# Removing Camera Shake Blur and Unwanted Occluders from Photographs

Oliver Whyte

▶ **To cite this version:**

Oliver Whyte. Removing Camera Shake Blur and Unwanted Occluders from Photographs. Computer Vision and Pattern Recognition [cs.CV]. École normale supérieure de Cachan - ENS Cachan, 2012. English. NNT : . tel-01063340

**HAL Id: tel-01063340**

**https://theses.hal.science/tel-01063340**

Submitted on 11 Sep 2014

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

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

EDSP
École Doctorale
Sciences Pratiques

ENS
CACHAN

Inria
INVENTEURS DU MONDE NUMÉRIQUE

# THÈSE DE DOCTORAT
# DE L'ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

présentée par

**OLIVER WHYTE**

pour obtenir le grade de

**DOCTEUR DE L'ECOLE NORMALE SUPERIEURE DE CACHAN**

Domaine :

**MATHÉMATIQUES APPLIQUÉES**

Sujet de la thèse :

## Restauration des images par l'elimination du flou et des occlusions

---

## Removing Camera Shake Blur and Unwanted Occluders from Photographs

Thèse présentée et soutenue à Cachan le 15 mars 2012
devant le jury composé de :

| | | |
|---|---|---|
| Fredo DURAND | Professeur, Massachusetts Institute of Technology | Rapporteur |
| Rob FERGUS | Professeur, New York University | Rapporteur |
| Nikos PARAGIOS | Professeur, Ecole Centrale de Paris | Examinateur |
| Sylvain PARIS | Chercheur Associé, Adobe Systems Inc. | Examinateur |
| Jean PONCE | Professeur, Ecole Normale Supérieure, Paris | Directeur de thèse |
| Josef SIVIC | Chargé de Recherche, INRIA Paris-Rocquencourt | Directeur de thèse |
| Andrew ZISSERMAN | Professeur, University of Oxford | Directeur de thèse |

Thèse préparée au sein de l'équipe Willow du laboratoire d'informatique de l'École Normale Supérieure, Paris (INRIA/ENS/CNRS UMR 8548). 23 avenue d'Italie, 75214 Paris.

# Acknowledgements

I would like to thank my supervisors Josef Sivic, Andrew Zisserman and Jean Ponce for their expertise, guidance and enthusiasm through these three years, without which this thesis would not have been possible. I would also like to thank all the members of the Willow team, temporary and permanent, for sharing many interesting discussions and providing helpful ideas and expertise throughout my time there.

I am particularly grateful to Fredo Durand and Rob Fergus for graciously accepting to review this thesis, and for their helpful and insightful feedback, and to Sylvain Paris and Nikos Paragios for accepting to participate in my defence as members of the jury.

Finally, I would like to thank my parents and brother for their continual encouragement during this endeavour, and above all Lindsey for her patience, guidance and enduring support, for which I am exceptionally grateful.

This thesis was supported financially by ANR project ANR-07-BLAN-0331-01, the MSR-INRIA laboratory, the EIT-ICT labs (activity 10863), and ERC grant VideoWorld.

# Abstract

This thesis investigates the removal of spatially-variant blur from photographs degraded by camera shake, and the removal of large occluding objects from photographs of popular places. We examine these problems in the case where the photographs are taken with standard consumer cameras, and we have no particular information about the scene being photographed.

Most existing deblurring methods model the observed blurry image as the convolution of a sharp image with a uniform blur kernel. However, we show that blur from camera shake is in general mostly due to the 3D rotation of the camera, resulting in a blur that can be significantly non-uniform across the image. We model this blur using a weighted set of camera poses, which induce homographies on the image being captured. The blur in a particular image is parameterised by the set of weights, which provides a compact global descriptor for the blur, analogous to a convolution kernel. This descriptor fully captures the spatially-variant blur at all pixels, and is able to model camera shake more accurately than previous methods.

We demonstrate direct estimation of the blur weights from single and multiple blurry images captured by conventional cameras. This permits a sharp image to be recovered from a blurry "shaken" image without any user interaction or additional information about the camera motion. For single image deblurring, we adapt an existing marginalisation-based algorithm and a maximum a posteriori-based algorithm, which are both compatible with our model of spatially-variant blur.

In order to reduce the computational cost of our homography-based model, we in-

troduce an efficient approximation based on local-uniformity of the blur. By grouping pixels into local regions which share a single PSF, we are able to take advantage of fast, frequency domain convolutions to perform the blur computation. We apply this approximation to single image deblurring, obtaining an order of magnitude reduction in computation time with no visible reduction in quality.

For deblurring images with saturated pixels, we propose a modification of the forward model to include this non-linearity, and re-derive the Richardson-Lucy algorithm with this new model. To prevent ringing artefacts from propagating in the deblurred image, we propose separate updates for those pixels affected by saturation, and those not affected. This prevents the loss of information caused by clipping from propagating to the rest of the image.

In order to remove large occluders from photos, we automatically retrieve a set of exemplar images of the same scene from the Internet, using a visual search engine. We extract multiple homographies between each of these images and the target image to provide pixel correspondences. Finally we combine pixels from several exemplars in a seamless manner to replace the occluded pixels, by solving an energy minimisation problem on a conditional random field.

Experimental results are shown on both synthetic images and real photographs captured by consumer cameras or downloaded from the Internet.

# Contents

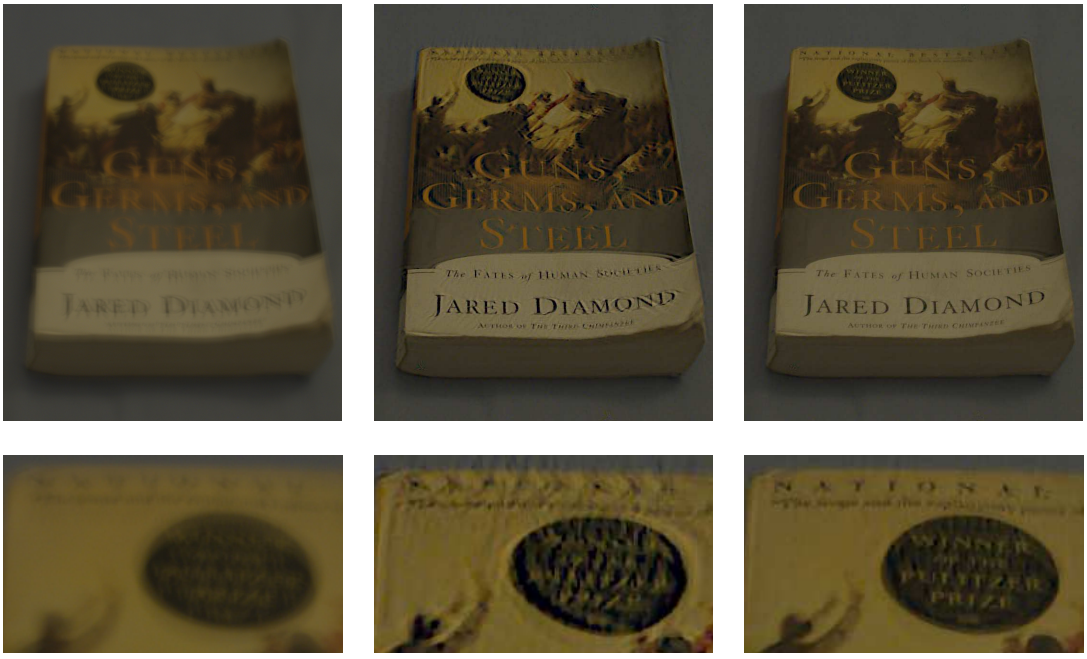
# List of Figures

# Chapter 1

# Introduction

With the explosion of digital photography in recent years, many of us take large numbers of digital photos with cameras or camera-phones. When we review our photos later however, there is sometimes a divergence between what we remember seeing at the time, and what our cameras recorded. This disparity can perhaps be explained by the old adage that we see with our brains, not with our eyes. Whether our photos contain a luridly-dressed tourist that our brain had filtered out at the time, or our photos appear blurry due to camera shake in low light, it is not uncommon that the photos we find ourselves with do not capture what we wanted to record.

In this thesis, we develop models and methods aimed at "restoring" photographs to bring them closer to the images we hoped to record at the time of their taking. For many people, existing photos capture important and fleeting moments that it may be impossible to recapture. As opposed to proposing new hardware or methods for taking future photographs, we are predominantly concerned with handling images that have already been captured. By using accurate models of the image formation process, and incorporating strong prior information on the images we would like to recover, we endeavour to make software post-processing the "brain" to the digital "eyes" of our cameras. Specifically, our objective in this thesis is to automatically restore photographs, when these photographs contains unwanted occluders as shown in Figure 1.2, or when



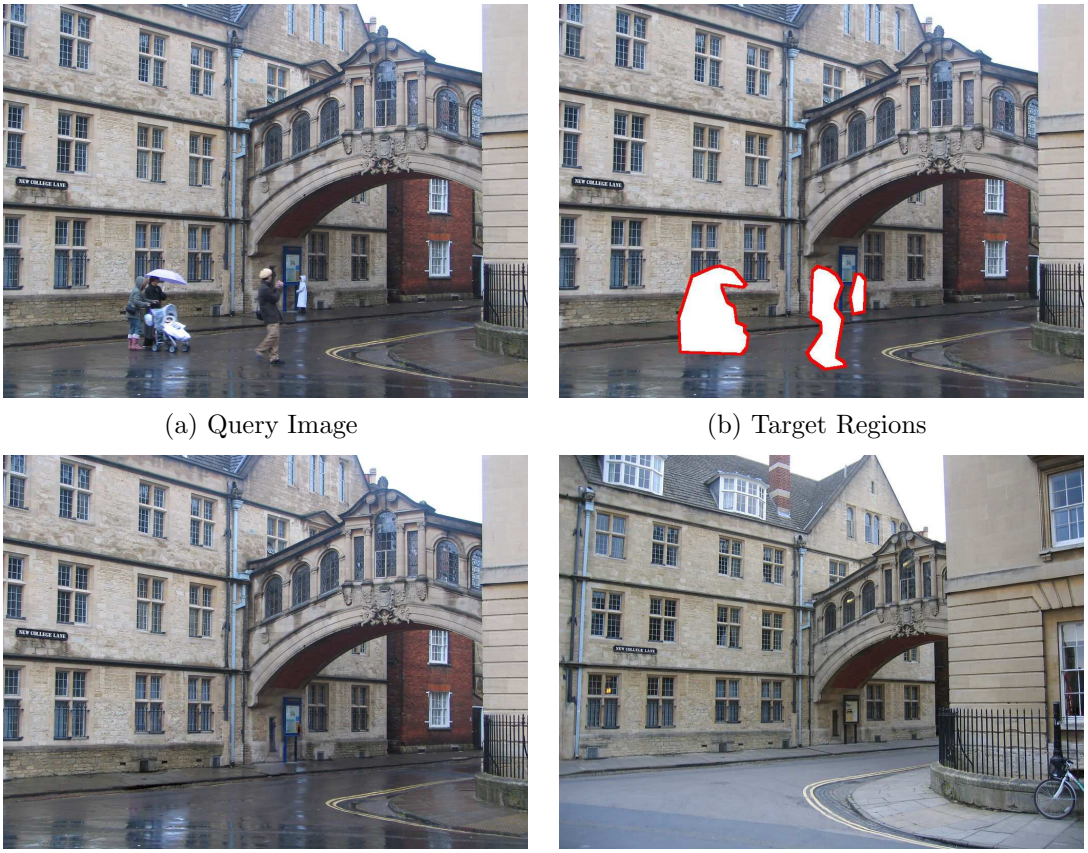

(a) Blurry input image

(b) Deblurred output image using spatially-invariant blur model

(c) Deblurred output image using our blur model

Figure 1.1. **Removing camera shake blur from photos.** The blur caused by camera shake, such as in the blurry image shown here (a), is typically *spatially-variant*. Most previous work on removing camera shake has assumed a *spatially-invariant* blur model, leading to deblurred images such as that shown in (b). Using the model for spatially-variant camera shake blur proposed in Chapter 3 and the blind deblurring algorithms described in Chapter 4, we are able to model the spatially-variant blur correctly and obtain superior deblurred results (c). Despite the large blur (around 30 pixels), much of the text that is illegible in the input image, and which is not restored sufficiently with the spatially-invariant blur model, can be read clearly in our deblurred image.

they are blurry due to camera shake, as shown in Figure 1.1. Besides the emotional motivation for improving peoples' photos, more concrete motivations for these tasks might be the need to recover visual information (e.g. vehicle license plate numbers) from blurry photos, to reduce the cost of obtaining high-quality sharp images (by using cheap computational power instead of expensive camera hardware), or to remove trademarked or otherwise sensitive objects from photos before publishing them.



(a) Query Image (b) Target Regions

(c) Our Result (d) A reference view of the same scene

Figure 1.2. **Removing occluders from photos.** This figure shows an example result produced by the system described in Chapter 7. The tourists are removed and replaced with a faithful rendition of the underlying scene, as can be seen by comparison with another image of the scene, shown in (d).

## 1.1 Problem Statement

The types of image degradation considered in this work lead to inherently ill-posed image restoration problems. Starting from only a degraded image, we wish to recover a good image of the same scene. If there is a loss of information, or we have more unknowns than observations, there may be a large family of valid solutions, which we must somehow choose amongst when producing the "restored" image. This is the case in both the problems discussed here. For deblurring images, where the blur is unknown, we

must estimate both the parameters describing the blur and the sharp image. Since the sharp image has the same number of pixels as the blurry image, we evidently have more unknowns than observations. When we wish to remove occluders from photographs, the recorded image contains no information about what is behind the occluder; it could be concealing a building, a tree, a patch of grass, or a group of people. Making the right choice and producing visually-pleasing results requires good models of the image formation process, and equally importantly, good prior information about the unknowns.

## 1.2 Contributions

The main contributions of this thesis can be divided broadly into deblurring of camera shake, and removing large occluders from photographs.

### 1.2.1 Restoring Photographs Blurred Due to Camera Shake

The main contributions with respect to deblurring camera shake are first to demonstrate that camera shake is mainly caused by 3D rotation, as opposed to 3D translation of the camera, causing spatially-variant blur, and second to derive a geometric model for the blur process based on this. We propose a formulation of this model which is directly applicable in existing blind deblurring algorithms, and consequently demonstrate the ability to remove spatially-variant camera shake blur from photographs. Figure 1.1 demonstrates an example result on a real image blurred by camera shake, and shows that the spatially-variant blur model allows us to recover a significantly better result than only modelling spatially-invariant blur.

We propose an efficient approximation for this model which significantly reduces the computational burden associated with using it, and makes spatially-variant blind deblurring of camera shake practical for real images. Finally we address deblurring of images containing clipped, or saturated pixels. We propose a forward model that includes sensor saturation, and propose a non-blind deblurring algorithm that incorporates this model while preventing artefacts from appearing in the deblurred results. Figure 1.3

(a) Original blurry image (b) Our deblurred result (c) Canon's "Second Shot"

Figure 1.3. **Deblurring a real saturated image.** Taken from Canon's "Your Second Shot" advertising campaign. The original image (a) is very blurry, however using blind PSF estimation (Chapter 4) to estimate how the image was blurred, followed by non-blind deblurring, handling the saturated pixels (Chapter 6) we are able to obtain a much better image (b). For the advertising campaign, Canon flew the couple back to Barcelona to take the photo again (c) with a new camera.

shows a result deblurring a real image, with saturated highlights in the background. The image is taken from Canon's "Second Shot" advertising campaign[1], in which the owners of badly blurred or degraded images were given the chance to go back and retake their photographs with a new camera. Our result is of course not as good as Canon's retaken photo, but is a little more achievable for those without the means or desire to return to the scene of every blurry photo they ever took.

### 1.2.2 Removing Occluders from Photographs

Our contribution with respect to removing large occluders from photographs is to propose an automatic system for replacing occluded regions of photographs using photographs of the same scene, retrieved from the Internet. Our system is able to take an input image with the occluders marked by a user, and return a restored image, where the occluded region has been seamlessly replaced with realistic image content that corresponds to the true underlying scene. Figure 1.2 demonstrates an example result produced by our system. Note that the underlying scene is realistically rendered, without any knowledge of the 3D structure of the scene or of the environmental conditions.

[1] `http://yoursecondshot.usa.canon.com/`

## 1.3 Thesis Outline

We begin in Chapter 2 with some background on image restoration and a discussion of relevant work from the literature.

In Chapter 3 we examine blur caused by camera shake. We begin by considering the geometric relationship between motion of the camera and the apparent motion of the image. Following this, we propose and discuss a novel formulation for camera shake blur, which handles spatially-variant blur naturally. We present the discrete equivalent of our model, and provide a comparison to other models for camera shake blur. Finally we discuss some practical implementation considerations.

In Chapter 4 we demonstrate blind deblurring of shaken images, demonstrating that our model is easily applicable within existing deblurring paradigms. We first apply our model within two algorithms for single-image deblurring, using two different approaches to estimating the parameters of the blur. We present results of these algorithms and discuss some of the advantages and disadvantages of using our model instead of the classical spatially-invariant blur model. Secondly we demonstrate deblurring of images when there is a additional noisy image available of the scene. We conclude with some implementation issues relevant to all the algorithms presented.

In Chapter 5 we introduce an efficient approximation scheme for our spatially-variant blur model. We begin by examining the bottlenecks in the deblurring process, before presenting our approximation scheme and showing how our approximation greatly reduces the computational burden of these steps. We present results using the approximation, and compare to those produced using the exact model.

In Chapter 6 we approach deblurring of images containing clipped / saturated pixels. We discuss possible ways of handling such pixels in the deblurring process, and propose a modification to the image formation model which incorporates the saturation process. Further to this, we examine how saturated pixels cause artefacts in the deblurred results, and propose an algorithm which handles these clipped pixels explicitly to prevent visible artefacts being introduced. Finally we present results of our method and compare to

other methods.

In Chapter 7, we approach the removal of large occluders from photographs of famous landmarks. We discuss the visual search engine which we use to retrieve a set of example images of the same scene, and how we register these images geometrically and photometrically to an input image. Following this we present our algorithm for replacing the occluders, by combining pixels from several of the example images. We present results of our system on a range of input images.

Finally in Chapter 8 we summarise the contributions of this thesis and discuss potential directions for future work.

## 1.4 Publications

Parts of the work in this thesis have appeared in the following publications:

O. Whyte, J. Sivic, and A. Zisserman. Get out of my picture! Internet-based inpainting. In *Proceedings of the 20th British Machine Vision Conference*. London, 2009

O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 2010

O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. In *Proceedings of the IEEE Workshop on Color and Photometry in Computer Vision (CPCV 2011), with ICCV 2011*. Barcelona, Spain, 2011

O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision* 98(2), pp. 168–186, 2012

# Chapter 2

# Background and Related Work

Image restoration is a vast field of research, having been studied for many decades since the advent of digital imagery. No such thing as a perfect camera exists, and all images are, to some extent, affected by noise and blur. Classical work on digital image restoration tackled these problems, and are covered in depth by Gonzalez and Woods (1992). Recently, the field has expanded to include more diverse sources of degradation such as chromatic aberrations or missing pixels. Without doubt, the use of *a priori* information has been crucial in many recent advances, often allowing surprising amounts of of information to be recovered (or perhaps, hallucinated) from images degraded by noise, blur, or other sources of corruption. This information may concern the statistics of undegraded images, the properties of the underlying scene, or other factors such as idiosyncrasies of human perception. In this chapter we recap some of the most relevant work to the subjects covered in this thesis: deblurring and inpainting.

The problem of restoring images can be broken down into several components. First, a model is needed to relate the undegraded, "ideal" image to the observed image produced by the camera. Second, the parameters of this model must be estimated, and finally the sharp image can be estimated, given the model and the estimated parameters.

We begin by introducing a general forward model for image degradation in Sec-

tion 2.1, and discuss the concepts which underpin successful image restoration algorithms. Following from a probabilistic model, which relates the unknown (latent) ideal image with the observations and our prior knowledge, we discuss the two general approaches to image restoration by maximising or marginalising this probability. In Section 2.2 we discuss existing algorithms for deblurring images when the blur is known. Following this, in Section 2.3 we recap some of the existing work for deblurring images when the blur is unknown. In Section 2.4 we introduce the problem of parameterising spatially-variant blur, and discuss some recent work on this problem. Finally in Section 2.5 we address the relevant work and background to the problem of "inpainting".

## Notation

In this thesis we will use some notation consistently, which we introduce here for reference. We use bold lower-case letters to denote vectors, e.g. $\mathbf{f}$, and bold upper-case letters to denote matrices, e.g. $\mathbf{A}$. We use subscripts on non-bold letters to index into vectors and matrices, e.g. $f_j$ indicates the $j^{\text{th}}$ element of the vector $\mathbf{f}$, while $A_{ij}$ indicates element $(i, j)$ of matrix $\mathbf{A}$.

Calligraphic letters such as $\mathcal{U}$ denote sets or domains, depending on the context. We denote functions with lower-case letters, i.e. $f : \mathbb{R} \to \mathbb{R}_+$.

To denote 2D discrete images we use 1D vectors, e.g. $\mathbf{f} \in \mathbb{R}^N$ for an image of height $H$ and width $W$, where $N = H \times W$. In this context we denote 2D convolutions between two images as operating directly on the 1D vectors, i.e. $\mathbf{u} * \mathbf{v}$, and likewise we denote the 2D discrete Fourier transform simply as a function mapping a vector to a vector $\mathcal{F} : \mathbb{R}^N \to \mathbb{C}^N$. We denote the Hadamard (element-wise) product between two vectors by $\mathbf{u} \circ \mathbf{v}$. We abuse standard notation somewhat and write the element-wise division of two vectors simply using a fraction $\frac{\mathbf{u}}{\mathbf{v}}$.

When several objects of the same type are collected into a set, we use numeric superscripts in parentheses to index into this set, e.g. $\mathbf{d}^{(q)} \in \{\mathbf{d}^{(\mathbf{q})}\}$. When objects of the same type are identified by a symbolic property, we use superscripts without parentheses, e.g. $\mathbf{d}^x$, $\mathbf{d}^y$ to indicate derivative filters in the $x$ and $y$ directions.

## 2.1 Image Degradation and Restoration

To frame the problems discussed in this thesis, we begin with a general forward model of image degradation. There are four important components in the image degradation process:

- The (observed) degraded image $g : \Omega \to \mathbb{R}_+$, which is the image output by the camera's sensor. The domain $\Omega \subset \mathbb{R}^2$ is the 2D plane of the camera's sensor.
- The (latent) sharp image $f : \Omega' \to \mathbb{R}_+$. This is the underlying, ideal, sharp image of the scene, which we would like to recover. In most cases, the domain $\Omega' \subset \mathbb{R}^2$ is chosen to be the same as $\Omega$.
- The degradation operator $H$ that acts on $f$, and which describes how light from the sharp image $f$ is distributed in the observed image $g$. Depending on the situation, $H$ may be known in advance (e.g. from optical properties of the camera) or may be unknown (e.g. arbitrary camera motion during camera shake).
- The random noise $N$ that perturbs the recorded image after the sharp image has been degraded by $H$.

These components are combined in the generic image degradation model

$$g(\mathbf{x}) = N\big((Hf)(\mathbf{x})\big), \tag{2.1}$$

where $\mathbf{x}$ is a point $(x, y) \in \Omega$. We will denote the "noiseless" degraded image $Hf$ by $g^*$, and also assume that the degradation operator $H$ is linear, allowing us to write

$$g^*(\mathbf{x}) = (Hf)(\mathbf{x}) = \int_{\Omega'} h(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \, \mathrm{d}\mathbf{x}', \tag{2.2}$$

where $\mathbf{x}'$ is a point $(x', y') \in \Omega'$, and $h : \Omega \times \Omega' \to \mathbb{R}_+$ is referred to as the *impulse response* of $H$, or the *point-spread function* (PSF). For a point light source with magnitude 1 at a particular point $\mathbf{x}'_0 \in \Omega'$, the 2D function $h(\,\cdot\,, \mathbf{x}'_0)$ is the response produced by $H$, and

describes how the light from this point is spread over the observed image (Gonzalez and Woods, 1992). In Section 2.1.2 we discuss possible forms for the PSF, and in Section 2.1.4 we discuss several noise models that are useful in practice.

It is important to note that in general, for a given degraded image $g$, both the sharp image $f$ *and* the PSF $h$ may be unknown. This makes the image restoration problem particularly difficult. The forward model plays an important role in solving this problem, but some additional information is typically needed to recover a good estimate of the latent image. This additional information can come in the form of statistical priors, which encourage the latent image to look realistic, and improve the conditioning of the problem. In Section 2.1.5 we discuss priors for the latent image and for the PSF.

#### Spatially-invariant Blur

A common assumption when modelling image blur is that the PSF is spatially-invariant, which is to say that there exists a function $a : \mathbb{R}^2 \to \mathbb{R}_+$ such that $h(\mathbf{x}, \mathbf{x}') = a(\mathbf{x} - \mathbf{x}')$. The function $a$ is typically referred to as a *convolution kernel*, and in this case the dimensionality of the PSF is reduced from four to two, and Equation (2.2) reduces to a 2D convolution of $f$ with $a$:

$$g^*(\mathbf{x}) = (a * f)(\mathbf{x}) = \int_{\Omega'} a(\mathbf{x} - \mathbf{x}') f(\mathbf{x}') \, \mathrm{d}\mathbf{x}'. \tag{2.3}$$

In general, blur may be spatially-variant, and we will return to this issue in Section 2.4, and again in Chapter 3.

### 2.1.1 The Discrete Setting

Real cameras are equipped with a discrete set of pixels, and output a discrete set of samples of the degraded image, denoted by the vector $\mathbf{g} \in \mathbb{R}_+^N$, where $N = H \times W$ pixels for an image with $H$ rows and $W$ columns. We consider the sharp image also to be discrete: $\mathbf{f} \in \mathbb{R}_+^N$. We use $i$ to index into the degraded image $\mathbf{g}$, i.e. $g_i = g(\mathbf{x}_i)$, where $\mathbf{x}_i \in \Omega$ is the coordinate of the $i^{\text{th}}$ pixel. Likewise, we use $j$ to index into the

sharp image $\mathbf{f}$, such that $f_j = f(\mathbf{x}'_j)$ for a coordinate $\mathbf{x}'_j \in \Omega'$. Finally, we note that to evaluate an image at arbitrary (sub-pixel) locations, we interpolate from nearby pixels. In this work, we use linear interpolation schemes, whereby sub-pixel values of an image, say $g(\mathbf{x})$ are interpolated as a linear combination of nearby pixels:

$$g(\mathbf{x}) = \sum_i b(\mathbf{x}, \mathbf{x}_i) g_i, \tag{2.4}$$

where the coefficients $b(\mathbf{x}, \mathbf{x}_i)$ are calculated using a standard method such as bilinear or bicubic interpolation.

In this discrete setting, we can write Equation (2.2) as

$$g_i^* = \sum_j A_{ij} f_j, \tag{2.5}$$

or in matrix-vector notation, $$\mathbf{g}^* = \mathbf{A}\mathbf{f}, \tag{2.6}$$

where the $N \times N$ matrix $\mathbf{A}$ captures the discrete PSF. Each column of the matrix $\mathbf{A}$ contains the PSF for the corresponding pixel in the latent image $\mathbf{f}$. In most cases of blur, the light from each pixel in $\mathbf{f}$ is spread over a relatively small number of nearby pixels in $\mathbf{g}$. As a result, the PSF matrix $\mathbf{A}$ for an image is usually sparse (contain a relatively small number of non-zero values). When the PSF is spatially-invariant, we denote the discrete convolution kernel by $\mathbf{a}$, and write

$$\mathbf{g}^* = \mathbf{a} * \mathbf{f}. \tag{2.7}$$

### 2.1.2 Types of Degradation

In this thesis we are concerned with two types of PSF: those arising from camera shake, which causes image blur, and those arising when some pixels from the observed image are missing or deleted.

- **Image Blur.** When an image is degraded by blur, light from a single point in $f$ is spread across a region in $g$. In this case, the PSF $\mathbf{A}$ will have many non-zeros in

each column. When the PSF is unknown, the problem is typically referred to as "blind" deblurring. If the PSF is known *a priori*, or has already been estimated, the problem of recovering the sharp image is, by contrast, referred to as "non-blind" deblurring.

- **Missing or Deleted Pixels.** When the image has been corrupted in certain regions, or contains occluders that we wish to remove, the PSF is simply the identity, i.e. $\mathbf{A} = \mathbf{I}$, since we assume that no blurring has occurred. We model the missing pixels as noise with very high uncertainty, such that their observed intensity is unrelated to their latent intensity.

### 2.1.3 Probabilistic Formulation

Given the general model of image degradation discussed above, a natural starting point for image restoration algorithms is to write down a probabilistic generative model for the observed image. If we know the type of noise affecting the observed image, we can write down the likelihood of the observed image $p(\mathbf{g}|\mathbf{f}, \mathbf{A})$, which is the probability density of $\mathbf{g}$, conditioned on $\mathbf{f}$ and $\mathbf{A}$. If we then wish to find the latent image and PSF which best match the observed image $\mathbf{g}$, an obvious choice would be to find the $\mathbf{f}$ and $\mathbf{A}$ which maximise this likelihood. Due to the loss of information that occurs in the degradation process, however, image restoration algorithms which simply maximise this likelihood are known to be ill-conditioned. Algorithms of this sort are prone to producing results containing artefacts, and in which noise is amplified.

If we have some prior knowledge about the latent variables being recovered, then using Bayes' rule, we can formulate the posterior distribution for the unknowns (the sharp image $\mathbf{f}$ and also, if unknown, the PSF $\mathbf{A}$). The posterior incorporates the likelihood, which arises from the random noise in the observed image, and also prior knowledge about the unknowns, and in our case is given by

$$p(\mathbf{f}, \mathbf{A}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{A})p(\mathbf{f})p(\mathbf{A}) \qquad \text{for an unknown PSF} \tag{2.8}$$

$$p(\mathbf{f}|\mathbf{g}, \mathbf{A}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{A})p(\mathbf{f}) \qquad \text{for a known PSF.} \tag{2.9}$$

The priors $p(\mathbf{f})$ and $p(\mathbf{A})$ can either be manually defined, or learnt from example data. By incorporating this prior knowledge about the unknown variables, the posterior ameliorates the instability of estimating $\mathbf{f}$ or $\mathbf{A}$ from the likelihood alone. We will discuss several popular choices for these terms in the following sections.

