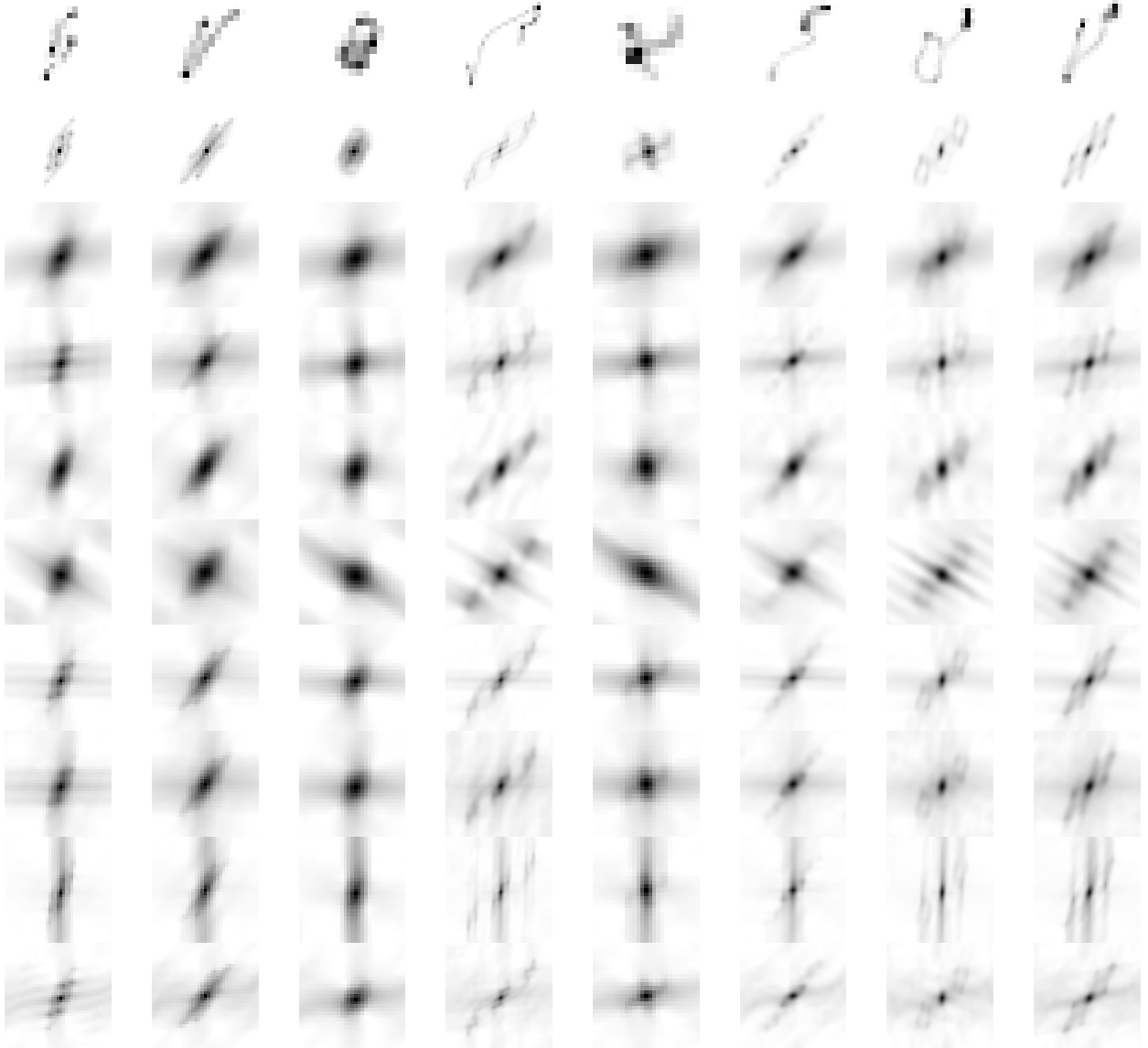


Figure 5.8: Symmetrized blur kernels estimated using the IID prior model from synthetically blurred images without additional noise. The top two rows shows the true blur kernel $\mathbf{k}$ and the corresponding symmetrized kernel $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$. The remaining eight rows plot $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ estimated from blurred images using the IID model; i.e., $g_\omega \equiv 1$ in (5.5). As the blurred images have no noise, we take $\eta^2 = 0$.

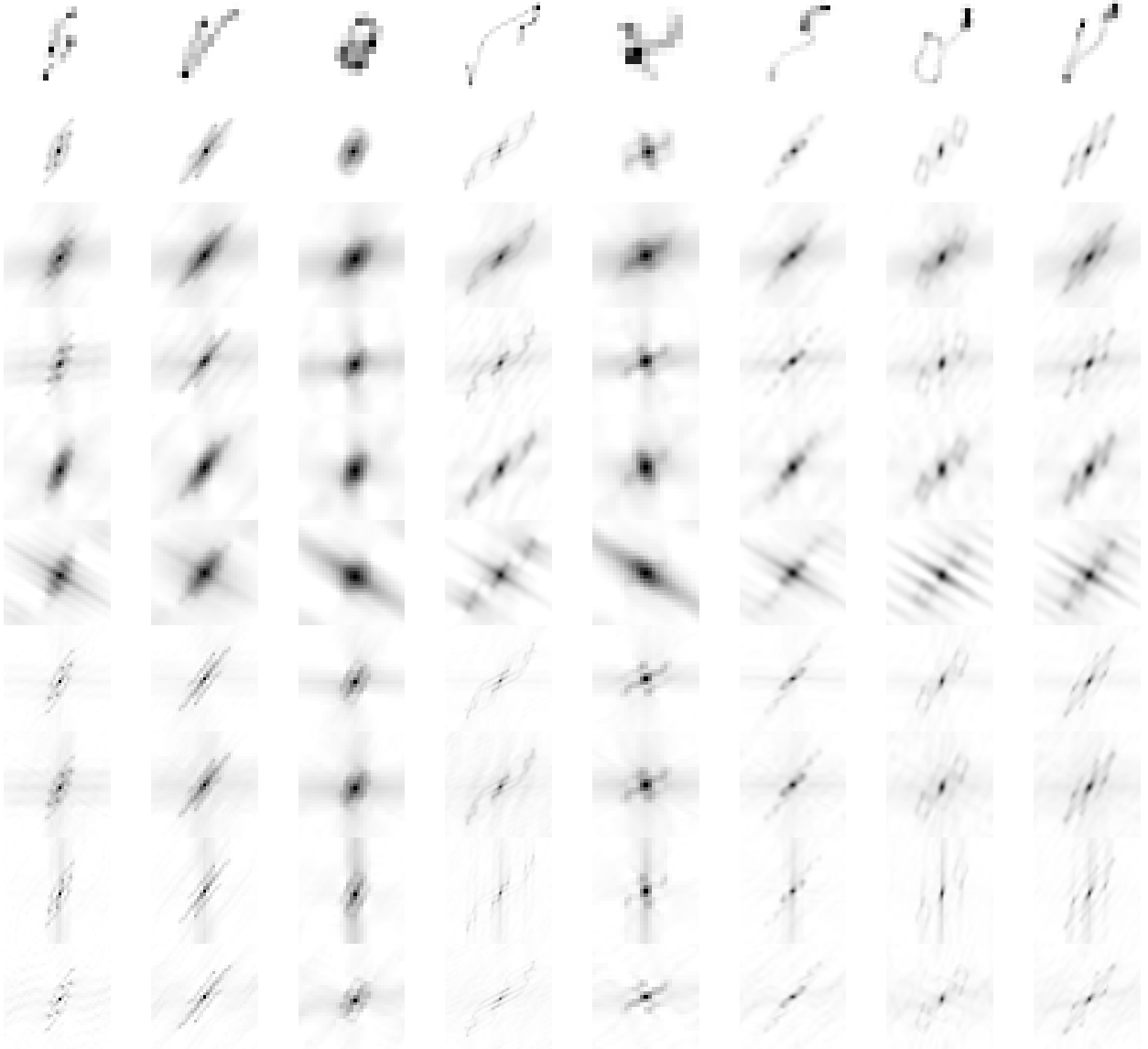


Figure 5.9: Symmetrized blur kernels estimated using the dependent prior model from synthetically blurred images without additional noise. The top two rows shows the true blur kernel $\mathbf{k}$ and the corresponding symmetrized kernel $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$. The remaining eight rows plot $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ estimated from blurred images using g_ω obtained from the auto-regressive model in (5.5). As the blurred images have no noise, we take $\eta^2 = 0$.

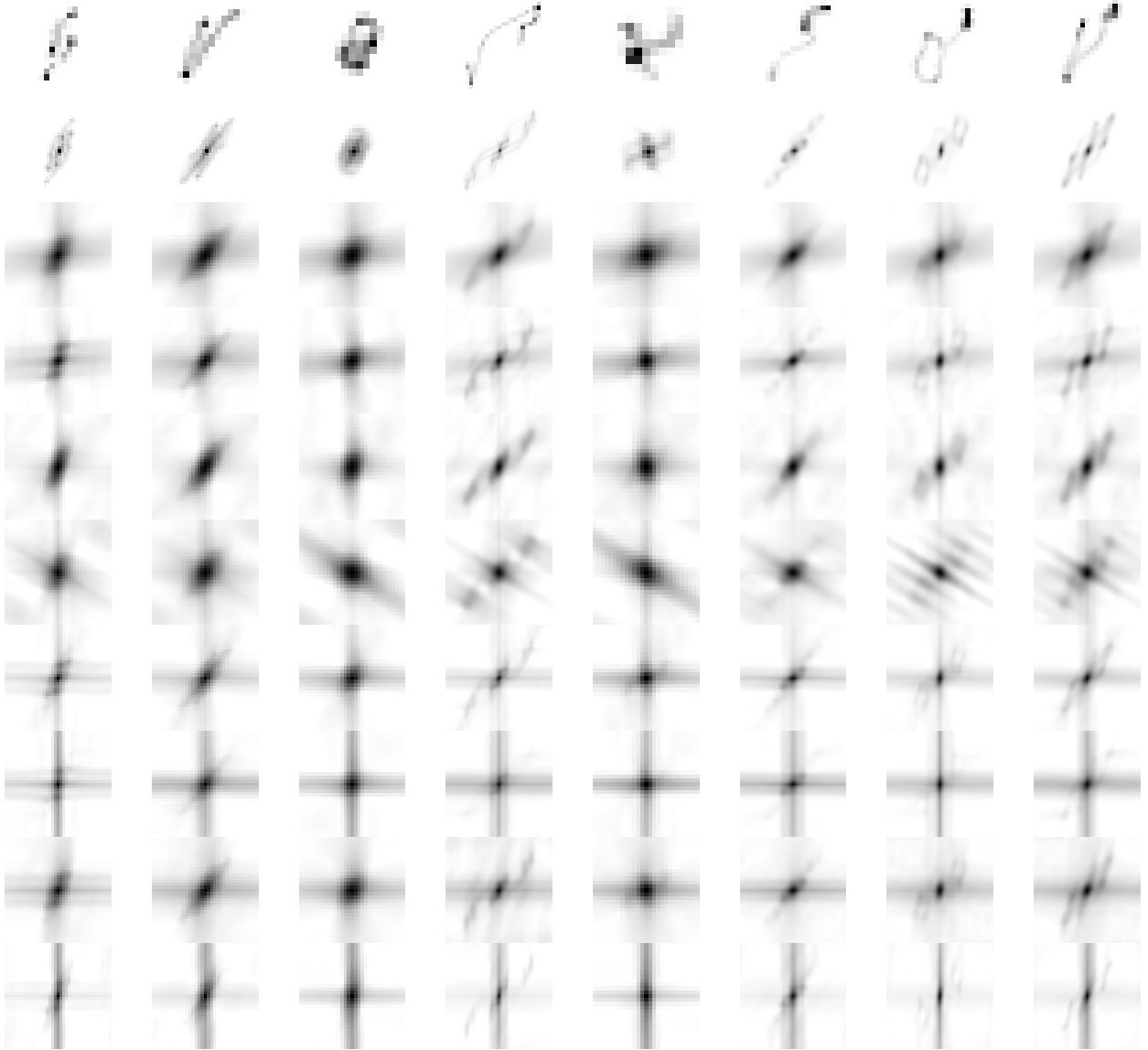


Figure 5.10: Symmetrized blur kernels estimated using the IID prior model from synthetically blurred images with additional noise ($S = 100$). As before, the top two rows shows the true blur kernel $\mathbf{k}$ and the corresponding symmetrized kernel $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$. The remaining eight rows plot $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ estimated from noisy blurred images using the IID model. The noise variance is taken to be $\eta^2 = 0.01^2$.

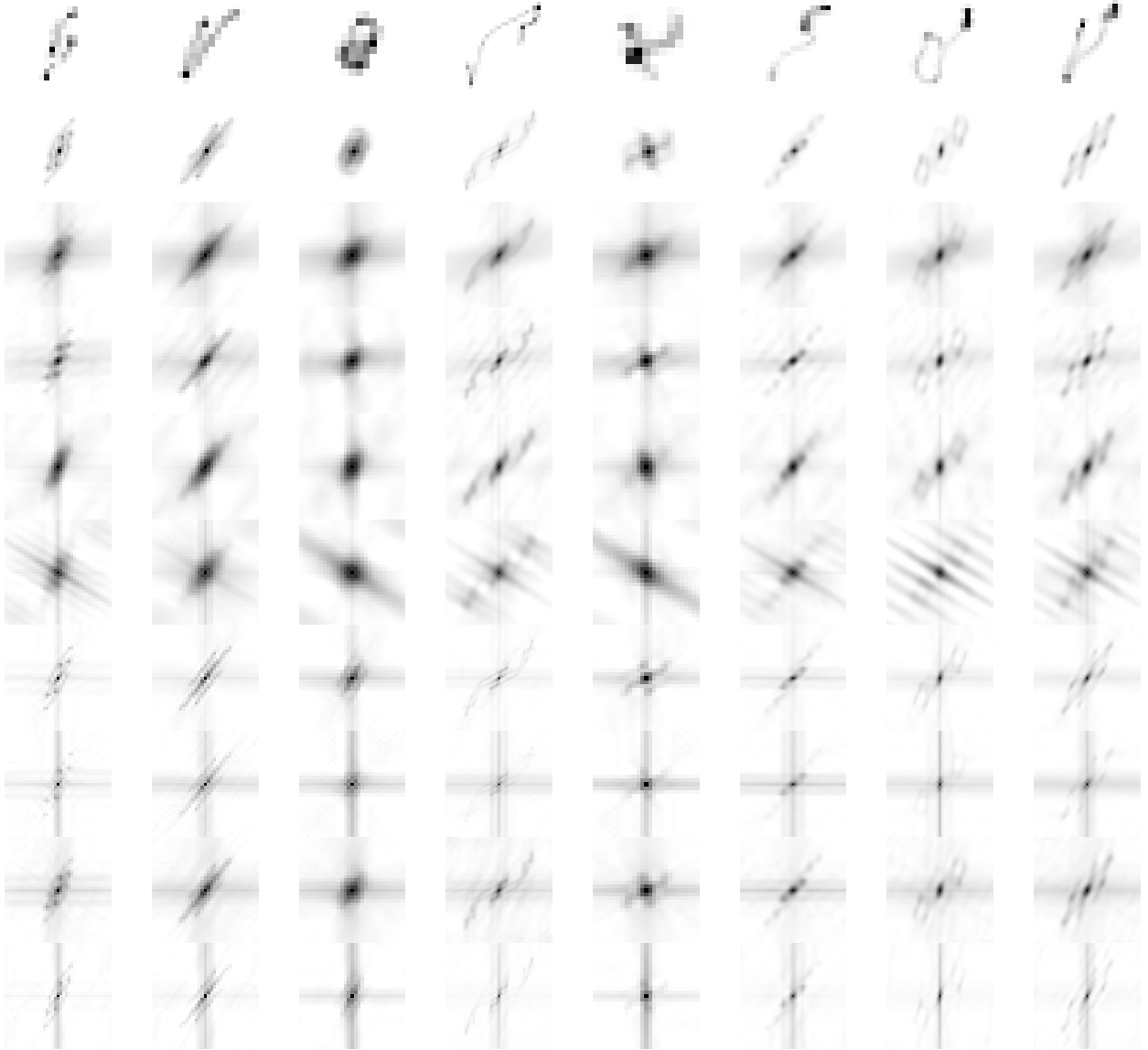


Figure 5.11: Symmetrized blur kernels estimated using the dependent prior model from synthetically blurred images with additional noise ($S = 100$). As before, the top two rows shows the true blur kernel $\mathbf{k}$ and the corresponding symmetrized kernel $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$. The remaining eight rows plot $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ estimated from noisy blurred images using the auto-regressive model with $\rho_1 = 0.3$ and $\rho_2 = 0.8$. The noise variance is taken to be $\eta^2 = 0.01^2$.

Chapter 6

Non-blind deconvolution and super-resolution using dependent prior

The main contribution of this thesis is to demonstrate how the locally dependent gradient prior motivated in the previous chapter can be used for various image reconstruction problems. In this chapter, we describe applications of the generalized prior in the context of non-blind deconvolution and super resolution problems. We start by describing a unified approach that incorporates both these problems, and outline our proposed method in the subsequent sections. Chapter 7 discusses the more difficult blind deconvolution problem, where the challenges are of a different nature. Chapter 8 discusses a few natural but simple extensions such as missing observations in the input (inpainting) and robust loss functions.

6.1 A unified approach

Several standard image reconstruction problems can be formulated as linear error minimization problems that are by nature under-constrained, and require some kind of regularization or prior to solve. Such problems typically involve an unknown $m \times n$ latent image ℓ , an observed $M \times N$ image $\mathbf{y}$, and a linear process represented by a $MN \times mn$ matrix A (assumed to be known),

such that $\mathbf{y}$ can be modeled as

$$\text{vec}(\mathbf{y}) = A \text{vec}(\boldsymbol{\ell}) + \text{vec}(\boldsymbol{\varepsilon})$$

where $\boldsymbol{\varepsilon}$ represents image noise added when observing $\mathbf{y}$, and $\text{vec}(\mathbf{y})$, $\text{vec}(\boldsymbol{\varepsilon})$, and $\text{vec}(\boldsymbol{\ell})$ denote corresponding vectorized forms of matrices. Examples of such problems we consider in this chapter are:

- *Non-blind image deconvolution.* Here the observed image $\mathbf{y}$ is assumed to have been distorted, usually due to camera shake during exposure or lens imperfections, by a blurring process that is modeled as a convolution with a blur kernel $\mathbf{k}$. Here we consider the case when the blur kernel is known. A is determined by $\mathbf{k}$, and the dimension of $\boldsymbol{\ell}$ is only slightly larger than that of $\mathbf{y}$, so as to include all the pixels in $\boldsymbol{\ell}$ that contributed to $\mathbf{y}$.
- *Super-resolution.* Here $\boldsymbol{\ell}$ is a high-resolution image, from which the low-resolution image $\mathbf{y}$ is assumed to have been obtained through a process of blurring and downsampling. Here the disparity in the sizes of $\boldsymbol{\ell}$ and $\mathbf{y}$ is typically much higher, controlled by the downsampling factor.
- *Image denoising.* This can be considered as a special case of both the problems above. A can be the identity matrix, and $\boldsymbol{\ell}$ and $\mathbf{y}$ have the same dimensions, with the goal of finding a solution that is less noisy and visually more plausible, using some kind of image prior. It is often useful to allow for a small blur kernel as well.