Given the posterior distribution, it is the task of image restoration algorithms to find the "most probable" sharp image $\mathbf{f}$. The idea of what is "most probable" however is not clear: is it the image $\mathbf{f}$ that maximises the posterior probability over all possible $\mathbf{f}$ and $\mathbf{A}$, or is it the expected value of $\mathbf{f}$ under the distribution? In subsequent sections we will describe different approaches to the image restoration problem, based on different answers to this question. Finally we note that although not all image restoration algorithms have a probabilistic interpretation, this formulation can nevertheless provide some useful intuition into why they succeed.

#### Maximim A Posteriori

One popular method of finding the "most probable" values of the unknown variables in a system is to find those values which maximise the posterior probability. In this approach, for a known PSF, the estimated latent image $\hat{\mathbf{f}}$ is found as

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} p(\mathbf{f}|\mathbf{g}, \mathbf{A}), \tag{2.10}$$

while for unknown PSF, the latent image and PSF are estimated simultaneously

$$\{\hat{\mathbf{f}}, \hat{\mathbf{A}}\} = \arg\max_{\mathbf{f},\mathbf{A}} p(\mathbf{f}, \mathbf{A}|\mathbf{g}). \tag{2.11}$$

This maximisation is typically addressed by first transforming the the probability maximisation problem into an energy minimisation problem. The forms of the likelihood and priors in Equation (2.8) are generally chosen such that the posterior can be written

as a Gibbs distribution, with the form

$$p(\mathbf{f}, \mathbf{A}|\mathbf{g}) = \frac{1}{Z} \exp\left(-\tfrac{1}{T} U(\mathbf{f}, \mathbf{A})\right), \tag{2.12}$$

where $U$ is an energy function which depends on $\mathbf{g}$, and $Z$ is a normalising constant. For this distribution, the sharp image $\mathbf{f}$ and PSF $\mathbf{A}$ which maximise the posterior probability are those which minimise the energy function $U$. Given that $U(\mathbf{f}, \mathbf{A}) \propto -\log p(\mathbf{f}, \mathbf{A}|\mathbf{g}) +$ const., we can write the MAP problem for the posterior in Equation (2.8) as

$$\min_{\mathbf{f},\mathbf{A}} F\big(\mathbf{g}, \mathbf{A}\mathbf{f}\big) + \alpha\rho_{\mathbf{f}}(\mathbf{f}) + \beta\rho_{\mathbf{A}}(\mathbf{A}),, \tag{2.13}$$

where the function $F$ is derived as the negative log-likelihood, and penalises latent images or PSFs which do not agree with the observed data, while the functions $\rho_{\mathbf{f}}$ and $\rho_{\mathbf{A}}$ are derived as the negative log-priors, and penalise latent images or PSFs which are unlikely under those priors. The function $F$ is referred to as the data fidelity term, while $\rho_{\mathbf{f}}$ and $\rho_{\mathbf{A}}$ are referred to as the regularisation terms.

The difficulty in the MAP approach is that the search space is very large, with potentially millions of unknowns. Furthermore the energy function may not be convex, in which case it is not generally possible to reach the globally optimal solution. Other problems include the unknown noise variance in the observed image, which affects the appropriate regularisation weights $\alpha$ and $\beta$ to use, and furthermore the fact that under some commonly used image priors, the blurry image is itself a local minimum of (Levin et al., 2009).

### Marginalisation

An alternative approach to maximising the posterior probability is to attempt to find the expected (mean) values of the unknowns under the posterior distribution. In this approach, for a known PSF the estimated latent image is calculated as

$$\hat{\mathbf{f}} = \int \mathbf{f}\ p(\mathbf{f}|\mathbf{g}, \mathbf{A})\, \mathrm{d}\mathbf{f}. \tag{2.14}$$

while for unknown PSF, the latent image and PSF would be estimated as

$$\hat{\mathbf{f}} = \int\int \mathbf{f}\ p(\mathbf{f}, \mathbf{A}|\mathbf{g})\, \mathrm{d}\mathbf{f}\, \mathrm{d}\mathbf{A} \tag{2.15}$$

$$\hat{\mathbf{A}} = \int\int \mathbf{A}\ p(\mathbf{f}, \mathbf{A}|\mathbf{g})\, \mathrm{d}\mathbf{f}\, \mathrm{d}\mathbf{A}. \tag{2.16}$$

This approach sometimes has advantages over maximising the posterior, since the MAP solution may find a peak which has high probability density but has very little probability mass below it. Marginalising will find a solution which is a combination of all posssible solutions, weighted by their probability density, and so a wide peak with a lower maximum density may have more influence than a narrow but higher one.

The difficulty in estimating the unknowns by marginalisation is that the expectations in Equations (2.14) to (2.16) are analytically intractable. To evaluate these integrals some approximation strategy must be used, of which there are in general two types: stochastic approximations and parametric approximations. Stochastic approximation methods, such as a Markov Chain Monte Carlo (Neal, 1993), attempt to evaluate the integrals stochastically by drawing samples from the true posterior distribution. Such algorithms expend most of their effort on drawing samples in a way that ensures convergence to the true distribution. Parametric approximation methods, such as ensemble learning (Lappalainen and Miskin, 2000), attempt to find a parametric approximation of the posterior for which the integrals become tractable. These algorithms spend most of their computational effort estimating the parameters of the approximating distribution, while the final marginalisation under the parametric distribution is often trivial or easy to compute.

### 2.1.4 Noise Models

The likelihood term $p(\mathbf{g}|\mathbf{f}, \mathbf{A})$ in the posterior distribution is defined by the type of noise that is present in the observed image. Here we recap three noise models which are used in this work: Gaussian noise, Poisson noise, and uniform noise. In all cases we assume that all pixels $g_i$ in the observed image are independent, conditioned on the sharp image

$\mathbf{f}$ and the PSF $\mathbf{A}$, i.e. $p(\mathbf{g}|\mathbf{f}, \mathbf{A}) = \prod_i p(g_i|\mathbf{f}, \mathbf{A})$. In deblurring, Gaussian and Poisson noise are two widely applicable models (Boncelet, 2005). Uniform noise on the other hand, is useful for modelling corrupted or deleted pixels, such as in inpainting.

### Poisson Noise

Poisson noise is a realistic noise model to use for images, since the arrival of photons at the sensor is naturally modelled as a Poisson process. Indeed, in low light, "quantum noise" (noise caused by the random arrival times of individual photons) dominates over other types of noise in a digital camera. For a Poisson random variable $z$ with mean $\lambda$, we write $z \sim \text{Pois}(\lambda)$, and $z$ has the probability density function (PDF)

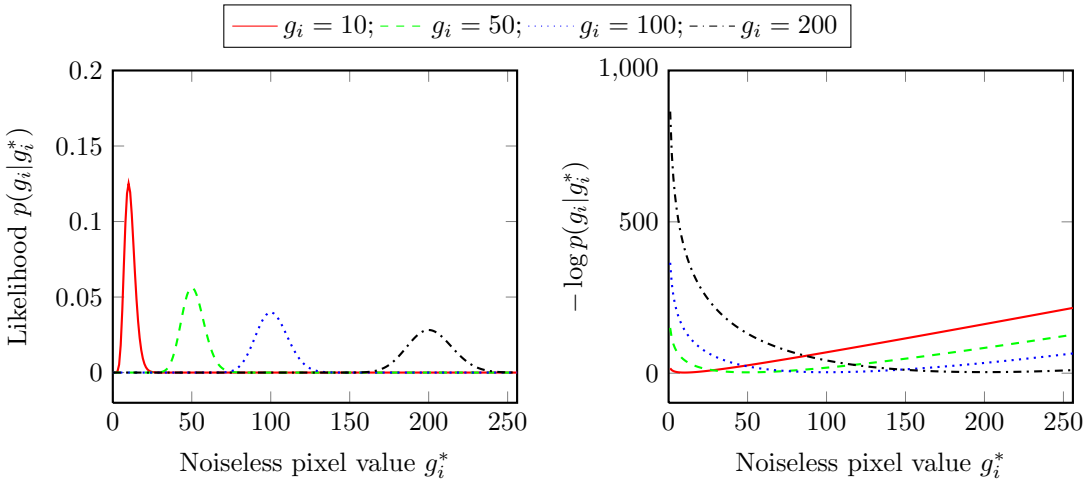
$$p(z) = \frac{\lambda^z e^{-\lambda}}{z!}. \tag{2.17}$$

Applying this in our degradation model, we have $g_i \sim \text{Pois}(g_i^*)$:

$$p(g_i|g_i^*) \propto \frac{{g_i^*}^{g_i} e^{-g_i^*}}{g_i!}, \tag{2.18}$$

$$\text{and} \qquad -\log p(g_i|g_i^*) = -g_i \log g_i^* - g_i^* - \log g_i! + \text{const.} \tag{2.19}$$

Figure 2.1 plots the likelihood $p(g_i|g_i^*)$ as a function of the unknown $g_i^*$, and the corresponding negative log-likelihood, which is used as the data fidelity term in the MAP problem in Section 2.1.3. As can be seen, the peaks of $p(g_i|g_i^*)$ become wider at higher $g_i$, indicating greater uncertainty in $g_i^*$. In image restoration tasks, where we wish to estimate $g_i^*$, this corresponds to penalising errors less at bright pixels than at dark ones. Also, as $g_i^* \to 0$, $-\log p(g_i|g_i^*) \to \infty$, implicitly enforcing positivity on $g_i^*$. Finally, we note that the negative log-likelihood (Equation (2.19) and Figure 2.1 (b)) is convex, allowing a global minimum to be found if we use this noise model in a MAP approach.




(a) Likelihood of a blurry pixel $g_i$ as a function of $g_i^*$ under a Poisson noise model, for several different values $g_i$

(b) Negative log-likelihood for the likelihoods in (a)

Figure 2.1. **The Poisson noise distribution.** The likelihood (a) and negative log-likelihood (b) of different observed pixel values $g_i$, as a function of the noiseless value $g_i^*$.

### Gaussian Noise

Gaussian noise is perhaps the most common noise model used in image processing, due to its tractability and wide applicability. For a Gaussian random variable $z$ with mean $\mu$ and variance $\sigma^2$, we write $z \sim \mathcal{N}(\mu, \sigma^2)$, and $z$ has the PDF

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(z-\mu)^2}{2\sigma^2}\right). \tag{2.20}$$

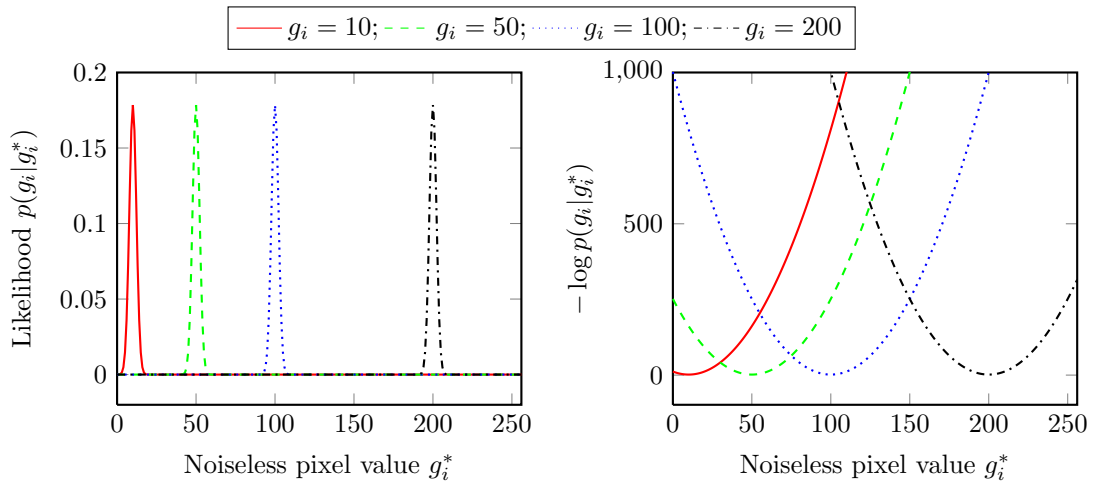
Applying this in our degradation model, we have $g_i \sim \mathcal{N}(g_i^*, \sigma^2)$:

$$p(g_i|g_i^*) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(g_i - g_i^*)^2}{2\sigma^2}\right) \tag{2.21}$$

and

$$-\log p(g_i|g_i^*) = \frac{1}{2\sigma^2}(g_i - g_i^*)^2 + \text{const.} \tag{2.22}$$

Figure 2.2 plots the likelihood $p(g_i|g_i^*)$ for several values of $g_i$ and the corresponding negative log-likelihoods. In contrast to Poisson noise, the variance remains constant at




(a) Likelihood of a blurry pixel $g_i$ as a function of $g_i^*$ under a Gaussian noise model with variance $\sigma^2 = 5$, for several different values $g_i$

(b) Negative log-likelihood for the likelihoods in (a)

Figure 2.2. **The Gaussian noise distribution.** The likelihood (a) and negative log-likelihood (b) of different observed pixel values $g_i$, as a function of the noiseless value $g_i^*$.

all brightnesses, and the likelihood does not enforce positivity on $g_i^*$. Minimisation of the negative log-likelihood is a linear least-squares problem, which is typically easier to solve than that in Equation (2.19), and for which many good algorithms exist. Furthermore it is convex, making it a good candidate for MAP approaches.

### Uniform Noise

As well as the random fluctuations in the light arriving at the sensor, we can also model corrupted or deleted pixels in the image as noise with a uniform distribution. For a uniform random variable $z$, we write $z \sim \mathrm{Unif}(a, b)$, with $a < b$, and $z$ has the PDF

$$p(z) = \begin{cases} 0 & z < a \\ \frac{1}{b-a} & a \leq z \leq b \\ 0 & b < z \end{cases} \tag{2.23}$$

Applying this in our degradation model, we have $g_i \sim \text{Unif}(0, 256)$, and

$$-\log p(g_i|g_i^*) = \begin{cases} \infty & g_i < 0 \\ \text{const.} & 0 \leq g_i \leq 256 \\ \infty & 256 < g_i \end{cases} \cdot \tag{2.24}$$

Under this noise model, an observed pixel $g_i$ is independent of the noiseless pixel $g_i^*$, and is thus decoupled from the latent image $\mathbf{f}$. This noise model is thus applicable when the degraded image contains pixels that we would like to replace, since we would like the recovered image to be independent of the replaced pixels. Recently, Cho et al. (2011) have used the uniform noise model in deblurring, to handle outlier pixels in the blurry image, e.g. due to "hot pixels", or salt and pepper noise.

### 2.1.5 Priors

As mentioned earlier, image restoration problems are frequently ill-posed, such that methods which use the data fidelity term alone may amplify noise in the solutions, or fail completely to find a good solution. As a result, some prior knowledge of the unknowns is often included in the restoration problem to improve the conditioning and stabilise the solution. In this section we discuss several such priors for the latent image and the PSF.

#### Priors for the Sharp Image

An important aspect of image restoration is the prior on the latent image $p(\mathbf{f})$. Many different image priors have been proposed for image reconstruction tasks such as deblurring, denoising, and super-resolution, often based on the statistics of natural images. Early work on the statistics of natural images observed that the amplitude spectrum, when radially averaged over orientation, approximately follows a power law (Field, 1987), or that the entropy of sharp images is higher than that of blurred images (Gull and Skilling, 1984).

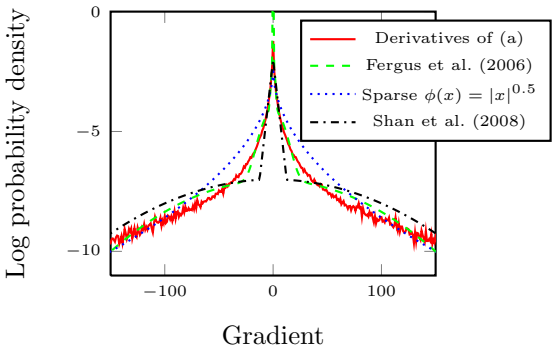
Recent work has focused particularly on the distribution of the responses to dictionaries of filters. For a set of filters $\{\mathbf{d}^{(\mathbf{q})}\}$, these priors model the statistics of the image responses $\{\mathbf{f}^{(\mathbf{q})}\}$, where $\mathbf{f}^{(q)} = \mathbf{d}^{(q)} * \mathbf{f}$. The simplest variant of these considers the response of the image to horizontal and vertical derivative filters $\{\mathbf{d}^x, \mathbf{d}^y\}$, essentially modelling the strength of the correlation between neighbouring pixel values. This use of spatial correlations has a long history (Molina and Ripley, 1989), however early work used Gaussian distributions to model these correlations, which leads to overly-smoothed results when applied to image restoration. It has been observed that the derivatives of natural images in fact generally follow heavy-tailed sparse distributions, rather than a Gaussian distribution. These distributions have a large probability mass near zero, reflecting the fact that most pixels are strongly correlated with their neighbours. However, the heavy tail is also necessary to model sharp edges, which have large gradient magnitudes, and which are not adequately captured by a Gaussian distribution. Beginning with the use of total variation and related norms (Rudin et al., 1992), priors of this nature have been used extensively in image restoration tasks (Fergus et al., 2006; Krishnan and Fergus, 2009; Krishnan et al., 2011; Levin et al., 2007; Shan et al., 2008; Tappen et al., 2003). Priors of this type have proven to be effective at suppressing noise in restored images while encouraging sharp edges to appear. The disadvantage of these priors is that the corresponding regularisers $\rho_{\mathbf{f}}$ in the MAP problem (Section 2.1.3) are generally non-convex. This implies that we cannot find the globally optimal latent image. However, in practice this does not generally cause visible degradation in the restored image.

Fergus et al. (2006) proposed a prior on the sharp image which assumes that the derivatives for all pixels are independent and follow a mixture of $C$ zero-mean Gaussians with mixture weights $\pi_c$ and variances $v_c$ (also known as a Gaussian scale mixture):

$$p(\mathbf{f}^{(q)}) = \prod_j \sum_{c=1}^{C} \pi_c \exp\left(-\frac{{f_j^{(q)}}^2}{2v_c}\right). \tag{2.25}$$

Fergus et al. learn the parameters $\pi_c$ and $v_c$ from real images; for example, the parameters

(a) A street scene




(b) Logarithm of gradient histogram and of various priors

Figure 2.3. **Statistics of, and priors for image gradients in photographs.** (a) shows a photograph of a street scene, from Fergus et al. (2006). In (b), the logarithm of the histogram of gradients is plotted, along with several sparse-derivatives priors.

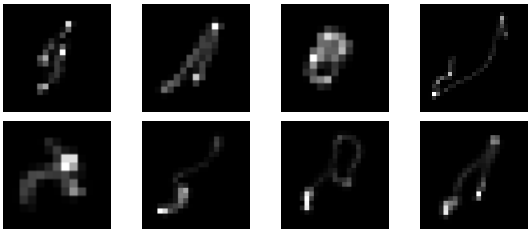
of the prior shown in Figure 2.3 (b) have been learnt on the image shown in Figure 2.3 (a).

Other authors have used priors with the form

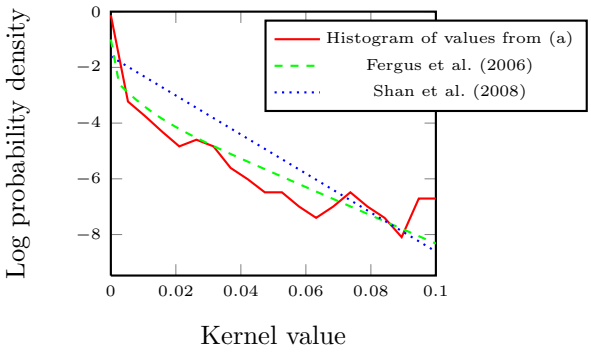
$$p(\mathbf{f}^{(q)}) \propto \prod_j \exp\left(-\phi(f_j^{(q)})\right) \tag{2.26}$$

Krishnan and Fergus (2009); Levin et al. (2007); Tappen et al. (2003) explore the use of the hyper-Laplacian distribution where $\phi(x) = |x|^p$, for some exponent $p$, typically chosen to be $0.5 \leq p \leq 0.8$. Shan et al. (2008) use a piecewise prior where $\phi(x) = \min(k\,|x|\,, ax^2 + b)$, where the parameters $k$, $a$, and $b$ are set manually. Figure 2.3 plots the derivatives of a natural image, and various distributions which have been used to fit this empirical distribution. The histogram is characterised by a sharp peak at zero, with a slow fall-off.

While priors on the derivatives of images concern pairs of neighbouring pixels, other authors have focussed on the statistics of small image patches. Some authors have proposed to use simple sparse distributions with large dictionaries of filters to model this distribution, such as wavelets (Mallat, 1999) or learnt dictionaries (Aharon et al., 2006; Mairal et al., 2008; Olshausen and Field, 1996). Others have proposed to use more



(a) Some typical camera shake blur kernels from Levin et al. (2009)




(b) Logarithm of kernel histograms and of two priors

Figure 2.4. **Statistics of, and priors for camera shake blur kernels.** Levin et al. (2009) estimated a set of "true" camera shake convolution kernels (a) using sharp / blurry pairs of images. In (b), the logarithm of the histogram of all the kernel values in (a) is plotted, along with two sparsity-inducing priors.

complex distributions and learn their parameters (Roth and Black, 2005; Zoran and Weiss, 2011).

### Priors for the PSF

When we wish to estimate the PSF for a blurry image, we also need priors for the parameters of the PSF. For example, when the blur is spatially-invariant and parameterised by a convolution kernel $\mathbf{a}$, we use the prior $p(\mathbf{a})$. Since in this work we are primarily concerned with blur caused by camera shake, we will discuss priors related to this type of PSF here.

Successful algorithms for estimating camera shake PSFs generally share the same two pieces of prior information about the blur kernel being estimated. First, all its elements are non-negative, since the image formation process is additive, with sensor elements accumulating photons, i.e. $a_k \geq 0$ for $k = 1, \ldots, K$. The second, and arguably more important fact to observe about a blur kernel for camera shake is that it should be sparse, i.e. contain relatively few non-zero elements. Figure 2.4 shows some typical camera shake kernels, which clearly exhibit these properties.

The nonnegative sparsity prior has been a prominent feature of previous camera shake removal algorithms, and has also been leveraged for the alignment of blurred image with non-blurred images (Yuan et al., 2007a). Fergus et al. (2006) encourage sparsity by placing a mixture of exponentials prior on the kernel values, such that

$$p(\mathbf{a}) = \prod_k \sum_{d=1}^{D} \pi_d \exp(-\lambda_d a_k), \tag{2.27}$$

where $\pi_d$ is the mixture weight for the $d^{\text{th}}$ exponential, with rate parameter $\lambda_d$. Fergus et al. (2006) learn the parameters $\{\pi_\mathbf{d}\}$ and $\{\lambda_\mathbf{d}\}$ from a set of real blur kernels. Shan et al. (2008) use a single exponential distribution for each of the kernel elements, such that

$$p(\mathbf{a}) = \prod_k \lambda \exp(-\lambda a_k). \tag{2.28}$$

Figure 2.4 (b) plots these priors along with a histogram of values from some real camera shake blur kernels.

In a contrasting approach, Cai et al. (2009) choose to construct the blur kernel as a linear combination of a predefined set of "curvelets", i.e. $\mathbf{a} = \mathbf{\Phi b}$ where $\mathbf{\Phi}$ is the matrix of curvelets, and place an exponential prior on the coefficients $\mathbf{b}$ of the curvelets, rather than on the kernel elements directly. By forcing the kernel to be constructed from a set short curves, this approach encourages the kernel to contain thin linear structures, which intuitively correspond to the camera's motion path.

Cho and Lee (2009) and Yuan et al. (2007b) use simple Gaussian priors to regularise the kernel values, followed by thresholding such that most of the kernel elements are set to zero.

## 2.2 Algorithms for Non-Blind Image Deblurring

When the PSF for a blurry image is known, the problem of estimating the sharp image $\mathbf{f}$ is typically referred to as "non-blind" image restoration. This problem has also

been studied for a long time, e.g. (Wiener, 1949). Many algorithms exist for non-blind deblurring, perhaps most famously the Richardson-Lucy algorithm (Lucy, 1974; Richardson, 1972), which recovers the maximum-likelihood estimate of the sharp image under a Poisson noise model. For a Gaussian noise model, the maximum-likelihood solution can be found using standard least-squares methods such as conjugate gradient descent (Shewchuk, 1994), and for this reason, Gaussian noise has been favoured in the recent literature. Much work has revolved around the incorporation of strong priors, such as those described in Section 2.1.5, to suppress noise in the output while encouraging realistic image features to appear. In theory, the image priors are independent of the noise model used, so we can mix and match. In practice, particular authors have considered certain combinations. In general, maximum likelihood and maximum a posteriori algorithms are used to estimate the latent image, as opposed to marginalisation algorithms, and in this section we do not consider marginalisation approaches to non-blind deblurring, although this is feasible, as mentioned later in Section 4.3.3.

### 2.2.1 Non-Blind Deblurring With Poisson Noise

One very popular algorithm for non-blind deblurring under a Poisson noise model is the Richardson-Lucy algorithm (Lucy, 1974; Richardson, 1972). This algorithm has been shown to converge to the maximum-likelihood solution under this noise model (Shepp and Vardi, 1982). Variants of this algorithm have also appeared for general inversion and matrix-factorisation problems (Lee and Seung, 2001). We can write the likelihood of the blurry image $\mathbf{g}$ as a function of $\mathbf{f}$ as follows:

$$p(\mathbf{g}|\mathbf{f}) = \prod_i \frac{g_i^{*g_i} e^{-g_i^*}}{g_i!}, \quad \text{where} \tag{2.29}$$

$$g_i^* = \sum_j A_{ij} f_j. \tag{2.30}$$

The negative log-likelihood is then:

$$\mathcal{L}_{\text{pois}} = \sum_i g_i^* - g_i \log g_i^* + \log(g_i!). \tag{2.31}$$

As a maximum-likelihood algorithm, the Richardson-Lucy algorithm minimises $\mathcal{L}_{\text{pois}}$ over $\mathbf{f}$. The Karush-Kuhn-Tucker conditions for the minimum are, for all $f_j$:

$$f_j \frac{\partial}{\partial f_j} \mathcal{L}_{\text{pois}} = 0, \tag{2.32}$$

$$\text{and} \qquad \frac{\partial}{\partial f_j} \mathcal{L}_{\text{pois}} \geq 0 \qquad \text{if } f_j = 0. \tag{2.33}$$

The Richardson-Lucy update rule can be derived as a fixed-point iteration from the condition in Equation (2.32):

$$f_j \sum_i \frac{\partial}{\partial f_j} \left(g_i^* - g_i \log g_i^* + \log(g_i!)\right) = 0 \tag{2.34}$$

$$f_j \sum_i \left( \frac{\partial g_i^*}{\partial f_j} - \frac{g_i}{g_i^*} \frac{\partial g_i^*}{\partial f_j} \right) = 0 \tag{2.35}$$

$$f_j \sum_i A_{ij} = f_j \sum_i A_{ij} \frac{g_i}{g_i^*}, \tag{2.36}$$

Assuming that the blur is conservative, i.e. $\sum_i A_{ij} = 1$, we obtain the update rule by replacing the unknowns $f_j$ and $g_i^*$ with our estimates $\hat{f}_j$ and $(\mathbf{A}\hat{\mathbf{f}})_i = \sum_{j'} A_{ij'} \hat{f}_{j'}$, and adding iteration indices:

$$\hat{f}_j^{t+1} = \hat{f}_j^t \sum_i A_{ij} \frac{g_i}{(\mathbf{A}\hat{\mathbf{f}}^t)_i}, \tag{2.37}$$

where $t$ indexes the iteration. In matrix-vector form, this update equation is

$$\hat{\mathbf{f}}^{t+1} = \hat{\mathbf{f}}^t \circ \mathbf{A}^\top \Big( \frac{\mathbf{g}}{\mathbf{A}\hat{\mathbf{f}}^t} \Big), \tag{2.38}$$

where $\circ$ represents the Hadamard (element-wise) product and the fraction represents element-wise division of two vectors. Note that the convergence of the Richardson-Lucy algorithm can be improved significantly with acceleration schemes, such as that of Biggs and Andrews (1997), however for simplicity we do not discuss these here.

### Incorporating Image Priors

In the form in Equation (2.38), the Richardson-Lucy algorithm corresponds to a maximum-likelihood algorithm, and as such noise is amplified in the deblurred result. The most popular way of mitigating this problem is to stop the algorithm before it has converged, typically after 20-50 iterations. This "early-stopping" technique is a form of regularisation, although it does not correspond to a probabilistic prior that can be written down (Bishop, 2006). Probabilistic priors however can also be incorporated in the algorithm by starting from the posterior $p(\mathbf{f}|\mathbf{g}, \mathbf{A}) = p(\mathbf{g}|\mathbf{f}, \mathbf{A})p(\mathbf{f})$, which leads to the following regularised update rule:

$$\hat{\mathbf{f}}^{t+1} = \frac{\hat{\mathbf{f}}^t}{1 - \alpha \nabla \rho_{\mathbf{f}}(\hat{\mathbf{f}}^t)} \circ \mathbf{A}^\top \Big( \frac{\mathbf{g}}{\mathbf{A}\hat{\mathbf{f}}^t} \Big), \tag{2.39}$$

where $\alpha$ is the regularisation weight, and $\nabla \rho_{\mathbf{f}}(\hat{\mathbf{f}}^t)$ is the vector of partial derivatives of the regulariser $\rho_{\mathbf{f}}$ with respect to each latent pixel, evaluated at the current estimate $\hat{\mathbf{f}}^t$, i.e. $\big(\nabla \rho_{\mathbf{f}}(\hat{\mathbf{f}}^t)\big)_j = \frac{\partial \rho_{\mathbf{f}}}{\partial f_j}\Big|_{\mathbf{f}=\hat{\mathbf{f}}^t}$. Using this formulation, Tai et al. (2011) have investigated the use of various types of regularisation with the Richardson-Lucy algorithm in detail. Figure 2.5 shows a synthetic example of an image blurred and corrupted by Poisson noise, along with the non-blind deblurring results using both the standard and regularised Richardson-Lucy algorithm. Note that this method limits the magnitude of the regularisation weight $\alpha$, as the update vector may become negative if $\alpha$ is too large. Welk (2010) avoids this limitation by splitting the regularisation gradient into positive and negative contributions:

$$\hat{\mathbf{f}}^{t+1} = \frac{\hat{\mathbf{f}}^t}{1 - \alpha [\nabla \rho_{\mathbf{f}}(\hat{\mathbf{f}}^t)]_-} \circ \left( \mathbf{A}^\top \Big( \frac{\mathbf{g}}{\mathbf{A}\hat{\mathbf{f}}^t} \Big) + \alpha [\nabla \rho_{\mathbf{f}}(\hat{\mathbf{f}}^t)]_+ \right), \tag{2.40}$$

(a) Original image

(b) Blurred with shown kernel, and Poisson noise added

(c) Deblurred with Richardson-Lucy

(d) Deblurred with regularised RL (using prior from (2.26) with $\phi(x) = |x|^{0.8}$)

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

Figure 2.5. **Non-blind deblurring with Poisson noise.** In this synthetic example, the original sharp image (a) is blurred and corrupted with Poisson noise to produce the degraded image (b). The deblurred result (c) is produced using the Richardson-Lucy algorithm. Since no regularisation is used, noise is amplified and artefacts appear in the deblurred result (e.g. the vertical stripes to the right of the table). In the regularised Richardson-Lucy result (d), the noise and artefacts are reduced, although some image features are also suppressed.

where $[z]_{\pm} = \frac{1}{2}(z \pm |z|)$. This update rule ensures that the non-negativity constraint on $\hat{\mathbf{f}}$ is not violated, regardless of the value of $\alpha$.