Assuming an IID Gaussian distribution for the noise $\boldsymbol{\varepsilon}$, the problem is easily identified with the standard least squares problem

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where the $n \times p$ model matrix X , equivalent to A in the formulation above, has $p > n$, and consequently the normal equations

$$X^T X \boldsymbol{\beta} = X^T \mathbf{y}$$

have infinitely many solutions. A common approach to resolve this problem is ridge regression or Tikhonov regularization, which can be viewed as a Bayesian approach that assumes an IID Gaussian distribution for the elements of β , or equivalently the latent image ℓ in the earlier formulation. In the context of natural images, i.e., photographic images of natural scenes, it has been more useful to consider IID priors on image gradients instead. For IID Gaussian priors, the regularised estimation problem reduces to solving linear equations of the form

$$\left(A^T A + \lambda \sum_j B_j^T B_j \right) \text{vec}(\ell) = A^T \text{vec}(\mathbf{y}),$$

where B_j -s represent gradient operators. Non-Gaussian priors such as the hyper-Laplacian family can be handled by simple IRLS (iteratively re-weighted least squares) algorithms that solve weighted versions of the Gaussian problem in each iteration.

This approach is not new, and has been used for both non-blind deconvolution [42] and super-resolution [16]. Solving the linear equations directly is difficult; for example, a deconvolution problem with a 200×200 input image $\mathbf{y}$ will lead to a linear problem with at least $200^2 = 40000$ variables, in turn leading to a 40000×40000 coefficient matrix. Similarly, up-scaling a 200×200 image by a factor of two will involve solving for $400^2 = 160000$ variables. In practice, iterative optimization methods such as conjugate gradient are used instead to solve the linear equations [19, 60].

We describe three novel contributions in the context of this problem. First, we show that generalizing the IID prior on image gradients by introducing local dependence between gradients leads to improved estimates of the latent image, both in quantitative terms and in terms of visual plausibility.

Further, for super-resolution or denoising where misspecification of the blur kernel can be more problematic than the choice of prior, we demonstrate that the simple estimates of the blur kernel described in Section 5.3, assuming symmetry of the kernel, are usually quite effective.

Finally, we implement an alternative numerical approach that solves the linear equations *directly*, making use of existing numerical sparse matrix libraries, taking advantage of the fact that the coefficient matrix in these problems are typically sparse. The performance of this

approach is comparable in terms of speed and memory, and in certain cases, especially for non-blind deconvolution with complicated kernels, this approach often works better than the conjugate-gradient approach.

6.2 Mathematical formulation and derivations

Suppose we have a latent image ℓ , from which we obtain an observed image $\mathbf{y}$. The problem is fully defined through the prior on ℓ and the conditional distribution of $\mathbf{y}$ given ℓ . We describe the second component first. We henceforth interpret ℓ and $\mathbf{y}$ to denote both the matrix and vectorized forms (with successive columns stacked on top of each other), depending on context.

Conditional distribution of $\mathbf{y}$

We assume that the elements of ℓ represent grayscale image intensities on a scale between 0 and 1 (color images can be handled by processing the red, green, and blue channels separately). The observed image $\mathbf{y}$ is obtained from ℓ through the following mechanism. First, ℓ undergoes convolution with a blur kernel $\mathbf{k}$; the resulting image is denoted by $\mathbf{k} \otimes \ell$. We use the term convolution somewhat loosely, and sometimes use $\otimes$ to mean filtering, where $(\mathbf{k} \otimes \mathbf{x})_{kl} = \sum_{i,j} \mathbf{x}_{(k+i-1)(l+j-1)} \mathbf{k}_{ij}$, as opposed to strict convolution, where $(\mathbf{k} \otimes \mathbf{x})_{kl} = \sum_{i,j} \mathbf{x}_{(k-i+1)(l-j+1)} \mathbf{k}_{ij}$. It is easy to see that convolution with $\mathbf{k}$ is same as filtering with $\text{flip}(\mathbf{k})$, and vice versa, where $\text{flip}(\mathbf{k})$ is obtained from $\mathbf{k}$ by reversing the order of both rows and columns. Optionally, in the case of super-resolution, the resulting image is downsampled by an (integral) factor of f by retaining every f -th row and column and dropping the rest; the resulting image is denoted by $\text{d}_f^\downarrow(\mathbf{k} \otimes \ell)$. Finally, the observed $\mathbf{y}$ is obtained as

$$\mathbf{y} = \text{d}_f^\downarrow(\mathbf{k} \otimes \ell) + \varepsilon, \quad (6.1)$$

where ε represents independent noise that is assumed to be Gaussian.

It is easy to see that the mapping $\mathbf{x} \mapsto \text{d}_f^\downarrow(\mathbf{k} \otimes \mathbf{x})$ is linear when viewed as a matrix to matrix function. Thus, the map can be represented by left multiplication by a matrix, denoted by $T_{\mathbf{k},f}$,

operating on the vectorized version of the input $\mathbf{x}$. Both valid convolution (see Section 2.4) and downsampling reduce the size of the output relative to the input, so $T_{\mathbf{k},f}$ will in general have fewer rows than columns. Without downsampling ($f = 1$), $T_{\mathbf{k}} \equiv T_{\mathbf{k},1}$ is an almost Toeplitz matrix, with each element of $\mathbf{k}$ appearing exactly once in each row. Downsampling is achieved by dropping rows from $T_{\mathbf{k},1}$. The conditional distribution of $\mathbf{y}$ given $\boldsymbol{\ell}$ can thus be written as

$$\mathbf{y} \mid \boldsymbol{\ell} \sim \mathcal{N}(T_{\mathbf{k},f} \boldsymbol{\ell}, \eta^2 I) \quad (6.2)$$

where $T_{\mathbf{k},f}$ is a substantially sparse matrix and η^2 is the unknown error variance. We assume here that $\mathbf{k}$ is known. This is not always a reasonable assumption, and in practice $\mathbf{k}$ may need to be estimated in advance.

Prior distribution of $\boldsymbol{\ell}$

Needless to say, obtaining good results depends on formulating suitable priors for the latent image. For the problems discussed in this chapter, we focus on the hyper-Laplacian prior family with distribution given by (with unit scaling)¹

$$f_{\alpha}(z) = \frac{\alpha}{2\Gamma(1/\alpha)} e^{-|z|^{\alpha}}, z \in \mathbb{R}, \quad (6.3)$$

with $\alpha \in [0.5, 0.8]$ empirically found to be appropriate for natural images [43, 37].

The horizontal gradients $\mathbf{x}^{(h)} = [-1, 1] \otimes \boldsymbol{\ell}$ and vertical gradients $\mathbf{x}^{(v)} = [-1, 1]^T \otimes \boldsymbol{\ell}$ are assumed to be independent for simplicity, even though clearly they are not as they must integrate to the same image $\boldsymbol{\ell}$. Although the original latent image $\boldsymbol{\ell}$ is expected to have high spatial correlation, it is common to assume that taking gradients serves as an effective decorrelating filter, and thus the individual pixel elements $\mathbf{x}_i^{(h)}$ and $\mathbf{x}_i^{(v)}$ of the image gradients are also taken to be *independent*. However, as discussed in Chapter 5, this assumption is often unreasonable, and a simple generalized lag-1 auto-regressive model (Model 1 in Chapter 5), with correlation

¹For the blind deconvolution problem discussed in Chapter 7, we use a mixture-of-Gaussian prior as an approximation.

given by

$$\text{Cor}(\mathbf{x}_{i_1, i_2}, \mathbf{x}_{j_1, j_2}) = \rho_1^{|i_1 - j_1|} \rho_2^{|i_2 - j_2|}, \quad (6.4)$$

better models the joint distribution of image gradients in practice. Instead of estimating ρ_1 and ρ_2 from the image, we interpret these as hyperparameters of the prior that must be provided by the user. For the results presented here, we have used the values 0.3 and 0.6 along and across the direction of the gradient. It is useful to recall that just as a one-dimensional $AR(1)$ process with lag-1 correlation ρ can be decorrelated by applying the filter $\begin{bmatrix} -\rho & 1 \end{bmatrix}$, a two-dimensional stationary process with correlation given by (6.4) can be decorrelated using the filter

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_1 \rho_2 & -\rho_1 \\ -\rho_2 & 1 \end{bmatrix}$$

We denote by $\mathbf{z}^{(h)}$ and $\mathbf{z}^{(v)}$ the results of applying the corresponding filters $\boldsymbol{\rho}^{(h)}$ and $\boldsymbol{\rho}^{(v)}$ on $\mathbf{x}^{(h)}$ and $\mathbf{x}^{(v)}$ respectively, and assume that the individual elements of $\mathbf{z}^{(h)}$ and $\mathbf{z}^{(v)}$ are independent. The prior model for $\boldsymbol{\ell}$, in its generalized form, is thus given by

$$f(\boldsymbol{\ell}) = \left(\prod_i \frac{1}{\sigma} f_\alpha(\mathbf{z}_i^{(h)} / \sigma) \right) \left(\prod_j \frac{1}{\sigma} f_\alpha(\mathbf{z}_j^{(v)} / \sigma) \right) \quad (6.5)$$

where f_α is given by (6.3), σ is a scale parameter controlling the (common) variance of $\mathbf{z}^{(h)}$ and $\mathbf{z}^{(v)}$, and the indices i and j run over elements of $\mathbf{z}^{(h)}$ and $\mathbf{z}^{(v)}$ respectively. The crucial observation here is that the maps $\boldsymbol{\ell} \mapsto \mathbf{z}^{(h)} = \boldsymbol{\rho}^{(h)} \otimes [-1, 1] \otimes \boldsymbol{\ell}$ and $\boldsymbol{\ell} \mapsto \mathbf{z}^{(v)} = \boldsymbol{\rho}^{(v)} \otimes [-1, 1]^T \otimes \boldsymbol{\ell}$ are both linear, and given in vectorized form by

$$\begin{aligned} \mathbf{z}^{(h)} &= T_{\boldsymbol{\rho}^{(h)}} T_{[-1, 1]} \boldsymbol{\ell} = B_h \boldsymbol{\ell} \\ \mathbf{z}^{(v)} &= T_{\boldsymbol{\rho}^{(v)}} T_{[-1, 1]^T} \boldsymbol{\ell} = B_v \boldsymbol{\ell} \end{aligned}$$

Here, as with $T_{\mathbf{k}, f}$, B_h and B_v are substantially sparse, almost Toeplitz matrices.

The posterior mode of $\ell \mid \mathbf{y}$

For $\alpha = 2$, the hyper-Laplacian distribution in (6.5) reduces to the Gaussian distribution

$$B_h \ell, B_v \ell \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{2}\sigma^2 I\right)$$

Combining with (6.2), it is easy to verify that the posterior distribution of $\ell \mid \mathbf{y}$ is Gaussian with mean (and hence mode)

$$\left(T_{\mathbf{k},f}^T T_{\mathbf{k},f} + \frac{2\eta^2}{\sigma^2} (B_h^T B_h + B_v^T B_v)\right)^{-1} T_{\mathbf{k},f}^T \mathbf{y}. \quad (6.6)$$

The general ($\alpha \neq 2$) case can be handled by iteratively computing weighted versions of (6.6). To see this, it is simplest to phrase the problem as an explicit optimization problem maximizing the posterior

$$f(\ell \mid \mathbf{y}) \propto \exp\left\{-\frac{1}{2\eta^2}\|\mathbf{y} - T_{\mathbf{k},f}\ell\|^2\right\} \cdot \exp\left\{-\sum_i |\mathbf{z}_i^{(h)}|/\sigma\right\} \cdot \exp\left\{-\sum_j |\mathbf{z}_j^{(v)}|/\sigma\right\}$$

or equivalently, minimizing the negative log-posterior (ignoring terms not dependent on ℓ)

$$M(\ell) = \frac{1}{2\eta^2} (\ell^T T_{\mathbf{k},f}^T T_{\mathbf{k},f} \ell - 2\mathbf{y}^T T_{\mathbf{k},f} \ell) + \frac{1}{\sigma^\alpha} \sum_i |\mathbf{z}_i^{(h)}|^\alpha + \frac{1}{\sigma^\alpha} \sum_j |\mathbf{z}_j^{(v)}|^\alpha$$

where $\mathbf{z}^{(h)} = B_h \ell$ and $\mathbf{z}^{(v)} = B_v \ell$. To write the score equations obtained by setting $\nabla M = \mathbf{0}$, it is convenient to express $M(\ell)$ as

$$M(\ell) = \frac{1}{2\eta^2} (\ell^T T_{\mathbf{k},f}^T T_{\mathbf{k},f} \ell - 2\mathbf{y}^T T_{\mathbf{k},f} \ell) + \frac{1}{\sigma^\alpha} \Psi_\alpha(B_h \ell) + \frac{1}{\sigma^\alpha} \Psi_\alpha(B_v \ell)$$

where $\Psi_\alpha(\mathbf{z}) = \sum_k |\mathbf{z}_k|^\alpha$. Using the chain rule, we can then write

$$(\nabla M(\ell))^T = \mathbf{J}_M(\ell) = \frac{1}{2\eta^2} (2\ell^T T_{\mathbf{k},f}^T T_{\mathbf{k},f} - 2\mathbf{y}^T T_{\mathbf{k},f}) + \frac{1}{\sigma^\alpha} \mathbf{J}_{\Psi_\alpha}(B_h \ell) B_h + \frac{1}{\sigma^\alpha} \mathbf{J}_{\Psi_\alpha}(B_v \ell) B_v$$

where $\mathbf{J}_h(\mathbf{x})$ denotes the Jacobian matrix of a function h evaluated at $\mathbf{x}$. It is easy to verify that $\mathbf{J}_{\Psi_\alpha}^T(\mathbf{z}) = \nabla \Psi_\alpha(\mathbf{z})$ is a column vector with k th element given by

$$\frac{\partial}{\partial \mathbf{z}_k} |\mathbf{z}_k|^\alpha = \alpha |\mathbf{z}_k|^{\alpha-1} \text{sign}(\mathbf{z}_k),$$

which simplifies to $2\mathbf{z}_k$ when $\alpha = 2$ (the Gaussian case). This leads to the score equations (setting $\nabla M(\boldsymbol{\ell}) = \mathbf{0}$)

$$T_{\mathbf{k},f}^T T_{\mathbf{k},f} \boldsymbol{\ell} + \frac{\eta^2}{\sigma^\alpha} \left[B_h^T \nabla \Psi_\alpha(\mathbf{z}^{(h)}) + B_v^T \nabla \Psi_\alpha(\mathbf{z}^{(v)}) \right] = T_{\mathbf{k},f}^T \mathbf{y}.$$

For $\alpha = 2$, $\nabla \Psi_\alpha(\mathbf{z}^{(h)}) = 2B_h \boldsymbol{\ell}$ and $\nabla \Psi_\alpha(\mathbf{z}^{(v)}) = 2B_v \boldsymbol{\ell}$, so the score equations reduce to

$$T_{\mathbf{k},f}^T T_{\mathbf{k},f} \boldsymbol{\ell} + \frac{2\eta^2}{\sigma^2} \left[B_h^T B_h + B_v^T B_v \right] \boldsymbol{\ell} = T_{\mathbf{k},f}^T \mathbf{y}.$$

matching the posterior mean in (6.6) above. For general α , we can write the k th element of $\nabla \Psi_\alpha(\mathbf{z})$ as

$$\alpha |\mathbf{z}_k|^{\alpha-1} \text{sign}(\mathbf{z}_k) = \alpha \frac{|\mathbf{z}_k|^{\alpha-1} \text{sign}(\mathbf{z}_k)}{\mathbf{z}_k} \mathbf{z}_k = \alpha |\mathbf{z}_k|^{\alpha-2} \mathbf{z}_k.$$

Equivalently, in matrix notation,

$$\nabla \Psi_\alpha(\mathbf{z}) = W(\mathbf{z}) \mathbf{z},$$

where $W(\mathbf{z})$ is a diagonal matrix with k th diagonal element given by $\alpha |\mathbf{z}_k|^{\alpha-2}$. We can thus rewrite the score equations as

$$\left(T_{\mathbf{k},f}^T T_{\mathbf{k},f} + \frac{\alpha \eta^2}{\sigma^\alpha} \left[B_h^T W(B_h \boldsymbol{\ell}) B_h + B_v^T W(B_v \boldsymbol{\ell}) B_v \right] \right) \boldsymbol{\ell} = T_{\mathbf{k},f}^T \mathbf{y}. \quad (6.7)$$

This leads to an iteratively reweighted least squares scheme where the $(t+1)$ th iterate $\boldsymbol{\ell}^{(t+1)}$ is obtained by solving the weighted least squares problem (6.7) with $W(B_h \boldsymbol{\ell})$ and $W(B_v \boldsymbol{\ell})$ replaced by $W(B_h \boldsymbol{\ell}^{(t)})$ and $W(B_v \boldsymbol{\ell}^{(t)})$ respectively. We henceforth use $\lambda = \frac{\alpha \eta^2}{\sigma^\alpha}$ to denote the regularization or tuning parameter; in the Bayesian interpretation, λ^{-1} is a measure of the signal-to-noise ratio.

Solving the linear equations in practice

As indicated earlier, the set of linear equations that need to be solved are so large in practice that conventional direct methods do not work. The Gaussian case can be solved efficiently in the Fourier domain. However, this approach cannot incorporate weights, and so cannot be used for $\alpha \neq 2$.

Levin et al. [42] use the conjugate gradient method [19], [60], an iterative method with low memory requirements that solves the linear equations $Ax = b$ by evaluating Ax alone, not even requiring A to be explicitly computed. As the only linear operations involved in (6.7) are filters, subsampling, and multiplication by diagonal matrices, this can be done efficiently.

The conjugate gradient approach is usually quite effective, but requires a fixed number of iterations for practical use, and performance depends on the problem. Specifically, we see relatively poor performance with complicated blur kernels. As the large matrices involved are substantially sparse, we implemented an alternative approach that computes a direct solution using efficient sparse matrix computation techniques. In particular, we make use of the CHOLMOD library [10] through the Matrix package in R [3]. We provide an R package that implements both the direct solution and conjugate gradient method [54].

6.3 Results

In this section, we present some results using our proposed methods. We start with some real image examples, mainly to visually illustrate the effect of the various priors and kernels on the resulting estimates. More extensive results are available at a supplementary website.² We also present quantitative results using synthetic noisy images as input, generated from known latent images to which the reconstructed images can be compared.