The Richarson-Lucy is an iterative algorithm and may require a large number of iterations to converge. As an alternative method for quickly finding the MAP estimate of the latent image with total-variation (TV) regularisation, Getreuer (2010) propose a fast algorithm using a "split Bregman" method.

Yuan et al. (2008) propose a method for suppressing ringing artefacts in non-blind deblurring.

### 2.2.2 Non-Blind Deblurring With Gaussian Noise

Many approaches to non-bind deblurring are based on an additive uniform Gaussian noise model, rather than a Poisson noise model, since it is very amenable to optimization, and is a realistic model in practice. For Gaussian noise, the likelihood is

$$p(\mathbf{g}|\mathbf{f}) = \frac{1}{\sqrt{2\pi}\sigma} \prod_i \exp\left(\frac{(g_i - g_i^*)^2}{2\sigma^2}\right). \tag{2.41}$$

The negative log-likelihood is then

$$\mathcal{L}_{\text{gauss}} = \log\sqrt{2\pi}\sigma + \sum_i \frac{(g_i - g_i^*)^2}{2\sigma^2}. \tag{2.42}$$

Since $\hat{\mathbf{g}}$ is linear in the sharp image $\mathbf{f}$, minimising $\mathcal{L}_{\text{gauss}}$ is a linear least-squares minimisation problem, which is convex and can be solved with almost any standard least-squares algorithm, e.g. conjugate gradient descent. The solution to this problem, which is the maximum-likelihood sharp image, can be written using the pseudo-inverse of $\mathbf{A}$ as

$$\hat{\mathbf{f}} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{g} \tag{2.43}$$

One particular case of interest under a Gaussian noise model is that of spatially-invariant blur, i.e. $\mathbf{g}^* = \mathbf{a} * \mathbf{f}$. In this case, using Parseval's theorem, $\mathcal{L}_{\text{gauss}}$ can be transformed into $N$ independent 1D quadratic minimisations in the frequency domain, allowing the maximum-likelihood solution to be obtained directly by pixel-wise division in the frequency domain (Gamelin, 2001):

$$\hat{\mathbf{f}} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{a})^* \circ \mathcal{F}(\mathbf{g})}{\mathcal{F}(\mathbf{a})^* \circ \mathcal{F}(\mathbf{a})}\right). \tag{2.44}$$

#### Incorporating Image Priors

The maximum-likelihood solution for Gaussian noise in non-blind deblurring is typically very ill-conditioned. This is most clearly seen in Equation (2.44), where the denominator

is the power spectrum of the blur kernel $\mathbf{a}$. If this power spectrum has some small or zero values, those frequencies will be unstable and may be amplified in the solution. If the observed image were noise-free and we had infinite computational precision available, this would not be a problem, however in practice this solution is ill-conditioned. To avoid this, most non-blind deblurring uses some form of regularisation to smooth the solution. The resulting regularised least-squares problem is

$$\min_{\mathbf{f}} \ \|\mathbf{A}\mathbf{f} - \mathbf{g}\|_2^2 + \alpha\rho_{\mathbf{f}}(\mathbf{f}), \tag{2.45}$$

where $\alpha$ is a weight that balances the data-fidelity term and the regularisation term.

The simplest form of regularisation to add to a Gaussian noise model is $\ell_2$ (Tykhonov) regularisation directly on the elements of $\mathbf{f}$, such that $\rho_{\mathbf{f}}(\mathbf{f}) = \|\mathbf{f}\|_2^2$. This produces a cost function which remains quadratic, and is thus straightforward to solve. Unfortunately, in the case of image restoration, such a regulariser does not correspond to a realistic prior for natural images.

Many authors have proposed algorithms to exploit the priors discussed in Section 2.1.5 to regularise least-squares solutions, which amounts to adding a smoothness penalty to the negative log likelihood. One of the most popular regularisers is derived from the sparse gradient prior of Equation (2.26), which penalises the derivatives of the deblurred image, which we write as the set of convolutions $\{\mathbf{d}^{(\mathbf{q})} * \mathbf{f}\}$. In this notation, the set of filters $\{\mathbf{d}^{(\mathbf{q})}\} = \{\mathbf{d}^x, \mathbf{d}^y\}$ are $1^{\text{st}}$ order horizontal and vertical finite-difference derivative filters. While it is possible to extend the regularisation to higher-order derivatives $\mathbf{d}^{xx}$, $\mathbf{d}^{xy}$, $\mathbf{d}^{yy}$, *etc.* this is not generally done in practice. This kind of regularisation can be written as $\rho_{\mathbf{f}}(\mathbf{f}) = \Phi\big(\{\mathbf{d}^{(\mathbf{q})} * \mathbf{f}\}\big)$, where the function $\Phi$ penalises the derivatives of the deblurred image. Using priors of the form in Equation (2.26), $\Phi$ has the form of a sum of 1D functions of each derivative at each pixel: $\Phi(\{\mathbf{v}^{(\mathbf{q})}\}) = \sum_q \sum_j \phi(v_j^{(q)})$. Levin et al. (2007) use $\phi(x) = |x|^p$ with $p = 0.8$, and use iteratively-reweighted least squares (IRLS) to minimise Equation (2.45) with this regulariser. This method involves solving a series of weighted least-squares problems, e.g. using conjugate gradient descent (CG).

Although the minimisation problem is non-convex for $p < 1$, the algorithm typically produces good results.

One problem with the IRLS method is that it typically takes a long time to run, sometimes tens of minutes for a megapixel image, as the conjugate gradient descent must be performed multiple times. To mitigate this, several authors have proposed to use so called "half-quadratic" optimisation schemes, whereby auxiliary variables are introduced to represent the derivatives of the deblurred image, and an alternating minimisation can be performed to obtain the deblurred result quickly (Geman and Yang, 1995). For the sparse gradient prior, a pair of auxiliary images $\{\mathbf{v}^{(\mathbf{q})}\} = \{\mathbf{v}^x, \mathbf{v}^y\}$ is introduced, corresponding to the derivatives of $\mathbf{f}$, and Equation (2.45) (where $\mathbf{f}$ is regularised directly) is replaced with the following problem, wherein $\mathbf{f}$ is indirectly regularised via the auxiliary variables:

$$\min_{\mathbf{f},\{\mathbf{v}^{(\mathbf{q})}\}} \|\mathbf{A}\mathbf{f} - \mathbf{g}\|_2^2 + \lambda \sum_q \left\|\mathbf{d}^{(q)} * \mathbf{f} - \mathbf{v}^{(q)}\right\|_2^2 + \alpha\Phi\big(\{\mathbf{v}^{(\mathbf{q})}\}\big). \tag{2.46}$$

For sufficiently large $\lambda$, this problem is approximately equal to the standard regularised least squares problem in Equation (2.45) with $\rho_{\mathbf{f}}(\mathbf{f}) = \Phi\big(\{\mathbf{d}^{(\mathbf{q})} * \mathbf{f}\}\big)$. The optimisation alternates between solving for $\mathbf{f}$ while holding $\{\mathbf{v}^{(\mathbf{q})}\}$ fixed, and vice versa, gradually increasing $\lambda$. For fixed $\{\mathbf{v}^{(\mathbf{q})}\}$, the problem is quadratic in $\mathbf{f}$ (hence the name "half-quadratic"), and for spatially-invariant blur can be solved directly in the frequency domain:

$$\hat{\mathbf{f}} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{a})^* \circ \mathcal{F}(\mathbf{g}) + \lambda \sum_q \mathcal{F}\left(\mathbf{d}^{(q)}\right)^* \circ \mathcal{F}\left(\mathbf{v}^{(q)}\right)}{\mathcal{F}(\mathbf{a})^* \circ \mathcal{F}(\mathbf{a}) + \lambda \sum_q \mathcal{F}\left(\mathbf{d}^{(q)}\right)^* \circ \mathcal{F}\left(\mathbf{d}^{(q)}\right)}\right). \tag{2.47}$$

Holding $\mathbf{f}$ fixed, each optimal $\mathbf{v}^{(q)}$ can be found by $2N$ independent 1D minimisations (1 for each pixel of each auxiliary image), of the form

$$\min_{v_j^{(q)}} \lambda\big((\mathbf{d}^{(q)} * \mathbf{f})_j - v_j^{(q)}\big)^2 + \alpha\phi\big(v_j^{(q)}\big). \tag{2.48}$$

For the hyper-Laplacian prior, where $\phi(x) = |x|^p$, Krishnan and Fergus (2009) demon-

(a) Original image (b) Blurred with shown kernel, and Gaussian noise with $\sigma = 2.5/256$ added (c) Levin et al. (2007) (d) Krishnan and Fergus (2009)

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

Figure 2.6. **Non-blind deblurring with Gaussian noise.** In this synthetic example, the original sharp image (a) is blurred and corrupted with Gaussian noise to produce the degraded image (b). The deblurred results are produced using the algorithms of (c) Levin et al. (2007), and (d) Krishnan and Fergus (2009).

strated analytic solutions to these minimisations for particular exponents $p = \frac{1}{2}$ and $p = \frac{2}{3}$, and showed how to use a lookup table to solve this problem quickly for arbitrary exponents. Shan et al. (2008) also use the half-quadratic method for their piecewise prior $\phi(x) = \min(k\,|x|\,, ax^2 + b)$. Using this choice for $\phi$, Equation (2.48) can be minimised separately for each piece of $\phi$, before taking the minimum amongst them.

In a related approach, Wang et al. (2008) use the half-quadratic method to apply total variation regularisation to the deblurring problem, where $\Phi(\{\mathbf{v}^{(\mathbf{q})}\})$ does not penalise each derivative independently but couples them such that $\Phi(\{\mathbf{v}^{(\mathbf{q})}\}) = \sum_j \sqrt{v_j^{x\,2} + v_j^{y\,2}}$. In this case, the optimal $\{\mathbf{v}^{(\mathbf{q})}\}$ are found as the solutions of $N$ independent 2D minimisations, and can still be found in closed-form. Recently, Zoran and Weiss (2011) have applied the half-quadratic method with a regulariser that models the statistics of small patches of the image rather than the derivatives. Several authors have investi-

gated variants of these split-variable approaches, and accelerated algorithms for solving them (Afonso et al., 2010; Osher et al., 2011).

### 2.2.3 Non-Blind Deblurring With Other Noise Models

As seen in previous sections, the data fidelity term used for non-blind deblurring can be derived as the negative log-likelihood for a particular noise distribution. However, when the appropriate noise model is not known *a priori*, or there are outliers to the real noise distribution in the blurry image, artefacts can appear in the deblurred image. In pursuit of deblurring algorithms that are robust against such problems other authors have proposed algorithms using robust data terms. These data fidelity terms reduce the risk of over-penalising a large data fidelity error. For examples, Xu and Jia (2010) use an $\ell_1$ data term in non-blind deblurring to be robust to non-Gaussian noise, with TV regularisation, while Welk (2010) proposes a robust version of the Richardson-Lucy algorithm.

## 2.3 Algorithms for Blind PSF Estimation

In general, the PSF for a blurry image may not be known. For some causes of blur, e.g. optical aberrations, the parameters may be estimated once per camera, however for others such as camera shake, the PSF is unique to each image and must be estimated on a per-image basis. "Blind" estimation of the PSF parameters directly from images has a long history, including methods based on entropy maximisation (Gull and Skilling, 1984), iterative projection on convex sets (Ayers and Dainty, 1988), and modification of the Richardson-Lucy algorithm (Fish et al., 1995). The particular task of deblurring "shaken" images has received considerable attention recently. Fergus et al. (2006) proposed the first robust algorithm for blind estimation of a camera shake blur kernel directly from a single blurry image, using an approximate marginalisation approach based on the method of Miskin and MacKay (2000) (also closely related to the more general blind deblurring algorithm of Molina et al. (2006)). Shan et al. (2008) and Cho and Lee

(2009) subsequently proposed maximum a posteriori (MAP) type algorithms capable of tackling this difficult problem in a much shorter amount of time. Several authors have also proposed similar MAP-type algorithms (Cai et al., 2009; Gupta et al., 2010; Krishnan et al., 2011; Xu and Jia, 2010), while most recently Levin et al. (2011) propose a more efficient formulation of the marginalisation approach.

While non-blind deblurring may be posed as a convex problem, or at least a well-conditioned one, the problem of finding the sharp image and PSF that best reconstruct the observed image is in general ill-posed, since we have more unknowns than equations. In fact, for a given blurry image $\mathbf{g}$, there are an infinite number of sharp image / PSF pairs that can reconstruct $\mathbf{g}$ equally well, as demonstrated in Figure 2.7. To obtain a useful solution, it is necessary to add some regularisation and/or constraints on both the sharp image and the PSF.

For now, we will assume that the blur is spatially-invariant, such that the PSF $\mathbf{A}$ is parameterised by the single convolution kernel $\mathbf{a}$.

### 2.3.1 Single-Image PSF Estimation

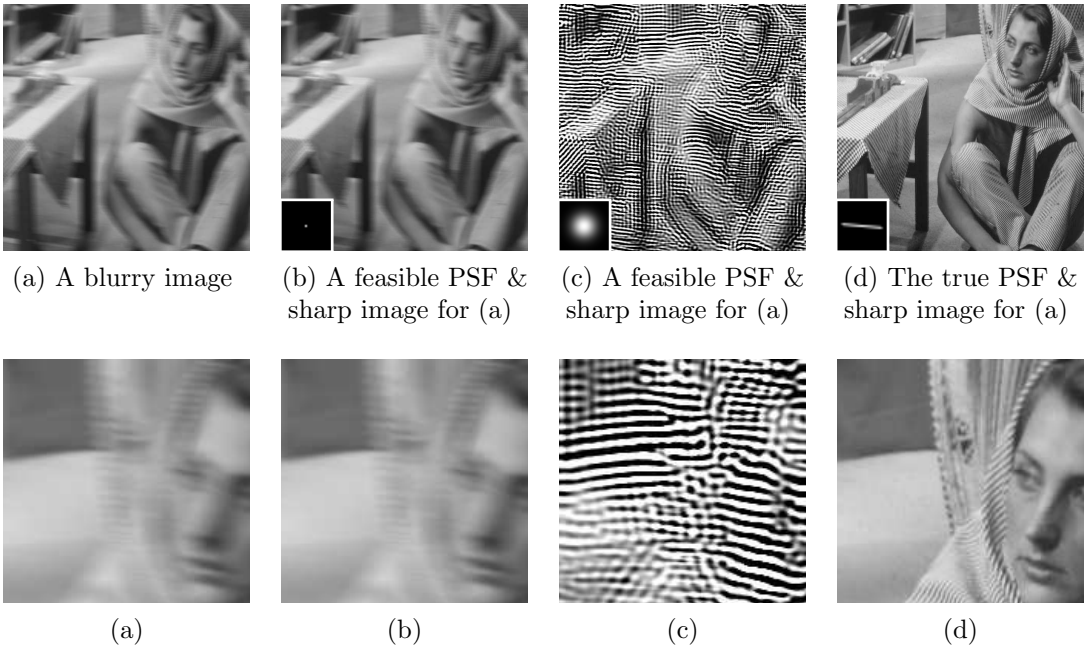
As discussed in Section 2.1.5, good priors for $\mathbf{f}$ and $\mathbf{a}$ are necessary for blind deblurring to be successful, so approaches to single-image PSF estimation generally take the posterior distribution for $\mathbf{f}$ and $\mathbf{a}$ as their starting point:

$$p(\mathbf{f}, \mathbf{a}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{a})p(\mathbf{f})p(\mathbf{a}), \tag{2.49}$$

where the likelihood is derived from an assumption of uniform Gaussian noise, as discussed in Section 2.1.4:

$$p(\mathbf{g}|\mathbf{f}, \mathbf{a}) \propto \prod_i \exp\left(-\frac{((\mathbf{a} * \mathbf{f})_i - g_i)^2}{2\sigma^2}\right), \tag{2.50}$$

where $\sigma$ is the standard deviation of the noise, and the priors $p(\mathbf{f})$ and $p(\mathbf{a})$ may be taken from those discussed in Section 2.1.5. Given this posterior, two options for estimating



(a) A blurry image (b) A feasible PSF & sharp image for (a) (c) A feasible PSF & sharp image for (a) (d) The true PSF & sharp image for (a)

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

Figure 2.7. **Infinite solutions to the blind deblurring problem.** Given a blurry image (a), there are an infinite number of possible PSFs and sharp images (b), (c), (d) that could have produced the blurry image. It is particularly challenging to find the correct solution amongst all these possible solutions, and strong priors on both the sharp image and PSF are required.

$\mathbf{a}$ are to (a) marginalise out $\mathbf{f}$ and find the expected value of $\mathbf{a}$ under $p(\mathbf{a}|\mathbf{g})$, or (b) attempt to find the values of $\mathbf{f}$ and $\mathbf{a}$ that jointly maximise $p(\mathbf{f}, \mathbf{a}|\mathbf{g})$. In this section we will review two popular approaches to blind PSF estimation, which follow these two broad directions. Figure 2.8 demonstrates an example result deblurring an image with these two algorithms.

### 2.3.2 The Marginalisation Approach

Fergus et al. (2006) proposed a marginalisation-based algorithm for blind deconvolution of a single image, blurred by camera shake. The algorithm is based on the variational inference approach of Miskin and MacKay (2000), originally designed for simultaneous deblurring and source separation of cartoon images.

The algorithm proposed by Miskin and MacKay (2000) attempts to approximate the

(a) Blurry image

(b) Marginalisation algorithm of Fergus et al. (2006)

(c) MAP algorithm of Cho and Lee (2009)

Figure 2.8. **An example of blind deblurring by marginalisation or MAP algorithms.** For the blurry image (a), the single-image deblurring algorithms of Fergus et al. (2006) and Cho and Lee (2009) estimate a convolution kernel, and subsequently deblur the image using a non-blind deconvolution algorithm, such as Richardson-Lucy. For this image the results produced by the marginalisation approach of Fergus et al. and the MAP approach of Cho and Lee produce similar results.

posterior distribution for both the kernel and the sharp image $p(\mathbf{f}, \mathbf{a}|\mathbf{g})$ by a simpler, factorised distribution. The factorised form of this distribution means that it is straightforward to marginalise out the sharp image $\mathbf{f}$ and find the expected value of $\mathbf{a}$. Fergus et al. (2006) successfully adapted this algorithm to the removal of camera shake blur from photographs by applying it in the gradient domain, within a multiscale framework. By working in the gradient domain, the latent variable $f_j$ for the intensity of a pixel is replaced by the $x$ and $y$ derivatives $f_j^x$ and $f_j^y$ at that pixel, which are treated as separate variables. They use the mixture of exponentials prior on the kernel described in Equation (2.27), and the mixture of Gaussians prior from Equation (2.25) on the sharp image. For simplicity, within the context of this algorithm, we use $\mathbf{f}^\partial$ to denote the concatenation of the derivative images $\mathbf{f}^x$ and $\mathbf{f}^y$, and use $j$ to index over this, i.e. $j \in \{1, \ldots, 2N\}$. Finally, to free the user from manually tuning the noise variance $\sigma^2$, the inverse variance $\beta_\sigma = \sigma^{-2}$ is also considered as a latent variable.

Following Miskin and MacKay (2000), we denote the set of latent variables $\mathbf{f}^\partial$, $\mathbf{a}$,

and $\beta_\sigma$ as the "ensemble" $\boldsymbol{\Theta}$. The aim is to find the factorised distribution

$$q(\boldsymbol{\Theta}) = q(\beta_\sigma) \prod_j q(f_j^\partial) \prod_k q(a_k) \tag{2.51}$$

that best approximates the true posterior $p(\boldsymbol{\Theta}|\mathbf{g})$. Note that in this context, the latent image pixels $f_j^\partial$, kernel elements $a_k$, and noise precision $\beta_\sigma$ are random variables. Since the distribution is factorised, the posterior for $\mathbf{a}$, $p(\mathbf{a}|\mathbf{g})$ is approximated simply by $q(\mathbf{a})$. Thus, having found the optimal $q(\boldsymbol{\Theta})$, the expected value of $\mathbf{a}$ under $q(\mathbf{a})$ is taken to be the optimal blur kernel, i.e., $\hat{\mathbf{a}} = \langle \mathbf{a} \rangle_{q(\mathbf{a})}$. Since $q(\mathbf{a})$ is itself factorised, i.e. $q(\mathbf{a}) = \prod_k q(a_k)$, this expectation is generally trivial to compute. Fergus et al. discard the latent image distribution $q(\mathbf{f})$ and estimate the sharp image using a non-blind deblurring algorithm (although as shown in Figure 4.2 (d), $\langle \mathbf{f} \rangle_{q(\mathbf{f})}$ does in fact provide a useful estimate of the sharp image).

To find the optimal $q(\boldsymbol{\Theta})$, Miskin and MacKay minimise the following functional (Miskin and MacKay, 2000, Eqn. (10)) over both the form and the parameters of $q(\boldsymbol{\Theta})$:

$$C_{\mathrm{KL}} = \int q(\boldsymbol{\Theta}) \left[ \ln \frac{q(\boldsymbol{\Theta})}{p(\boldsymbol{\Theta})} - \ln p(\mathbf{g}|\boldsymbol{\Theta}) \right] \mathrm{d}\boldsymbol{\Theta}. \tag{2.52}$$

Minimising this functional is equivalent to minimising the Kullback-Leibler divergence between the posterior and the approximating distribution (Bishop, 2006), and this is tackled by first using the calculus of variations to derive the optimal forms of $q(f_j^\partial)$, $q(a_k)$ and $q(\beta_\sigma)$, then iteratively optimising their parameters. We omit a complete explanation of the details of the optimisation, however the main feature of the iterative process is as follows: Each distribution, for example $q(a_k)$, is parameterised by two sufficient statistics, which we will denote $a_k^{(1)}$ and $a_k^{(2)}$, essentially a mean and variance for $a_k$. Miskin and MacKay (2000) first derive equations for the optimal values of every unknown (i.e. $a_k^{(1)}$, $a_k^{(2)}$, $f_k^{(1)}$, $f_k^{(2)}$, *etc.*) in terms of the other unknowns. By applying these optimality conditions as fixed-point update rules, Miskin and MacKay (2000) aim to decrease the Kullback-Leibler divergence between the true and approximate posterior.

### 2.3.3 The Maximum a Posteriori Approach

In contrast to the marginalisation approach of the previous section, several authors have proposed algorithms which estimate a convolution kernel $\mathbf{a}$ by maximising the posterior $p(\mathbf{f}, \mathbf{a}|\mathbf{g})$ (Cai et al., 2009; Cho and Lee, 2009; Shan et al., 2008; Xu and Jia, 2010). These maximum a posteriori (MAP) approaches vary in the priors they use, but generally follow the same structure, alternating between estimating $\mathbf{a}$ and $\mathbf{f}$ and proceeding from coarse to fine in a multi-scale framework.

#### Updating the blur kernel

Cho and Lee (2009) propose a method of kernel estimation which uses simple Gaussian priors on the kernel elements. Simply optimising $\mathbf{a}$ using this simple prior would almost certainly fail to produce a reasonable result, since it is not realistic. However, the introduction of non-linear filtering and thresholding steps into the process encourages the algorithm to find a latent image with sparse gradients and a blur kernel with sparse non-zero elements, such as discussed in Section 2.1.5.

To update the kernel, Cho and Lee (2009) first filter the current estimate $\hat{\mathbf{f}}$ of the sharp image to predict step edges. A bilateral filter (Tomasi and Manduchi, 1998) is applied to denoise the image, followed by a shock filter (Osher and Rudin, 1990) to enhance sharp edges, producing an image $\hat{\mathbf{f}}'$. The derivatives $\mathbf{d}^x * \hat{\mathbf{f}}'$ and $\mathbf{d}^y * \hat{\mathbf{f}}'$ of this filtered image are then computed. Finally, these derivatives are thresholded to produce sparse derivatives maps $\mathbf{p}^x$ and $\mathbf{p}^y$. The threshold is chosen so as to retain only a small number of non-zero entries, while ensuring that sufficient non-zeros are retained from all edge orientations.

Having predicted and retained salient sharp edges from the latent image, the set of sparse derivatives maps is augmented with higher order derivatives to form the set $\{\mathbf{p}^{(\mathbf{q})}\} = \{\mathbf{p}^x, \mathbf{p}^y, \mathbf{d}^x * \mathbf{p}^x, \frac{1}{2}(\mathbf{d}^x * \mathbf{p}^y + \mathbf{d}^y * \mathbf{p}^x), \mathbf{d}^y * \mathbf{p}^y\}$, and used to estimate the blur kernel by solving

$$\min_{\mathbf{a}} \sum_q \omega_q \|\mathbf{a} * \mathbf{p}^{(q)} - \mathbf{d}^{(q)} * \mathbf{g}\|_2^2 + \beta \|\mathbf{a}\|_2^2, \tag{2.53}$$

where $\mathbf{d}^{(q)} \in \{\mathbf{d}^x, \mathbf{d}^y, \mathbf{d}^{xx}, \mathbf{d}^{xy}, \mathbf{d}^{yy}\}$, $\beta$ is a weight for the Tykhonov regularisation, and the weights $\omega_q$ weight each partial derivative. This is simply a linear least squares problem, which can be solved directly in the frequency domain, analogously to Equation (2.47):

$$\hat{\mathbf{a}} = \mathcal{F}^{-1}\left(\frac{\sum_q w_q \mathcal{F}\left(\mathbf{p}^{(q)}\right)^* \circ \mathcal{F}\left(\mathbf{g}^{(q)}\right)}{\sum_q w_q \mathcal{F}\left(\mathbf{p}^{(q)}\right)^* \circ \mathcal{F}\left(\mathbf{p}^{(q)}\right) + \beta}\right). \tag{2.54}$$

Having found the kernel $\hat{\mathbf{a}}$ which minimises Equation (2.53), the values are thresholded, such that any element whose value is smaller than $\frac{1}{20}$ the largest element's value is set to zero. This encourages sparsity in the kernel, and ensures that all the elements are positive.

Xu and Jia (2010) take essentially the same approach to kernel estimation as Cho and Lee (2009), with the addition of an extra filtering step aimed at suppressing unhelpful non-zeros in the sparse derivative maps $\{\mathbf{p}^x, \mathbf{p}^y\}$. Xu and Jia show that non-zeros in $\mathbf{p}^x$ and $\mathbf{p}^y$ that arise from narrow peaks in the image (as opposed to step edges) actually reduce the quality of the estimated kernel, and as such should be removed before solving Equation (2.53).

In a contrasting approach, Shan et al. (2008) do not perform any image filtering, and replace the Tykhonov regularisation in Equation (2.53) with $\ell_1$ regularisation, solving

$$\min_{\mathbf{a}} \sum_q \omega_q \|\mathbf{a} * \mathbf{d}^{(q)} * \hat{\mathbf{f}} - \mathbf{d}^{(q)} * \mathbf{g}\|_2^2 + \beta \|\mathbf{a}\|_1 . \tag{2.55}$$

Here, instead of the Tykhonov regularisation used by Cho and Lee, $\ell_1$ regularisation is used, corresponding to an exponential prior and encouraging sparsity directly. Using this regulariser however, precludes direct minimisation in the frequency domain, and Shan et al. use an interior point method (Kim et al., 2007) to solve Equation (2.55).

### Updating the latent image

In order to update the latent image, the current estimate of the blur kernel $\hat{\mathbf{a}}$ is used to deconvolve the blurry image and obtain an improved estimate of the sharp image. Cho and Lee (2009) do this by solving

$$\min_{\mathbf{f}} \sum_{q} \omega_q \|\hat{\mathbf{a}} * \mathbf{d}^{(q)} * \mathbf{f} - \mathbf{d}^{(q)} * \mathbf{g}\|_2^2 + \alpha \|\mathbf{d}^x * \mathbf{f}\|_2^2 + \alpha \|\mathbf{d}^y * \mathbf{f}\|_2^2, \tag{2.56}$$

where $\alpha$ is a regularisation weight, and now, the partial derivatives include the zeroth order: $\mathbf{d}^{(q)} \in \{1, \mathbf{d}^x, \mathbf{d}^y, \mathbf{d}^{xx}, \mathbf{d}^{xy}, \mathbf{d}^{yy}\}$. Note that in this problem, the data term involves higher-order derivatives of the image, although the regularisation does not. This is solved in closed-form by pixel-wise division in the frequency domain. Shan et al. (2008) update the latent image using their non-blind deblurring algorithm mentioned in Section 2.2.2, which uses a sparsity prior on the image gradients. Xu and Jia (2010) choose to add $\ell_2$ regularisation that encourages the latent image gradients to be close to the sparse derivative maps $\{\mathbf{p}^{(\mathbf{q})}\}$ predicted from the previous iteration:

$$\min_{\mathbf{f}} \sum_{q} \omega_q \|\hat{\mathbf{a}} * \mathbf{d}^{(q)} * \mathbf{f} - \mathbf{d}^{(q)} * \mathbf{g}\|_2^2 + \alpha \|\mathbf{d}^x * \mathbf{f} - \mathbf{p}^x\|_2^2 + \alpha \|\mathbf{d}^y * \mathbf{f} - \mathbf{p}^y\|_2^2. \tag{2.57}$$

This reduces the problem of over-smoothing caused by the $\ell_2$ regularisation, since sharp edges are encouraged to appear where they exist in $\{\mathbf{p}^{(\mathbf{q})}\}$, while still permitting closed-form minimisation in the frequency domain.

As shown by Shan et al. (2008), the appropriate values for the relative weights of the different orders of partial derivatives are $\omega_q = \frac{1}{2^{\kappa(q)}}\omega_0$, where $\kappa(q)$ denotes the order of derivative $\mathbf{d}^q$.