²<https://www.isid.ac.in/~deepayan/rip/html/>

Visual effect of correlation

We first give examples of the super-resolution problem. Efrat et al. [16] show that the choice of the blur kernel $\mathbf{k}$ often plays a more important role than the choice of prior in super-resolution. This blur is typically an inherent property of the camera that is present even if there is no motion blur, and is not negligible even though it is usually small. When one has access to the camera, Efrat et al. [16] outline a procedure to estimate the camera blur, but this is usually not the case.

The effect of kernel misspecification is illustrated in Figure 6.1 using two images from the supplementary website³ of Efrat et al. [16] upscaled by a factor of $f = 2$ using an Epanechnikov kernel with bandwidth $h = 2$. Both upscaled images (cropped to show central part) exhibit ringing artifacts when using the IID Gaussian prior, which is typical when using a blur kernel more diffuse than it should be. The sparse priors (both IID and AR) lead to less prominent artifacts, although the effect of kernel misspecification is evident in the text contained in the first image. The sparse priors tend to produce smoother results as expected, but the IID prior tends to oversmooth, producing more “blocky” results.

The results are substantially improved if we use the kernels that are estimated from the input images, as seen in Figure 6.2. The Epanechnikov kernel also gives better results with smaller bandwidth; results for $h = 1.5$ and $h = 1$ (as well $\lambda = 0.01$) are included in the supplementary website.

An interesting observation here is that the Gaussian prior actually gives visually better results in parts of the image that contain texture. On the other hand, the sparse prior performs better in regions that are smooth, such as skin or uniform backgrounds.

As the input images in these two examples are themselves quite sharp, we can possibly obtain even better results with a lower value of λ , making the prior less important. The choice of prior becomes more important for images that are less sharp. Figure 6.3 shows portions of a frame from the Buster Keaton film “*The Electric House*” (downloaded from the Internet Archive), upscaled using different priors; see the supplementary website for the full frame.

³<https://webee.technion.ac.il/people/anat.levin/papers/supres/>

Original images with Epanechnikov kernel ($h = 2$) inset



IID Gaussian ($\alpha = 2$) prior, $\lambda = 0.001$, Epanechnikov kernel with $h = 2$



IID sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, Epanechnikov kernel with $h = 2$



AR sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, Epanechnikov kernel with $h = 2$



Figure 6.1: Examples of super-resolution with user-supplied kernel. Here we use the Epanechnikov kernel, defined in terms of bandwidth h as $k_h(x, y) = (3/4h)^2(1-|x|/h)^2(1-|y|/h)^21\{|x| \leq h, |y| \leq h\}$, discretized by integrating over pixels. The blur kernel used ($h = 2$) is more diffuse than it should be, leading to noticeable ringing artifacts with the IID Gaussian prior. Uncropped results and results for other values of h are available in the supplementary website.

Original images with estimated kernels inset



IID Gaussian ($\alpha = 2$) prior, $\lambda = 0.001$, estimated kernel



IID sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, estimated kernel



AR sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, estimated kernel



Figure 6.2: Examples of super-resolution with estimated kernel, applied on the same images shown in Figure 6.1. The results are substantially better, although in this case similar results can be obtained using an Epanechnikov kernel with a smaller bandwidth. There is no clear advantage to using a sparse prior over a Gaussian prior, but the IID sparse prior clearly oversmooths compared to the correlated sparse prior.

Original image with estimated kernel inset (resized using cubic interpolation)



IID Gaussian ($\alpha = 2$) prior, $\lambda = 0.001$, estimated kernel



IID sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, estimated kernel



AR sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, estimated kernel



Figure 6.3: Super-resolution of a cropped frame from the Buster Keaton film *The Electric House*, using blur kernel estimated from the input image. The input image is substantially blurry, and noticeably better results are obtained with with the correlated sparse prior, while the IID Gaussian prior produces ringing artifacts.

The Gaussian prior produces prominent artifacts in this case, which are absent for the sparse prior. Again, the IID sparse prior oversmooths, whereas the correlated prior gives more realistic gradients.

These same observations generally hold for denoising with simple blur kernels as well. As we saw in Figure 5.7, our proposed method for estimating symmetric blur kernels suggests that the first three test images from Figure 1.4 have small amounts of blurring. Figure 6.4 shows the effect of performing non-blind deconvolution on these images using the corresponding estimated blur kernels.

Examples of more realistic blurring due to mild lack of focus are shown in Figure 6.5, which shows the result of denoising two real-life images that are slightly blurred, using blur kernels estimated from the input image. More examples are included in supplementary website.

Synthetic images

We next present quantitative comparisons of various priors in terms of how accurately they recover the true image when it is known. We use the eight test images shown in Figure 1.4 as latent images, and generate synthetic input images to evaluate performance in deconvolution as well as super-resolution.

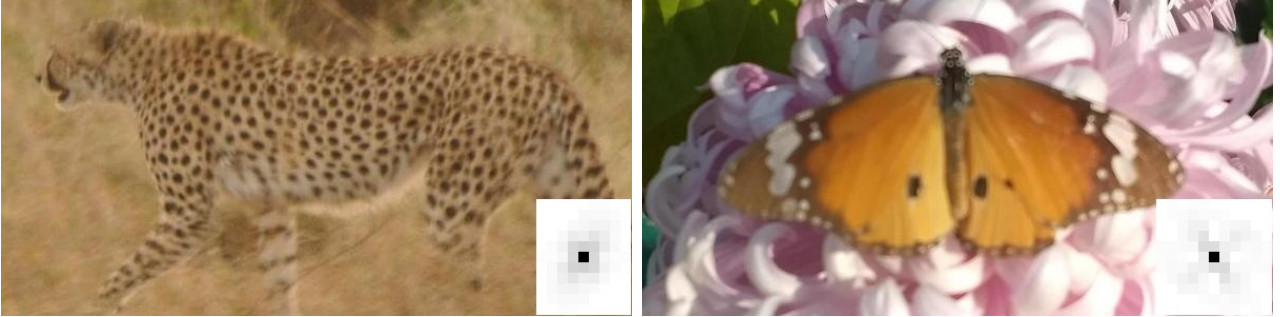
Non-blind deconvolution. The first simulation experiment assesses the performance of various priors for the non blind deconvolution problem. It uses the usual eight test images along with eight test kernels from Levin et al. [44], shown in Figure 1.5. For each of the 64 resulting combinations, we first compute a blurred image by blurring the test image with the test kernel. To investigate the effect of noise, these blurred images then have further noise added to them as described in Section 1.3. Specifically, if the intensity of a pixel in the blurred image is $L \in [0, 255]$, it is replaced by X/S , where X is a Poisson random variable with mean SL . In the results reported here, we use $S = 100$ (less noisy) and $S = 10$ (more noisy). This Poisson model is a natural noise model for the observed image, although it is not the noise model assumed during estimation.

Deconvolution is performed for each synthetic image with the true kernel, slightly altered



Figure 6.4: Deconvolution of the first three test images using the blur kernels estimated from the input images, shown in Figure 5.7.

Original image with estimated kernel inset



IID Gaussian ($\alpha = 2$) prior, $\lambda = 0.001$, estimated kernel



IID sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, estimated kernel



AR sparse ($\alpha = 0.8$) prior, $\lambda = 0.001$, estimated kernel



Figure 6.5: Denoising or deconvolution of real-life images using blur kernel estimated from the input image. See supplementary website for further examples.

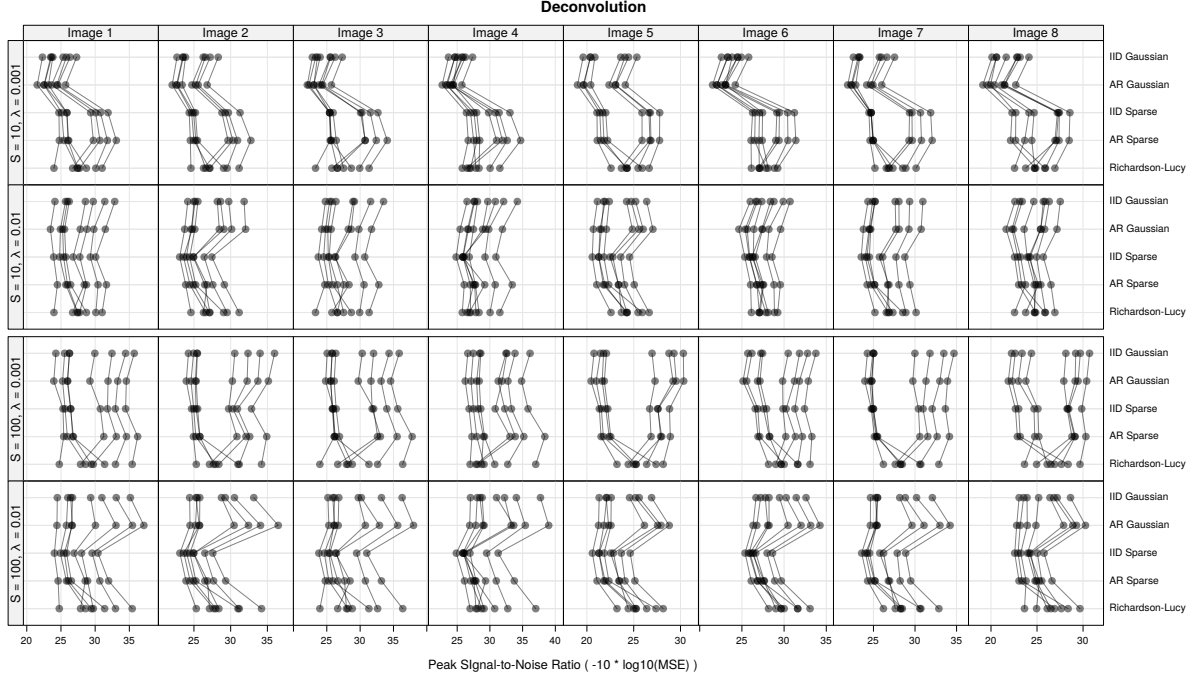


Figure 6.6: Performance of non-blind deconvolution for synthetic images, quantified through Peak Signal-to-Noise Ratio (PSNR) on the x -axis. Columns represent test images. Rows represent combinations of noise-level (S) and regularization parameter (λ). PSNR values corresponding to all eight test kernels are shown in each panel, with lines connecting values corresponding to the same kernel.

by setting small elements (less than 1% of the maximum) to zero. Four priors are used for image gradients: Gaussian ($\alpha = 2$) and sparse ($\alpha = 0.8$), both with IID and auto-regressive with correlation parameters 0.3 and 0.6 along and across the direction of the gradient. Two values of the tuning parameter λ are used, 0.01 and 0.001. In addition, the results of using the Richardson-Lucy algorithm (with 25 iterations) are also included.

The synthetic images are smaller than the input test images as boundary elements that require information from outside the image are dropped. The recovered images are of the same size as the generated synthetic image, and are compared with correspondingly cropped versions of the true latent image. Performance is quantified using the Peak Signal to Noise Ratio or PSNR, which for image values between 0 and 1 is defined as $-10 \log(\text{MSE})$, with MSE denoting the mean squared error. Higher values of PSNR, corresponding to lower values of MSE, indicate better performance in recovering the latent image. Results are summarized in Figure 6.6.

Somewhat surprisingly, the sparse priors do not always lead to better quantitative perfor-

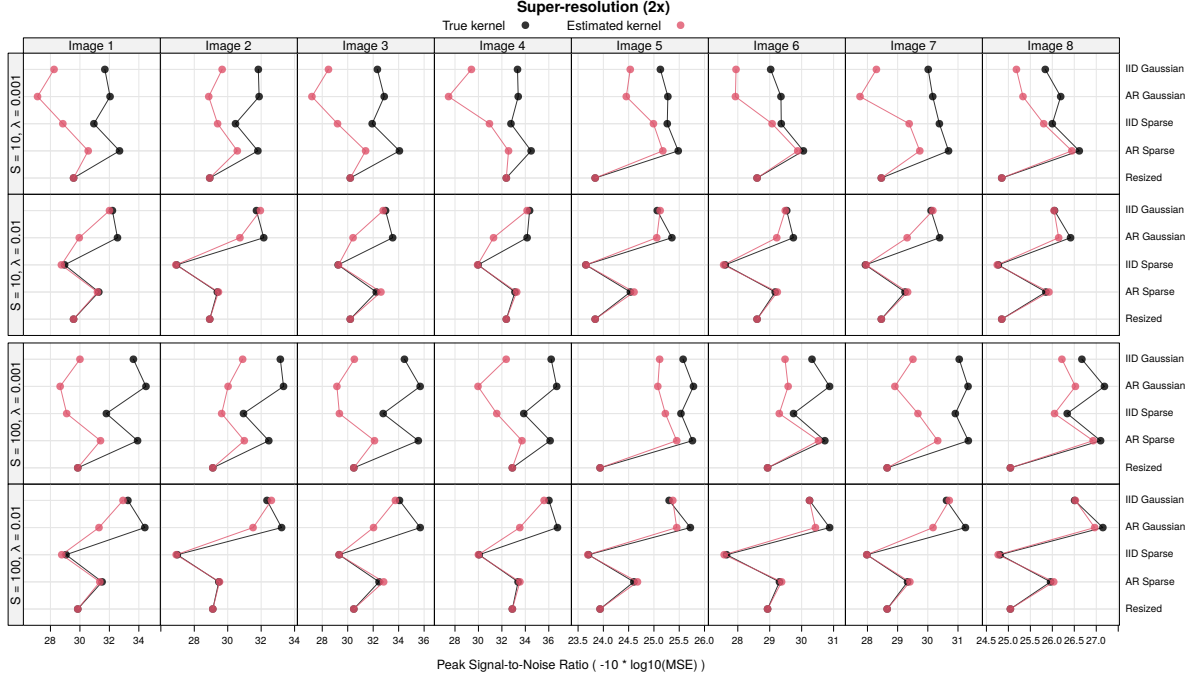


Figure 6.7: Performance of super-resolution for synthetic images, quantified through PSNR on the x -axis. Columns represent test images. Rows represent combinations of noise-level (S) and regularization parameter (λ). PSNR values corresponding to both the true kernel and estimated kernels are shown in each panel.

mance, even though they more closely reflect the true gradient distribution. One case where the sparse priors do perform better is with more noise ($S = 10$) and small regularization penalty ($\lambda = 0.001$). For the same images, similar performance is seen with $\lambda = 0.01$ and the IID Gaussian priors. For less noisy images ($S = 100$), all priors give roughly similar performance with $\lambda = 0.001$, but the AR Gaussian prior is the clear front-runner with $\lambda = 0.01$. Between the two sparse priors, the correlated AR model seems to be slightly better, as it is usually as good as and often better than the IID prior.

Super-resolution. The second simulation experiment similarly assesses performance for the super-resolution problem. The eight test images are downsampled by a factor of two after being blurred by a simple 3×3 rectangular kernel with bandwidth $h = 1.25$. The resulting images then have further noise added to them using the Poisson model as before, again using $S = 10$ and $S = 100$.

Super-resolution is performed for each of the 16 synthetic images thus obtained using the four gradient priors as before, with tuning parameter $\lambda = 0.01$ and 0.001 . In one set of experiments

the true kernel is used, and in another the kernel is estimated from the downsampled input image after upscaling it using linear interpolation. The estimated kernels are cropped to have size 5×5 pixels. The performance of the various combinations, quantified using PSNR, are summarized in Figure 6.7.

Here again the sparse priors are not necessarily better, although they do perform marginally better in the presence of more noise with a smaller regularization parameter. The correlated priors generally perform better than their IID counterparts. For $\lambda = 0.001$, the proper specification of the kernel seems to have an impact, whereas the estimated kernel works as well as the true kernel for $\lambda = 0.01$.

Of course, quantitative comparisons do not completely capture performance, and visual assessment is also important. The supplementary website includes the actual restored images, along with results for more example images processed with various parameter combinations. Our general observation is that $\lambda = 0.001$ with the AR sparse prior usually gives the best visually plausible results with the fewest artifacts, closely followed by $\lambda = 0.01$ with either the IID Gaussian or AR Gaussian priors.

Chapter 7

Blind deconvolution using dependent prior

In this chapter, we describe applications of the generalized prior model proposed in Chapter 5 in the context of blind deconvolution. We build upon the approach of Levin et al. [45] reviewed in Chapter 4. Although this is not the only approach that can be used with the prior we consider, it has the advantage that a well-documented code implementation is available, making it easier to assess improvements in results, if any.

In the previous chapter, we used the generalized prior formulated as a simple first-order auto-regressive model (Model 1 in Chapter 5) for non-blind image deconvolution and super resolution. Although this is the more natural generalization of the IID prior, it is difficult to incorporate into the EM approach considered here. Instead, in this chapter we consider the alternative model specified by the variance function g_ω of the DFT coefficients of image gradients (Model 2). For the specific parametric model derived from a first-order auto-regressive model, g_ω is determined by two parameters ρ_1 and ρ_2 that determine correlation along and across the direction of the image gradient.

7.1 Estimating blur kernel using generalized prior

We first discuss how the kernel estimation procedure can be extended to incorporate the generalized prior model. Results obtained from synthetic and real life images are given in subsequent sections.

Gaussian prior For the Gaussian prior given by (4.2), incorporating the generalization into the estimation process is relatively simple. Recall from Chapter 4 that the crucial step in the process is the E-step, where $\boldsymbol{\mu} = \mathbb{E}_{\mathbf{k}(t)}(\mathbf{x} | \mathbf{y})$ and $\boldsymbol{\Sigma} = \text{Var}_{\mathbf{k}(t)}(\mathbf{x} | \mathbf{y})$ are computed. These calculations can be performed efficiently in the Fourier domain, as described earlier in Section 4.3.

For the generalization given by Model 2, the underlying $\mathbf{z}$ matrix has IID $N(0, \sigma^2)$ components, and consequently the DFT coefficients $\mathbf{Z}_\omega$ are also IID complex Gaussian, i.e.,

$$\mathbf{Z}_\omega \sim \mathbb{CN}(0, \sigma^2),$$

with the same variance σ^2 provided $\mathbf{Z}$ is suitably normalized. By Model 2, $\mathbf{X}_\omega = \sqrt{g_\omega} \mathbf{Z}_\omega$, and we thus have, analogous to the calculations in Section 4.3, $\mathbf{X}_\omega \sim \mathbb{CN}(0, g_\omega \sigma^2)$ and $\mathbf{Y}_\omega | \mathbf{X}_\omega, \mathbf{K}_\omega \sim \mathbb{CN}(\bar{\mathbf{K}}_\omega \cdot \mathbf{X}_\omega, \eta^2)$, with suitably normalized $\mathbf{X}$ and $\mathbf{Y}$. It easily follows that for the generalized prior, the analogue of (4.13) is given by

$$\mathbf{X}_\omega | \mathbf{Y}_\omega, \mathbf{K}_\omega \sim \mathbb{CN} \left(\frac{\sigma^2 g_\omega \mathbf{K}_\omega \mathbf{Y}_\omega}{\eta^2 + \sigma^2 g_\omega |\mathbf{K}_\omega|^2}, \frac{\eta^2 \sigma^2 g_\omega}{\eta^2 + \sigma^2 g_\omega |\mathbf{K}_\omega|^2} \right). \quad (7.1)$$

$\boldsymbol{\mu} = \mathbb{E}(\mathbf{x} | \mathbf{y}, \mathbf{k})$ is then obtained by simply taking the inverse DFT of $\mathbb{E}(\mathbf{X} | \mathbf{Y}, \mathbf{K})$. As before, $\boldsymbol{\Sigma}$ is assumed to be diagonal. All diagonal entries are identical and are given by

$$\frac{1}{\Sigma_{ii}} = \frac{1}{\sigma^2 \bar{g}} + \frac{\sum \mathbf{k}_j^2}{\eta^2}.$$

where $\bar{g}$ is the average of g_ω taken over all DFT frequencies.

Sparse MOG prior Unfortunately, the modification is not as straightforward for the sparse MOG prior (4.3). In this case, the implementation of the E-step depends heavily on compu-

tations in the image domain, and the specification of the generalization in the Fourier domain does not easily translate to the image domain, which would require modeling the dependence between the components of the transformed image derivatives $\mathbf{x}$, obtained from $\mathbf{z}$ by transformation, in terms of the IID sparse MOG distribution of the components of $\mathbf{z}$.

The approach we take here is different, and although approximate, it is potentially more useful in other similar contexts. Instead of directly modifying the E-step, we proceed in two steps. First, we transform the input image derivative $\mathbf{y}$ so that the assumptions of the original IID model are approximately satisfied. This step can be viewed as indirectly *imputing* $\mathbf{z}$ (which has IID components) from $\mathbf{x}$ (which has dependent components). Second, we apply the original E-step on the transformed problem.

Specifically, recall that in Model 2, the unblurred image gradient $\mathbf{x}$ is a transformation of another unobserved process $\mathbf{z}$ that satisfies the IID prior model $\mathbf{z}_i \sim \pi(\mathbf{z})$. Had we observed $\mathbf{u} = \mathbf{k} \otimes \mathbf{z} + \mathbf{n}_u$, we would already know how to estimate $\mathbf{k}$; however, we instead observe $\mathbf{y} = \mathbf{k} \otimes \mathbf{x} + \mathbf{n}_y$. To estimate $\mathbf{k}$ from $\mathbf{y}$, we modify the E-step as follows:

1. Given the current estimate $\mathbf{k}^{(t)}$ of $\mathbf{k}$, compute $\mathbf{u}^{(t)} = \mathbb{E}(\mathbf{u} | \mathbf{y}, \mathbf{k}^{(t)})$. This step can be executed efficiently in the Fourier domain, as shown below.
2. Proceed to compute $\boldsymbol{\mu} = \mathbb{E}_{\mathbf{k}^{(t)}}(\mathbf{z} | \mathbf{u}^{(t)})$ and $\boldsymbol{\Sigma} = \text{Var}_{\mathbf{k}^{(t)}}(\mathbf{z} | \mathbf{u}^{(t)})$ (i.e., replace $\mathbf{y}$ by $\mathbf{u}^{(t)}$ in the original E-step) and use them to compute $Q(\mathbf{k} | \mathbf{k}^{(t)})$ based on the IID process $\mathbf{z}$, which we already know how to do.

The computations required for the first step above are facilitated by the following theorem.

Theorem 7.1.1 *Let $Z \sim \mathbb{CN}(0, \sigma^2)$, and given Z , let U and Y be defined as $U = KZ + \varepsilon_u$ and $Y = \alpha KZ + \varepsilon_y$ where $\alpha \in \mathbb{R}$ and $\varepsilon_u, \varepsilon_y$ are independently distributed as $\mathbb{CN}(0, \eta^2)$. Then, the conditional expectation of U given Y is*

$$\mathbb{E}(U | Y) = \frac{\alpha |K|^2 \sigma^2}{\eta^2 + \alpha^2 |K|^2 \sigma^2} Y.$$

Proof: Clearly (U, Y) has a joint Complex Normal distribution with $\mathbb{E}(U) = 0$ and $\mathbb{E}(Y) = 0$,

and thus the required conditional expectation is given by

$$E(U | Y) = \frac{\text{Cov}(U, Y)}{\text{Var}(Y)} Y$$

The variance of Y is given by

$$\begin{aligned} \text{Var}(Y) &= E(Y\bar{Y}) \\ &= E E[(\alpha K Z + \varepsilon_y)(\overline{\alpha K Z + \varepsilon_y}) | Z] \\ &= E E[\alpha^2 K \bar{K} Z \bar{Z} + \alpha K Z \bar{\varepsilon}_y + \alpha \varepsilon_y \bar{K} \bar{Z} + \varepsilon_y \bar{\varepsilon}_y | Z] \\ &= \alpha^2 |K|^2 E[Z \bar{Z}] + 0 + 0 + E[\varepsilon_y \bar{\varepsilon}_y] \\ &= \alpha^2 |K|^2 \sigma^2 + \eta^2 \end{aligned}$$

Similarly, the covariance term is

$$\begin{aligned} \text{Cov}(U, Y) &= E(U\bar{Y}) \\ &= E E[(K Z + \varepsilon_u)(\overline{\alpha K Z + \varepsilon_y}) | Z] \\ &= E E[\alpha K \bar{K} Z \bar{Z} + K Z \bar{\varepsilon}_y + \alpha \varepsilon_u \bar{K} \bar{Z} + \varepsilon_u \bar{\varepsilon}_y | Z] \\ &= \alpha |K|^2 E[Z \bar{Z}] + 0 + 0 + 0 \\ &= \alpha |K|^2 \sigma^2 \end{aligned}$$

The result immediately follows.

One slightly unsatisfactory assumption in the previous result is that the errors for U and Y are assumed to be independent. As our primary interest is in recovering the effect of the scaling of the mean by $\alpha = \sqrt{g_\omega}$, it may be more reasonable to assume that the errors are related when predicting U from Y . We can extend the above result by relaxing the independence assumption, and instead assume that $E(\varepsilon_u \bar{\varepsilon}_y) = \rho \eta^2$ for some correlation ρ . This leads to the following theorem.

Theorem 7.1.2 *Let $Z \sim \mathbb{CN}(0, \sigma^2)$, and given Z , let U and Y be defined as $U = KZ + \varepsilon_u$ and*

$Y = \alpha KZ + \varepsilon_y$ where $\alpha \in \mathbb{R}$ and $(\varepsilon_u, \varepsilon_y)$ are jointly Complex normal with mean 0, variance η^2 , and $E(\varepsilon_u \bar{\varepsilon}_y) = \rho \eta^2$. Then, the conditional expectation of U given Y is

$$E(U | Y) = \frac{\alpha |K|^2 \sigma^2 + \rho \eta^2}{\eta^2 + \alpha^2 |K|^2 \sigma^2} Y.$$

The proof is identical to that of Theorem 7.1.1, except that the last term in the calculation of $\text{Cov}(U, Y)$ is $\rho \eta^2$ instead of 0.

We use the above results as follows. In the imputation step in the modified algorithm described above, we need to compute

$$\mathbf{u}^{(t)} = E(\mathbf{u} | \mathbf{y}, \mathbf{k}^{(t)})$$

given the current estimate $\mathbf{k}^{(t)}$ of $\mathbf{k}$. We instead compute

$$\mathbf{U}_\omega^{(t)} = E(\mathbf{U}_\omega | \mathbf{Y}_\omega, \mathbf{K}_\omega^{(t)})$$

for each frequency pair ω in the Fourier domain using Theorem 7.1.2, putting $\alpha = \sqrt{g_\omega}$, and the correlation parameter ρ to some user-supplied value. We use $\rho = 0.75$ in the results we report, and this seems to provide better results than $\rho = 0$. Once we impute the values of $\mathbf{U}_\omega$, we apply inverse DFT to obtain the imputed value of $\mathbf{u}$. One additional heuristic we apply is to scale the values of the resulting $\mathbf{u}$ to match the variance of $\mathbf{y}$, as this seems to give more stable results.

We end this section with a remark. Note that a naïve alternative to imputing $\mathbf{u}$ at every step of the iteration (using the current estimate of $\mathbf{k}$) is to make an initial one-time transformation of $\mathbf{Y}_\omega \mapsto \mathbf{U}_\omega = \mathbf{Y}_\omega / \sqrt{g_\omega}$. This is equivalent to applying Theorem 7.1.2 with $\eta^2 = 0$, i.e., if there is no noise. In fact, this approach gives good results in synthetic images where there is no additional noise. However, as expected, the results are not as good in the presence of noise, and in that case imputing in each step becomes important.

7.2 Performance comparison for synthetic images

To assess the performance of the generalized prior, we performed a study comparing our proposed method with the sparse IID method of Levin et al. [45], largely following their evaluation scheme. The study uses the eight test images and test kernels to create synthetic blurred images as described in Section 1.3 in Chapter 1, and then use the two methods to recover the blur kernels. The study consists of three experiments with different noise levels: The first replicates the scheme used by Levin et al. [45] to generate test images by convolving the test image and the test kernel without introducing any additional noise. The second experiment uses test images with additional noise, as outlined earlier in Section 1.3, with the amount of Poisson-type noise controlled by parameter $S = 100$. The third experiment uses test images with even more noise, generated with $S = 10$. The last two sets are the same as those used for testing the performance of non-blind deconvolution in Chapter 6.

Specifically, for the first set of experiments, 64 test input images are obtained and stored in PNG format by combining each of the 8 unblurred test images with each of the 8 test blur kernels. In this experiment, which we refer to as the “exact” setup, the blurred images have no additional noise apart from the integer rounding inherent in storing 8-bit images. The actual estimation procedure, as implemented, assumes that image intensities are scaled to lie between 0 and 1. With this scaling, the noise introduced in the image gradients by rounding, aggregated over all test images, has a standard deviation of approximately 0.0016. This noise is modeled as $N(0, \eta^2)$, and the default value of η used by Levin et al. [45] is 0.01.

To investigate the effect of noise, two further sets of 64 test images have noise added to them as follows: if the intensity of a pixel in the corresponding “exact” blurred image is an integer L between 0 and 255, it is replaced by X/S , where X is a Poisson random variable with mean SL , rounded to the nearest integer between 0 and 255. We refer to this experiment as the “noisy” setup. This is a natural noise model for the raw image, and we use two values of S , $S = 100$ (less noisy) and $S = 10$ (more noisy). The resulting noise in the image gradients, aggregated over all test images, has a standard deviation of approximately 0.005, half the default value of η , for $S = 100$, and of approximately 0.015 for $S = 10$.

Performance is evaluated for each test image as follows: (1) The blur kernel is estimated using the EM approach as outlined above. (2) The estimated blur kernel is then used in a non-blind deconvolution algorithm using the methods outlined in the previous chapter with three priors: (a) IID Gaussian ($\alpha = 2$) with $\lambda = 0.01$, (b) AR Sparse ($\alpha = 0.8$) with $\lambda = 0.001$, as well as (c) IID Sparse ($\alpha = 0.8$) with $\lambda = 0.001$. (3) Finally, the resulting recovered image is compared with the true unblurred image, and their similarity measured by a quantity SSDE (the sum of squared differences after finding the best shifted match) as described by Levin et al. [45]. Lower values of SSDE indicate a closer match between the true and recovered image.

Specifically, for the IID prior the sparse MOG model used is the default model used by Levin et al. [45] (mixture probabilities 0.3047, 0.4344, 0.2609 with corresponding variances $1/7021.68, 1/471.84, 1/41.85$). The noise parameter is taken to be $\eta = 0.01$ for the exact setup, $\eta = 0.0125$ for the first noisy set with $S = 100$, and $\eta = 0.0225$ for the second noisy set with $S = 10$ (these choices are somewhat arbitrary, chosen by trial and error, as they give better overall performance than the default). For the generalized prior, the additional model parameter g_ω is calculated for the auto-regressive model with $\rho_1 = 0.3$ (correlation parameter along the direction of the gradient) and $\rho_2 = 0.6$ (across the direction of the gradient). For the imputation step, correlation between the errors for $\mathbf{U}$ and $\mathbf{Y}$ is taken to be 0.75. To avoid potential scaling problems affecting the validity of the variance parameters, the imputed derivatives $\mathbf{u}$ are scaled so that their overall standard deviation matches that of the untransformed derivatives $\mathbf{y}$.

We naturally expect worse performance in the presence of noise. However, even for a given level of noise, the overall performance is affected by two factors: the choice of prior in the estimation of the *kernel*, and the choice of prior in the subsequent *non-blind* deconvolution step. The effect of the first factor, the choice of prior used in estimating the blur kernel, is summarized in Figure 7.1 for all three sets of images, and three different choices for the non-blind deconvolution step.

Some interesting observations can be drawn from this figure. For the exact setup, using the dependent (AR) prior seems to be beneficial for most images (the points lie below the

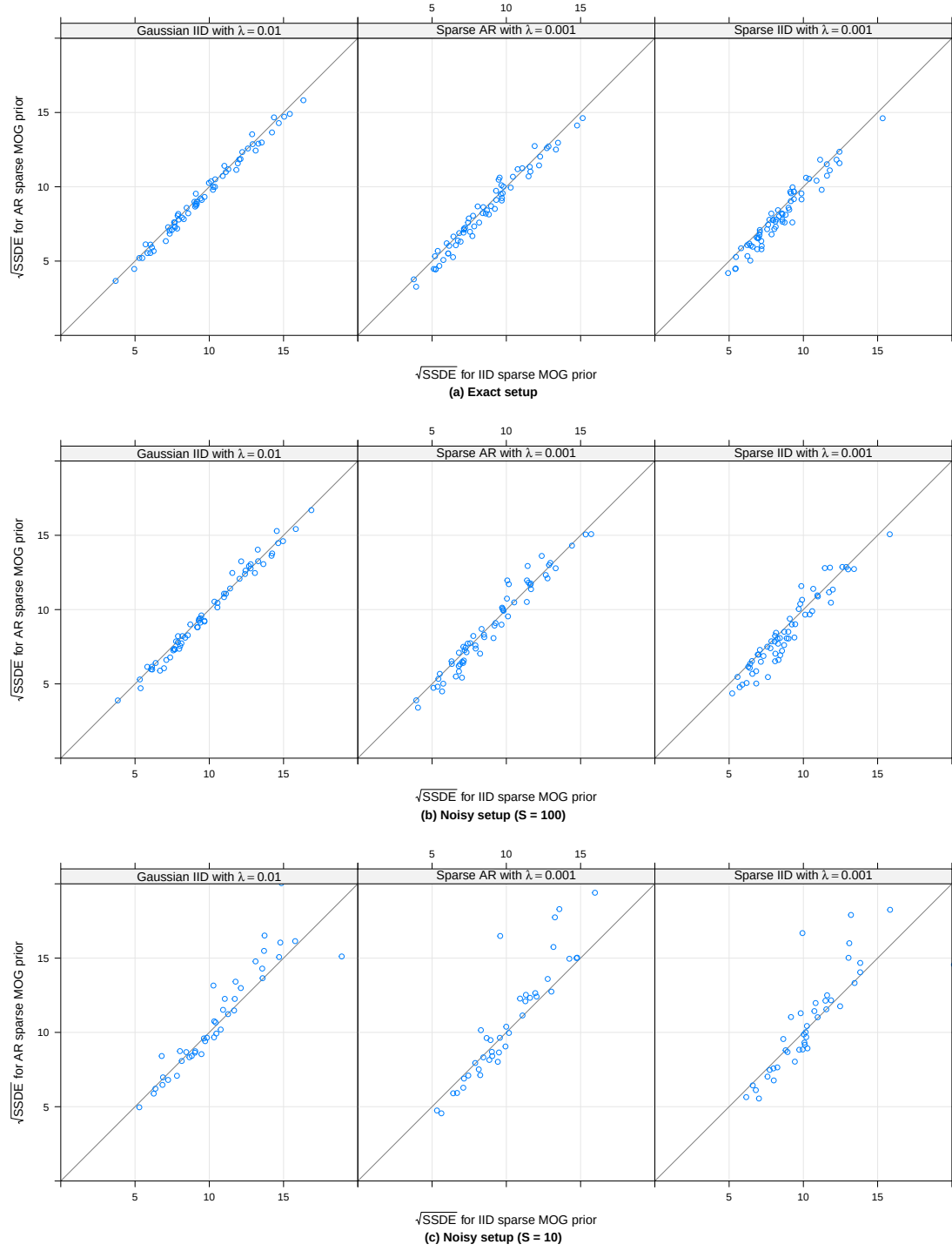


Figure 7.1: Plot summarizing performance of auto-regressive model prior with respect to performance using the IID prior when estimating the blur kernel. The three rows represent the type of blurred input image: (a) “exact” setup, (b) “noisy” setup with $S = 100$, (c) more “noisy” setup with $S = 10$. Each point in the scatter plot represents one of the 64 image-kernel combinations. The x axis plots the square root of the sum of squared differences (SSDE) between the original true image and the recovered image (for the best alignment) using the IID prior. The y axis plots the square root of the SSDE using the g_ω corresponding to an auto-regressive model with $\rho_1 = 0.3$ and $\rho_2 = 0.6$. Columns in the plots represent the prior used during *non-blind* deconvolution, after the blur kernel has been estimated. A diagonal line is added for reference.

diagonal), but the improvement in SSDE is minimal when deconvolution is done using the IID Gaussian prior. The improvement is slightly better when using the AR sparse prior, and even more when using the IID sparse prior in the non-blind deconvolution step. There is, however, a subtle point here. As observed in the previous chapter, deconvolution using the IID sparse prior leads to oversmoothed blocky results that are unrealistic, even though this is not reflected by SSDE as a measure of “goodness”.

The behaviour for noisy images are more interesting. Even though the dependent prior leads to lower SSDE for many images, in many cases it leads to higher SSDE, and this happens more when the SSDE value is high to begin with. In other words, when the IID prior is unable to estimate the kernel well, using the dependent prior usually makes things worse, and conversely it improves the result when the IID prior performs well. As seen in the subsequent figures showing the estimated kernels, higher noise sometimes leads to complete failure to estimate the kernel in some such cases; the resulting SSDE values are not shown in Figure 7.1, which excludes SSDE value larger than 400.

This leads to a natural question: given a method of estimating the blurred kernel, how much does the choice of prior used during the non-blind deconvolution step affect the result are measured by SSDE? This question is answered in Figure 7.2, which shows (for the exact setup) that the improvement due to the choice of the AR sparse prior during the non-blind deconvolution step (compared to the IID Gaussian prior) far exceeds the improvement due to differences in the estimated kernel. We exclude the IID sparse prior from this comparison because the results are visually unappealing, although it is numerically competitive with the AR sparse prior.