These steps are applied iteratively, working from coarse to fine in a multi-scale framework. The iterative process generally converges quickly at each scale, and 5-7 iterations are typically sufficient.

### Why does the MAP approach work?

Clearly the effectiveness of the MAP approach is not dependent on the exact priors used to regularise the least-squares estimation steps, since Cho and Lee (2009) use very weak Gaussian priors. What appears to be the common key features of these algorithms are that sharp edges are encouraged to appear in the latent image, and that the kernel is encouraged to be sparse. While Shan et al. (2008) achieve this using sparse priors combined with an iterative edge / non-edge segmentation process, Cho and Lee (2009) avoid the costly optimisation problems and achieve the same aim instead with simple non-linear filtering and thresholding steps, encouraging sharp edges only at predicted step edge locations. Recent authors have argued that a straight-forward MAP approach to blind deblurring using a sparse-gradients image prior should fail in general (Fergus et al., 2006; Levin et al., 2009). The sparse gradient prior in fact favours blurry images over sharp ones in most cases, and is thus unsuited for encouraging sharp edges to appear in the deblurred image. The sharp edge prediction / segmentation components of the algorithms proposed by Joshi et al. (2008), Cho and Lee (2009) and Shan et al. (2008) avoid this problem by limiting the scope of the sparse gradient action to particular regions of the image.

Although this explanation of why the MAP approach works is somewhat heuristic, Weiss (2011) recently highlighted similarities with well-established algorithms in communications for reconstruction of signals transmitted via imperfect transmission channels (Godard, 1980; Shalvi and Weinstein, 1990), which are based on the fact that degrading a signal always moves its marginal statistics closer to a Gaussian distribution. Assumimg that the samples in the latent signal are independent and identically distributed, the reconstruction problem can be posed as a global optimisation, where the objective is to maximise the non-Gaussianity of the latent signal. Although the independence assumption is not generally satisfied in natural images, the MAP-type blind deblurring algorithms discussed here (particularly that of Krishnan et al. (2011)) can be thought of as maximising the non-Gaussianity of the latent image in this way, offering

a possible route to theoretically explaining their success.

Recently, Levin et al. (2009) have analysed the differences between the marginalisation and MAP approaches to blind deblurring, and conclude that in order to estimate the PSF, it is advantageous to marginalise over the sharp image $\mathbf{f}$, while maximising over the kernel $\mathbf{a}$. In a subsequent work, Levin et al. (2011) propose an algorithm which does this explicitly, maximising the posterior for the kernel alone $p(\mathbf{a}|\mathbf{g})$, which is obtained by marginalising over the unknown sharp image $\mathbf{f}$, i.e. $p(\mathbf{a}|\mathbf{g}) = \int p(\mathbf{a}, \mathbf{f}|\mathbf{g})\, \mathrm{d}\mathbf{f}$. To do this, Levin et al. (2011) adopt a strategy that is similar in spirit to the marginalisation algorithm of Fergus et al. (2006), described in the previous section.

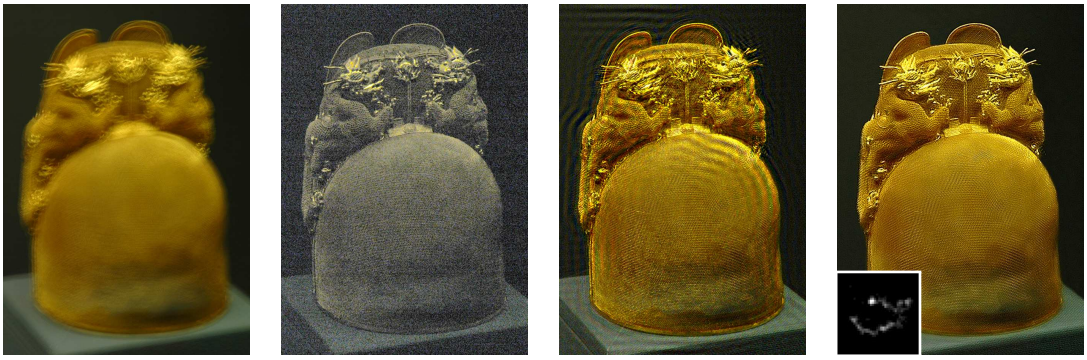
### 2.3.4 Deblurring With Noisy / Blurry Image Pairs

Yuan et al. (2007b) proposed a method for blind deblurring where, in addition to a blurry image $\mathbf{g}$, a sharp but noisy image $\mathbf{f}_N$ of the same scene is also available. The motivation for this is that in low light, blurry images occur at long shutter speeds, however it is often also possible to use a short exposure at a high ISO setting to obtain a sharp but noisy image of the same scene. While the noisy image may be degraded too badly to allow the direct recovery of a good sharp image by denoising, it can initially be used as a proxy for the sharp image, allowing us to estimate the PSF. Following this, with the PSF fixed, the noisy image can also be used to improve the final non-blind deconvolution step, using a modified version of the Richardson-Lucy algorithm, which uses $\mathbf{f}_N$ to suppress ringing artefacts in the result.

In order to estimate the blur kernel, Yuan et al. (2007b) solve

$$\min_{\mathbf{a}} \; \|\mathbf{a} * \mathbf{f}_N - \mathbf{g}\|_2^2 + \beta \|\mathbf{a}\|_2^2 . \tag{2.58}$$

Having estimated the PSF, Yuan et al. propose several modifications to the Richardson-Lucy (RL) algorithm to perform the non-blind deblurring, which take advantage of the fact that it is possible to recover much of the low-frequency content of $\mathbf{f}$ from a denoised version of $\mathbf{f}_N$. Images deblurred with the standard RL algorithm often exhibit "ringing"



(a) Blurry image (b) Noisy image (c) Deblurred using Richardson-Lucy algorithm (d) Estimated blur kernel and Yuan et al.'s deblurred result

Figure 2.9. **An example result of the algorithm of Yuan et al. (2007b).** Using a blurry image (a) and a noisy image (b) as inputs, Yuan et al. (2007b) estimate a spatially-invariant blur kernel (shown in (d)). Performing non-blind deconvolution using the standard Richardson-Lucy algorithm produces a deblurred image (c) containing visible ringing artefacts. Yuan et al. (2007b) propose several modifications to the Richardson-Lucy algorithm, using the noisy image as a guide to produce a deblurred image (d) free from ringing.

artefacts – low-frequency ripples spreading across the image, such as in Figure 2.9 (c) – but using the denoised image it is possible to disambiguate the true low frequencies from these artefacts, and largely remove them from the result. Doing this significantly improves the deblurred results compared to the standard RL algorithm. We refer the reader to (Yuan et al., 2007b) for full details of the augmented RL algorithm. Figure 2.9 shows an example result deblurred using this approach.

## 2.4 Modelling Spatially-Variant Blur

Until this point, we have only considered spatially-invariant blur, where the image formation model in Equation (3.13) can be written as a 2D convolution. In most of the work mentioned above, and in general through the deblurring literature, convolution is used as a simple model for image blur. In general however, the blur in an image may be spatially-variant, and the matrix $\mathbf{A}$ will contain a different PSF for each pixel of the

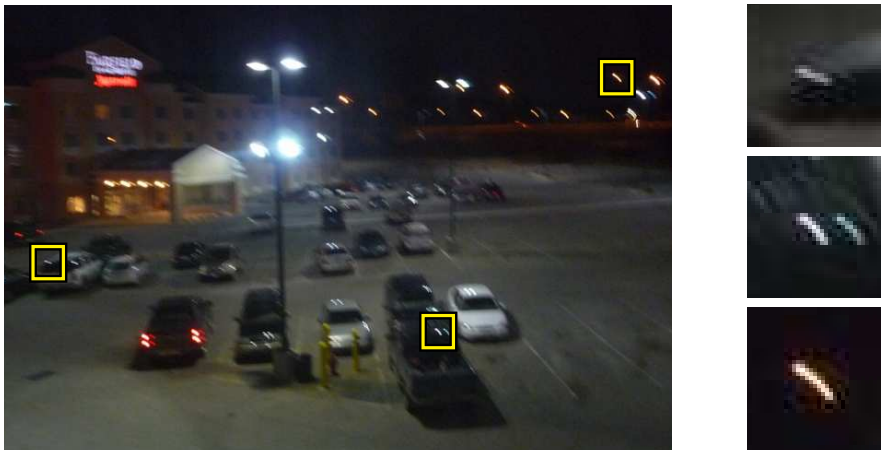
image. Figure 2.10 shows an example of spatially-variant blur due to camera shake.

For real images of more than a few hundred pixels across, it is not practical to compute and store every element of the large matrix $\mathbf{A}$ explicitly, and instead a more compact parameterisation of the matrix is required. For spatially-invariant blur, such a parameterisation comes in the form of a convolution kernel. For spatially-variant blur, recent work has investigated analogous parameterisations of the matrix $\mathbf{A}$ (a notable exception is the work of Seitz and Baker (2009), who attempt to estimate the elements of $\mathbf{A}$ itself, with no underlying parameterisation). The first distinction to draw is between spatially-variant blur caused by moving objects in the scene, and that caused by the image capture process. When the blur is caused by moving objects, the spatial-variance is largely due to the fact that different objects move differently, and the blur changes abruptly at object boundaries. Within each object, simple linear blurs are often assumed (Chakrabarti et al., 2010; Cho et al., 2007; Levin, 2006), such that the matrix $\mathbf{A}$ is parameterised by a line segment representing the blur, and a segmentation delineating the object.

When the blur is caused by the image capture process, rather than scene motion, the blur is more likely to vary smoothly across the image. Previous approaches to modelling this kind of blur can largely be divided into two categories: global models, which use a single set of parameters to describe the blur at all points in the image, and local models, which divide the image into a set of regions and use a separate set of blur parameters for each region. Although in this thesis we are concerned with blur caused by camera motion, we note that models for spatially-variant blur caused by optical aberrations of the camera have also recently been proposed by (Kee et al., 2011; Schuler et al., 2011).

### 2.4.1 Global Models

On of the contributions of this thesis (see Chapter 3) is to propose a model for spatailly-varying camera shake blur. Concurrently with this work, several models have been proposed to parameterise spatially-variant blur caused by unconstrained camera motion (Gupta et al., 2010; Joshi et al., 2010; Tai et al., 2011). These approaches are



(a) An image blurred by camera shake

(b) Details from the blurry image

Figure 2.10. **An example of spatially-variant blur due to camera shake.** This real hand-held image exhibits a blur due to camera shake which is spatially-variant. This can be seen in the close-ups (b).

derived from the relationship between motion of the camera and transformations of the sharp image. The general form of these models is

$$\mathbf{g}^* = \sum_{k=1}^{K} w_k \mathbf{T}^{(k)} \mathbf{f}, \tag{2.59}$$

where each matrix $\mathbf{T}^{(k)}$ applies some transformation to the sharp image, which then contributes to the blurry image with a weight $w_k$. In this case, the weights (which we concatenate into the $K$ element vector $\mathbf{w}$) provide the compact parameterisation of the PSF.

The models proposed by various authors vary in their choice of the set of image transformations $\{\mathbf{T}^{(\mathbf{k})}\}$. Joshi et al. (2010) model the image motion using homographies which correspond to a planar scene. These homographies are parameterised by the pose of the camera, which is recorded using inertial measurement sensors attached to the camera. Tai et al. (2011) also model the image motion using homographies, although without constraining them to correspond to scene planes or camera rotations. In this

case, the homographies are estimated by applying optical flow to the output of a second, high-speed camera attached to the main camera. Both Joshi et al. (2010) and Tai et al. (2011) apply their models only to non-blind deblurring, where the parameters $\mathbf{w}$ are estimated using additional hardware attached to the camera. Gupta et al. (2010) propose to use the set of image-plane translations and rotations, and estimate the weights $\mathbf{w}$ directly from the blurry image, i.e. in a blind deblurring setting.

While the models described above are largely agnostic of the scene content, Šorel and Flusser (2008) tackled simultaneous blind deblurring and depth map estimation from multiple blurry images. Global models of spatially-variant blur have been also applied under various constrained motion models (Klein and Drummond, 2005; Sawchuk, 1974; Shan et al., 2007; Tai et al., 2010b), and although global descriptors are used to parameterise these continuously-varying blurs, the constraints are often restrictive.

In Chapter 3 we propose a global model for camera shake blur with the form in Equation (2.59). Our method is distinguished from previous approaches by its foundation in the geometry of camera rotations, which provides a minimal parameterisation of the blur while remaining accurate at all possible camera settings.

### 2.4.2 Local Models

An alternative method for parameterising a spatially-variant blur matrix $\mathbf{A}$ is to assume that instead of each pixel having a unique PSF, nearby pixels can (approximately) share a single PSF. In this way, a spatially-variant blur can be parameterised using only a small number of convolution kernels, each assigned to a different region of the image. In contrast to "global" models such as those mentioned above, which use a complex motion model and a single set of parameters to describe the blur at every point in the image, these approaches are local, using a simple motion model but separate sets of parameters for each region in the image. Much previous work has relied on this idea, modelling the blur as being locally uniform (Hirsch et al., 2010; Nagy and O'Leary, 1998; Tai et al., 2010a; Vio et al., 2005).

The two general forms for these local models are

$$\mathbf{g}^* = \sum_{r=1}^{P} \mathbf{m}^{(r)} \circ (\mathbf{a}^{(r)} * \mathbf{f}) \tag{2.60}$$

$$\mathbf{g}^* = \sum_{r=1}^{P} \mathbf{a}^{(r)} * (\mathbf{m}^{(r)} \circ \mathbf{f}), \tag{2.61}$$

where the blur within each region $r$ is modelled using the convolution filter $\mathbf{a}^{(r)}$ and a windowing function $\mathbf{m}^{(r)}$. The windowing functions $\mathbf{m}^{(r)}$ are usually fixed manually in advance, and the set of kernels $\{\mathbf{a}^{(\mathbf{r})}\}$ parameterises the PSF. In the first form, $\mathbf{m}^{(r)}$ controls the region of influence of the filter $\mathbf{a}^{(r)}$ (i.e. which parts of the blurry image are affected by the filter $\mathbf{a}^{(r)}$), while in the second form, $\mathbf{m}^{(r)}$ controls its region of dependence (i.e. which parts of the sharp image are blurred with the filter $\mathbf{a}^{(r)}$). This locally-uniform approach has proven to be powerful enough to model complex spatially-variant blur with reasonable accuracy, while enjoying computational advantages over global models, since the FFT can be used to compute the convolutions efficiently. In both cases, the windows $\mathbf{m}^{(r)}$ are typically chosen to sum to unity everywhere, i.e. $\sum_r \mathbf{m}^{(r)} = \mathbf{1}$, to ensure that the blur is conservative.

Nagy and O'Leary (1998) consider local blur models of the form in Equation (2.60), using either rectangular or triangular regions, for both the case where the regions are disjoint, and where they overlap. Tai et al. (2010a) also adopt a spatially-variant blur model of the form in Equation (2.60). However, instead of fixing the regions of influence $\mathbf{m}^{(r)}$ and estimating the local kernels $\mathbf{a}^{(r)}$ which best fit the observed blur, they fix the kernels $\mathbf{a}^{(r)}$ using a pre-determined basis, and estimate the regions of influence $\mathbf{m}^{(r)}$ which best reconstruct the observed blur.

Hirsch et al. (2010) propose a model of the form in Equation (2.61), using a coarse grid of overlapping rectangular regions to cover the image. They propose to use the Bartlett-Hann window as $\mathbf{m}^{(r)}$ for each region $r$. This gives a smooth fall-off from one region to the next, allowing the blur to vary smoothly across the image, instead of changing abruptly at region boundaries. The fact that each rectangular region is small

and has a spatially-invariant blur allows the forward model to be computed using $P$ small convolutions. To take advantage of the small support of each region, it is possible to define a matrix $\mathbf{C}^{(r)}$ to crop out the $r^{\text{th}}$ rectangular region, and re-write the forward model in (2.61) as

$$\mathbf{g}^* = \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^\top \left( \mathbf{a}^{(r)} * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}) \right). \tag{2.62}$$

Here, all regions use the same Bartlett-Hann windowing function $\mathbf{m}$, and we note that the matrix ${\mathbf{C}^{(r)}}^\top$ re-inserts the $r^{\text{th}}$ region at its correct location. This model can be computed efficiently in the frequency domain:

$$\mathbf{g}^* = \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^\top \mathcal{F}^{-1} \left( \mathcal{F}\big(\mathbf{a}^{(r)}\big) \circ \mathcal{F}\big(\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}\big) \right), \tag{2.63}$$

where $\mathcal{F}(\,\cdot\,)$ takes the 2D discrete Fourier transform (computed using the fast Fourier transform), and $\mathcal{F}^{-1}(\,\cdot\,)$ the inverse Fourier transform.

This region-based approach to convolution was originally proposed by Stockham, Jr. (1966), as a means of computing *spatially-invariant* convolutions and correlations more efficiently. When the filters are all identical, i.e. $\mathbf{a}^{(r)} = \mathbf{a}$ for all $r$ and have small support, then the following equality holds exactly:

$$\sum_{r=1}^{P} {\mathbf{C}^{(r)}}^\top \left( \mathbf{a} * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}) \right) = \mathbf{a} * \mathbf{f}. \tag{2.64}$$

The two main advantages of computing a spatially-invariant convolution in sections is that the entire input signal $\mathbf{f}$ does not all need to be stored in memory at once, and that it is cheaper to compute several small FFTs than to compute one large FFT. The FFT has computational complexity $\mathcal{O}\big(N \log N\big)$ for a signal of length $N$, and thus computing two FFTs of size $N/2$ has complexity $\mathcal{O}\big(2N/2 \log N/2\big) = \mathcal{O}\big(N \log N/2\big)$. This advantage in speed also holds true for the spatially-variant case, allowing spatially-variant blur to be computed in a similar amount of time as spatially-invariant blur.

Although in this thesis we are concerned only with blur caused by camera motion, we note that such local models are also applicable to blur caused by moving objects in the scene, where the object (which occupies one region of the image) is often modelled using a single convolution filter, and the background (which occupies the rest of the image) is assumed to be static, or modelled using a different convolution folter.

## 2.5 Inpainting

The task of replacing missing or undesirable pixels in an image is known as "inpainting" (Bertalmío et al., 2000), or sometimes "image completion" or "scene completion". When we are concerned with replacing missing or deleted pixels in an image, the restoration problem is quite different from deblurring, where all of the latent pixels are, in some sense, observed. Here, the missing latent pixels are completely decoupled from the observed image, and so cannot be recovered at all using maximum-likelihood techniques which estimate the latent image from the data alone. Instead, we require informative priors on the latent image.

In the problem of inpainting, we are generally not concerned with blur, so the PSF is simpyl the identity, i.e. $\mathbf{A} = \mathbf{I}$. The occlusion / corruption that we wish to replace is modelled as noise with a uniform distribution, and the other observed pixels are assumed to be noiseless, i.e. exactly equal to their counterparts in the latent image. We refer to the unknown / corrupted region to be filled as the *target region*, and denote this by the set $\Psi$, such that $g_i \sim \text{Unif}(0, 256)$ for $i \in \Psi$. The known / good region is denoted by the set $\mathcal{M}$, and with no noise, we have $\mathbf{f}_\mathcal{M} = \mathbf{g}_\mathcal{M}$, as shown in Figure 2.11. The pixels in $\Psi$ which lie along the boundary with $\mathcal{M}$ are denoted by $\delta\Psi$. In a probabilistic formulation, the likelihood here provides no information about the unknown pixels $\mathbf{f}_\Psi$, as they are decoupled from the known pixels in $\mathcal{M}$. Thus we require some prior information about the latent image before we can begin to replace the missing pixels.

Previous work on the problem of inpainting is diverse, varying significantly according to the size of the region to be filled, the content of the image surrounding it, and the

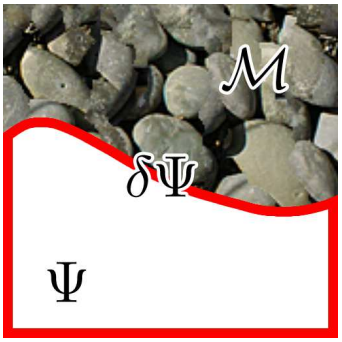



Figure 2.11. **The inpainting problem.** This image shows the notation for the inpainting problem. The known region of the images is denoted by $\mathcal{M}$, while the unknown target region is denoted by $\Psi$. The pixels in $\Psi$ that lie along the boundary with $\mathcal{M}$ are denote by $\delta\Psi$.

amount of user interaction required.

Early work on the problem relied on the notion that the latent image should belong to certain convex sets. If projection operators $\mathcal{P}$ for these sets are available, then identities of the form $\mathbf{f} = \mathcal{P}\mathbf{f}$ can be used to derive inpainting algorithms which find a solution which lies in the intersection of all the sets. Ferreira and Pinho (1994) used this approach to solve the inpainting problem as interpolation of a band-limited signal, and proposed an algorithm of this type capable of filling small holes, while (Hirani and Totsuka, 1996) proposed a projection-based method capable of filling a textured region containing straight-line edge structures, by constraining the power spectrum of the region to match a manually-selected reference region.

The problem of filling small regions has generally been approached using local models of images, i.e. models which account for the behaviour of images in areas a few pixels across. Bertalmío et al. (2000) approached the problem using the equations of fluid dynamics, following the intuition that lines of constant colour (isophotes) should continue into the region to be filled and connect smoothly within it. Their algorithm iteratively propagates colour into the target region from around the boundary, leading to a smooth continuation of colours.Recent work has focussed on the use of learnt priors on the statistics of local patches (Mairal et al., 2008; Roth and Black, 2005; Zoran and Weiss, 2011). The strength of these approaches is that the local structure in natural images

is reasonably predictable, such that a pixels' value is closely related to its neighbours' values. The general form for these approaches is essentially a MAP problem with no data fidelity term (since the observed pixels contain no information about the missing pixels):

$$\min_{\mathbf{f}} \; -\log p(\mathbf{f}) \tag{2.65}$$

In this formulation the unknown pixels are decoupled from the observed pixels in the PSF. The priors provide this coupling, and essentially provide a way of producing statistically probable (and thus hopefully convincing) hallucinations for the filled pixels. By looking at the pixels around the missing region, the priors are able to "suggest" likely content in the region.

Except in some cases, these local approaches fail when the region is larger than a few pixels across. When the pixels in the middle of the region are remote from those at the edge, it is no longer possible to infer their likely content from the surrounding pixels. One exception is when filling a region with homogeneous texture, for which a local texture synthesis algorithm can be applied (Efros and Leung, 1999).

When filling a region with large-scale structures however, purely-local methods fail to capture both the structure and the fine-scale texture. To cope with this, subsequent work has considered combining texture synthesis methods with higher-level structural information about strong edges inside the region. This structural information is either inferred from the surrounding image (Bertalmío et al., 2003; Criminisi et al., 2003; Jia and Tang, 2003), or manually provided by the user (Sun et al., 2005). In all these works, the texture synthesis algorithm replaces individual pixels or patches one at a time, selecting a pixel (or patch) from $\delta\Psi$ using some criteria, then replacing it. By combining texture synthesis with information about the large-scale content of the region, more complex image content can be produced. However, these algorithms are still constrained to draw their information from the image itself, or from simple manually-input guides. Thus there is a limit to the kind of regions they can reconstruct.

While methods of texture-synthesis and edge-continuation aim to hallucinate realistic low-level details of an image and simple structures, a separate line of work has investigated synthesising images from multiple input images of the same scene. (Agarwala et al., 2004; Rasmussen and Korah, 2005; Wilczkowiak et al., 2005) propose methods for combining multiple images, captured in quick succession by the same camera. An integral component of these methods is "Poisson blending" (Pérez et al., 2003), a method for seamlessly stitching together regions from different images without introducing visible boundaries.

On the other hand, there has been significant interest recently in organising, visualising, and searching the growing number of images available in online photo collections (Chum et al., 2007; Jégou et al., 2008; Philbin et al., 2007; Snavely et al., 2006). For popular locations, several authors have demonstrated that there exist sufficient images to perform accurate 3D reconstructions (Furukawa et al., 2010; Goesele et al., 2007) and 3D scene visualisations (Shahrokni et al., 2008). Hays and Efros (2007) first proposed to exploit this abundance of information to perform inpainting, using a large photo collection retrieved from the Internet to remove large objects from photographs. Their algorithm finds images with similar semantic content to that being restored, and directly copies the contents into the target image using Poisson blending. While the results are often convincing, they do not necessarily correspond to the true underlying scene, since the source images may depict an entirely different part of the world. Garg et al. (2009) also demonstrated the use of online photo collections for inpainting, by performing a dense 3D reconstruction of the scene and learning an appearance model for its surface.

These approaches to inpainting can be distinguished from the approaches discussed above by their use of direct exemplars, as opposed to statistical priors on the local behaviour of natural images. This idea goes back at least to Hirani and Totsuka (1996), who require the user to select a "reference" patch, which is then used as a constraint to ensure that the filled region of the latent image to has a similar spectrum to the reference region.

In Chapter 7 we propose a system which builds on this work, by retrieving reference

images of the same scene as the image to be restored. We replace the occluded region using these exemplars, without an explicit 3D reconstruction of the scene or any knowledge of the camera positions. As opposed to filling the missing region with content synthesised from statistical information or semantically similar scenes, we use these reference images to produce a realistic output that corresponds to the true occluded scene.

# Chapter 3

# Modelling Spatially-Variant Camera Shake Blur

## 3.1 Introduction

Everybody is familiar with camera shake, since the resulting blur spoils many photos taken in low-light conditions. Often in such conditions, the shutter speed selected by the camera is slow, and unless the photographer is able to hold the camera perfectly still, the camera's motion causes the photograph to be blurry. While significant progress has been made recently towards removing such blur from images, most approaches to this problem assume that the blurred image can be modelled as the 2D convolution of a sharp image with a *spatially-invariant* filter (Chan and Wong, 1998; Fergus et al., 2006; Shan et al., 2007; Yuan et al., 2007b), as discussed in Section 2.3. However, real camera shake does not, in general, cause such spatially-invariant blur (Levin et al., 2009). Figure 3.1 shows an example of a typical spatially-variant blur caused by camera shake.

In this chapter we propose a geometrically motivated model of *spatially-variant* image blur due to camera shake. We develop a global descriptor for such blur, analogous to a convolution kernel, and demonstrate its ability to model a more general class of blurs than previous approaches, including spatially-invariant blur as a special case. In this

chapter, we limit our scope to describing and discussing the model itself. We defer the application of the model to the following and later chapters.

We begin by considering the relative blurring effect of different camera motions, and deriving a geometric model for camera shake in Section 3.2. In Section 3.3 we develop this into a practical model for deblurring real images, and provide a comparison to other related models proposed recently. We conclude in Section 3.4.

## 3.2 A Geometric Model for Camera Shake

When we say that a photograph is blurry due to camera shake, it is generally understood that the camera moved during the exposure, and that this motion is to blame for the blur. More precisely, while the shutter is open the camera passes through a sequence of different poses, each of which gives a different view of the scene. The sensor accumulates all of these views, summing them up to form the recorded image. Intuitively, summing up many different views of a scene will result in a blurry image, and the more the views differ, the larger the blur will be. However, a camera may move in several different ways, and it is not necessarily obvious which kinds of motion cause large changes in the view (and hence a large blur), and which cause relatively small changes. Furthermore, even if the camera's motion is fully known for a given photograph, we need a model to translate this physical 3D motion into image-domain motion before we can begin to deblur the photograph.

In this section we address the challenges mentioned above. We examine the causes of camera shake blur, and derive a geometric model for the process, which fits into the linear degradation model presented in the previous chapter. In this work we limit our scope to photographs of static scenes, i.e. the blur is solely due to the motion of the camera.

### 3.2.1 Components of Camera Motion

To begin our discussion of camera motion, we note that the pose of a camera incorporates two components: *position* and *orientation*. Intuitively, the position tells us where the camera is, while the orientation tells us which way it is pointing. Both of these components may vary while the camera's shutter is open. In this section, we discuss the contribution of each component to the image blur, and conclude that in most cases of camera shake, the changes in orientation (rotation) of the camera during exposure have a significantly larger blurring effect than the changes in position (translation).

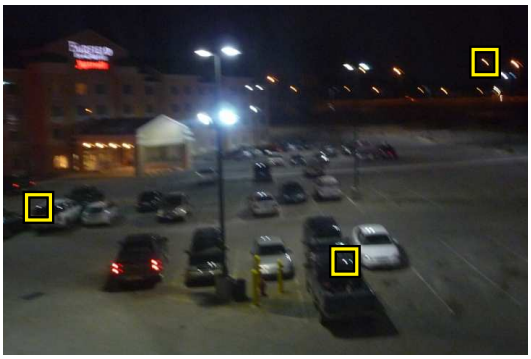
Consider the simplified case shown in Figure 3.2 of a scene point $P$, at a distance $D$ from the camera, being imaged at the centre of the camera's retina / sensor. During the exposure the image of the point is blurred through a distance $\delta$ pixels. In (a) the camera translates through a distance $X$ parallel to the image plane, while in (b) the camera rotates through an angle $\theta$ about its optical centre. By simple trigonometry, we can see that in (a) the camera must translate by

$$X = \frac{\delta}{F} D, \tag{3.1}$$

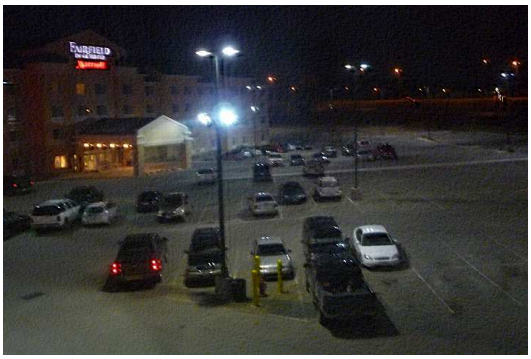
where $F$ is the camera's focal length, while in (b) the camera must rotate through an angle

$$\theta = \tan^{-1}\left(\frac{\delta}{F}\right). \tag{3.2}$$

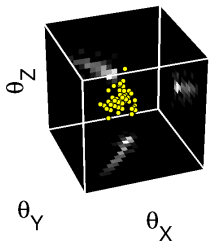
If we make the common assumption that the camera's focal length $F$ is approximately equal to the width of the sensor, say 1000 pixels, then to cause a blur of $\delta = 10$ pixels by translating the camera, we can see from Equation (3.1) that $X = \frac{1}{100} D$. Thus the required translation grows with the subject's distance from the camera, and for a subject just 1 metre away, we must move the camera by $X = 1\,\text{cm}$ to cause the blur. When photographing a subject 30 metres away, such as a large landmark, we would have to move the camera by 30 cm!



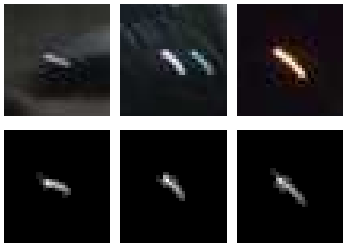
(a) An image blurred by camera shake



(b) Deblurred using the estimated blur descriptor




(c) Estimated global blur descriptor

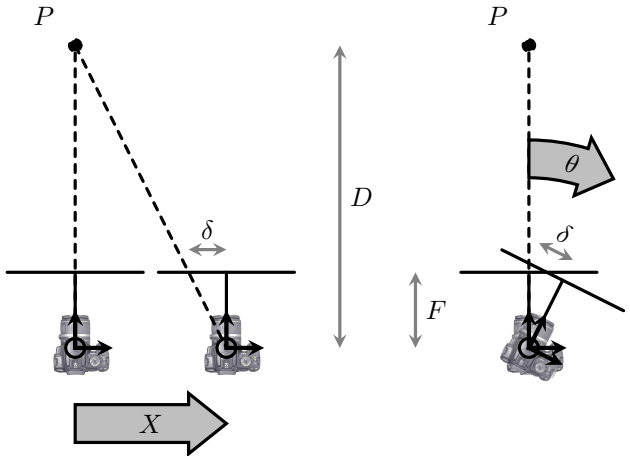


(d) Top: Details from blurry image. Bottom: corresponding local PSFs generated by (c)

(e) Details from the blurry (top) and deblurred (bottom) images

Figure 3.1. **Modelling non-uniform blur in a shaken image.** The blurry image (a) clearly exhibits blur which is non-uniform, as highlighted at different locations in the image. Using the model proposed in this work, we can describe this blur using a single global descriptor (c), which in this case has been estimated from the blurry image itself, simply by modifying existing algorithms for blind deblurring (see Chapter 4 for details). Close-ups of different parts of the image (d) show the variation in the shape of the blur, which can be accurately reproduced using our model, as shown by the local point spread functions generated from it. As can be seen in the deblurred image in (b) (see Chapter 6 for details on deblurring images like this with saturated pixels) and the close-ups in (e), different parts of the image, blurred in different ways, can be deblurred to recover a sharp image.

To cause the same amount of blur by rotating the camera, on the other hand, we can see from Equation (3.2) that we would need to rotate the camera by $\theta = \tan^{-1}\left(\frac{1}{100}\right) \simeq 0.6°$, independent of the subject's distance from the camera. To put this in terms of the motion of the photographer's hands, then for example if the camera body is 10 cm wide, such a rotation could be caused by moving one hand just 1 mm forwards or backwards relative to the other. Provided the subject is more than 1 metre from the camera, this



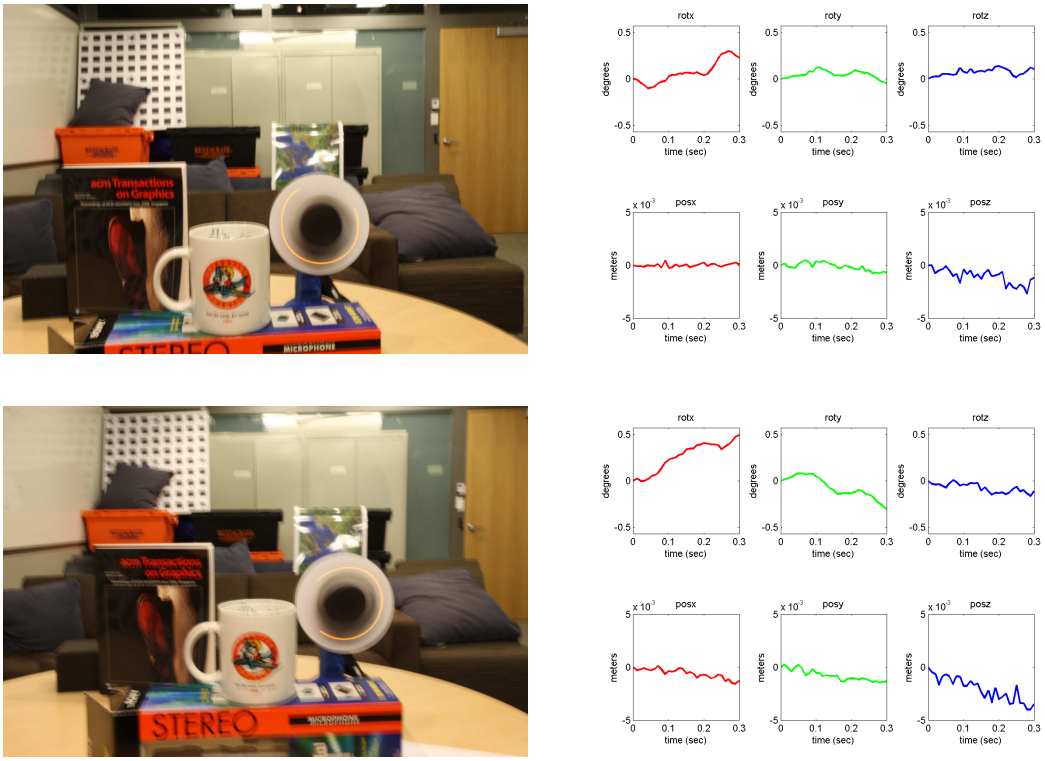

(a) Camera translation (b) Camera rotation

Figure 3.2. **Blur due to translation or rotation of the camera.** In this simplified example, we consider capturing a blurry image by either (a) translating the camera through a distance $X$ parallel to the image plane, or (b) rotating the camera through an angle $\theta$ about its optical centre. We consider the scene point $P$ at a distance $D$ from the camera, whose image is blurred by $\delta$ pixels as a result of either of the two motions. In most cases, for a given blur size $\delta$ the rotation $\theta$ constitutes a significantly smaller motion of the photographer's hands than the translation $X$ (see text for details).

motion is at least an order of magnitude smaller than for a translation of the camera causing an equivalent amount of blur.

In reality, both the position and orientation of the camera vary simultaneously during the exposure. However, if the camera only undergoes small changes in position (translations), then following the discussion above, we can assert that the variations in the camera's orientation (rotations) are the only significant cause of blur. Figure 3.3 supports this assumption, showing some shaken images captured by Joshi et al. (2010), for which the motion of the camera during the exposure was recorded. As can be seen, the camera translates by a few millimetres, while rotating by up to 0.5°. For these magnitudes of motion, the blur for subjects more than 1 m away from the camera can almost entirely be attributed to the rotation.

From now on, we assume that the translational component of camera motion *does not cause any blur*. Furthermore, we assume that all rotations occur about the camera's optical centre. Note, however, that a camera rotation about a centre that is not the



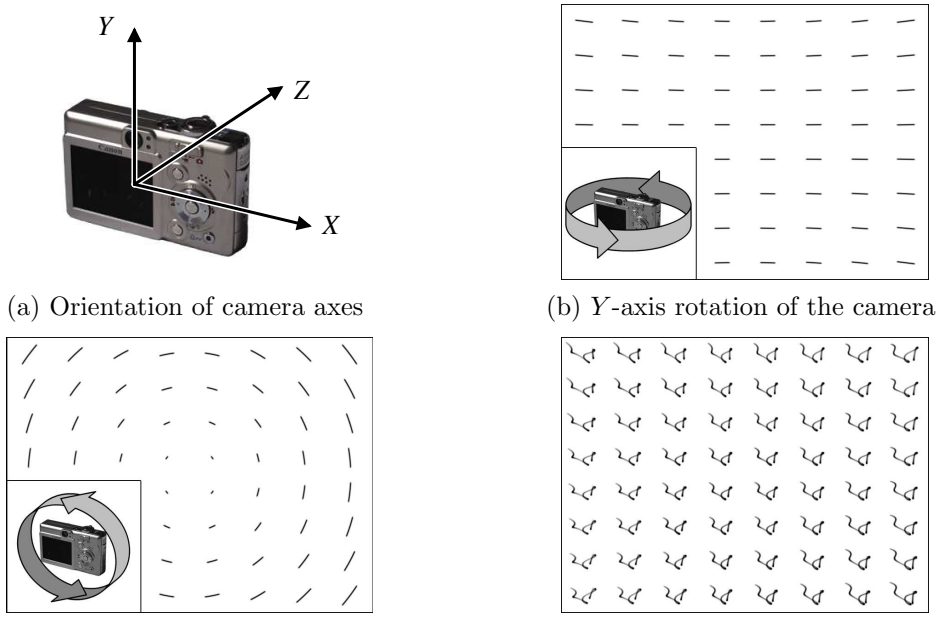

(a) Blurry image (b) Motion plots

Figure 3.3. **Real measurements of camera motion during a long exposure.** These photographs, from (Joshi et al., 2010), are taken with a 1/3 second exposure, and show how the pose of the camera varied during the exposure. The motion is recovered using an additional high-speed camera attached to the main camera – by performing structure-from-motion on the high-speed frames, the camera's pose can be recovered. From the motion plots, we see that the camera rotates between $\pm 0.5°$, and translates between $\pm 4\,\text{mm}$ while the shutter is open.

optical centre can be written as a rotation about the optical centre composed with a translation; these translations should generally be small for rotation centres that are not far from the optical centre.

### 3.2.2 Motion Blur and Homographies

Under a pinhole camera model, and assuming that the scene being photographed is static, rotations of a camera about its optical centre induce projective transformations of the image being observed. In other words, the image observed at one camera orientation is




(a) Orientation of camera axes

(b) $Y$-axis rotation of the camera

(c) $Z$-axis rotation of the camera

(d) Arbitrary sequence of rotations

Figure 3.4. **Our coordinate frame with respect to initial camera orientation, and the paths followed by image points under different camera rotations.** We define our coordinate frame (a) to have its origin at the camera's optical centre, with the $X$ and $Y$ axes aligned with those of the camera's sensor, and the $Z$ axis parallel to the camera's optical axis. Under single-axis rotations of the camera, for example about its $Y$-axis (b), or its $Z$-axis (c), the paths traced by points in the image are visibly curved and non-uniform across the image. This non-uniformity remains true for general camera shakes (d), which do not follow such simple single-axis rotations, but rather take arbitrary paths through camera pose space. The focal length of the camera in this simulation is equal to the width of the image, the principal point is at the image's centre, and the pixels are assumed to be square.

related to the image at any other by a 2D projective transformation, or homography. For an uncalibrated camera, this is a general 8-parameter homography, but for a camera with known internal parameters, the homography $\mathbf{H}$ is given by

$$\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}, \tag{3.3}$$

where the $3 \times 3$ matrix $\mathbf{R}$ is a rotation matrix describing the orientation of the camera, and $\mathbf{K}$ is the camera's internal calibration matrix (Hartley and Zisserman, 2004). In this work, we assume that the calibration matrix $\mathbf{K}$ is known (see Section 3.2.3).

The rotation matrix $\mathbf{R}$ has only 3 parameters. We adopt here the "angle-axis"

parameterisation, in which a rotation is described by the angle $\theta$ moved about an axis $\mathbf{a}$ (a unit-norm 3-vector). This can be summarised in a single 3-vector $\boldsymbol{\theta} = \theta\mathbf{a} = (\theta_X, \theta_Y, \theta_Z)$. $\mathbf{R}$ is then given by the matrix exponential

$$\mathbf{R}_{\boldsymbol{\theta}} = e^{[\boldsymbol{\theta}]_\times}, \quad \text{where} \tag{3.4}$$

$$[\boldsymbol{\theta}]_\times = \begin{bmatrix} 0 & -\theta_Z & \theta_Y \\ \theta_Z & 0 & -\theta_X \\ -\theta_Y & \theta_X & 0 \end{bmatrix}. \tag{3.5}$$

We fix our 3D coordinate frame to have its origin at the camera's optical centre. The axes are aligned with the camera's initial orientation, such that the $XY$-plane is aligned with the camera sensor's coordinate frame and the $Z$-axis is parallel to the camera's optical axis, as shown in Figure 3.4 (a). In this configuration, $\theta_X$ describes the "pitch" of the camera, $\theta_Y$ the "yaw", and $\theta_Z$ the "roll", or in-plane rotation, of the camera.

Having defined the type of image transformations we expect to occur while the shutter is open, we can write out the image degradation model. Let $T$ denote the exposure time of the photograph. While the shutter is open, the camera passes through a sequence of orientations $\boldsymbol{\theta}_t$, $t \in [0, T]$. As discussed above, at each pose $\boldsymbol{\theta}_t$, the sensor is exposed to a projectively transformed version of the sharp image $f$, where the projective transformation $\mathbf{H}_t$ is given by Equations (3.3) to (3.5). The noiseless blurry image $g^*$ is then modelled as the integral over the exposure time $T$ of all the transformed versions of $f$:

$$g^*(\mathbf{x}) = \int_0^T f\big(\mathbf{H}_t\mathbf{x}\big)\,\mathrm{d}t, \tag{3.6}$$

where, with a slight abuse of notation, we use $g^*(\mathbf{x})$ to denote the value of $g^*$ at the 2D image point represented by the homogeneous vector $\mathbf{x}$, and similarly for $f$.

Under this model, the apparent motion of scene points may vary significantly across the image. Figure 3.4 demonstrates this, showing the paths followed by points in an image under a $Y$-axis rotation, a $Z$-axis rotation, or an arbitrary sequence of rotations

of the camera. Under the (in-plane) $Z$-axis rotation, the paths vary significantly across the image. Under the (out-of-plane) rotation about the $Y$-axis, the paths, while varying considerably less, are still non-uniform. It should be noted that the degree of non-uniformity of this out-of-plane motion is dependent on the focal length of the camera, decreasing as the focal length increases. However, it is typical for consumer cameras to have focal lengths of the same order as their sensor width, as is the case in Figure 3.4. In addition, it is common for camera shake to include an in-plane rotational motion. From this, it is clear that modelling camera shake as a convolution with a spatially-invariant kernel is insufficient to fully describe its effects (see also Figure 3.1).

In general, a blurry image has no temporal information associated with it, so it is convenient to replace the temporal integral in Equation (3.6) by a weighted integral over a set of camera orientations:

$$g^*(\mathbf{x}) = \int f\big(\mathbf{H}_{\boldsymbol{\theta}}\mathbf{x}\big) w(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}, \tag{3.7}$$

where the weight function $w(\boldsymbol{\theta})$ encodes the camera's trajectory in a time-agnostic fashion. The weight will be zero everywhere except along the camera's trajectory, and the value of the function at a point $\boldsymbol{\theta}$ along the trajectory corresponds to the duration the camera spent at the orientation $\boldsymbol{\theta}$.

### 3.2.3 Camera Calibration

In order to compute the homography in Equation (3.3) that is induced by a particular rotation of the camera, we need to know the camera's calibration matrix $\mathbf{K}$. For the results shown in this thesis, we assume that $\mathbf{K}$ takes the standard form

$$\mathbf{K} = \begin{bmatrix} F & 0 & x_0 \\ 0 & F & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.8}$$

This corresponds to a camera whose sensor has square pixels, and whose optical axis intersects the sensor at $(x_0, y_0)$, referred to as the principal point. To estimate $\mathbf{K}$, we recover the pixel size and focal length of the camera from the image's EXIF tags, and assume that the principal point is at the centre of the image (although this does not hold exactly true in general, this simplified model is typically sufficient for modelling homographies caused by camera rotations (Szeliski, 2004)).

The radial distortion present in many consumer-grade digital cameras can represent a significant deviation from the pinhole camera model. Rather than incorporating the distortion explicitly into our model, we pre-process images with the commercially available PTLens software[1], which uses a database of lens and camera parameters to correct for the distortion.

A second distortion present in many digital images comes from the fact that the pixel values stored in, for example, a jpeg file, do not correspond linearly to the scene radiance. Most cameras apply a compression curve before storing the values, sometimes referred to as "gamma correction". Where possible we avoid this problem by using raw camera output images, such that the pixel values correspond linearly to scene radiance. In other cases, where the compression curve is known (e.g. having been calibrated), we preprocess the blurry images with the inverse of this curve to recover the linear values, and where it is unknown, we apply a generic sRGB curve.

In practice, there are several potential pitfalls in relying on the EXIF information to recover the internal calibration of the camera. For example, if the image has been cropped or resized, the focal length and pixel size recovered from the EXIF tags will no longer be correct. In addition, many image editors delete or overwrite the EXIF tags, and while almost all cameras include the focal length in the tags, some e.g. camera-phones, do not include sufficient information to estimate the pixel size. In the case that the EXIF tags are incorrect or incomplete, we set the focal length to the width of the image, and the principal point to the center of the image. Although this is unlikely to be an accurate estimate, it is typically sufficient for deblurring purposes.

[1] `http://epaperpress.com/ptlens/`

### 3.2.4 Uniform Blur As a Special Case

One consequence of our model for camera shake is that it includes uniform blur as a special case, and thus gives the conditions under which a uniform blur model is applicable. From the definition of the matrix exponential, $e^A = I + A + \frac{1}{2!}A^2 + \ldots$, we can see that if $\theta_Z = 0$ and $\theta_X$, $\theta_Y$ are small, Equation (3.4) can be approximated by discarding the $2^{\text{nd}}$ and higher order terms:

$$\mathbf{R}_{\boldsymbol{\theta}} \approx \begin{bmatrix} 1 & 0 & \theta_Y \\ 0 & 1 & -\theta_X \\ -\theta_Y & \theta_X & 1 \end{bmatrix}. \tag{3.9}$$

Combining this with Equations (3.3) and (3.8), it can be shown that as $F \to \infty$,

$$\mathbf{H}_{\boldsymbol{\theta}} \to \begin{bmatrix} 1 & 0 & F\theta_Y \\ 0 & 1 & -F\theta_X \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.10}$$

which simply amounts to a translation in the image plane of $(F\theta_Y, -F\theta_X)^\top$. Note that for typical camera shakes, $\theta_X$ and $\theta_Y$ will indeed be small. Thus we can see that if the focal length of the camera is large (e.g. if the camera is zoomed-in) and there is no in-plane rotation, a uniform blur model may be sufficient to describe the blur.

## 3.3 A Computational Model for Camera Shake

So far, our model has been defined in terms of the continuous functions $f$ and $g$, and the weight function $w$. We discretise the camera orientation space into a 3D volumetric grid of size $N_X \times N_Y \times N_Z$, and assign each orientation $\boldsymbol{\theta}^{(k)}$ a weight $w_k$, for $k \in \{1, \ldots, K\}$, where $K = N_X N_Y N_Z$. The set of weights $\mathbf{w}$ forms a global descriptor for the camera shake blur in an image, and by analogy with convolutional blur, we refer to $\mathbf{w}$ as the *blur kernel*.

Figure 3.1 (c) shows a visualisation of $\mathbf{w}$, where the cuboidal volume of size $N_X \times N_Y \times N_Z$ is shown, with the yellow points inside representing the non-zero elements of $\mathbf{w}$ in 3D. The kernel has also been projected onto the 3 back faces of the cuboid to aid visualisation, with white corresponding to a large value, and black corresponding to zero.

Each element $w_k$ corresponds to a camera orientation $\boldsymbol{\theta}^{(k)}$, and consequently to a homography $\mathbf{H}_k$, so that in the discrete setting, the blurry image $\mathbf{g}^*$ is modelled as a weighted sum of a set of projectively-transformed versions of $\mathbf{f}$:

$$\mathbf{g}^* = \sum_k w_k \mathbf{T}^{(k)} \mathbf{f}, \tag{3.11}$$

where $\mathbf{T}^{(k)}$ is the $N \times N$ matrix which applies homography $\mathbf{H}_k$ to the sharp image $\mathbf{f}$. The matrix $\mathbf{T}^{(k)}$ is very sparse. For example, if bilinear interpolation is used when transforming the image, each row has only 4 non-zero elements. Section 3.3.2 describes how to calculate the matrix $\mathbf{T}^{(k)}$. By writing out Equation (3.11) for a single pixel, we obtain the discrete analog of Equation (3.7):

$$g_i^* = \sum_k w_k \left( \sum_j T_{ij}^{(k)} f_j \right), \tag{3.12}$$

where $i$ and $j$ index the pixels of the blurry image and the sharp image, respectively. For a blurry pixel $g_i^*$ with coordinate vector $\mathbf{x}_i$, the sum $\sum_j T_{ij}^{(k)} f_j$ interpolates the value of $f(\mathbf{H}_k \mathbf{x}_i)$. Figure 3.5 shows an example of this, where a blurry pixel with homogeneous coordinate vector $\mathbf{x}_i$ is mapped under a homography $\mathbf{H}_k$ to the point $\mathbf{H}_k \mathbf{x}_i$ in the sharp image. The value of $\mathbf{f}$ at the point $\mathbf{H}_k \mathbf{x}_i$ is then interpolated as a weighted sum of the nearby pixels of $\mathbf{f}$. Section 3.3.2 gives further details of this interpolation.

Due to the bilinear form of Equation (3.12), note that when either the blur kernel or the sharp image is known, the blurry image is linear in the remaining unknowns, i.e.

$$\text{given } \mathbf{w}, \qquad \mathbf{g}^* = \mathbf{A}\mathbf{f}, \qquad \text{where } A_{ij} = \sum_k T_{ij}^{(k)} w_k, \tag{3.13}$$

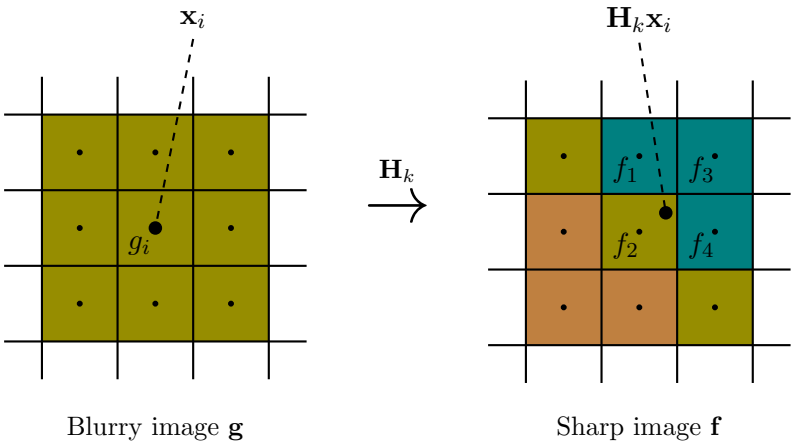



Figure 3.5. **Interpolation of sub-pixel locations in the sharp image.** In general, a homography $\mathbf{H}_k$ will not map a pixel (e.g. $\mathbf{x}_i$) in the blurry image $\mathbf{g}$ to a single pixel in the sharp image $\mathbf{x}$. Instead, the value of $\mathbf{f}$ at the point $\mathbf{H}_k\mathbf{x}_i$ is interpolated as a weighted sum of nearby pixels. Using bilinear interpolation, the value of $\mathbf{f}$ at $\mathbf{H}_k\mathbf{x}_i$ is computed as a linear combination of $f_1$, $f_2$, $f_3$, and $f_4$.

$$\text{given } \mathbf{f}, \qquad \mathbf{g}^* = \mathbf{B}\mathbf{w}, \qquad \text{where } B_{ik} = \sum_j T_{ij}^{(k)} f_j. \tag{3.14}$$

In the first form, $\mathbf{A} \in \mathbb{R}^{N \times N}$ is a large sparse matrix, whose rows each contain a local blur filter acting on $\mathbf{f}$ to generate a blurry pixel. In the second form, when the sharp image is known, each column of $\mathbf{B} \in \mathbb{R}^{N \times K}$ contains a projectively transformed copy of the sharp image. We will use each of these forms in the following chapters.

### 3.3.1 Comparison to Other Non-Uniform Blur Models

Concurrently with our proposal of this model for camera shake blur, several other authors have proposed global models of spatially-variant blur. In common with our approach, they generally model the blurry image as a sum of transformed versions of the sharp image. Tai et al. (2011) model the image motion using a temporally-ordered sequence of unconstrained 8-parameter homographies, which is known in advance. Joshi et al. (2010) also use a temporally-ordered sequence of 6-parameter homographies which correspond to a fronto-parallel planar scene, parameterised by the camera's pose. Gupta et al. (2010) propose a model which is similar in spirit to our own, recovering a set of weights over a

3D parameter space which describes transformations of the sharp image. However, they consider image plane translations and rotations, rather than the camera pose-induced homographies used in this work.

Gupta et al. (2010) provide some analysis of the accuracy of their model as the focal length of the camera varies, and show that as the focal length decreases, their model becomes less accurate. Starting from our model, this is not unexpected, as it is possible to show that their model is a special case of ours, obtained by letting $F \to \infty$. By following essentially the same derivation as in Section 3.2.4, except with $\theta_Z \neq 0$, it is possible to show that for small $\theta_X$ and $\theta_Y$, as $F \to \infty$,

$$\mathbf{H}_{\boldsymbol{\theta}} \to \begin{bmatrix} \cos\theta_Z & -\sin\theta_Z & F\theta_Y \\ \sin\theta_Z & \cos\theta_Z & -F\theta_X \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.15}$$

which covers the set of 2D translations and rotations considered in their work. By including the focal length in our model, we ensure accuracy even at short focal lengths. Of course, the need to know the focal length for an image is an additional requirement, compared to the model of Gupta et al. (2010). However, given that we recover their model by setting $F$ to a large number, and that it is often possible to extract the focal length of a photo from the EXIF tags, we do not believe this is a significant disadvantage of our model.

### 3.3.2 Computation of Interpolation Coefficients

Here we give details of how to calculate the matrix $\mathbf{T}^{(k)}$ that applies a homography $\mathbf{H}_k$ to an image, using bilinear interpolation. Note that these are standard interpolation weights, and are not specific to this deblurring application.

Using bilinear interpolation, the value of an image $f$ at a sub-pixel location $(x', y')$ is interpolated from the 4 pixels surrounding $(x', y')$ as follows:

$$\delta_x = x' - \lfloor x' \rfloor \tag{3.16}$$

$$\delta_y = y' - \lfloor y' \rfloor \tag{3.17}$$

$$\begin{aligned} f(x', y') = (1 - \delta_x)(1 - \delta_y) \quad & f\big(\lfloor x' \rfloor, \lfloor y' \rfloor\big) \\ + (1 - \delta_x)\delta_y \quad & f\big(\lfloor x' \rfloor, \lfloor y' \rfloor + 1\big) \\ + \delta_x(1 - \delta_y) \quad & f\big(\lfloor x' \rfloor + 1, \lfloor y' \rfloor\big) \\ + \delta_x\delta_y \quad & f\big(\lfloor x' \rfloor + 1, \lfloor y' \rfloor + 1\big) \end{aligned} \tag{3.18}$$

where $\lfloor \cdot \rfloor$ takes the integer part of a positive scalar.

In Equation (3.12), we write this interpolation operation as the sum

$$f(\mathbf{H}_k \mathbf{x}_i) = \sum_j T_{ij}^{(k)} f_j. \tag{3.19}$$

Using bilinear interpolation as described above, we can see that the elements $T_{ij}^{(k)}$ take the following values:

| $j$ | $T_{ij}^{(k)}$ |
|---|---|
| $\texttt{sub2ind}(\lfloor x' \rfloor, \lfloor y' \rfloor)$ | $(1 - \delta_x)(1 - \delta_y)$ |
| $\texttt{sub2ind}(\lfloor x' \rfloor, \lfloor y' \rfloor + 1)$ | $(1 - \delta_x)\delta_y$ |
| $\texttt{sub2ind}(\lfloor x' \rfloor + 1, \lfloor y' \rfloor)$ | $\delta_x(1 - \delta_y)$ |
| $\texttt{sub2ind}(\lfloor x' \rfloor + 1, \lfloor y' \rfloor + 1)$ | $\delta_x\delta_y$ |
| all others | 0 |

where the MATLAB function `sub2ind` provides the correspondence between the $(x, y)$ coordinates of a pixel and its index $j$.

### 3.3.3 Sampling the Set of Rotations

One important detail to consider is how finely to discretise the orientation parameter $\boldsymbol{\theta}$. Undersampling the set of orientations will affect our ability to accurately reconstruct the blurred image, but sampling it too finely will lead to unnecessary calculations. Since the kernel is defined over the 3 parameters $\theta_X$, $\theta_Y$ and $\theta_Z$, doubling the sampling resolution increases the number of kernel elements by a factor of 8. In practice, we have found that

a good choice of grid spacing is that which corresponds to a maximum displacement of 1 pixel in the image. Since we are fundamentally limited by the resolution of our images, reducing the spacing further leads to redundant orientations, which are indistinguishable from their neighbours. Setting the grid spacing in terms of pixels also has the advantage that our 3D blur kernels are defined on a grid which allows direct comparison to the pixel grid of the image. We set the size of our kernel along each dimension in terms of the size of the blur we need to model, typically a few degrees along each dimension of $\boldsymbol{\theta}$, e.g. $[-5°, 5°]$.

## 3.4 Conclusion

In this chapter we have proposed a geometrically-derived model for blur caused by camera shake. For a static scene and a camera with known focal length, we have shown that the blur caused by camera rotation can be modelled using a weighted set of homographies, and have proposed a practical formulation of this model in which the blurry image is bilinear in the sharp image and the weights. The model assumes that the motion of the camera during exposure is limited to rotations about its optical centre, and is temporally-agnostic to the distribution over camera orientations. Our model is not applicable for non-static scenes, or nearby scenes with large camera translations where parallax effects may become significant.

# Chapter 4

# Estimating and Removing Spatially-Variant Camera Shake Blur

## 4.1 Introduction

In this chapter, we demonstrate the effectiveness of the spatially-variant blur model presented in Chapter 3 by using it to replace the uniform blur model in three existing approaches to camera shake removal. We show quantitative and qualitative improvements in the deblurred results.

In Section 4.3 we consider single-image deblurring, where only a blurry image is available, applying our model within the algorithms proposed by Fergus et al. (2006) (Section 4.3.1) and Cho and Lee (2009) (Section 4.3.2). We compare our results to those of the original algorithms (which handle only spatially-invariant blur), and discuss the advantages and limitations of using our model. In Section 4.4 we apply our model to the case where an additional sharp but noisy image of the same scene is available, as proposed by Yuan et al. (2007b). In Section 4.5 we discuss some implementation considerations of using our model in these algorithms, before concluding in Section 4.6.