To enable a more detailed comparison, individual recovered kernels along with their SSDE values are shown in Figures 7.3 through 7.8. These show that despite the nominal improvements in SSDE, the use of a dependent prior does lead to sharper and visually improved kernel estimates in most cases. When the estimates are reasonably good, the kernel estimated using the dependent prior usually performs slightly better, but a much larger improvement is facilitated by use of a sparse dependent prior in the non-blind deconvolution step. However, when the

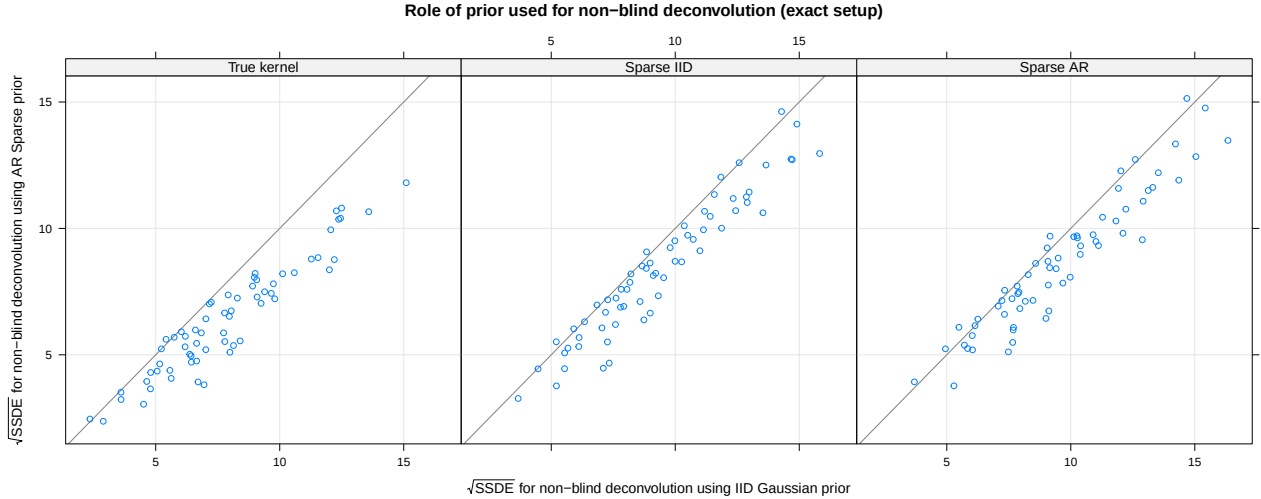


Figure 7.2: Role of prior used during non-blind deconvolution on performance using various kernel estimation methods. The first column shows performance when deconvolution is done using the true kernel. The second column uses the kernels estimated using the sparse IID prior, and the third column uses the kernels estimated using the sparse AR prior. The x axis plots the square root of SSDE between the true latent image and the recovered image (for the best alignment) using the IID Gaussian prior ($\alpha = 2, \lambda = 0.01$). The y axis plots the square root of the SSDE using the dependent (AR) sparse prior ($\alpha = 0.8, \lambda = 0.001$) with $\rho_1 = 0.3$ and $\rho_2 = 0.6$. A diagonal line is added for reference. The improvement due to the use of the dependent sparse prior is much larger than the differences due to prior used during estimation of the kernel.

estimates are *bad*, the dependent prior tends to have worse performance in both respects.

It is important to note that results are sensitive to the choice of parameters, and in particular the noise parameter η . Figure 7.9 shows the effect of using a value of η lower than it should be; specifically, using $\eta = 0.0125$ to estimate blur kernels from the more noisy set of images ($S = 10$) using the sparse IID prior. The results are clearly overly sparse and disconnected. The corresponding estimates for the AR Sparse prior are not shown but similar. The optimal value of η is not known in advance, and needs to be chosen by trial and error on a case-by-case basis.

7.3 Results on real-life examples

In this section we present results obtained from six real-life blurred images shown earlier in Figure 1.1. It is usually computationally impractical to use a full image to estimate the blur

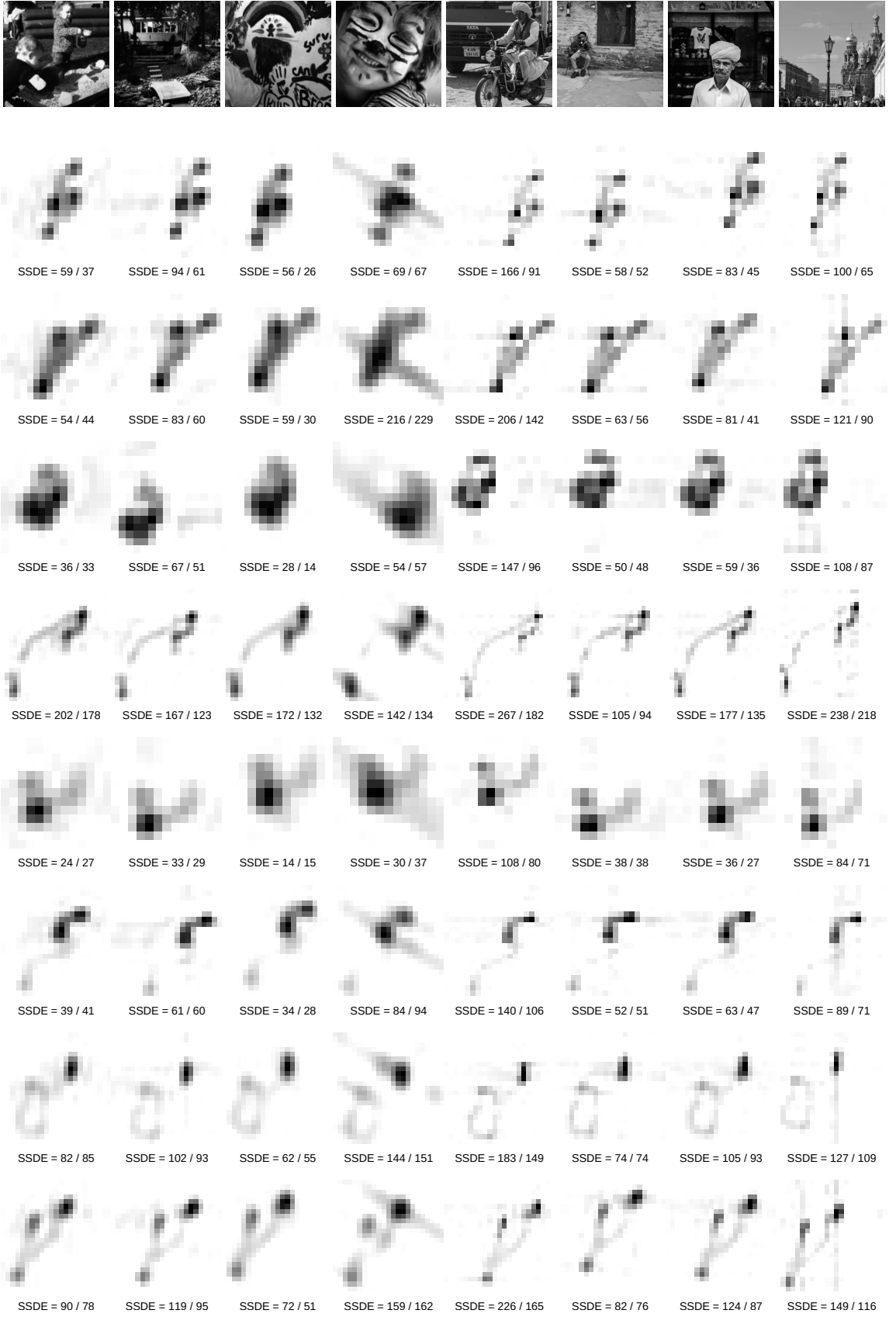


Figure 7.3: Estimated kernels in the exact setup using the IID sparse prior model. The two SSDE values correspond to subsequent non-blind deconvolution using the IID Gaussian prior and the AR Sparse prior.

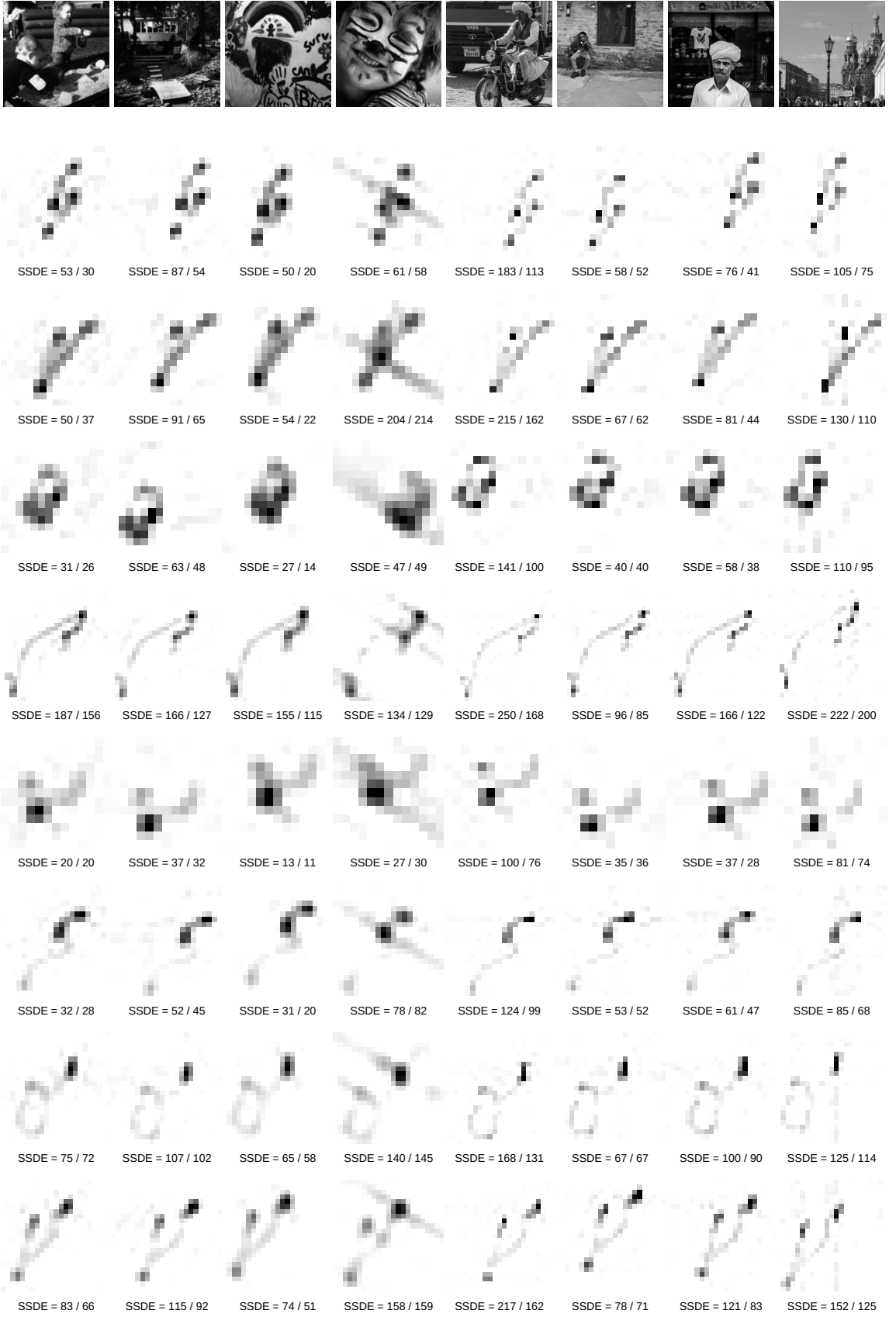


Figure 7.4: Estimated kernels in the exact setup using the AR sparse prior model. The two SSDE values correspond to subsequent non-blind deconvolution using the IID Gaussian prior and the AR Sparse prior.

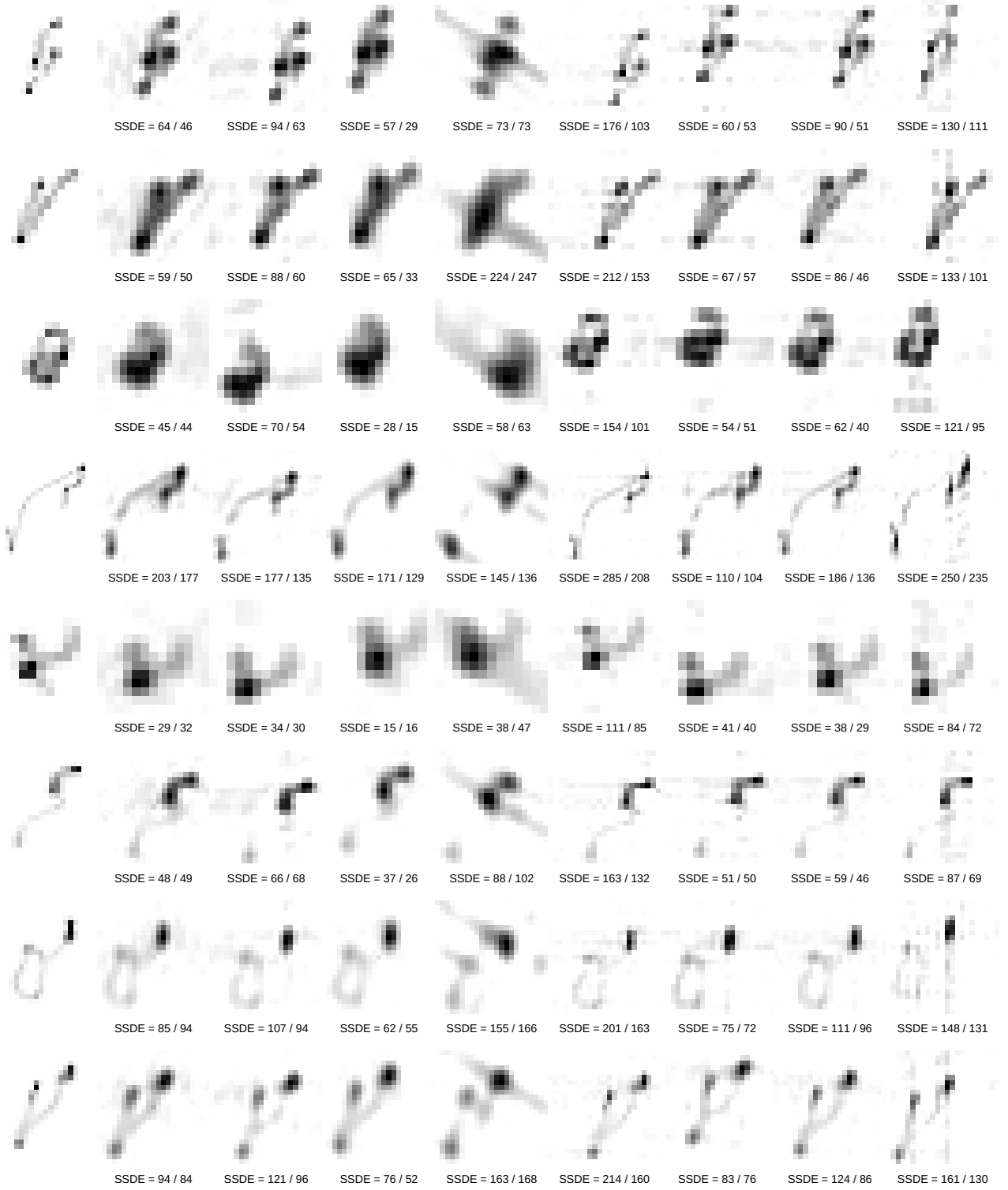


Figure 7.5: Estimated kernels in the less noisy setup ($S = 100$) using the IID sparse prior model. The two SSDE values correspond to subsequent non-blind deconvolution using the IID Gaussian prior and the AR Sparse prior.

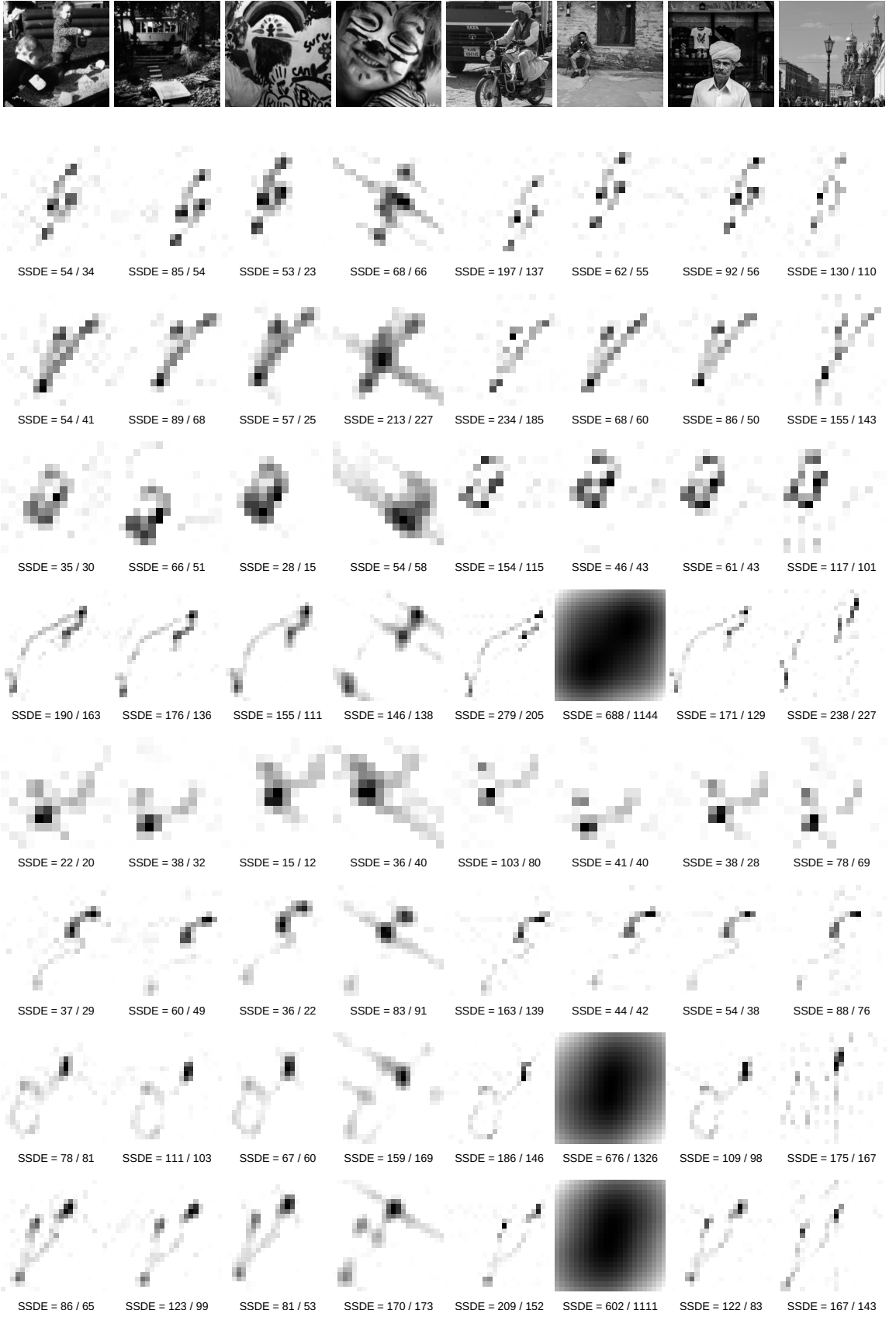


Figure 7.6: Estimated kernels in the less noisy setup ($S = 100$) using the AR sparse prior model. The two SSDE values correspond to subsequent non-blind deconvolution using the IID Gaussian prior and the AR Sparse prior.

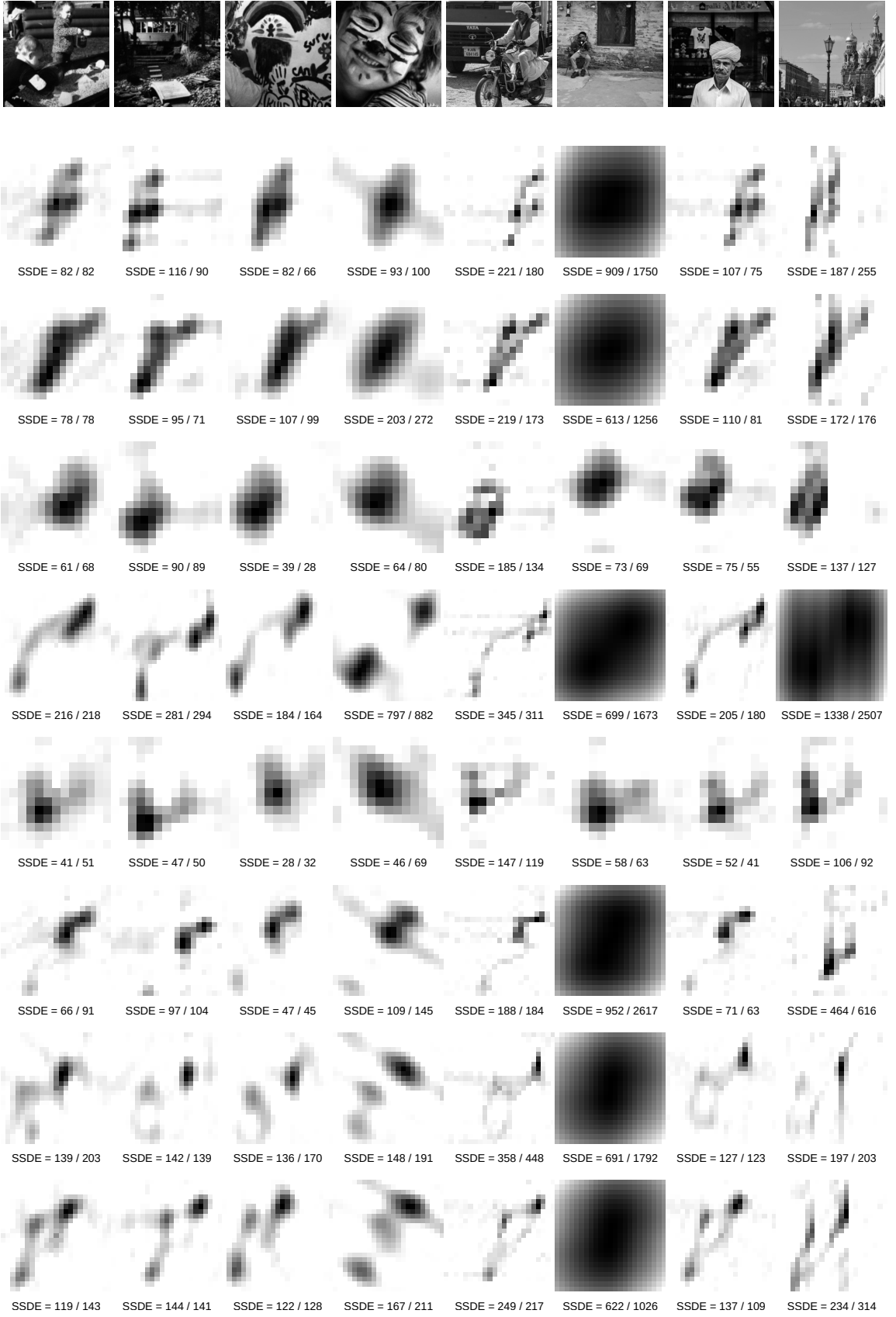


Figure 7.7: Estimated kernels in the more noisy setup ($S = 10$) using the IID sparse prior model. The two SSDE values correspond to subsequent non-blind deconvolution using the IID Gaussian prior and the AR Sparse prior.

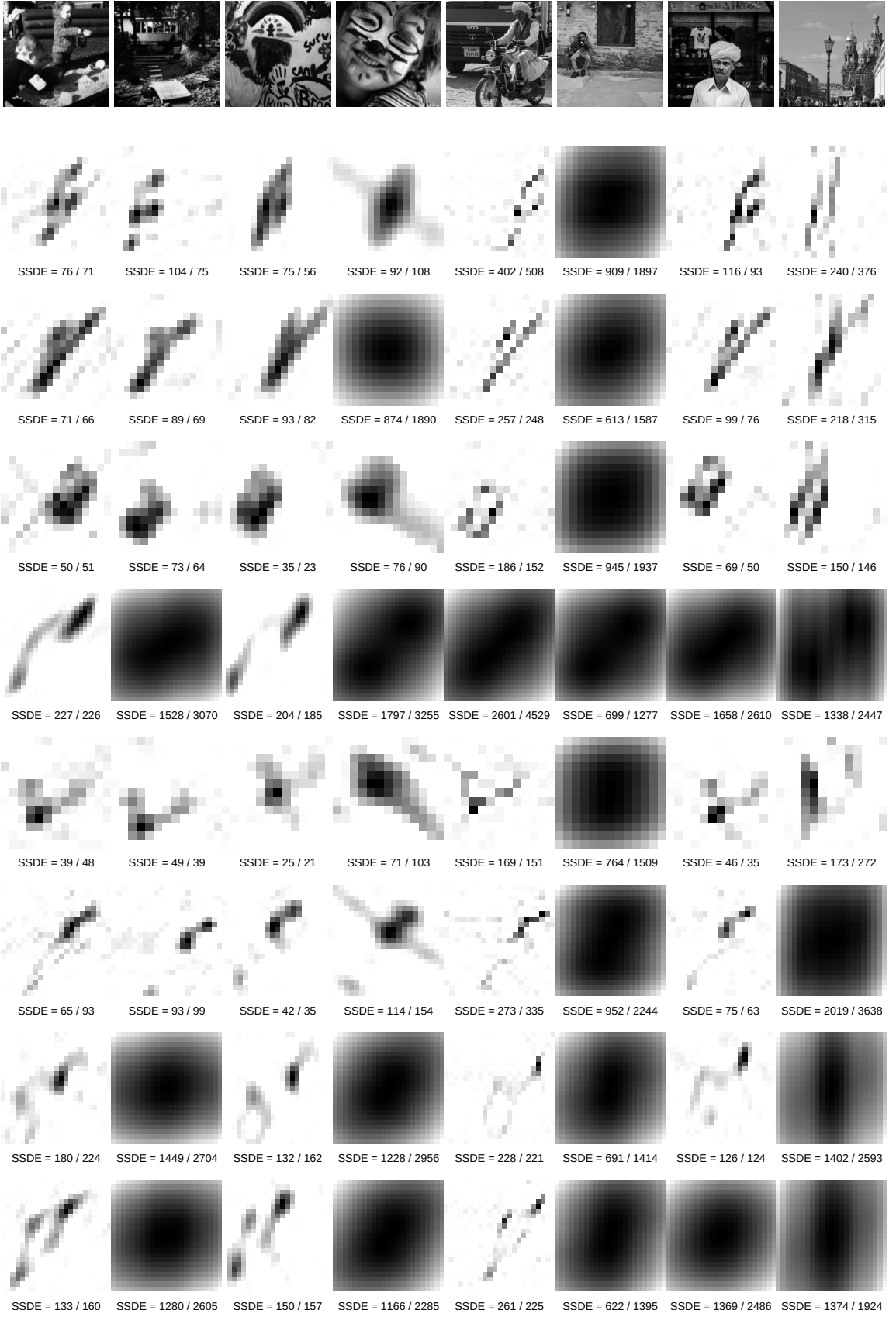


Figure 7.8: Estimated kernels in the more noisy setup ($S = 10$) using the AR sparse prior model. The two SSDE values correspond to subsequent non-blind deconvolution using the IID Gaussian prior and the AR Sparse prior.

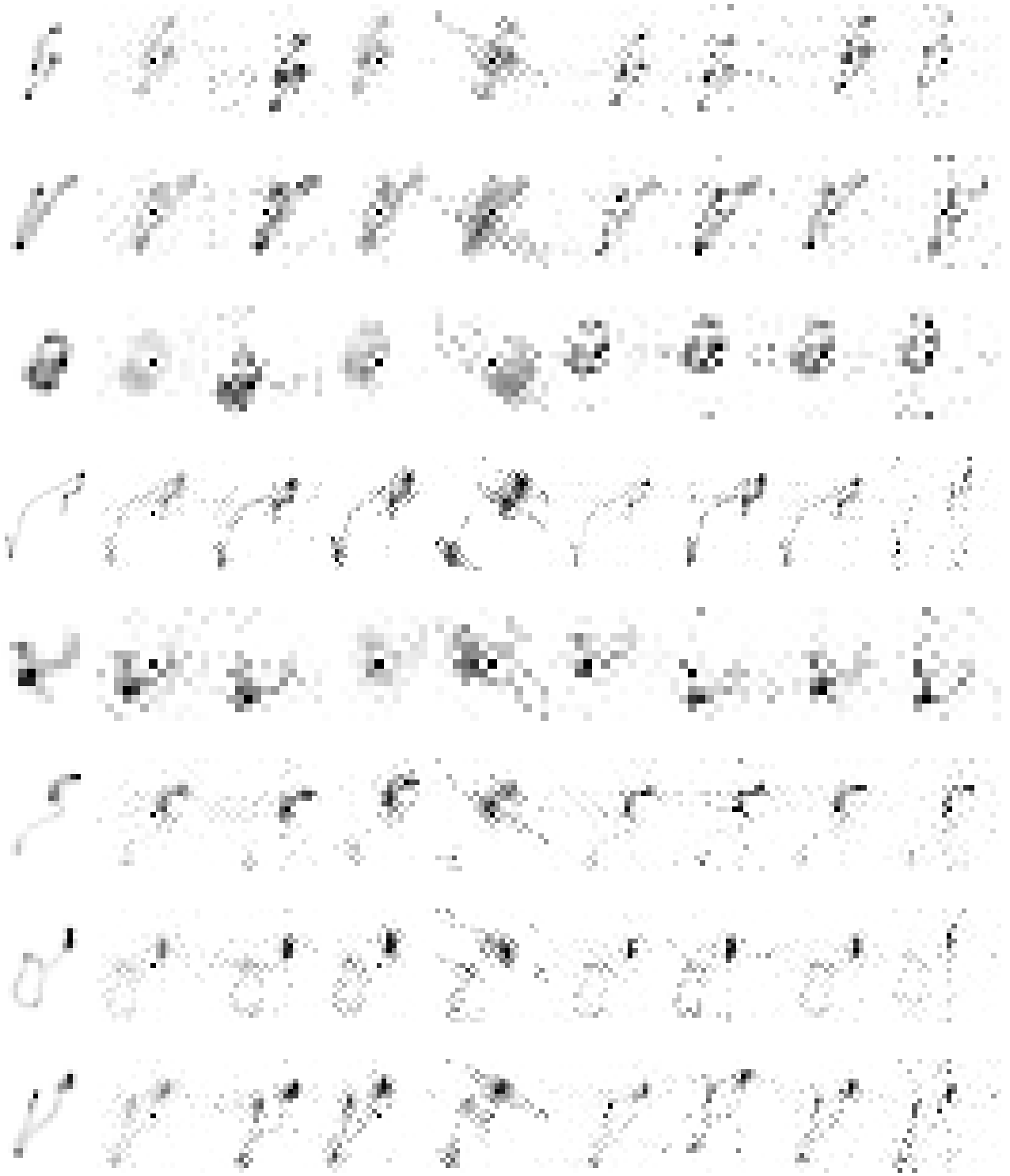


Figure 7.9: Estimated kernels in the more noisy setup ($S = 10$) using the IID Sparse prior model, when the noise parameter value is $\eta = 0.0125$, lower than it should be. This leads to overly sparse and disconnected estimates of the blur kernel.

kernel, and in fact the usefulness of the estimated kernel often depends on carefully choosing a sub-image. Figure 7.10 shows the six blurry sub-images used to estimate blur kernels. The first two images are from Fergus et al. [17], and the remaining four are from our personal collection. Figure 7.10 also shows estimates of the symmetrized kernels $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$, as discussed earlier, estimated for the IID model using (5.1) and the auto-regressive model using (5.5) with $\rho_1 = \rho_2 = 0.5$. As we saw in the synthetically blurred images, here also the estimated kernels using the AR model are sparser and less diffuse. The kernel size for each blurred image is chosen manually by trial and error to provide a snug visual fit.

We then estimated the blur kernels using both the IID sparse prior and the AR sparse prior. For each blurred image, we systematically tried several choices of $\eta \in \{0.008, 0.01, 0.012\}$ and $\rho_1, \rho_2 \in \{0.0, 0.3, 0.6\}$. Rather than showing all the resulting estimates, we choose the value of η which gave the cleanest set of estimated kernels, and compare the kernel estimated using the IID prior with the “sparsest” kernel among those estimated using the auto-regressive model prior (based on a visual inspection). The estimated kernels and the corresponding deblurred images (using the Richardson-Lucy algorithm) are shown in Figure 7.11 for the two images from Fergus et al. [17] and in Figure 7.12 for the four images from our personal collection.

As the true latent image is unknown in these cases, not much can be done beyond a visual assessment of the results. The improvement in the resulting deblurred images, if any, is not obvious from a visible inspection. This is not surprising, as the IID prior already performs remarkably well, and as seen in the systematic study of synthetically blurred images, the absolute improvement in SSDE when the true image is known is relatively small. However, in most cases, the kernel estimated using the generalized prior we propose is visibly sparser, indirectly validating the generalization and our method of incorporating it into the estimation process.

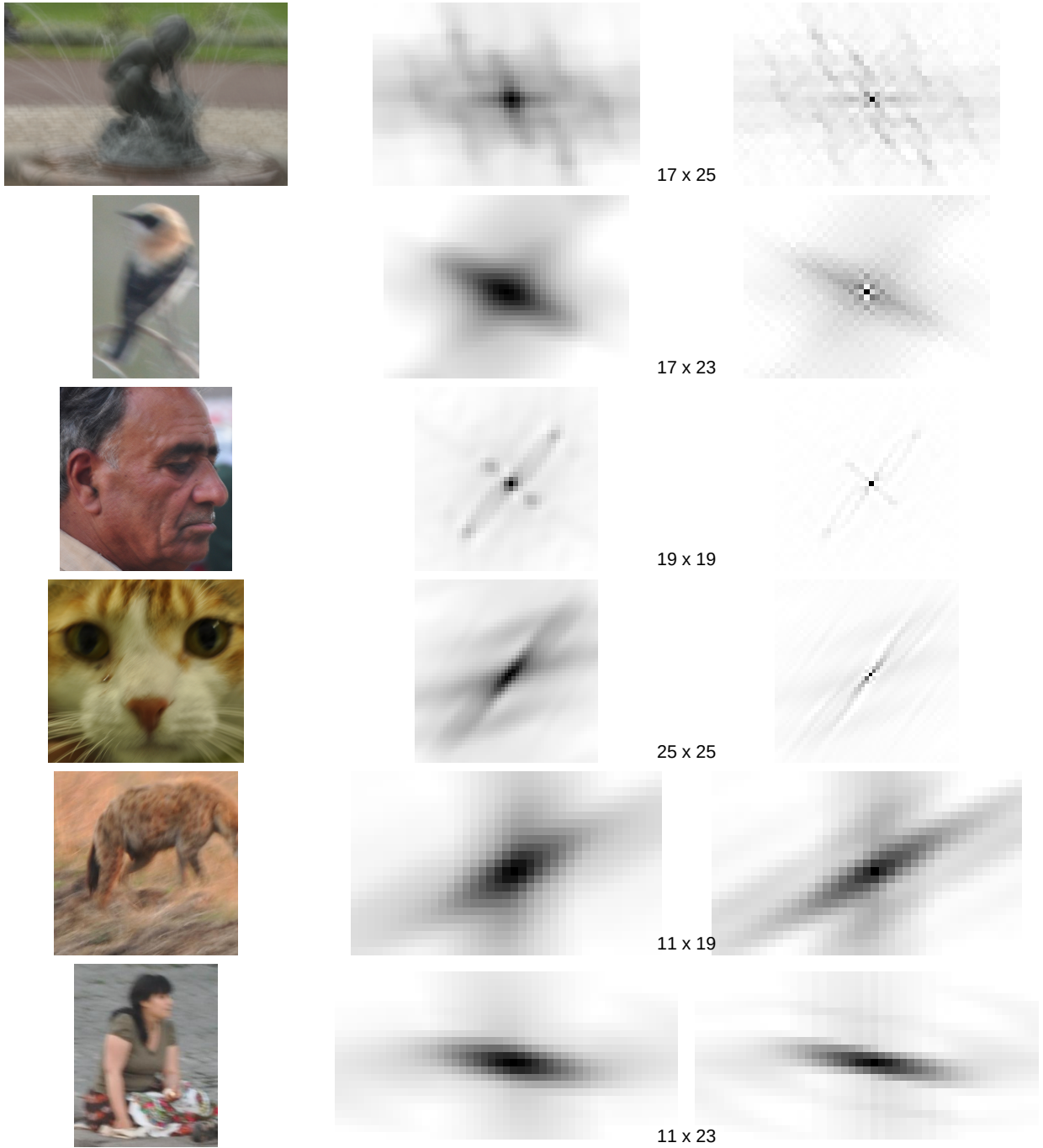


Figure 7.10: Sub-portions of real images used to estimate blur kernel, including preliminary estimation of kernel size. The leftmost column gives the full color images. The middle column gives estimates of the symmetrized kernel $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ using the IID prior, computed using (5.1). The rightmost column gives estimates of the symmetrized kernel $\text{flip}(\mathbf{k}) \otimes \mathbf{k}$ using the AR prior, computed using (5.5), with g_ω computed setting $\rho_1 = \rho_2 = 0.5$. The noise parameter η is taken to be 0.01 in both cases. The kernel sizes shown are chosen by trial and error to “fit” the estimated symmetrized kernels, and the plots shown here are appropriately zoomed.