## 4.2 Application to Existing Deblurring Algorithms

The fact that Equation (3.12) (p. 66) is bilinear in the sharp image and blur kernel is the key feature that allows our model to be applied within existing deblurring algorithms previously applied only to uniform blur. Since convolution is also a bilinear operation on the sharp image and the blur kernel, it can often be replaced with the general bilinear form in Equation (3.12) without significant modification to the algorithm. Here we discuss some of the general issues related to achieving this.

The constraint that all the PSF parameters be positive is intuitively applicable to our model, since each kernel element $w_k$ corresponds directly to the length of time spent at a camera orientation $\boldsymbol{\theta}_k$. If the camera passed through orientation $\boldsymbol{\theta}_k$ during the exposure, $w_k$ will be positive, and if not, $w_k = 0$.

The sparsity prior is intuitively applicable to blur kernels for our model too, since the camera follows a path $\boldsymbol{\theta}(t)$ through the space of camera orientations, and thus will only pass through a small subset of all possible orientations while the shutter is open.

It should be noted that an important case where our model cannot easily be substituted in place of convolution is when an algorithm relies on the ability to work in the frequency domain, e.g. the Wiener filter (Wiener, 1949). When taking the Fourier transform, convolution becomes an element-wise multiplication of the frequency components of the image and kernel, however this is not the case for the more general bilinear form in our model (see Chapter 5 for an efficient approximation to our model which benefits from this frequency-domain property).

## 4.3 Single-Image Deblurring

In this section, we examine the case where we have only a single blurry input image $\mathbf{g}$ from which to estimate $\hat{\mathbf{f}}$. We substitute our model into two successful algorithms for uniform blur, allowing them to handle non-uniform blur: those of Fergus et al. (2006) (Section 4.3.1) and Cho and Lee (2009) (Section 4.3.2).

### 4.3.1 The Marginalisation Approach

In this section we adapt the algorithm proposed by Fergus et al. (2006) for blind deconvolution of a single image, introduced in Section 2.3.2. We show that the convolutional blur model in the original algorithm can be replaced with our non-uniform blur model, leading to new update equations for the optimisation process, and we show in Section 4.3.3 that doing so improves the deblurred results.

Recall from Section 2.3.2 that the aim is to find the factorised distribution

$$q(\boldsymbol{\Theta}) = q(\beta_\sigma) \prod_j q(f_j^\partial) \prod_k q(w_k) \tag{4.1}$$

that best approximates the true posterior $p(\boldsymbol{\Theta}|\mathbf{g})$. In Section 2.3.2, the "ensemble" $\boldsymbol{\Theta}$ incorporated the latent image gradients $\mathbf{f}^\partial$, a convolution kernel $\mathbf{a}$ and the noise precision $\beta_\sigma$. In our case we replace the convolution kernel with the blur kernel $\mathbf{w}$ for our model. For our blur model, the optimal $q(\boldsymbol{\Theta})$ has the same form as in (Miskin and MacKay, 2000). However the equations for the optimal parameter values differ significantly and we have calculated these afresh (the derivation is provided in Appendix A). For our non-uniform blur model, we find the following optimal values for the parameters, *cf*. (Miskin and MacKay, 2000, Eqns. (46)–(49)):

$$w_k^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big\langle \Big( \sum_j T_{ij}^{(k)} f_j^\partial \Big)^2 \Big\rangle_{q(\mathbf{f}^\partial)} \tag{4.2}$$

$$w_k^{(1)} w_k^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big( g_i \sum_j T_{ij}^{(k)} \langle f_j^\partial \rangle_{q(f_j^\partial)} - \sum_{k' \neq k} \Big\langle \Big( \sum_j T_{ij}^{(k)} f_j^\partial \Big) \Big( \sum_j T_{ij}^{(k')} f_j^\partial \Big) \Big\rangle_{q(\mathbf{f}^\partial)} \langle w_{k'} \rangle_{q(w_{k'})} \Big) \tag{4.3}$$

$$f_j^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big\langle \Big( \sum_k T_{ij}^{(k)} w_k \Big)^2 \Big\rangle_{q(\mathbf{w})} \tag{4.4}$$

$$f_j^{(1)} f_j^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big( g_i \sum_k T_{ij}^{(k)} \langle w_k \rangle_{q(w_k)} - \sum_{j' \neq j} \langle f_{j'}^\partial \rangle_{q(f_{j'}^\partial)} \Big\langle \Big( \sum_k T_{ij'}^{(k)} w_k \Big) \Big( \sum_k T_{ij}^{(k)} w_k \Big) \Big\rangle_{q(\mathbf{w})} \Big), \tag{4.5}$$

where $w_k^{(1)}$ and $w_k^{(2)}$ are the parameters of $q(w_k)$, $f_j^{(1)}$ and $f_j^{(2)}$ are the parameters of $q(f_j^\partial)$, $q(\mathbf{f}^\partial) = \prod_j q(f_j^\partial)$, $q(\mathbf{w}) = \prod_k q(w_k)$, and $\langle \cdot \rangle_q$ represents the expectation with respect to the distribution $q$. For spatially-invariant blur (i.e. when the matrices $\mathbf{T}^{(k)}$ apply 2D translations of the image), these equations reduce to those given by Miskin and MacKay (2000). Note that these equations cannot be implemented directly in this form, and must be re-written as computations involving only the parameters $f_j^{(1)}$, $f_j^{(2)}$, $w_k^{(1)}$, and $w_k^{(2)}$, as follows. These are exactly the parameter updates computed in our implementation of the marginalisation algorithm described in Chapter 2. First, let

$$v_k^w = \left\langle w_k^2 \right\rangle - \langle w_k \rangle^2 \tag{4.6}$$

$$v_j^f = \left\langle f_j^{\partial 2} \right\rangle - \left\langle f_j^\partial \right\rangle^2 \tag{4.7}$$

$$\langle A_{ij} \rangle = \sum_k T_{ij}^{(k)} \langle w_k \rangle \tag{4.8}$$

$$\langle B_{ik} \rangle = \sum_j T_{ij}^{(k)} \left\langle f_j^\partial \right\rangle \tag{4.9}$$

$$\langle g_i^* \rangle = \sum_k \left( \sum_j T_{ij}^{(k)} \left\langle f_j^\partial \right\rangle \right) \langle w_k \rangle \,. \tag{4.10}$$

Then,

$$w_k^{(2)} = \langle \beta_\sigma \rangle \sum_{i,j} {T_{ij}^{(k)}}^2 v_j^f + \langle \beta_\sigma \rangle \sum_i \langle B_{ik} \rangle^2 \tag{4.11}$$

$$w_k^{(1)} w_k^{(2)} = \langle \beta_\sigma \rangle \sum_i \langle B_{ik} \rangle \left( g_i - \langle g_i^* \rangle \right) - \langle \beta_\sigma \rangle \sum_{i,j} T_{ij}^{(k)} \langle A_{ij} \rangle \, v_j^f + \langle w_k \rangle \, w_k^{(2)} \tag{4.12}$$

$$f_j^{(2)} = \langle \beta_\sigma \rangle \sum_{i,k} {T_{ij}^{(k)}}^2 v_k^w + \langle \beta_\sigma \rangle \sum_i \langle A_{ij} \rangle^2 \tag{4.13}$$

$$f_j^{(1)} f_j^{(2)} = \langle \beta_\sigma \rangle \sum_i \langle A_{ij} \rangle \left( g_i - \langle g_i^* \rangle \right) - \langle \beta_\sigma \rangle \sum_{i,k} T_{ij}^{(k)} \langle B_{ik} \rangle \, v_k^w + \left\langle f_j^\partial \right\rangle f_j^{(2)}. \tag{4.14}$$

Finally, in order to compute the expectation $\langle \beta_\sigma \rangle$ of the noise variance, it is necessary to evaluate the expected value of the squared reconstruction error (Miskin and MacKay, 2000, Eqn. (40)), which can be computed as follows:

$$\left\langle (g_i - g_i^*)^2 \right\rangle = (g_i - \langle g_i^* \rangle)^2 + \sum_{j,k} {T_{ij}^{(k)}}^2 v_j^f v_k^w + \sum_j \langle A_{ij} \rangle^2 v_j^f + \sum_k \langle B_{ik} \rangle^2 v_k^w . \tag{4.15}$$

Section 4.3.3 shows results of the marginalisation algorithm using these modified update equations and our spatially-variant blur model, compared to the original algorithm of Fergus et al. (2006).

### 4.3.2 The Maximum a Posteriori Approach

In this section we adapt the single image PSF estimation algorithm of Cho and Lee (2009), introduced in Section 2.3.3, to use our spatially-variant blur model. Like the marginalisation algorithm of the previous section, this algorithm can be readily adapted to handle non-uniform blur, substituting our model in place of convolution.

The main changes to the algorithm described in Chapter 2 are simply to replace the convolution in Equation (2.53) with our blur model, so that instead of estimating a convolution kernel $\hat{\mathbf{a}}$ by solving Equation (2.53), we instead estimate the spatially-variant blur descriptor $\hat{\mathbf{w}}$ by solving

$$\min_{\mathbf{w}} \sum_q \omega_q \Big\| \sum_k w_k \mathbf{T}^{(k)} \mathbf{p}^{(q)} - \mathbf{d}^{(q)} * \mathbf{g} \Big\|_2^2 + \beta \left\| \mathbf{w} \right\|_2^2 , \tag{4.16}$$

where $\omega_q$ are weights for the data fidelity term, $\mathbf{p}^{(q)}$ are thresholded derivative maps obtained by non-linear filtering of the current estimate of the sharp image, and $\mathbf{d}^{(q)}$ are derivative filters. Similarly, when updating the latent image, we modify Equation (2.56) to be

$$\min_{\mathbf{f}} \sum_q \omega_q \Big\| \sum_k \hat{w}_k \mathbf{T}^{(k)} (\mathbf{d}^{(q)} * \mathbf{f}) - \mathbf{d}^{(q)} * \mathbf{g} \Big\|_2^2 + \alpha \|\mathbf{d}^x * \mathbf{f}\|_2^2 + \alpha \|\mathbf{d}^y * \mathbf{f}\|_2^2 . \tag{4.17}$$

Since the spatially-variant blur model is linear in $\mathbf{w}$ and $\mathbf{f}$, these are still linear least squares problems. Although we are not able to take full advantage of the speed optimisations proposed by Cho and Lee (2009), due to their use of Fourier transforms to compute convolutions, the algorithm is generally able to estimate a blur kernel in a much shorter time than the marginalisation algorithm of Section 4.3.1.

### Modification for Non-Uniform Blur

When applying our model within this algorithm, we must take into account some important differences between our 3D kernels and 2D convolution kernels. First, we note that the local 2D point spread function (PSF) of a single pixel does not uniquely determine the global 3D kernel. This can be seen by considering a vertical blur at the left or right-hand side of the image. Such a blur could be explained either by a rotation of the camera about its $X$ axis, a rotation about its $Z$ axis, or some combination of the two. Thus in order to best constrain the kernel, we must ensure that the pixels used to estimate the kernel (the non-zeroes in $\{\mathbf{p}^x, \mathbf{p}^y\}$) do not only come from a small region of the image, in order for the kernel estimation step to be well-conditioned. To achieve this, we simply subdivide the image into $3 \times 3$ regions, and apply the gradient thresholding step independently on each. This ensures that we retain a set of gradients that are well distributed over both orientation and location.

A second observation is that our 3D kernels contain a certain degree of redundancy, arising largely from the in-plane rotation of the camera. As can be seen in Figure 3.4, a rotation of the camera about its $Z$ axis causes a very small displacement for pixels towards the centre of the image. Thus, in the kernel estimation step, the information provided by these pixels will be ambiguous with respect to this component of the camera's motion. Only pixels near the edge of the image will be able to provide detailed information concerning this motion. While the spatial binning mentioned above goes some way to ensuring that these pixels from the edge of the image are present in $\{\mathbf{p}^x, \mathbf{p}^y\}$, they may be greatly outnumbered by pixels from the interior. As a result, the kernels recovered by minimising Equation (4.16) with our model generally contain many

non-zeros spread smoothly throughout, and do not produce good deblurred outputs (see Figure 4.5).

If instead of the $\ell_2$ regularisation in Equation (4.16), we apply $\ell_1$ regularisation combined with non-negativity constraints, the optimisation is encouraged to find a sparse kernel and is more likely to choose between ambiguous camera orientations, as opposed to spreading non-zero values across all of them. This is equivalent to an exponential prior on the kernel values. In this case, the kernel estimation problem becomes

$$\min_{\mathbf{w}} \sum_q \omega_q \Big\| \sum_k w_k \mathbf{T}^{(k)} \mathbf{p}^{(q)} - \mathbf{d}^{(q)} * \mathbf{g} \Big\|_2^2 + \beta \sum_k w_k \quad \text{s.t.} \quad \forall k = 1, \ldots, K, \quad w_k \geq 0. \tag{4.18}$$

This is an instance of the *Lasso* problem (Tibshirani, 1996), for which efficient optimisation algorithms exist (Efron et al., 2004; Kim et al., 2007; Mairal et al., 2010). This problem is convex, so that we can be sure of obtaining a global minimum. The different results obtained using $\ell_2$ and $\ell_1$ regularisation are discussed in Section 4.3.3. With the use of the $\ell_1$ regularisation, we found that the best results were obtained with a lower value of $\beta$ than that given by Cho and Lee, and for the results in this work using $\ell_1$ regularisation, we set $\beta = 0.01$. In the remainder of the thesis, we refer to the original algorithm of Cho and Lee as MAP-$\ell_2$, and our $\ell_1$ regularised version as MAP-$\ell_1$.

### 4.3.3 Single-Image Deblurring Results

We show in this section results of single-image deblurring using the algorithms described in Section 4.3 to estimate the spatially-variant PSF, with comparisons to results obtained with the original algorithms of Fergus et al. (2006) and Cho and Lee (2009) on both synthetic and real data. The estimated PSF is used with the non-blind deblurring algorithm of Krishnan and Fergus (2009) to estimate the final deblurred image. This algorithm is easily adapted to non-uniform blur since it involves repeated minimisations of quadratic cost functions of the form (see Equation (2.46))

$$\min_{\mathbf{f}} \ \|\mathbf{A}\mathbf{f} - \mathbf{g}\|_2^2 + \alpha \big\|\mathbf{d}^x * \mathbf{f} - \mathbf{v}^x\big\|_2^2 + \alpha \big\|\mathbf{d}^y * \mathbf{f} - \mathbf{v}^y\big\|_2^2, \tag{4.19}$$

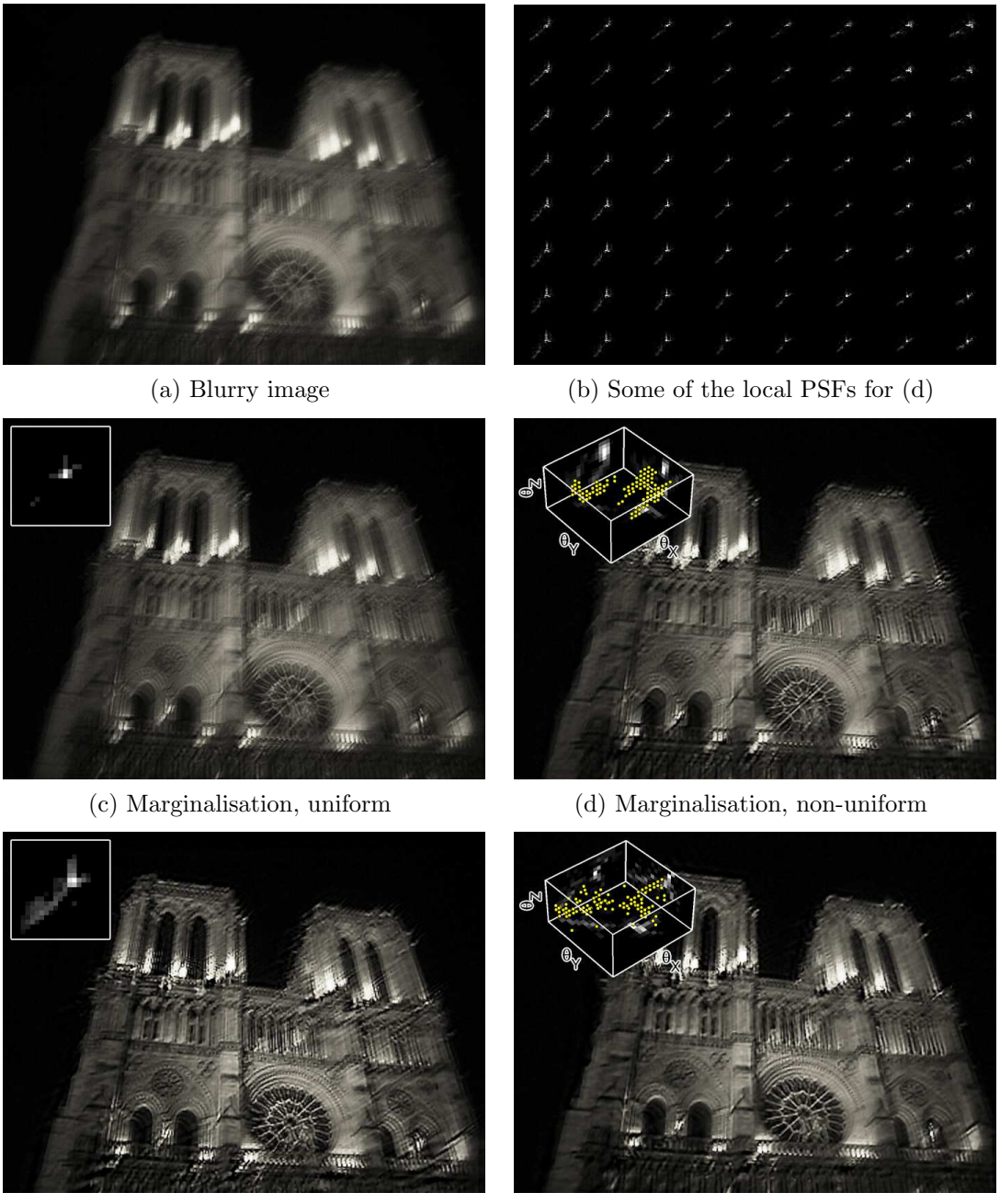
where $\mathbf{v}^x$ and $\mathbf{v}^y$ are intermediate variables of the optimisation scheme. For our non-uniform blur model, we use the conjugate gradient algorithm to minimise this cost function.

Figure 3.1 in Chapter 3 shows a result on a real camera shake blur, using the MAP-$\ell_1$ algorithm to estimate the kernel, and the Richardson-Lucy algorithm to perform the final deblurring. The blurry image has many saturated regions (e.g. the bright street lights), and in such cases we found the Richardson-Lucy algorithm to produce significantly better results than any least-squares based algorithms, such as that of Krishnan and Fergus (2009). We examine non-blind deblurring for images containing saturated pixels in Chapter 6.

Figures 4.1 and 4.2 show blind deblurring results on images blurred by real camera shake. Our model, used in both the marginalisation and MAP algorithms, is able to capture and remove the blur, while the original algorithms of Fergus et al. and Cho and Lee, using a uniform blur model, fail to find meaningful kernels or good deblurred results. This is explained by both the wide field of view, and the fact that the kernels estimated using our algorithm exhibit significant in-plane rotation.

In Figure 4.2 (d), we also demonstrate the use of the variational marginalisation algorithm of Fergus et al. to produce the deblurred output, as opposed to the algorithm of Krishnan and Fergus (2009), used in Figure 4.2 (c). Although, for computational simplicity, the kernel estimation step uses a grayscale image, at the convergence of this process the distributions $q(\mathbf{w})$ and $q(\beta_\sigma)$ for the kernel and noise variance can be fixed. The variational algorithm can then be run again to estimate $q(\mathbf{f}_c)$ for each color channel $c$ separately. In the final step, each color channel can be reconstructed from $q(\mathbf{f}_c)$ using Poisson reconstruction (Pérez et al., 2003), before matching the color histogram to that of the blurry image. As can be seen, a good deblurred image is produced, underlining the fact that our blur model is valid throughout the image, and that the kernel produced provides a good description of the true non-uniform blur in the image.

Figure 4.3 shows a third result of single-image deblurring, using the MAP algorithm. While the uniform blur kernel provides a reasonable estimate of the true blur, and allows




(a) Blurry image

(b) Some of the local PSFs for (d)

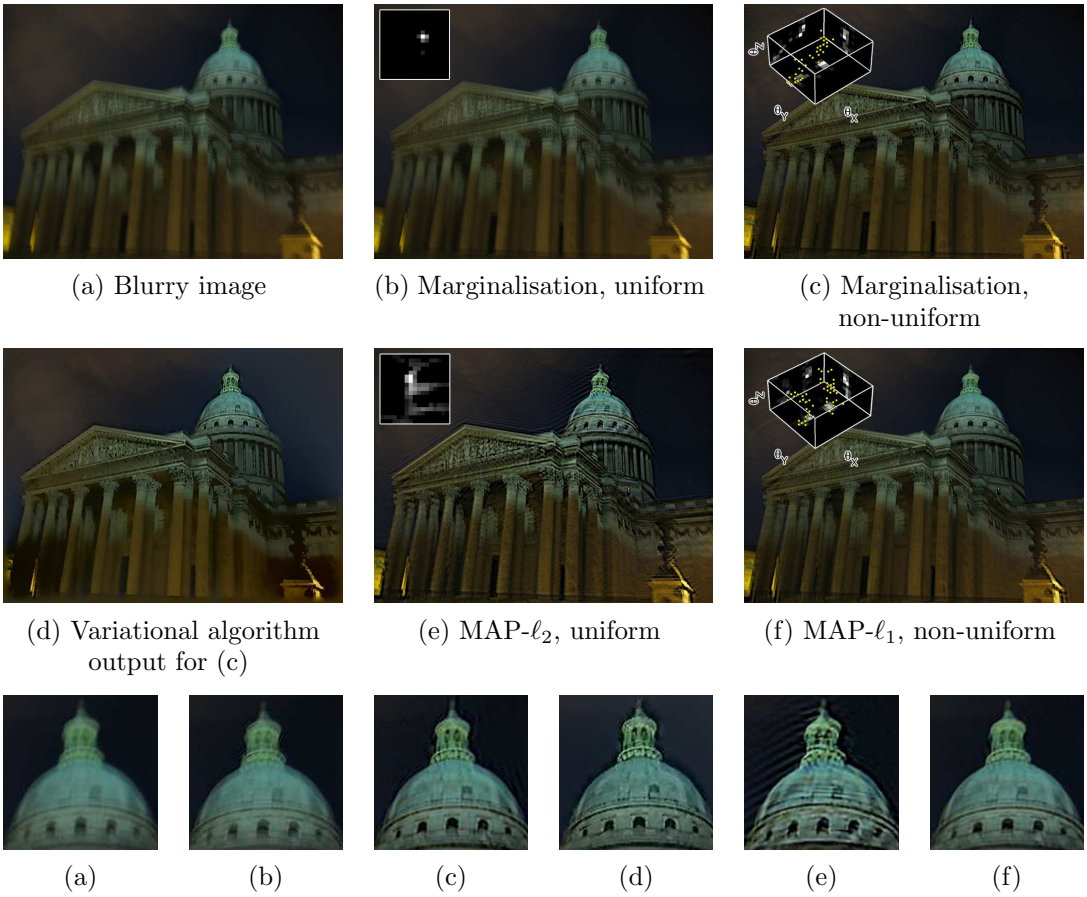
(c) Marginalisation, uniform

(d) Marginalisation, non-uniform

(e) MAP-$\ell_2$, uniform

(f) MAP-$\ell_1$, non-uniform

Figure 4.1. **Blind deblurring of real camera shake, example 1.** The result of blind deblurring on a real camera shake image (a), captured with a shutter speed of $\frac{1}{2}$ second, using both the marginalisation algorithm of Fergus et al. and the MAP approach of Cho and Lee with both the uniform and non-uniform blur models. Also shown in (b) are some of the local PSFs generated from the blur kernel in (d) at various points in the image. The marginalisation approach, when using our model (d) recovers a useful kernel and a good deblurred image, but when using the uniform model (c) does not. Using the MAP approach, the uniform model (e) finds a reasonable approximation to the non-uniform blur, which is valid on the left side of the image. However, on the right side, the error in the kernel leaves diagonal streaks on the deblurred output. Using our non-uniform model (f), however, avoids this problem. The blur kernels for our model in (d) and (f) cover $\pm 1.3^\circ$ along each dimension.



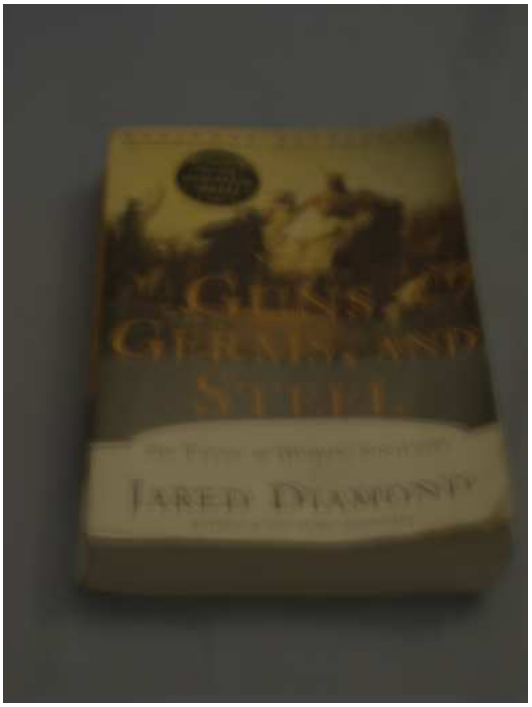
(a) Blurry image (b) Marginalisation, uniform (c) Marginalisation, non-uniform

(d) Variational algorithm output for (c) (e) MAP-$\ell_2$, uniform (f) MAP-$\ell_1$, non-uniform

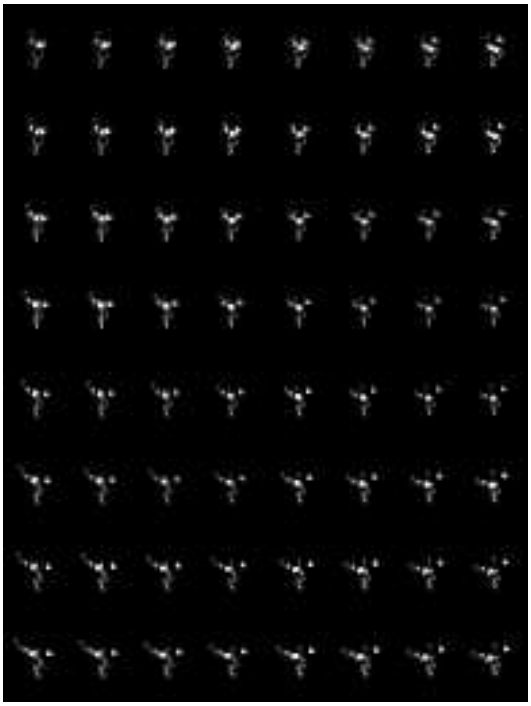


(a) (b) (c) (d) (e) (f)

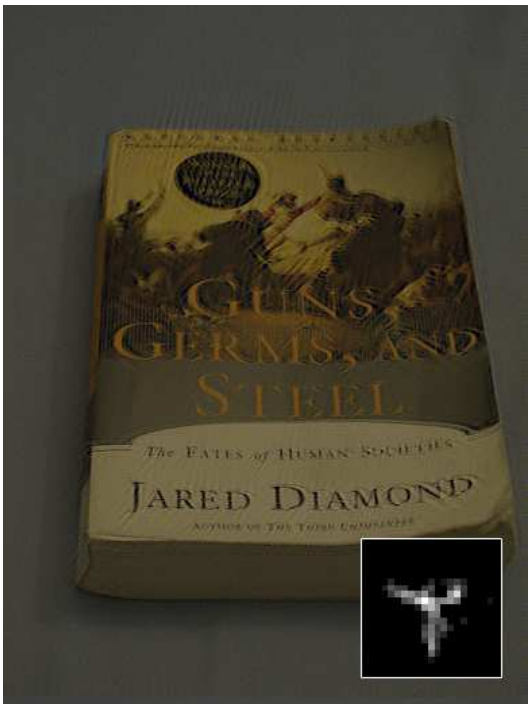
Figure 4.2. **Blind deblurring of real camera shake, example 2.** A hand-held image with camera shake (a), captured with a shutter speed of 1 second, with the results of blind deblurring using the marginalisation algorithm of Fergus et al. under both a uniform (b) and non-uniform (c)–(d) blur model, and the MAP algorithm of Cho and Lee with a uniform (e) and non-uniform (f) blur model. The estimated kernels for the two models are shown inset in the deblurred results. The variational output (d) is estimated using the marginalisation algorithm for the non-uniform case (calculated as $\langle \mathbf{f} \rangle_{q(\mathbf{f})}$ then converted from gradients to intensities using Poisson reconstruction (Pérez et al., 2003)). The results using our blur model show more detail and fewer artefacts than those using the uniform blur model, as can be seen in the zoomed-in portions shown in the last row. The rotational blur kernels in (c) and (f) cover $\pm 0.7°$ in $\theta_X$ and $\theta_Y$ and $\pm 1.4°$ in $\theta_Z$.