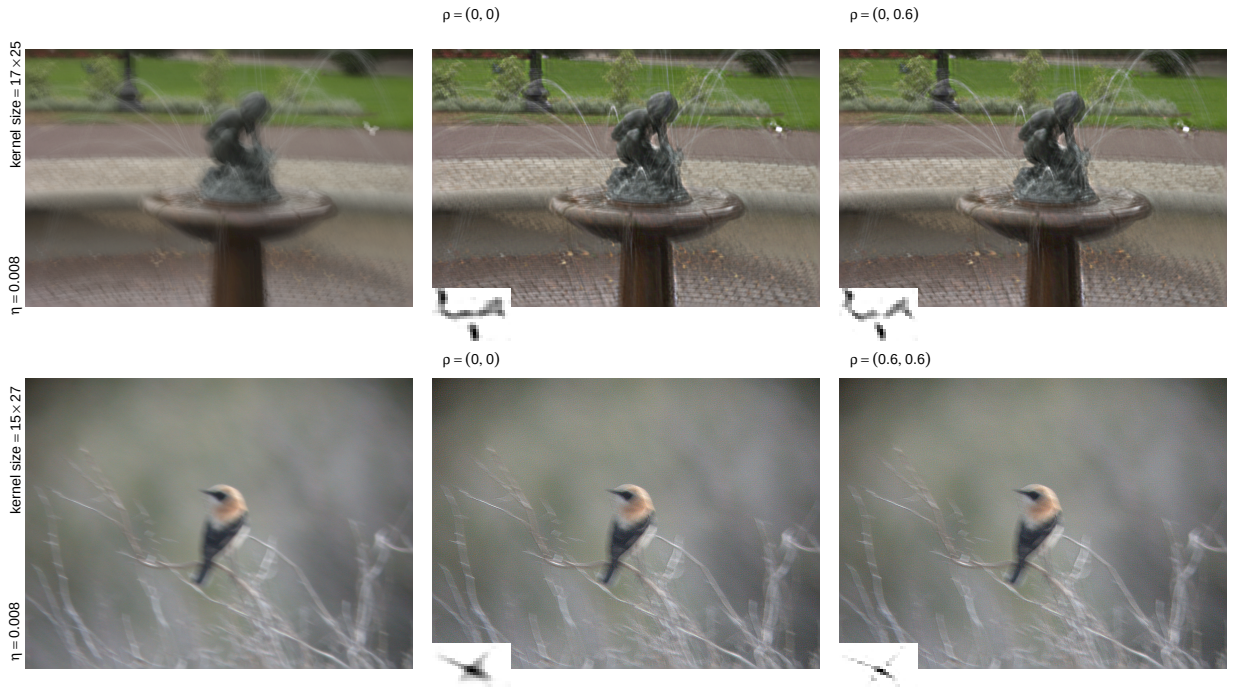


Figure 7.11: Original images from Fergus et al. [17] and their deblurred versions. The left column shows the original blurred image. The middle column shows the deblurred image recovered using the Richardson-Lucy algorithm, with the blur kernel estimated using the IID sparse MOG prior. Finally the third column shows the deblurred image recovered with the blur kernel estimated using our proposed generalized prior based on the auto-regressive model. In both cases, the estimated kernel is shown inset in the bottom-left corner, and the auto-regressive model parameters are shown on top.



Figure 7.12: Original images from our personal collection and their deblurred versions. The left column shows the original blurred image, the middle column shows the deblurred image corresponding to the blur kernel estimated using the IID sparse MOG prior, and the third column shows the deblurred image corresponding to the blur kernel estimated using our proposed generalized prior based on the auto-regressive model. In both cases, the estimated kernel is shown inset in the bottom-left corner, and the auto-regressive model parameters are shown on top. The non-blind deconvolution step is done using the Richardson-Lucy algorithm.

Chapter 8

Discussion and future work

8.1 Summary

Blurring of photographic images, usually due to lack of focus or camera shake during long exposures in low light situations, is one of the most common artifacts in amateur digital photography. Recovering an unblurred image from such a blurred photograph has been a long-standing research problem. In this thesis, we have attempted to study this problem and suggest some possible improvements to existing methods.

Although successful non-blind deconvolution methods have been around for many decades, with particular application to astronomical images, blind deconvolution methods for natural photographs, where the blur kernel is unknown, has been a more challenging problem. The problem is ill-posed in its general formulation, and requires additional assumptions to regularize it, usually in the form of a Bayesian prior on the true image. The use of effective priors that reflect the statistical properties of natural scenes has been explored in recent decades. In particular, a hyper-Laplacian prior for image gradients, approximated by a mixture-of-Gaussians, has been successfully used by Fergus et al. [17] to implement a MAP_k approach to estimate the posterior mode of the blur kernel $\mathbf{k}$ from a blurred image $\mathbf{b}$ which is modeled as

$$\mathbf{b} = \mathbf{k} \otimes \ell + \varepsilon,$$

where ℓ is the unobserved latent image and ε is noise. This approach has been further analyzed and simplified by Levin et al. [45], whose presentation we have largely followed in this thesis. Other approaches to blind deconvolution have also been developed, as summarized in Chapter 3. The hyper-Laplacian prior family has also been applied to the non-blind deconvolution [42] and super resolution [16] problems, among others.

Our contribution Our primary contribution in this context is the observation that the IID prior commonly used to model the derivative of the latent image does not fully reflect the statistical properties of natural images. In particular, the IID prior assumes that image derivatives are independent, whereas we demonstrate that although it is natural to assume that the derivatives follow a stationary process and thus have identically distributed components, they are in practice highly correlated. We demonstrate this empirically in two ways. First, we use the sample auto-correlation and partial auto-correlation functions along and across the direction of the gradients for a library of unblurred images to directly demonstrate that the derivative components are correlated. Secondly, we smooth the coefficients of the DFT of the gradient matrix to demonstrate that its variance is not flat, as would have been the case if the original gradients were independent, but rather the variances of the coefficients increase as the frequency nears 0 or 2π .

Based on these observations, a natural extension of the IID prior would have been to adopt a model analogous to ARMA models for time-series data. However, such models in general would potentially require many parameters to be estimated, and in addition, may be difficult to incorporate into the estimation process. Instead, we restrict our attention to a limited generalization where the image gradients are assumed to follow a simple auto-regressive model (product of two AR(1) models) parameterized by correlation parameters ρ_1 and ρ_2 controlling correlation along and across the direction of the gradient respectively. To make the model easy to incorporate into the estimation process, we also approximate it by an alternative generalization in terms of the variance of the Fourier coefficients.

We use this generalized locally dependent prior in the context of three specific image restoration problems: non-blind deconvolution, super resolution, and blind deconvolution. In Chap-

ter 6, we describe a unified approach that incorporates both non-blind deconvolution and super resolution, and show how the hyper-Laplacian prior with local dependence can be used to obtain the posterior mode of the latent image. In addition to the commonly used conjugate gradient approach, we have also implemented a direct solution based on efficient numerical sparse matrix libraries. Our implementation is available as an R package [54].

Blind deconvolution is a more challenging problem in general, but relatively easier if one assumes that the kernel is symmetric. In Chapter 5, we outline a simple frequency-domain estimator of the blur kernel incorporating local dependence. This estimator is found to be useful for simple blurring due to lack of focus or downsampling. Even for non-symmetric blur kernels, this approach is useful in estimation of the size of the blur kernel.

For the general blind deconvolution problem, we follow the approach of Levin et al. [45]. In Chapter 7, we describe the estimation process based on the expectation-maximization (EM) approach for both a Gaussian prior (which is easier to analyze) as well as the more realistic sparse mixture-of-Gaussians (MOG) prior. Although our generalized prior model is easily incorporated into the estimation process for the Gaussian prior, the MOG prior is more challenging. In this case, we use an indirect imputation approach, where in each E-step iteration we first replace the observed image gradients by a transformed process that more closely follows the IID prior model.

8.2 Results

To assess the usefulness of our proposed generalization, we have applied our methods to both synthetically blurred images and real images. The performance for both non-blind and blind deconvolution is assessed on synthetically blurred images obtained by pairwise convolution of a set of eight test images and eight test kernels, further corrupted with Poisson noise. Performance for super resolution is similarly assessed on two sets of eight test images. Results on real-life blurred images are also shown, with further examples for non-blind deconvolution and super-resolution available in a supplementary website.¹

¹<https://www.isid.ac.in/~deepayan/rip/html/>

For non-blind deconvolution and super resolution, our general observation is that sparse priors ($\alpha = 0.8$ in the hyper-Laplacian family) do not necessarily lead to quantitatively better results than the Gaussian approach. However, in terms of visual plausibility, the locally dependent auto-regressive prior usually gives better results compared to the IID sparse prior, although the Gaussian priors (both IID and dependent) can give better results in areas with texture.

For blind deconvolution, compared to the kernel estimated using the IID prior, the kernel estimated using the generalized dependent prior usually provides a better visual approximation the true kernel (when known). Both in that case and in the case of real blurred images, the kernels estimated using the generalized prior are usually sparser and cleaner compared to the kernels estimated using the IID prior. However, performance of the blind deconvolution method studied here degrades substantially as noise increases.

Despite the improved kernel estimates, the improvement in the recovered *unblurred* image is usually nominal, and in the case of the real-life examples, not easily identified in a visual inspection. In fact, comparing the corresponding unblurred images with the true latent image, in the case of synthetically blurred images where the true latent image is known, we find that based on a numerical measure of performance, the choice of prior used in the final non-blind deconvolution step plays a much bigger role than the prior used during the kernel estimation phase.

While disappointing, this is not unexpected as the IID prior already performs quite well, and there is not much scope for improvement. Nonetheless, the kernel estimated using the generalized prior we propose is visibly sparser in most cases, suggesting that the generalized prior leads to concrete improvement in the estimated kernels.

8.3 Extensions

Inpainting. As noted earlier, for non-blind deconvolution and super resolution, we have implemented an alternative to the usual conjugate gradient approach that obtains the posterior mode of the latent image by directly solving a sparse linear system. One advantage of the

direct method is that it trivially allows for missing values in the input image $\mathbf{y}$, by keeping the size of the latent image ℓ unchanged but dropping the missing values from $\mathbf{y}$ as well as the corresponding rows of $T_{\mathbf{k},f}$ in (6.6) or (6.7). Restoring images with missing values is known as *inpainting* [5, 79], and our approach can thus be extended to implement a Bayesian solution to the inpainting problem for the prior family considered in this article. This extension is implemented in our package, and although it is of limited use for large missing patches (as it does not account for image features beyond the boundary of the patch), it gives reasonable results for small patches.

Robust loss. Another simple extension to the IRLS approach allows us to replace the squared error loss function used to ensure image fidelity (i.e., $\mathbf{y} \approx \mathbf{d}_f^\dagger(\mathbf{k} \otimes \ell)$) by a robust loss function of the kind used in M -estimation. Specifically, along with the IRLS weights for the prior component in (6.7), we can also add weights for the observed pixels $\mathbf{y}$ in each IRLS step that are determined by the previous estimate of ℓ . This approach is implemented in our package for the Huber and Tukey bisquare loss functions [30, 4]. In practice, it is often more useful to use a robust loss as a post-processing denoising step in conjunction with an identity kernel and the AR sparse prior, after performing deconvolution or super-resolution using the relatively fast IID Gaussian prior, or the even faster Richardson-Lucy method. Examples of both inpainting and robust loss functions are available in the supplementary website.

General dependence structures. Although we have used an auto-regressive model with two additional correlation parameters to parameterize our generalized prior model, our generalization is essentially determined by the variance function g_ω of the DFT coefficients. A natural extension of our method would be to use g_ω obtained from other models, or from an unblurred image similar to the one being deblurred. Such images may be available if multiple photographs are taken of the same scene.

Kernel size Finally, preliminary estimation of the kernel size is a useful first step in blind deconvolution that is facilitated by our generalized prior model, as described in Section 5.4.

It may be interesting to explore the usefulness of a more formal (penalized) likelihood-based assessment of kernel size. The related frequency-domain kernel estimation procedure outlined in Section 5.3, for symmetric blur kernels, can also be extended to parametric models, with parameter estimates obtained numerically by maximizing likelihood in the Fourier domain. Such parametric models can be used to model common blur patterns such as those caused by lack of focus or uniform motion blur.

8.4 Future work

On a more general note, there is scope for exploring alternative approaches to the blind deconvolution problem, and this may lead to further improvements. We chose the approach of Levin et al. [45] as a starting point because of ready availability of a well-tested implementation, but it is by no means the only possible starting point.

Even within the basic EM framework of Fergus et al. [17] and Levin et al. [45], there is scope for alternative approaches. Their approach uses a mixture-of-Gaussian distribution to approximate the sparse hyper-Laplacian prior as this makes the variational Bayes approach more tractable. However, this becomes relevant only in the E-step, which is essentially a non-blind deconvolution step where the conditional mode of a Bayesian posterior is computed along with its conditional variance. The non-blind deconvolution step considered in Chapter 6 solves exactly this problem for the hyper-Laplacian problem, and as this is implemented as a weighted Gaussian problem through IRLS, an estimate of the conditional variance is also readily available. This would bypass both the mixture-of-Gaussian and the variational Bayes approximations, and the IRLS approach would also be easier to generalize to other prior families. It would also avoid the imputation approach we take, as the dependent prior can be directly incorporated in the IRLS-based non-blind deconvolution procedure.

Furthermore, the idea of formulating the blind deconvolution problem as two separate problems for horizontal and vertical gradients with a common blur kernel seems somewhat artificial. Instead, a single likelihood problem formulated in terms of the observed image, as is done for the non-blind deconvolution case in Chapter 6, is equally amenable to the EM approach. Al-

though we have not seen a discussion of this, it is possible that the reason that this alternative approach has not become popular is that it would lead to more severe boundary artifacts if one uses Fourier domain techniques, which implicitly assume periodicity in the image domain. This is less of an issue with image gradients which may reasonably be assumed to be stationary. An approach that handles boundary issues more carefully may resolve such problems, and in fact, we have had some success following this line of approach. However, while this approach is theoretically more appealing, the quality of the final results depend on many details of the computation, such as the choice of parameter values and the type of regularization applied on the blur kernel, and our approach has not yet produced estimates that are as good as the approach of Levin et al. [45]. However, this approach merits further exploration.

The basic idea of using priors to solve ill-posed problems is of course quite general and can be applied in contexts other than for natural images. One important context where similar methods can be useful is for medical imaging, where the corresponding super-resolution problem can be particularly important. Figure 8.1 shows the distribution of gradients and auto-correlation functions for four MRI images from the Visual Human Project,² suggesting that the locally dependent hyper-Laplacian prior family could model the gradients of these images as well.

The essential ideas of both non-blind and blind deconvolution as well as super-resolution can be applied in three dimensions as well. The main challenge is computational rather than theoretical, and would involve developing efficient frameworks for calculating three dimensional convolutions, gradients, etc. Such methods could be applied to video input (where the third dimension is time) as well as to three dimensional medical images.

8.5 Software

Our methods for non-blind deconvolution and super resolution are implemented as an R package [54]. Our implementation of blind deconvolution is based on the MATLAB code provided as supplementary material by Levin et al. [45].

²<https://www.nlm.nih.gov/research/visible/mri.html>

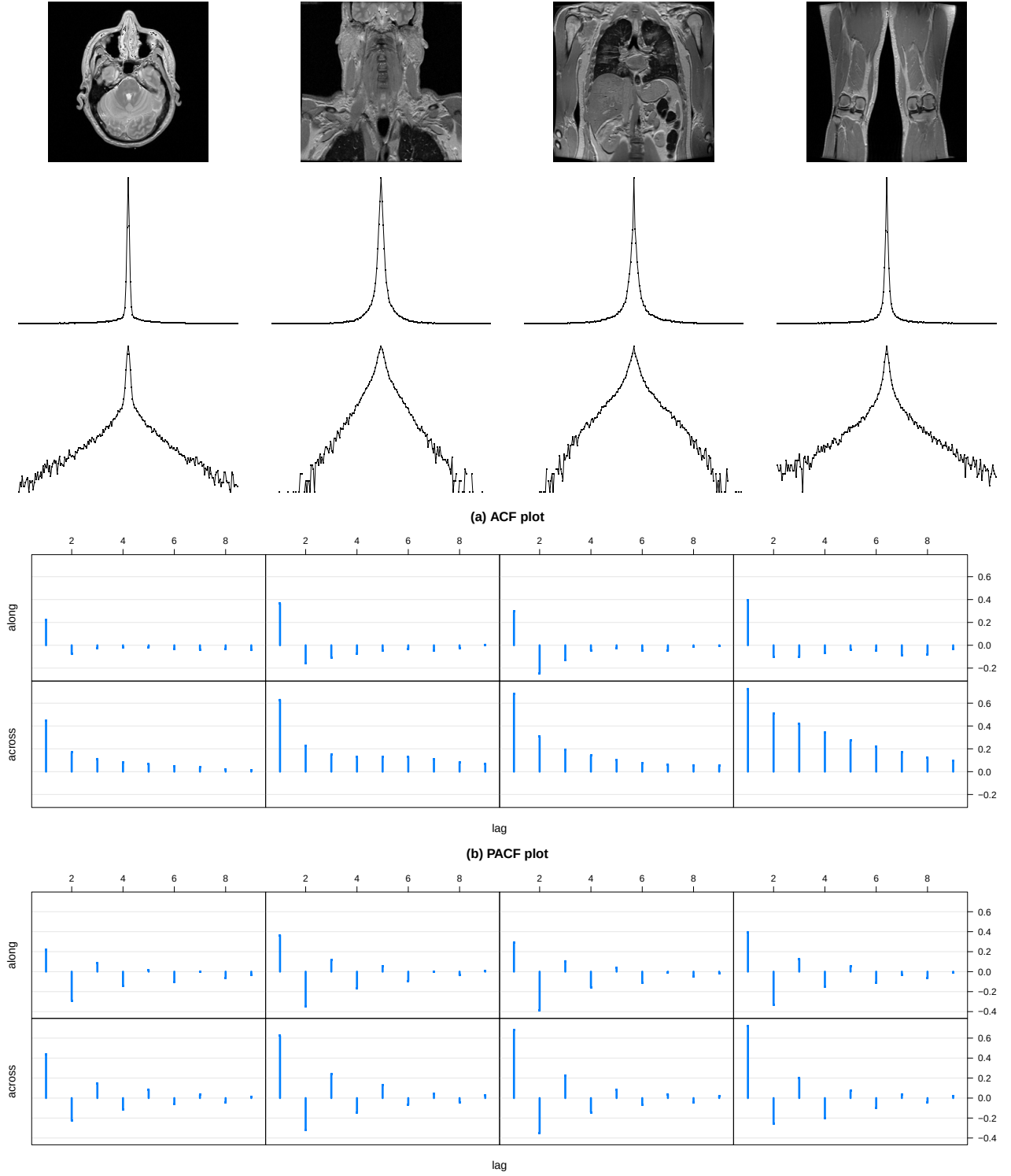


Figure 8.1: Analogues of Figure 1.8 (top, marginal distribution of gradients) and Figure 5.4 (bottom, ACF and PACF of horizontal gradients along / across the direction of the gradient averaged across rows / columns) for MRI images from the Visual Human Project. The similarity with the corresponding plots for natural images suggests that the locally dependent prior family considered in this thesis could be useful in the domain of medical imaging as well.