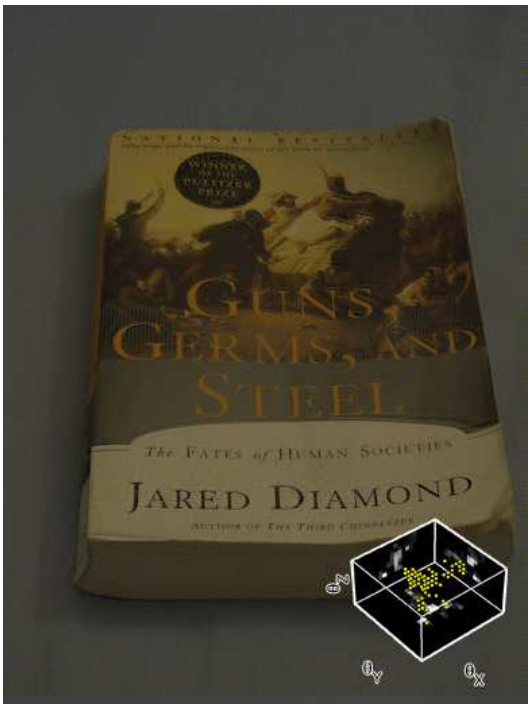
(a) Blurry image



(b) Some local PSFs for (d), magnified




(c) MAP-$\ell_2$, uniform




(d) MAP-$\ell_1$, non-uniform

Figure 4.3. **Blind deblurring of real camera shake, example 3.** The result of blind deblurring on a real camera shake image (a), captured with a shutter speed of 1 second, using the MAP approach of Cho and Lee with both the uniform and non-uniform blur models. Also shown in (b) are some of the local PSFs generated from the blur kernel in (d) at various points in the image. In the blurry image, most of the text on the book cover is too blurred to read. Deblurring the image with the uniform blur model (c) allows some of the text on the cover of the book to be read, however, after deblurring with our non-uniform model (d), all but the smallest text becomes legible. The estimated kernels for the two models are shown inset in the deblurred results. The blur kernel in (d) covers $\pm 0.4°$ in $\theta_X$ and $\theta_Y$, and $\pm 0.9°$ in $\theta_Z$.

| | Marginalisation | | | | MAP | | | |
|---|---|---|---|---|---|---|---|---|
| | 10px | | 20px | | 10px | | 20px | |
| **Y-axis** | Non-Uni. | Uni. | Non-Uni. | Uni. | Non-Uni. | Uni. | Non-Uni. | Uni. |
| $\sigma = 0$ | **23.1** (1.4) | 23.2 (1.4) | **27.2** (1.1) | 58.1 (2.4) | **18.3** (1.1) | 19.2 (1.2) | **25.6** (1.0) | 27.4 (1.1) |
| $\sigma = 5$ | **24.9** (1.3) | 25.8 (1.3) | **29.0** (1.1) | 56.8 (2.2) | **22.4** (1.1) | 25.9 (1.3) | **30.9** (1.2) | 31.4 (1.2) |
| $\sigma = 10$ | **27.0** (1.2) | 30.1 (1.3) | **30.7** (1.1) | 48.7 (1.8) | 37.3 (1.6) | **37.0** (1.6) | **38.3** (1.4) | 46.9 (1.7) |
| **Z-axis** | | | | | | | | |
| $\sigma = 0$ | **14.4** (1.3) | 21.8 (2.0) | **18.1** (1.1) | 26.1 (1.6) | **12.0** (1.1) | 27.7 (2.5) | **16.9** (1.0) | 44.1 (2.7) |
| $\sigma = 5$ | **17.4** (1.2) | 24.8 (1.7) | **23.2** (1.2) | 54.5 (2.8) | **17.5** (1.2) | 29.9 (2.1) | **24.0** (1.2) | 48.9 (2.5) |
| $\sigma = 10$ | **22.0** (1.1) | 50.9 (2.7) | **26.5** (1.1) | 55.8 (2.4) | **24.0** (1.2) | 35.5 (1.8) | **32.1** (1.4) | 52.7 (2.2) |

RMS errors between deblurred results and true sharp image
(with ratios to the error obtained with ground-truth kernel in parentheses)


Figure 4.4. **Blind deblurring of synthetic single-axis blurs.** A sharp image (top left) with examples of synthetic blur by rotation of the camera about its $Y$ and $Z$-axis, and the kernels and deblurred results for different cases. We compare the results of blind deblurring for two sizes of blur and three noise levels, and the reconstruction errors are summarised in the table at the top. For each single-axis blur, the table contains the root-mean-square (RMS) errors between the deblurred results and the ground-truth sharp image for blurs with a maximum size of 10 or 20 pixels in the image, using our non-uniform model and the uniform model. In each cell we also show, in parentheses, the ratio between the RMS error and the corresponding error for that blurry image deblurred with the ground-truth kernel. Note that to facilitate comparison without the influence of image priors, the deblurred images were all produced using the Richardson-Lucy algorithm.

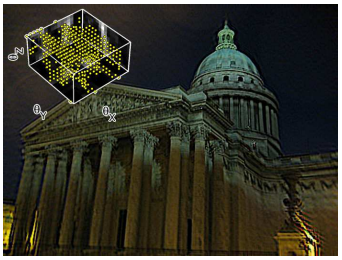


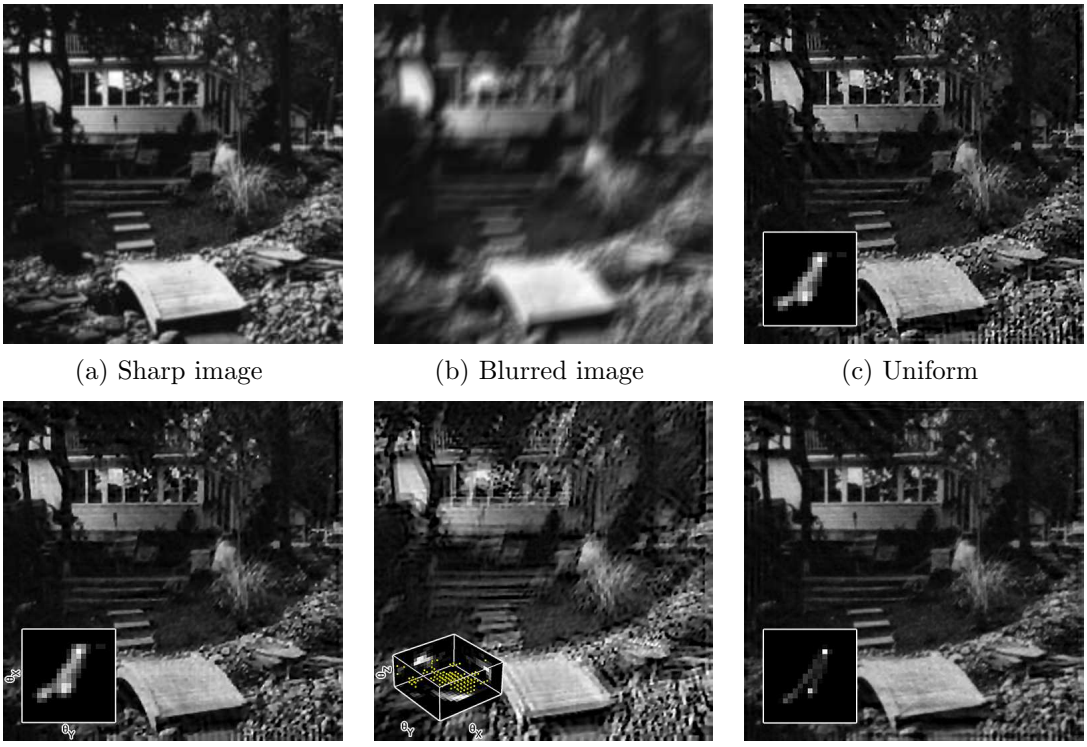
Figure 4.5. **Poor performance of MAP-$\ell_2$ with non-uniform blur model.** The corresponding blurry image can be seen in Figure 4.2. Shown is the estimated kernel and deblurred result when using our non-uniform blur model in the algorithm of Cho and Lee with $\ell_2$ regularisation on the kernel. As can be seen, the $\ell_2$ regularisation is not sufficient to produce a good estimate of the kernel, and results in a deblurred output containing many artefacts.

us to resolve some of the text on the book's cover, the use of our non-uniform blur model provides a clear improvement, and permits almost all of the text to be read.

Figure 4.4 shows results for blind deblurring of synthetically blurred images using the two methods (marginalisation and MAP), and demonstrates two important points: first, small out-of-plane (e.g. $Y$-axis) components of a blur are sufficiently uniform that the two models both perform well, although the rotational model performs better. Second, our approach is the only one capable of removing in-plane ($Z$-axis) blurs, which cannot be represented as convolutions. In this case, and also for the largest out-of-plane blurs, we are able to recover a good sharp image, whereas the uniform approach breaks down due to the blur's non-uniformity. The MAP and marginalisation algorithms exhibit similar performance across the different blur sizes and noise levels, although as demonstrated by the displayed kernels, the MAP-$\ell_1$ approach tends to find sparser, less contiguous kernels than the marginalisation approach.

Figure 4.5 shows the failure of the MAP algorithm to produce a good result (using the blurry image from Figure 4.2 (a)) when using the original $\ell_2$ regularisation proposed by Cho and Lee (2009) with our non-uniform blur model. As discussed in Section 4.3.2, the kernel produced is highly non-sparse despite the thresholding step, and the deconvolved output correspondingly exhibits many artefacts compared to the MAP-$\ell_1$ result in Figure 4.2 (f).

In Figure 4.6, we compare our approach to that of Fergus et al. (2006) on a real,



(a) Sharp image (b) Blurred image (c) Uniform

(d) Our model, with large focal length and $\theta_Z = 0$ (e) Our model, with large focal length (f) Ground-truth kernel

Figure 4.6. **Blind deblurring of a real uniform blur.** A real camera shake blur (a)–(b) from the dataset of Levin et al. (2009), deblurred using kernels estimated with the marginalisation algorithm. We show deblurred results and kernels for four cases; (c) uniform blur using original algorithm of Fergus et al., (d) our model with a large focal length $F$ and no in-plane rotation ($\theta_Z = 0$), (e) our approach with a large focal length $F$ but with $\theta_Z$ unconstrained, and (f) the ground-truth (uniform) kernel, provided with the dataset. Note that (d) is indistinguishable from (c), apart from a translation, and that the kernel in (e), while not perfect, does have the same diagonal shape as the true blur, with the non-zeros centered around a single value of $\theta_Z$.

uniformly blurred image, taken from the dataset of Levin et al. (2009), where the true blur is known, and also known to be uniform. This demonstrates the fact that our model includes uniform blur as a special case; by setting the focal length to be large and applying the constraint that $\theta_Z = 0$, we obtain results indistinguishable from those of Fergus et al. (2006). When we do not apply the constraint on $\theta_Z$, our algorithm still produces a good result, but unsurprisingly does not perform as well, since the number of kernel elements to be estimated is much larger ($K$ is increased by a factor of 8).

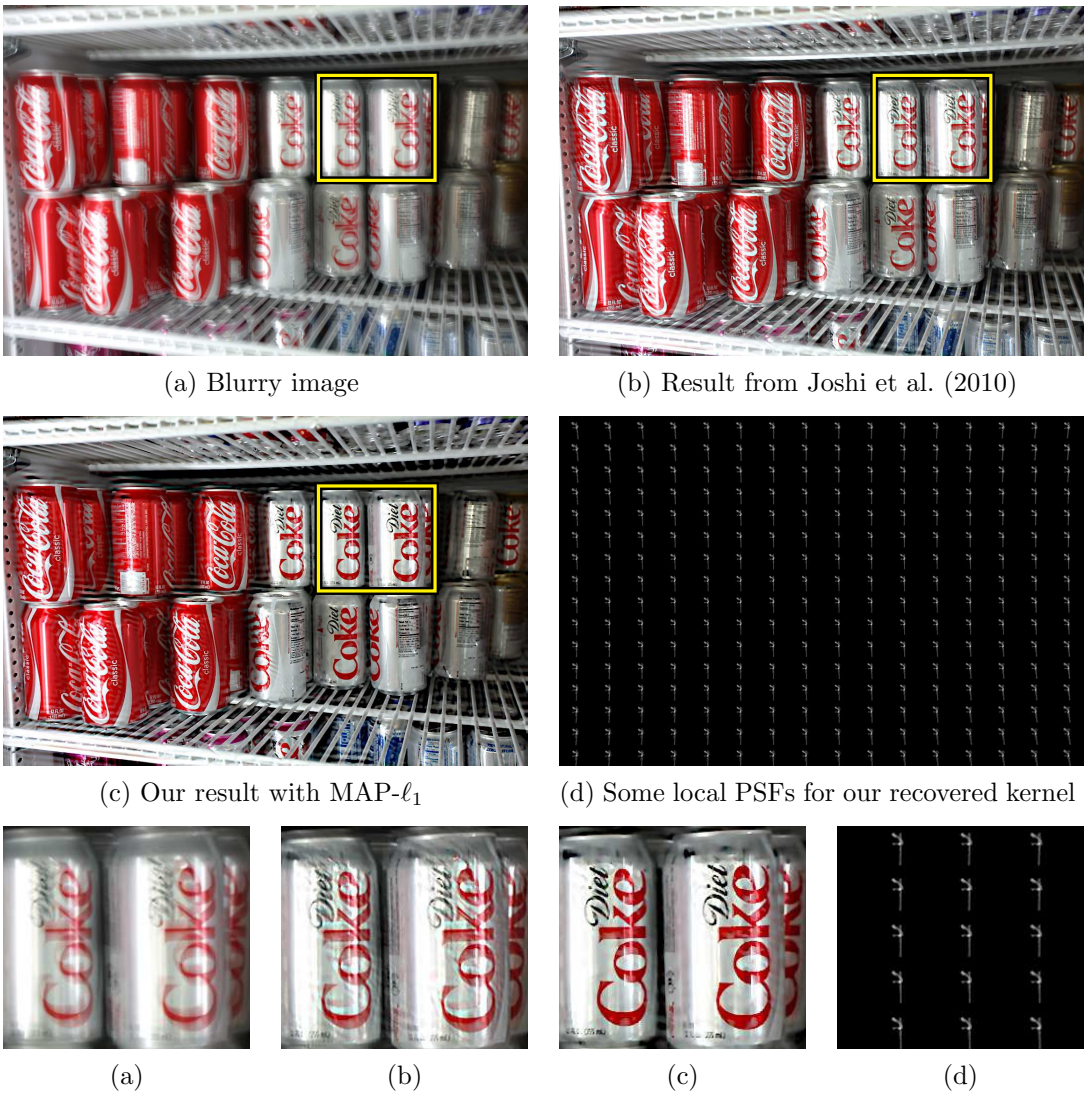
Figure 4.7 shows the result of the non-uniform MAP-$\ell_1$ approach on an image of

Joshi et al. (2010). Although the scene is close to the camera, we are able to obtain a comparable result to that of Joshi et al. without considering the camera's translational motion. This suggests that ignoring the camera's translation is reasonable in practice. Note also that we estimate the kernel directly from the blurry image, whereas Joshi et al. (2010) use additional hardware to record the camera's motion.

### 4.3.4 Discussion

Besides comparing the results of a given algorithm with either a uniform or non-uniform blur model, we can also compare the marginalisation and MAP approaches for a given model. In our experiments, we have observed that the MAP algorithm is generally more robust to the level of contrast in the input image. The parameters of the image prior provided by Fergus et al. (2006) are learnt from a single daytime image of a street scene, so the application of this prior to an image with a very different distribution of intensities (e.g. a photo taken at night) is liable to produce poor results. The MAP algorithm however only relies on the ability to predict step edges from a blurry image, and adapts its threshold for predicting these edges depending on the contrast of the image. On an image containing only low-contrast edges then, such as in Figure 3.1, the marginalisation approach (using the street scene prior) fails to find a useful kernel, while the MAP approach finds a good kernel, as demonstrated by the deblurred result in Figure 3.1 (b).

On the other hand, as discussed by Cho and Lee (2009), the performance of the MAP approach is sensitive to the values of the parameters $\alpha$ and $\beta$, which must be manually specified, while the marginalisation approach has almost no parameters to tune. The problem of setting these parameters automatically merits future study, as it often requires some amount of manual tuning before a good result is achieved. A principled way of setting $\alpha$ and $\beta$ would link their values to the noise variance in the image, however we defer this to future work. In this thesis, $\alpha$ and $\beta$ were set manually, although we have found that $\alpha = 0.0005$ and $\beta = 0.01$ to be generally good values.



(a) Blurry image (b) Result from Joshi et al. (2010)

(c) Our result with MAP-$\ell_1$ (d) Some local PSFs for our recovered kernel

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

Figure 4.7. **Comparison of MAP-$\ell_1$ method with that of Joshi et al. (2010).** A hand-held image with camera shake (a), from Joshi et al. (2010), with the deblurred results from the original work (b) and using the MAP-$\ell_1$ method with our blur model (c). The blur in this image is approximately 35 pixels long, and we obtain a comparable result without the use of additional hardware and without considering the camera's translation during the exposure, despite the scene being close to the camera. To estimate the kernel for this image, we found that was necessary to reduce the threshold on the kernel values (Section 2.3.3) to $\frac{1}{100}$ the maximum value, instead of $\frac{1}{20}$, in order to capture the long thin structure of the non-zeros. Without this, the vertical line of non-zeros was set to zero early in the MAP-$\ell_1$ algorithm and was not subsequently recovered (see Section 4.5.1 for more details).

**Running time.** An important difference between the two approaches is that the MAP algorithm typically takes a much shorter amount of time to run, since the parameter updates for the marginalisation algorithm, in Equations (4.11) to (4.15), are computationally expensive. Due to this expense, and the larger number of iterations required for our model compared to the uniform model (due to the larger number of parameters to estimate), the marginalisation algorithm with our non-uniform model can take several hours to deblur an image of several hundred pixels across under MATLAB on a Linux workstation with an Intel Xeon 2.93GHz CPU and 8GB of memory. Deblurring larger images with this method is not currently practical. Our MATLAB and C implementation of the MAP algorithm, on the other hand, can deblur the same images in under an hour, and megapixel images in 1 – 3 hours, depending on the size of the blur (see Chapter 5 for an approximation which allows us to substantially decrease the blind deblurring time further).

**Limitations and failures.** Since both algorithms (marginalisation and MAP) attempt to solve non-convex minimisation problems, neither can guarantee the ability to arrive at a globally optimal solution. However, in practice we have found them both to be capable of deblurring a wide range of images with large blurs – the blurs removed in this thesis are up to 35 pixels wide (e.g. Figures 1.1 and 6.1) – for both uniform and non-uniform blur. For our model, this corresponds to around $3^\circ$ – $5^\circ$ of rotation around each axis for a photograph whose width and focal length are both 1000 pixels. This is due in large part to the multi-scale approach; by finding a sequence of solutions at increasingly fine resolutions, the large scale structures in the blur kernel and sharp image are resolved before the fine details. In the case of the MAP algorithm, each of the individual minimisations over the sharp image $\mathbf{f}$ and the blur kernel $\mathbf{w}$ is convex, ensuring convergence to a local minimum, even though the overall problem is not jointly convex in both $\mathbf{f}$ and $\mathbf{w}$. The edge prediction step helps direct the optimisation process towards a desirable minimum, by encouraging the sharp image to contain step edges.

Nevertheless, failures do occur under both algorithms, and Figure 4.8 shows some

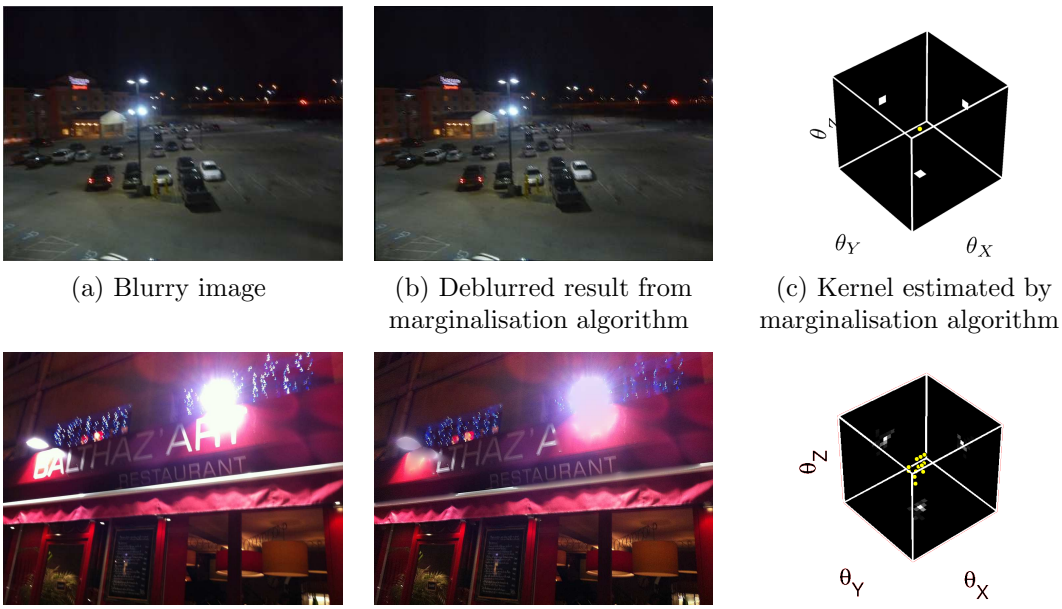
example cases. We have observed failures under both algorithm caused by several factors, including a high level of noise in the input images, an excessively large blur, or an unknown non-linear camera response function e.g. Figure 4.8 (bottom row). Also, both algorithms may fail when the priors and heuristics which guide the blind deblurring optimisations do not match the particular image being estimated e.g. Figure 4.8 (top row). This may happen if the image is too dark, for example, or contains only fine-scale texture with no large-scale step edges. Of the two algorithms, the MAP algorithm is sensitive to the values of its various parameters, and a slight change in these parameters can mean the difference between failure and success when deblurring an image.

Since we have assumed that camera translation has a negligible blurring effect, our model (and the uniform model too) is unlikely to produce good results on images for which this is not true, due to the depth-dependent blur produced. As discussed in Chapter 3, this is unlikely to be a problem on most shaken images, except for close-up photos where the subject is less than about 1 m from the camera.

In general, by monitoring the intermediate images produced by the algorithms during their alternating and coarse-to-fine optimisations, it is possible to see when the blind deblurring process will fail, before it completes. If the current estimate of the latent image does not look sharp at a given point, it is unlikely that the algorithm will recover. On the other hand, a good-looking latent image mid-way through the process does not guarantee success, as both algorithms may diverge from a good solution before completion, without any real predictability.

## 4.4 Deblurring With Noisy / Blurry Image Pairs

As for the single-image deblurring algorithms discussed in the previous sections, we directly substitute our blur model in place of convolution in the original kernel estimation problem of Yuan et al. (2007b) in Equation (2.58). As discussed in Section 2.1.5, some prior knowledge must be applied to recover a good kernel estimate. In their algorithm, Yuan et al. (2007b) constrain the kernel to have non-negative elements and unit $\ell_1$




(a) Blurry image (b) Deblurred result from marginalisation algorithm (c) Kernel estimated by marginalisation algorithm

(d) Blurry image (e) Deblurred result from MAP-$\ell_1$ algorithm (f) Kernel estimated by MAP-$\ell_1$ algorithm

Figure 4.8. **Blind deblurring failures.** Top row: A blurry image (a) with the deblurred result (b) and kernel (c) estimated by the marginalisation algorithm. In this image, the prior for the latent image (which is learnt from a day-time street scene) does not guide the optimisation to a meaningful solution, and instead the algorithm returns a delta as the kernel. See Figure 3.1 (p. 58) for a successful kernel estimation for this image using the MAP-$\ell_1$ algorithm. Bottom row: A blurry image (d) with the deblurred result (e) and kernel (f) estimated by the MAP-$\ell_1$ algorithm. This image contains a large amount of noise, and was captured with a camera-phone whose response function is unknown. As a result, the algorithm fails to estimate a good kernel, and instead returns a kernel which is close to a delta function.

norm, however they simultaneously penalise the $\ell_2$ norm of the kernel, reducing the sparsity-inducing effect of the constraint. To help find a sparse kernel, they propose a thresholding scheme with hysteresis which sets some kernel elements to zero at each iteration. In our approach, we opt to use the $\ell_1$ and positivity constraints alone, since they lead naturally to a sparse kernel (Tibshirani, 1996), a fact also exploited by Shan et al. (2007) for blur kernel estimation.

In order to estimate the blur kernel, we solve the following problem

$$\min_{\mathbf{w}} \left\| \sum_k w_k \mathbf{T}^{(k)} \mathbf{f}_N - \mathbf{g} \right\|_2^2 \quad \text{s.t.} \quad \forall\, k = 1, \ldots, K, \quad w_k \geq 0 \quad \text{and} \quad \sum_k w_k = 1. \quad (4.20)$$

Similar to Equation (4.18), this least-squares formulation with non-negative $\ell_1$ constraints can be solved efficiently (Kim et al., 2007; Mairal et al., 2010). Since the energy function is convex with convex constraints, we can be sure of reaching the global minimum.

For comparison, we have also implemented this algorithm for uniform blurs, using convolution in Equation (4.20).

To perform the non-blind deblurring using the estimated kernel, we apply the augmented RL algorithm of Yuan et al. (2007b) directly, replacing convolution with our spatially-variant blur model.

### 4.4.1 Geometric and Photometric Registration

For the case of noisy / blurry image pairs, the two images are simply taken one after the other with a hand-held camera, so they may not be registered with each other. Thus, we estimate an approximate registration $\boldsymbol{\theta}_0$ between them at the coarsest scale, using an exhaustive search over a large set of rotations, for example $\pm 10^\circ$ about all 3 axes using the same step size as for the blur kernel, and we remove this mis-registration from the noisy image. When applying the uniform blur model in this case, we manually estimate the in-plane rotation to best register the two images, as in (Yuan et al., 2007b).

To compensate for the difference in exposure between the noisy and blurry images, at each scale $s$, after computing $\hat{\mathbf{w}}_s$ for that scale, we estimate a linear rescaling $a$ by computing the linear least-squares fit between the pixels of $\mathbf{g}_s$ and those of $\hat{\mathbf{g}}_s(\hat{\mathbf{w}}_s, \mathbf{f}_{N,s})$, and apply this to the noisy image, i.e. $\mathbf{f}_N \leftarrow a\mathbf{f}_N$.

### 4.4.2 Results and Discussion

In this section, we present results with noisy / blurry image pairs, and refer the reader to Section 4.5 for implementation details. Figures 4.9 and 4.10 show a comparison between the uniform model and ours, using the algorithm described above to estimate the blur kernels. Having estimated the kernel, we deblur the blurred images using the augmented RL algorithm of Yuan et al. (2007b). As can be seen from the deblurred images obtained

with the two models, our results exhibit more detail and fewer artefacts than those using the uniform blur model.

Blind deblurring with the aid of a noisy image is significantly more robust than single image deblurring. This is because it involves solving two convex optimisation problems, such that we can be sure that we have reached the globally-optimal deblurred image. The solutions are also not particularly sensitive to the values of regularisation parameters, since the problems are well-conditioned, with many more observations than unknowns. However, this approach is only applicable when a noisy image $\mathbf{f}_N$ is available, which precludes most existing blurry images. As a method of capturing a good image with a standard camera in low-light however, the noisy / blurry image pair is a good technique for photographers to keep in mind.

## 4.5 Implementation

The implementation of the variational kernel estimation method presented in Section 4.3.1 is based on the code made available by Miskin and MacKay (2000) and by Fergus et al. (2006)[1]. We have modified the algorithm to use our blur model and replaced the parameter update equations with the corresponding versions derived for our bilinear blur model in Equations (4.11) to (4.15). A package containing our code is available online[2]. The implementation of the non-blind deblurring algorithm of Krishnan and Fergus (2009) is also based on MATLAB code made available online by the authors[3]. The implementations of the Richardson-Lucy algorithm, the algorithm of Cho and Lee (2009), and the augmented RL algorithm of Yuan et al. (2007b) are our own, and we use these implementations for both uniform and non-uniform blur models when comparing results. A binary executable for Cho and Lee's algorithm is available, however we did not observe an improvement in the results obtained, and thus use our own implementation to permit a fairer comparison between the results from the uniform and non-uniform blur models.

[1] `http://cs.nyu.edu/~fergus/research/deblur.html`
[2] `http://www.di.ens.fr/willow/research/deblurring/`
[3] `http://cs.nyu.edu/~dilip/research/fast-deconvolution/`

### 4.5.1 Multiscale Implementation

All of the kernel estimation algorithms presented here are applied within a multiscale framework, starting with a coarse representation of image and kernel, and repeatedly refining the estimated kernel at higher resolutions. In the case of single-image deblurring, this is essential to avoid poor local minima, however it is also important for computational reasons in both the single-image and noisy / blurry image pair cases. The kernel at the original image resolution may have thousands or tens of thousands of elements, however very few of these should have non-zero values. For example to solve Equation (4.20) directly at full resolution would involve transforming $\mathbf{f}_N$ for every possible rotation under consideration and storing all the copies simultaneously in a matrix $\mathbf{B}_N$. For a blur of size 10 pixels in an image 1000 pixels wide, this could be over 1000 copies of $\mathbf{f}_N$. This represents a significant amount of redundant computation, since most of these copies will correspond to zeros in the kernel, and furthermore $\mathbf{B}_N$ may have too many columns to fit in the computer's memory. The effect on the computation and memory requirements for single-image deblurring is comparable.

Thus, in all of the applications presented in this chapter, which estimate the kernel iteratively, we use our current estimate of the kernel $\hat{\mathbf{w}}_s$ at a scale $s$ to constrain our estimate at the next iteration. To do this, we define an "active region" where $\hat{\mathbf{w}}_s$ is non-zero, and constrain the non-zeros at the next iteration to lie within this region. By fixing many kernel elements to zero, we eliminate a large amount of computation and memory requirements associated with estimating those elements' values. In the example of the previous paragraph, this corresponds to discarding many columns of $\mathbf{B}_N$, clearly reducing the size of the problem. We first build Gaussian pyramids for the blurred image (and noisy image, if applicable), and at the coarsest scale $s = 0$, define the active region to cover the full kernel. At each iteration, we find the non-zero elements of our current estimate of the kernel $\hat{\mathbf{w}}_s$, and dilate this region using a $3 \times 3 \times 3$ cube to define the active region for the next iteration. When moving from one scale $s$ to the next scale $s + 1$, we upsample $\hat{\mathbf{w}}_s$ using bilinear interpolation, find the non-zero elements of this

upsampled kernel, and as before, dilate this region using a $3 \times 3 \times 3$ cube. This initialises the active region for our next estimate $\hat{\mathbf{w}}_{s+1}$. We repeat this process at each scale, until we have found the optimal kernel at the finest scale.

This approach is generally effective at reducing the computational burden of the kernel estimation without reducing accuracy, however, problems may occur if the blur kernel contains long faint structures, as it is possible for these to be clamped to zero at a coarse scale and never to be recovered, as mentioned in Figure 4.7.

## 4.6 Conclusion

We have applied our new model for spatially-variant camera shake blur within the frameworks of several existing camera shake removal algorithms. We have validated the model with experiments on real and synthetic data, demonstrating superior results compared to the uniform blur model.

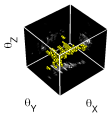
(a) Noisy image (b) Blurry image (c) Deblurred result, uniform (d) Deblurred result, non-uniform

Figure 4.9. **Deblurring real camera shake blur using a noisy / blurry image pair.** A noisy image (a) and a blurry image (b) captured with a hand-held camera, with the deblurred results for the uniform (c) and non-uniform (d) blur models. As can be seen in the close-ups (bottom two rows), our result contains more details and fewer artefacts than when using the uniform blur model, and reveals features not visible in either the noisy or the blurry image. The non-uniform kernel in (d) covers $\pm 3^\circ$ along each dimension.