Bibliography

- [1] G. Ayers and J. C. Dainty. Iterative blind deconvolution method and its applications. *Optics letters*, 13(7):547–549, 1988.
- [2] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *European Conference on Computer Vision*, pages 341–355. Springer, 2012.
- [3] D. Bates and M. Maechler. *Matrix: Sparse and Dense Matrix Classes and Methods*, 2019. URL <https://CRAN.R-project.org/package=Matrix>. R package version 1.2-18.
- [4] A. E. Beaton and J. W. Tukey. The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.
- [5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 417–424, 2000.
- [6] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.
- [7] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [8] M. M. Bronstein, A. M. Bronstein, M. Zibulevsky, and Y. Y. Zeevi. Blind deconvolution of images using optimal sparse representations. *IEEE Transactions on Image Processing*, 14(6):726–736, 2005.

- [9] A. Chakrabarti. A neural approach to blind motion deblurring. In *European Conference on Computer Vision*, pages 221–235. Springer, 2016.
- [10] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):1–14, 2008.
- [11] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics (TOG)*, 28(5):145, 2009.
- [12] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. Citeseer, 2007.
- [13] J. W. Cooley, P. A. Lewis, and P. D. Welch. The fast Fourier transform and its applications. *IEEE Transactions on Education*, 12(1):27–34, 1969.
- [14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [15] 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*, 38(2):295–307, 2015.
- [16] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2832–2839, 2013.
- [17] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (TOG)*, 25(3):787–794, 2006.
- [18] J. R. Fienup. Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769, 1982.

- [19] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computer Journal*, 7(2):149–154, 1964. doi: 10.1093/comjnl/7.2.149.
- [20] W. Freeman and C. Liu. Markov random fields for super-resolution and texture synthesis. *Advances in Markov Random Fields for Vision and Image Processing*, 1(155-165):3, 2011.
- [21] R. W. Gerchberg and W. O. Saxton. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237, 1972.
- [22] D. Goldfarb and A. Idnani. Dual and primal-dual methods for solving strictly convex quadratic programs. In *Numerical Analysis*, pages 226–239. Springer, 1982.
- [23] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.
- [24] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2017.
- [25] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *European Conference on Computer Vision*, pages 171–184. Springer, 2010.
- [26] P. Hall and P. Qiu. Blind deconvolution and deblurring in image analysis. *Statistica Sinica*, pages 1483–1509, 2007.
- [27] P. Hall and P. Qiu. Nonparametric estimation of a point-spread function in multivariate problems. *The Annals of Statistics*, 35(4):1512–1534, 2007.
- [28] M. Hayes. The reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(2):140–154, 1982.

- [29] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1–8, 2004. doi: 10.1109/CVPR.2004.1315043.
- [30] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematics Statistics*, 35(1):73–101, 1964.
- [31] B. R. Hunt. The application of constrained least squares estimation to image restoration by digital computer. *IEEE Transactions on Computers*, 100(9):805–812, 1973.
- [32] J. Jia. Single image motion deblurring using transparency. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [33] N. Joshi, R. Szeliski, and D. J. Kriegman. PSF estimation using sharp edge prediction. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [34] A. K. Katsaggelos and K.-T. Lay. Maximum likelihood blur identification and image restoration using the EM algorithm. *IEEE Transactions on Signal Processing*, 39(3):729–733, 1991.
- [35] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981. doi: 10.1109/TASSP.1981.1163711.
- [36] K. I. Kim and Y. Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, 2010.
- [37] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-Laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041, 2009.
- [38] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE signal processing magazine*, 13(3):43–64, 1996.

- [39] D. Kundur and D. Hatzinakos. Blind image deconvolution revisited. *IEEE signal processing magazine*, 13(6):61–63, 1996.
- [40] M. A. Lee, T. Budavari, R. L. White, and C. Gulian. Robust statistics for image deconvolution. *Astronomy and computing*, 21:15–21, 2017.
- [41] A. Levin. Blind motion deblurring using image statistics. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 841–848. MIT Press, 2007. URL <http://papers.nips.cc/paper/3085-blind-motion-deblurring-using-image-statistics.pdf>.
- [42] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Deconvolution using natural image priors. Available online, 2007. URL <https://groups.csail.mit.edu/graphics/CodedAperture/SparseDeconv-LevinEtAl07.pdf>.
- [43] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (TOG)*, 26(3):70, 2007.
- [44] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Computer Vision and Pattern Recognition, 2009.*, pages 1964–1971. IEEE, 2009.
- [45] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *2011 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2657–2664. IEEE, 2011.
- [46] A. C. Likas and N. P. Galatsanos. A variational approach for Bayesian blind image deconvolution. *IEEE Transactions on Signal Processing*, 52(8):2222–2233, 2004.
- [47] L. B. Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974.
- [48] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

- [49] E. Maeland. On the comparison of interpolation methods. *IEEE Transactions on Medical Imaging*, 7(3):213–217, 1988. doi: 10.1109/42.7784.
- [50] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.
- [51] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, pages 783–798. Springer, 2014.
- [52] J. Miskin and D. J. MacKay. Ensemble learning for blind image separation and deconvolution. In *Advances in independent component analysis*, pages 123–141. Springer, 2000.
- [53] R. Molina, A. K. Katsaggelos, J. Abad, and J. Mateos. A Bayesian approach to blind deconvolution based on dirichlet distributions. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2809–2812. IEEE, 1997.
- [54] K. Nandy and D. Sarkar. *rip.recover: Image Reconstruction Utilities*, 2020. URL <https://github.com/deepayan/rip/>. R package version 0.1-1.
- [55] M. Noroozi, P. Chandramouli, and P. Favaro. Motion deblurring in the wild. In *German conference on Pattern Recognition*, pages 65–77. Springer, 2017.
- [56] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.
- [57] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.
- [58] J. A. Parker, R. V. Kenyon, and D. E. Troxel. Comparison of interpolating methods for image resampling. *IEEE Transactions on Medical Imaging*, 2(1):31–39, 1983.
- [59] M. Peligrad and W. B. Wu. Central limit theorem for Fourier transforms of stationary processes. *The Annals of Probability*, pages 2009–2022, 2010.
- [60] E. Polak and G. Ribiere. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 3(R1):35–43, 1969.

- [61] P. Qiu. A nonparametric procedure for blind image deblurring. *Computational Statistics & Data Analysis*, 52(10):4828–4841, 2008.
- [62] S. Ramakrishnan, S. Pachori, A. Gangopadhyay, and S. Raman. Deep generative filter for motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2993–3000, 2017.
- [63] W. H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972.
- [64] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, volume 2, pages 860–867. IEEE, 2005.
- [65] P. Ruiz, X. Zhou, J. Mateos, R. Molina, and A. K. Katsaggelos. Variational Bayesian blind image deconvolution: A review. *Digital Signal Processing*, 47:116–127, 2015.
- [66] J. C. Russ. *The image processing handbook*. CRC press, 2015.
- [67] R. R. Schultz and R. L. Stevenson. A Bayesian approach to image expansion for improved definition. *IEEE Transactions on Image Processing*, 3(3):233–242, 1994.
- [68] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
- [69] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (TOG)*, 27(3):73, 2008.
- [70] E. P. Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *31st Asilomar Conf on Signals, Systems and Computers*, volume 1, pages 673–678, 1997.
- [71] E. P. Simoncelli. Bayesian denoising of visual images in the wavelet domain. In *Bayesian inference in wavelet-based models*, pages 291–308. Springer, 1999.

- [72] E. Spodarev, editor. *Stochastic Geometry, Spatial Statistics and Random Fields: Asymptotic Methods*. Springer, 2013.
- [73] J.-L. Starck, E. Pantin, and F. Murtagh. Deconvolution in astronomy: A review. *Publications of the Astronomical Society of the Pacific*, 114(800):1051, 2002.
- [74] J. Sun, N.-N. Zheng, H. Tao, and H.-Y. Shum. Image hallucination with primal sketch priors. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003*, volume 2, pages II-729. IEEE, 2003.
- [75] J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [76] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015.
- [77] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2013.
- [78] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *In IEEE Workshop on Statistical and Computational Theories of Vision*. Citeseer, 2003.
- [79] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004.
- [80] E. Thiébaud and J.-M. Conan. Strict a priori constraints for maximum-likelihood blind deconvolution. *Journal of the Optical Society of America A*, 12(3):485–492, 1995.
- [81] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos. The variational approximation for Bayesian inference. *IEEE Signal Processing Magazine*, 25(6):131–146, 2008.

- [82] M. Unser, A. Aldroubi, M. Eden, et al. Fast B-spline transforms for continuous image representation and interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):277–285, 1991.
- [83] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2):168–186, 2012.
- [84] N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: with Engineering Applications*. Technology Press, 1950.
- [85] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *European Conference on Computer Vision*, pages 157–170. Springer, 2010.
- [86] L. Xu, S. Zheng, and J. Jia. Unnatural L0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013.
- [87] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.
- [88] Y. Yang, N. P. Galatsanos, and H. Stark. Projection-based blind deconvolution. *Journal of the Optical Society of America A*, 11(9):2401–2409, Sep 1994. doi: 10.1364/JOSAA.11.002401.

Appendix A

Details of auto-regressive model

In this thesis, we make extensive use of an auto-regressive model with a simple correlation structure given by (5.2), namely,

$$\rho(i, j, k, l) = \rho_1^{|i-k|} \rho_2^{|j-l|}. \quad (\text{A.1})$$

Here we provide a few related details.

A.1 Calculation of g_ω for auto-regressive model

Here, we derive the corresponding variance of the DFT coefficients in terms of ρ_1 and ρ_2 . Recall that given a two-dimensional periodic spatial process y , its two-dimensional DFT coefficient for a frequency pair (ω_1, ω_2) is given by

$$Y(\omega_1, \omega_2) = \sum_{t=1}^M \sum_{s=1}^N y_{ts} \exp\left(-\frac{2\pi i \omega_1 t}{M}\right) \exp\left(-\frac{2\pi i \omega_2 s}{N}\right),$$

and its variance given by

$$g_{(\omega_1, \omega_2)} = \text{Var}(Y(\omega_1, \omega_2)) = \text{E}(Y(\omega_1, \omega_2) \bar{Y}(\omega_1, \omega_2)).$$

A simplified expression of this variance term can be derived from the variance of similar one-dimensional process. Define a process (y_t) that is jointly Gaussian with $E(y(t)) = 0$, with

$$\text{Var}(y(t)) = \frac{\sigma^2}{1 - \rho^2}$$

for all t , and

$$\text{Cov}(y(t), y(t')) = \frac{\rho^{|t-t'|}}{1 - \rho^2} \sigma^2$$

for all $t \neq t'$. The DFT of $y(t)$ is given by

$$Y(\omega) = \sum_{t=1}^N y(t) \exp\left(-\frac{2\pi i t \omega}{N}\right),$$

and its variance given by

$$\text{Var}(Y(\omega)) = E(Y(\omega) \bar{Y}(\omega)).$$

By a direct calculation, an expression for this variance can be obtained as follows.

$$\begin{aligned} \text{Var}(Y(\omega)) &= E \left[\left(\sum_{t=1}^N y(t) \exp\left(-\frac{2\pi i t \omega}{N}\right) \right) \left(\sum_{t=1}^N y(t) \exp\left(\frac{2\pi i t \omega}{N}\right) \right) \right] \\ &= \frac{\sigma^2}{1 - \rho^2} \sum_{t_1=1}^N \sum_{t_2=1}^N \rho^{|t_1-t_2|} \exp\left(-\frac{2\pi i t_1 \omega}{N}\right) \exp\left(\frac{2\pi i t_2 \omega}{N}\right) \\ &= \frac{\sigma^2}{1 - \rho^2} \sum_{t_1=1}^N \sum_{t_2=1}^N \rho^{|t_1-t_2|} a^{t_1-t_2} \quad \text{where, } a = \exp\left(-\frac{2\pi i \omega}{N}\right) \\ &= \frac{\sigma^2}{1 - \rho^2} \left(\sum_{t=1}^N 1 + \sum_{t_1=1}^N \sum_{t_2=1}^{t_1-1} \rho^{t_1-t_2} a^{t_1-t_2} + \sum_{t_1=1}^N \sum_{t_2=t_1+1}^N \rho^{t_2-t_1} a^{t_1-t_2} \right) \\ &= \frac{\sigma^2}{1 - \rho^2} \left(N + \sum_{t_1=1}^N (\rho a)^{t_1} \frac{1}{\rho a} \frac{1 - (\frac{1}{\rho a})^{t_1-1}}{1 - (\frac{1}{\rho a})} + \sum_{t_1=1}^N \left(\frac{a}{\rho}\right)^t \frac{\rho}{a} \frac{(\frac{\rho}{a})^{t_1} - (\frac{\rho}{a})^N}{1 - (\frac{\rho}{a})} \right) \\ &= \frac{\sigma^2}{1 - \rho^2} \left(N + \frac{1}{1 - \frac{1}{\rho a}} \frac{1 - (\rho a)^N}{1 - \rho a} - \frac{N}{1 - \frac{1}{\rho a}} + \frac{N \frac{\rho}{a}}{1 - \frac{\rho}{a}} - \frac{(\frac{\rho}{a})^N}{1 - \frac{\rho}{a}} \frac{1 - (\frac{a}{\rho})^N}{1 - \frac{a}{\rho}} \right) \\ &= \frac{\sigma^2}{1 - \rho^2} \left(N \left(1 - \frac{1}{1 - \frac{1}{\rho a}} + \frac{\frac{\rho}{a}}{1 - \frac{\rho}{a}} \right) + \frac{1}{1 - \frac{1}{\rho a}} \frac{1 - (\rho a)^N}{1 - \rho a} - \frac{(\frac{\rho}{a})^N}{1 - \frac{\rho}{a}} \frac{1 - (\frac{a}{\rho})^N}{1 - \frac{a}{\rho}} \right) \\ &= V_\rho(\omega) \text{ (say) }, \end{aligned}$$

noting that a in the above expression depends on ω . Using the simplified form of the 1-D case, the 2-D case can be derived easily as follows.

$$\begin{aligned}
& \text{Var}(Y(\omega_1, \omega_2)) \\
&= \mathbb{E} \left[\left(\sum_{t=1}^M \sum_{s=1}^N y_{ts} \exp \left(-\frac{2\pi i \omega_1 t}{M} \right) \exp \left(-\frac{2\pi i \omega_2 s}{N} \right) \right) \left(\sum_{t'=1}^M \sum_{s'=1}^N y_{t's'} \exp \left(\frac{2\pi i \omega_1 t'}{M} \right) \exp \left(\frac{2\pi i \omega_2 s'}{N} \right) \right) \right] \\
&= \sum_{t=1}^M \sum_{s=1}^N \sum_{t'=1}^M \sum_{s'=1}^N E(y_{ts} y_{t's'}) \exp \left(-\frac{2\pi i \omega_1 t}{M} \right) \exp \left(-\frac{2\pi i \omega_2 s}{N} \right) \exp \left(\frac{2\pi i \omega_1 t'}{M} \right) \exp \left(\frac{2\pi i \omega_2 s'}{N} \right) \\
&= \kappa \sum_{t=1}^M \sum_{s=1}^N \sum_{t'=1}^M \sum_{s'=1}^N \rho_1^{|t-t'|} \rho_2^{|s-s'|} \exp \left(-\frac{2\pi i \omega_1 t}{M} \right) \exp \left(-\frac{2\pi i \omega_2 s}{N} \right) \exp \left(\frac{2\pi i \omega_1 t'}{M} \right) \exp \left(\frac{2\pi i \omega_2 s'}{N} \right) \\
&= V_{\rho_1}(\omega_1) V_{\rho_2}(\omega_2).
\end{aligned}$$

Here κ is a shorthand for $\kappa(\rho_1, \rho_2, \sigma_1, \sigma_2)$ defined before.

A.2 Correlation among DFT coefficients

The generalization of the IID prior in Chapter 5 is specified in two alternative forms. The first form postulates an auto-regressive model for the image gradients in the image domain, and the second form postulates a non-constant variance structure for the corresponding DFT coefficients, viewing it as a technical approximation to the first form. The postulated variance structure in the second form is precisely the variance structure induced by the first form, but one inconsistency between the two is that in the second form the DFT coefficients are assumed to be uncorrelated, which is not true for the first form.

The actual correlations can be explicitly calculated (details not shown). Here we provide some summary results showing that this inconsistency between the two models is minor in practice. For various image sizes (20×20 , 50×50 , 100×100 , and 200×200), and various values of $\rho_1 = \rho_2 = \rho$ in the auto-regressive model, we explicitly calculated the maximum absolute correlation between any two DFT coefficients. These maximum values are shown in Figure A.1. As seen from the plot, although the DFT correlation increases with the auto-correlation, this effect is muted as the image size increases. For even moderate image sizes such

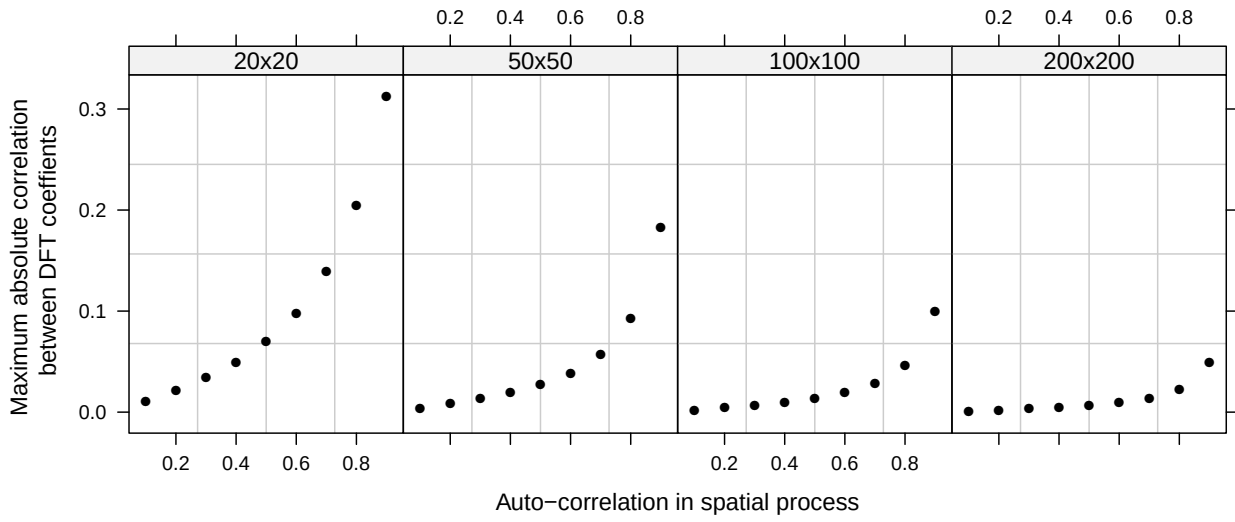


Figure A.1: Maximum correlation between DFT coefficients as function of image size and degree of auto-correlation.

as 100×100 , the maximum DFT correlation is only around 0.1 when $\rho = 0.9$. This is consistent with the remark in Chapter 5 that these correlations vanish asymptotically.