(a) Noisy image

(b) Blurry image

(c) Uniform kernel



(d) Non-uniform kernel

(e) Deblurred result, uniform

(f) Deblurred result, non-uniform

(g) Noisy

(h) Blurry

(i) Uniform

(j) Non-uniform

Figure 4.10. **Deblurring real camera shake blur using a noisy / blurry image pair.** A noisy image (a) and blurry image (b) captured with a hand-held camera, shown with the estimated kernels (c)–(d) and deblurred images (e)–(f) for the uniform and non-uniform blur models. Note in the close-up that the result using our model (j) has sharper edges and fewer artefacts than that using the uniform model (i). The non-uniform kernel in (d) covers $\pm 3^\circ$ along each dimension .

# Chapter 5

# Efficient Computation of the Spatially-Variant Blur Model

## 5.1 Introduction

Due to the additional computational expense incurred by using a spatially-variant blur model instead of a spatially-invariant one, both blind and non-blind deblurring under this model can be very time consuming. The number $K$ of homographies $\{\mathbf{T}^{(\mathbf{k})}\}$ in the forward model in Equation (3.11) can be large, and we must compute hundreds, or thousands of projective image transformations at each iteration of blind and non-blind deblurring algorithms. To reduce the running time of the whole deblurring process, in this chapter we propose an efficient approximation to the blur model in Equation (3.11), based on the locally-uniform "Efficient Filter Flow" proposed by Hirsch et al. (2010), introduced in Section 2.4.

In Section 5.2 we examine some of the bottlenecks in the spatially-variant blind deblurring process, before introducing an efficient approximation in Section 5.3 to the model proposed in Chapter 3. We compare results and deblurring times using the approximation, to those using the exact model, and conclude in Section 5.4.

## 5.2 Bottlenecks in Spatially-Variant Blind Deblurring

Although the model for a spatially-variant PSF introduced in Chapter 3 is parameterised by the vector $\mathbf{w}$, which may have a similar number of elements to a traditional convolution kernel, the computational cost of using this model in blind deblurring is substantially higher. This is due to several factors, for example that the transformations $\{\mathbf{T}^{(\mathbf{k})}\}$ involve sub-pixel interpolations, that the set of matrices $\{\mathbf{T}^{(\mathbf{k})}\}$ is too large to fit in memory and must be re-computed on-the-fly frequently, and that there is no equivalent to the frequency-domain convolution which allows spatially-invariant blur (2D convolution) to be computed in $\mathcal{O}\big(N \log N\big)$ time instead of $\mathcal{O}\big(N^2\big)$, where $N$ is the number of pixels in the image. These factors mean that blind PSF estimation using this model is significantly slower than for spatially-invariant blur. In this section we discuss the cost of the most computationally-intensive steps in the MAP blind PSF estimation algorithm. Recall from Chapter 4 that the MAP algorithm has two major steps: updating the blur kernel and updating the sharp image; we discuss each of these in turn.

### 5.2.1 Updating the kernel

To update the kernel in the MAP-$\ell_1$ blind deblurring algorithm, recall from Chapter 4 that we solve Equation (4.18) (p. 77). For ease of exposition, we note that the problem can be written in the general form

$$\min_{\mathbf{w}} \left\|\mathbf{P}\mathbf{w} - \mathbf{g}^{\partial}\right\|_2^2 + \beta \sum_k w_k \quad \text{s.t.} \quad \forall k = 1, \ldots, K, \quad w_k \geq 0, \tag{5.1}$$

where $K$ is the number of non-zero weights in the kernel, the vector $\mathbf{g}^{\partial}$ is the concatenation of all the derivatives of $\mathbf{g}$, and the matrix $\mathbf{P}$ incorporates all the transformed versions of the predicted sparse derivative maps $\{\mathbf{p}^{(\mathbf{q})}\}$ (as discussed in Section 2.3.3

(p. 39)). The matrix $\mathbf{P}$ has the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{T}^{(1)}\mathbf{p}^{(1)} & \mathbf{T}^{(2)}\mathbf{p}^{(1)} & \dots & \mathbf{T}^{(K-1)}\mathbf{p}^{(1)} & \mathbf{T}^{(K)}\mathbf{p}^{(1)} \\ \mathbf{T}^{(1)}\mathbf{p}^{(2)} & \mathbf{T}^{(2)}\mathbf{p}^{(2)} & \dots & \mathbf{T}^{(K-1)}\mathbf{p}^{(2)} & \mathbf{T}^{(K)}\mathbf{p}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \tag{5.2}$$

where in the algorithm of Cho and Lee (2009), $\mathbf{p}^{(1)} = \mathbf{p}^x$, $\mathbf{p}^{(2)} = \mathbf{p}^y$, and $\mathbf{p}^{(q)}$ for $3 \leq q \leq 5$ contain second-order derivatives.

In iterative minimisation algorithms used for solving such least-squares problems, e.g. LARS-lasso (Efron et al., 2004) or conjugate-gradient descent (Shewchuk, 1994), we must repeatedly compute matrix-vectors products such as $\mathbf{Px}$, $\mathbf{P}^\top\mathbf{Px}$ and $\mathbf{P}^\top\mathbf{y}$ for arbitrary vectors $\mathbf{x}$ and $\mathbf{y}$. The $5N \times K$ matrix $\mathbf{P}$ is generally too large to fit into memory (e.g. for a megapixel image $P$ could have size $5 \cdot 10^6 \times 10^3$ elements, requiring 37 GB of memory at double precision), so we must generate its elements on-the-fly when computing these matrix-vector products. Computing each element of $\mathbf{P}$ is essentially equivalent to performing a sub-pixel interpolation, which is computationally expensive. As a result, these matrix-vector products are by far the biggest bottleneck in the kernel estimation step, since they might involve computing a large number of image warps and multiplications at each iteration.

To solve Equation (5.1), we use the LARS-lasso algorithm of Efron et al. (2004), since it involves relatively few products of the type described above per iteration, and is guaranteed to reach the global minimum in a finite number of steps. To perform a single iteration of this algorithm, we have the choice of computing three $\mathbf{Px}$-type products and one $\mathbf{P}^\top\mathbf{y}$-type product, or computing one $\mathbf{P}^\top\mathbf{Px}$-type product. Here, note that since $K << N$, it is generally feasible to store the $K \times K$ matrix $\mathbf{P}^\top\mathbf{P}$, referred to as the *Gram matrix*, in memory. This matrix can be computed without requiring any element of $\mathbf{P}$ more than once, using the fact that $\mathbf{P}^\top\mathbf{P} = \sum_i \mathbf{p}_i\mathbf{p}_i^\top$, where $\mathbf{p}_i^\top$ is the $i^{\text{th}}$ row of $\mathbf{P}$. Thus we can perform the entire LARS-lasso kernel estimation process while computing each element of $\mathbf{P}$ only once. With $\mathbf{P}^\top\mathbf{P}$ stored in memory, the $\mathbf{P}^\top\mathbf{Px}$ product at each iteration can be performed with optimised linear algebra libraries. We have found

empirically that this approach (pre-computing $\mathbf{P}^\top\mathbf{P}$, as opposed to computing the $\mathbf{Px}$ and $\mathbf{P}^\top\mathbf{y}$ products on-the-fly) gives the shortest kernel estimation times. However, the pre-computation of $\mathbf{P}^\top\mathbf{P}$ may still by time-consuming. For megapixel images, or images with many non-zeros in $\{\mathbf{p}^{(\mathbf{q})}\}$, the pre-computation may take minutes, or even hours.

### 5.2.2 Updating the sharp image

To update the sharp image in the MAP-$\ell_1$ blind deblurring algorithm, we use conjugate gradient descent (CG) to solve the minimisation problem in Equation (4.17) (p. 75), which essentially has the form

$$\min_{\mathbf{f}} \ \|\mathbf{Af} - \mathbf{g}\|_2^2 + \alpha \|\mathbf{d}^x * \mathbf{f}\|_2^2 + \alpha \|\mathbf{d}^y * \mathbf{f}\|_2^2 , \tag{5.3}$$

where $\mathbf{A} = \sum_k w_k \mathbf{T}_k$. To compute the gradient of the cost function, we need to evaluate products of the type $\mathbf{A}^\top\mathbf{Af}$ at each iteration. As in the kernel estimation step, the image transformations $\mathbf{T}^{(k)}\mathbf{f}$ involved in computing these products are the most computationally-expensive operations. Unlike for the kernel estimation step, the $N \times N$ Gram matrix $\mathbf{A}^\top\mathbf{A}$ is too large to store, so cannot be pre-computed. However, we typically perform a small, fixed number of iterations of CG (e.g. 20 iterations), and as such the time spent performing this optimisation is typically much less than the kernel update step. Nevertheless, this CG minimisation may still take several minutes for a megapixel image, as shown in Figure 5.3 (b).

## 5.3 Locally-Uniform Approximation

In this section we describe an approximation to the forward model described in Chapter 3 that provides a fast means of calculating the spatially-variant forward model. More importantly however, we describe how this approximation can be used to (a) compute the kernel Gram matrix $\mathbf{P}^\top\mathbf{P}$ very quickly, allowing the kernel update step to be performed in seconds rather than hours, and (b) approximately minimise Equation (5.3) directly in two steps, allowing the sharp image to be updated in seconds rather than minutes.

This approximation is based on the "Efficient Filter Flow" model for spatially-variant blur proposed by Hirsch et al. (2010), discussed earlier in Section 2.4.2.

Hirsch et al. (2010) observe that in some cases of spatially-variant image blur, the blur may vary slowly and smoothly across the image. As a result, the PSFs of nearby pixels may be very similar, and it is reasonable to approximate spatially-variant blur as being locally-uniform. Following this observation, they propose an approximation of the forward model, whereby the sharp image $\mathbf{f}$ is covered with a coarse grid of $P$ overlapping patches, each of which is modelled as having a spatially-invariant blur. The overlap between patches ensures that the blur varies smoothly across the image, rather than changing abruptly at the boundary between two patches. The fact that each patch has a spatially-invariant blur allows the forward model to be computed using $P$ small convolutions. Hirsch et al. (2010) assign each patch $r$ a spatially-invariant blur filter $\mathbf{a}^{(r)}$, and the forward model is given by:

$$\mathbf{g}^* = \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^\top \left( \mathbf{a}^{(r)} * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}) \right), \tag{5.4}$$

where $\mathbf{C}^{(r)}$ is a matrix that crops out the $r^{\text{th}}$ patch from the image $\mathbf{f}$ (and thus ${\mathbf{C}^{(r)}}^\top$ reinserts it at its correct location). The vector $\mathbf{m}$ is a windowing function, e.g. the Bartlett-Hann window, which produces the smooth transition between neighbouring patches, and $\cdot \circ \cdot$ represents the Hadamard (element-wise) product between two vectors. This model can be computed efficiently in the frequency domain:

$$\mathbf{g}^* = \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^\top \mathcal{F}^{-1} \left( \mathcal{F}\big(\mathbf{a}^{(r)}\big) \circ \mathcal{F}\big(\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}\big) \right), \tag{5.5}$$

where $\mathcal{F}(\cdot)$ takes the 2D discrete Fourier transform (computed using the fast Fourier transform), and $\mathcal{F}^{-1}(\cdot)$ the inverse Fourier transform. The windowing function $\mathbf{m}$ is chosen such that the total weight at each pixel is 1, i.e. $\sum_r {\mathbf{C}^{(r)}}^\top \mathbf{m} = \mathbf{1}$.

### 5.3.1 A Globally-Consistent Approximation for Camera Shake

In their original work, Hirsch et al. (2010) estimate a separate filter $\mathbf{a}^{(r)}$ for each patch $r$. Likewise Harmeling et al. (2010a), who apply this approximation in the context of single-image camera shake removal, also estimate a separate filter per patch, using the MAP algorithm of Cho and Lee (2009). One weakness of this approach is that in regions with little texture or strong edges, it may not be possible to estimate a good kernel. Since the purely-local model has no notion of global consistency between the different regions' kernels, heuristics are needed to encourage neighbouring filters to be similar, and to detect failed regions and hypothesise kernels for those regions, based on their neighbours. However, given the forward blur model for camera shake in Equation (3.11), which is parameterised by a single set of weights $\mathbf{w}$, we can in fact write each $\mathbf{a}^{(r)}$ in terms of $\mathbf{w}$. For each patch $r$, we choose $\mathbf{a}^{(r)}$ to be the point spread function for the central pixel $i_r$, which is given by the $i_r^{\text{th}}$ row of $\mathbf{A}$. Since $\mathbf{A}$ is linear in $\mathbf{w}$, we can construct a matrix $\mathbf{J}^{(r)}$ such that $\mathbf{a}^{(r)} = \mathbf{C}^{(r)}\mathbf{J}^{(r)}\mathbf{w}$. The elements of each matrix $\mathbf{J}^{(r)}$ are simply a re-arrangement of the elements of the matrices $\mathbf{T}^{(k)}$: $J^{(r)}_{jk} = T^{(k)}_{i_r j}$. Figure 5.1 shows how the quality of the approximation varies with the number of patches being used, compared to the exact model. In all our experiments, we use a grid of $6 \times 8$ patches.

Having written each filter $\mathbf{a}^{(r)}$ in terms of $\mathbf{w}$, we can then substitute this into Equation (5.5) to obtain the following approximation of the forward model in Equation (3.11):

$$\mathbf{g}^* \simeq \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^\top \mathcal{F}^{-1}\left(\mathcal{F}\big(\mathbf{C}^{(r)}\mathbf{J}^{(r)}\mathbf{w}\big) \circ \mathcal{F}\big(\mathbf{m} \circ \mathbf{C}^{(r)}\mathbf{f}\big)\right). \tag{5.6}$$

These equations allow the forward model to be computed quickly using only a handful of frequency-domain convolutions. Furthermore, the derivatives of $\mathbf{g}^*$ with respect to $\mathbf{f}$ and $\mathbf{w}$ can also be computed using a small number of frequency-domain convolutions and correlations. As discussed in the previous section, computing the forward model and its derivatives are bottlenecks in both the blind PSF estimation algorithm of Cho and Lee (2009).

Equation (5.6) provides us with a fast approximate way of computing the forward

model for $\mathbf{g}^* = \mathbf{A}\mathbf{f} = \mathbf{B}\mathbf{w}$. By re-arranging this approximation, we can compute $\mathbf{B}^\top\mathbf{y}$ for arbitrary $\mathbf{y}$ and $\mathbf{B}^\top\mathbf{B}$ using 3 FFTs per patch as follows:

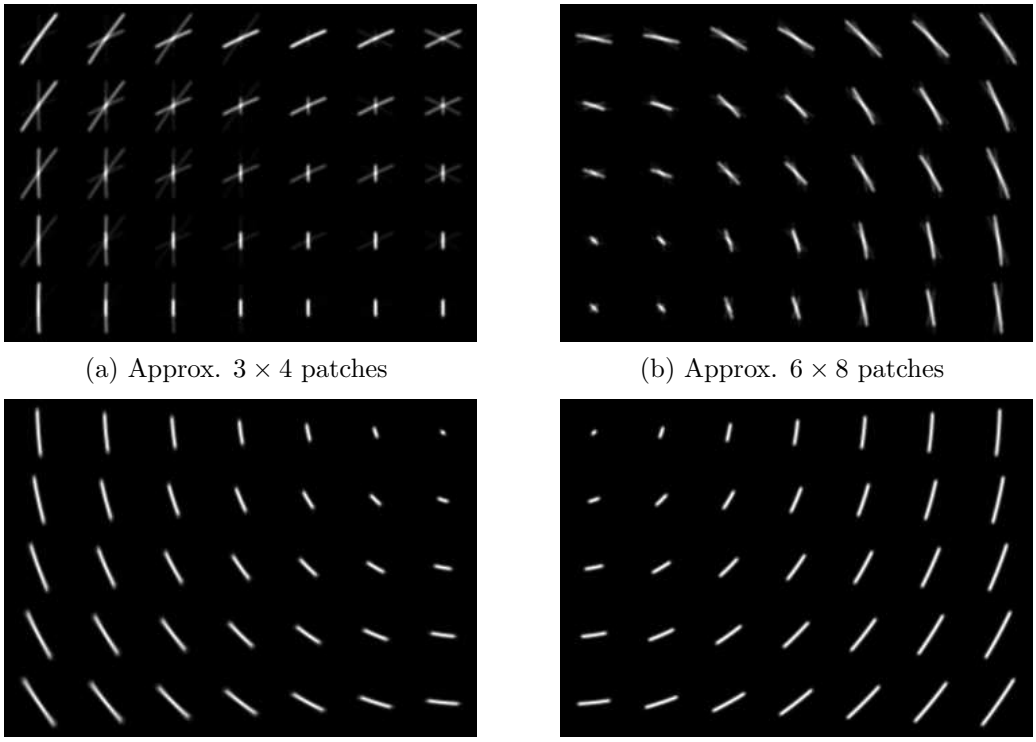
$$\mathbf{P}^\top\mathbf{y} \simeq \sum_{r=1}^{P} {\mathbf{J}^{(r)}}^\top {\mathbf{C}^{(r)}}^\top \mathcal{F}^{-1}\Big(\mathcal{F}\big(\mathbf{m} \circ \mathbf{C}^{(r)}\mathbf{f}\big)^* \circ \mathcal{F}\big(\mathbf{C}^{(r)}\mathbf{y}\big)\Big) \tag{5.7}$$

$$\mathbf{P}^\top\mathbf{P} \simeq \sum_{r=1}^{P} {\mathbf{J}^{(r)}}^\top {\mathbf{C}^{(r)}}^\top \text{XCorrMatrix}\Big(\mathcal{F}^{-1}\Big(\mathcal{F}\big(\mathbf{m} \circ \mathbf{C}^{(r)}\mathbf{f}\big)^* \circ \mathcal{F}\big(\mathbf{C}^{(r)}\mathbf{f}\big)\Big)\Big)\mathbf{C}^{(r)}\mathbf{J}^{(r)}. \tag{5.8}$$

The function XCorrMatrix constructs the full cross-correlation matrix from a cross-correlation vector between two signals. To clarify, consider a pair of 1D signals $\mathbf{u}$ and $\mathbf{v}$. We can compute the cross-correlation vector $\mathbf{c}$ for all relative offsets between them as $\mathbf{c} = \mathcal{F}^{-1}\big(\mathcal{F}\left(\mathbf{u}\right)^* \circ \mathcal{F}\left(\mathbf{v}\right)\big)$. If we define the matrix $\mathbf{S}_k$ which shifts a signal by $k$ elements, then the element $c_k$ stores the cross-correlation between $\mathbf{S}_k\mathbf{u}$ and $\mathbf{v}$, i.e. $c_k = (\mathbf{S}_k\mathbf{u})^\top\mathbf{v}$. With appropriate boundary conditions, we also have $c_k = (\mathbf{S}_{k-1}\mathbf{u})^\top(\mathbf{S}_1\mathbf{v}) = (\mathbf{S}_{k-2}\mathbf{u})^\top(\mathbf{S}_2\mathbf{v})$, and so on. Thus to compute the cross-correlation matrix $\mathbf{C}$ with elements $C_{k_1k_2} = (\mathbf{S}_{k_1}\mathbf{u})^\top(\mathbf{S}_{k_2}\mathbf{v})$, we need only re-arrange and duplicate the elements of the vector $\mathbf{c}$. This argument extends naturally to 2D, and this action is denoted by $\mathbf{C} = \text{XCorrMatrix}(\mathbf{c})$.

Using the approximation described above, computation of $\mathbf{P}^\top\mathbf{P}$ costs $\mathcal{O}\big(K+N\log(N)\big)$. This is in comparison to the cost of $\mathcal{O}\big(NK^2\big)$ to compute $\mathbf{P}^\top\mathbf{P}$ using the exact model. For camera shake, the blur kernel tends to consist of a line of non-zeros, so that the number of kernel elements $K$ grows (at best) linearly with the length of the blur. Since the length of the blur will grow linearly with the image width $W$, we can assert that $K$ grows as than $\sqrt{N}$. Thus the cost of computing $\mathbf{P}^\top\mathbf{P}$ with the exact model grows at least as $\mathcal{O}\big(N^2\big)$, while the approximate model grows as $\mathcal{O}\big(N\log N\big)$.

Figure 5.2 plots the blind PSF estimation times for several images with different sizes of blur, using both the exact forward model and the fast approximate forward model.



(a) Approx. $3 \times 4$ patches (b) Approx. $6 \times 8$ patches

(c) Approx. $12 \times 16$ patches (d) Exact

Figure 5.1. **Approximating spatially-variant blur by combining uniformly-blurred, overlapping patches.** Using the model described in Section 5.3.1, we can efficiently compute approximations to the spatially-variant blur model in Equation (3.11). With a small number of patches (a), the PSF at each pixel is visibly the sum of different blurs from overlapping patches. As more patches are used (b–c), the approximation becomes increasingly close to the exact model (d) – at $12 \times 16$ patches it is almost indistinguishable.

### 5.3.2 Fast Independent Non-Blind Deconvolution of Patches

Although the use of a locally-uniform approximation allows the forward model to be computed much faster using the FFT, we cannot immediately take full advantage of working in the frequency domain when it comes to non-blind deconvolution. For example, in the inner loop of the blind deblurring algorithm of Cho and Lee (2009), we must repeatedly solve problems of the form in Equation (5.3). When the blur is spatially-invariant, $\mathbf{Af} = \mathbf{a} * \mathbf{f}$, and Equation (5.3) can be minimised directly in a single-step by pixel-wise division in the frequency domain, as discussed in Section 2.2.2. However,

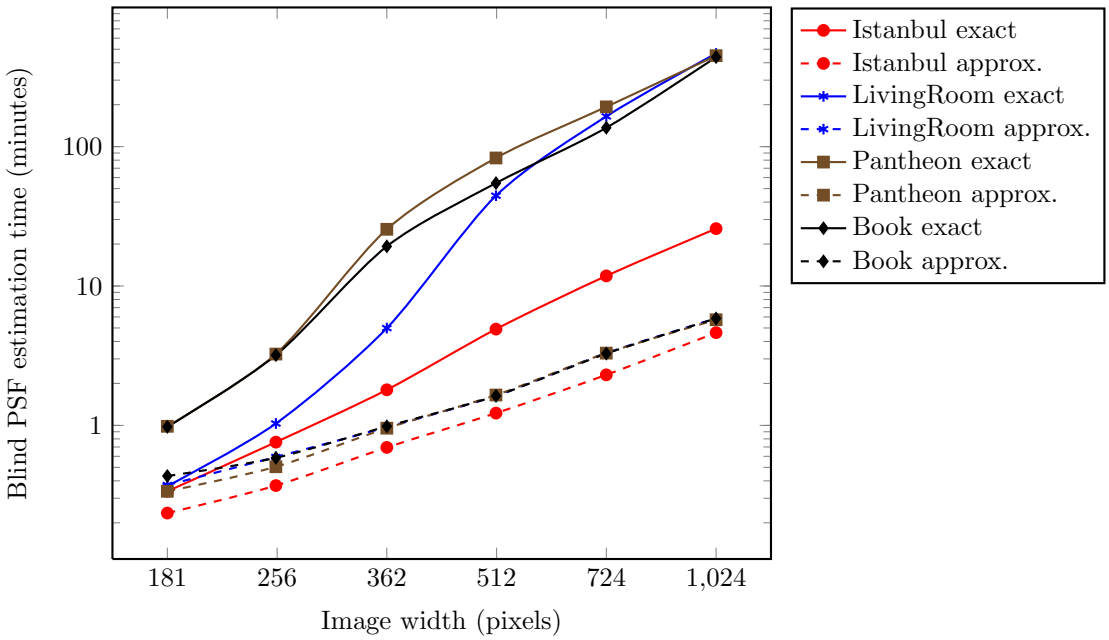



Figure 5.2. **Blind deblurring time using the exact and approximate model.** This log-log plot shows the time taken to perform blind deblurring of several images at various different sizes, using either the exact forward model (solid lines) or the approximate forward model (dashed lines). The blind deblurring algorithm is implemented in MATLAB, and the function to compute the exact forward model is implemented in compiled C, while the approximate model is implemented in MATLAB. The blind deblurring time using the exact model grows significantly faster with respect to the image size than using the approximate model, and for images 1024 pixels wide already takes almost 2 orders of magnitude longer. The images for these times are from the following figures: Istanbul – Figure 6.8 (p. 129), LivingRoom – Figure 5.3 (p. 108), Pantheon – Figure 4.2 (p. 80), Book – Figure 4.3 (p. 81).

for spatially-variant blur, the locally-uniform approximation does not permit this. Even though each patch has a spatially-invariant blur, the fact that the patches overlap means that they cannot be deconvolved independently. This can be seen by expanding the non-blind deblurring problem in Equation (5.3), where the sum over patches lies inside the data fidelity term of the cost function:

$$\min_{\mathbf{f}} \left\| \sum_{r=1}^{P} \mathbf{C}^{(r)^\top} \left( \mathbf{a}^{(r)} * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}) \right) - \mathbf{g} \right\|_2^2 + \alpha \left\| \mathbf{d}^x * \mathbf{f} \right\|_2^2 + \alpha \left\| \mathbf{d}^y * \mathbf{f} \right\|_2^2 . \tag{5.9}$$

If we note however that extracting and recombining the patches of an image has no

effect on an image, or on a filtering operation (Equation (2.64)), i.e.

$$\mathbf{g} = \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^{\top} (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{g}) \tag{5.10}$$

$$\mathbf{d}^x * \mathbf{f} = \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^{\top} (\mathbf{d}^x * \big(\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}\big)), \tag{5.11}$$

we can rewrite Equation (5.9) as

$$\begin{aligned} \min_{\mathbf{f}} \; & \Big\| \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^{\top} \big(\mathbf{a}^{(r)} * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}) - \mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{g}\big) \Big\|_2^2 \\ & + \alpha \Big\| \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^{\top} \big(\mathbf{d}^x * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f})\big) \Big\|_2^2 + \alpha \Big\| \sum_{r=1}^{P} {\mathbf{C}^{(r)}}^{\top} \big(\mathbf{d}^y * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f})\big) \Big\|_2^2. \end{aligned} \tag{5.12}$$

Although this cost function still does not permit a closed-form solution, we can obtain the following upper bound (up to a constant scale factor of $P$) using Jensen's inequality (Bishop, 2006):

$$\begin{aligned} \min_{\mathbf{f}} \; & \sum_{r=1}^{P} \Big\| {\mathbf{C}^{(r)}}^{\top} \big(\mathbf{a}^{(r)} * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f}) - \mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{g}\big) \Big\|_2^2 \\ & + \alpha \sum_{r=1}^{P} \Big\| {\mathbf{C}^{(r)}}^{\top} \big(\mathbf{d}^x * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f})\big) \Big\|_2^2 + \alpha \sum_{r=1}^{P} \Big\| {\mathbf{C}^{(r)}}^{\top} \big(\mathbf{d}^y * (\mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{f})\big) \Big\|_2^2. \end{aligned} \tag{5.13}$$

This formulation penalises each patch independently, and can be minimised using a split-variable approach, iterating between (a) estimating a deblurred patch $\hat{\mathbf{f}}^{(r)}$ for each patch $r$, and (b) estimating the complete deblurred image $\hat{\mathbf{f}}$ that best matches these patches. In practice a single iteration is sufficient to obtain a good result. As we show in the following, step (a) can be performed independently for each patch, and step (b) can be performed independently for each pixel of the deblurred image.

Specifically, we define the set of blurry patches $\mathbf{g}^{(r)}$ such that

$$\mathbf{g}^{(r)} = \mathbf{m} \circ \mathbf{C}^{(r)} \mathbf{g}, \tag{5.14}$$

and note that the multiplication by $\mathbf{C}^{(r)^\top}$ at the beginning of each term in Equation (5.13) only zero-pads an image, so has no effect on the cost. We first estimate a set of deblurred patches $\{\hat{\mathbf{f}}^{(\mathbf{r})}\}$ by solving

$$\min_{\mathbf{f}^{(r)}} \left\|\mathbf{a}^{(r)} * \mathbf{f}^{(r)} - \mathbf{g}^{(r)}\right\|_2^2 + \alpha\left\|\mathbf{d}^x * \mathbf{f}^{(r)}\right\|_2^2 + \alpha\left\|\mathbf{d}^y * \mathbf{f}^{(r)}\right\|_2^2 \tag{5.15}$$

for each patch. This can be done using the direct method of Equation (2.44). Secondly, to estimate the full deblurred image that best matches the deblurred patches $\hat{\mathbf{f}}^{(r)}$, we solve

$$\min_{\mathbf{f}} \sum_{r=1}^{P} \left\|\mathbf{m} \circ \mathbf{C}^{(r)}\mathbf{f} - \hat{\mathbf{f}}^{(r)}\right\|_2^2. \tag{5.16}$$

To minimise this, we first note that we can multiply through by $\mathbf{C}^{(r)^\top}$ without altering the cost (since $\mathbf{C}^{(r)^\top}$ simply applies zero-padding), and that $\mathbf{C}^{(r)^\top}(\mathbf{m} \circ \mathbf{C}^{(r)}\mathbf{f}) = (\mathbf{C}^{(r)^\top}\mathbf{m}) \circ \mathbf{f}$. Thus we obtain the following equivalent problem:

$$\min_{\mathbf{f}} \sum_{r=1}^{P} \left\|(\mathbf{C}^{(r)^\top}\mathbf{m}) \circ \mathbf{f} - \mathbf{C}^{(r)^\top}\hat{\mathbf{f}}^{(r)}\right\|_2^2. \tag{5.17}$$

This problem can be solved independently for each pixel of $\hat{\mathbf{f}}$, yielding the following solution:

$$\hat{\mathbf{f}} = \frac{\sum_r \mathbf{C}^{(r)^\top}(\mathbf{m} \circ \hat{\mathbf{f}}^{(r)})}{\sum_r \mathbf{C}^{(r)^\top}(\mathbf{m} \circ \mathbf{m})}. \tag{5.18}$$

Algorithm 1 summarizes the steps to performing this fast direct deconvolution, and in Figure 5.3 we compare this method of non-blind deblurring to other possibilities for solving Equation (5.3). The fast independent method produces results visually very similar to the exact model, in a significantly shorter amount of time. We use this direct deconvolution method to update the latent image in blind deblurring, as discussed in Section 5.2.2. This provides an additional speed improvement in blind deblurring, compared to the use of conjugate-gradient descent with the approximate forward model.

(a) Blurry image & estimated blur kernel

(b) Conjugate gradient descent (20 iterations), exact model, 442s

(c) Conjugate gradient descent (20 iterations), approximate model, 137s

(d) Direct independent deconvolution of patches, 4s

Figure 5.3. **Least-squares non-blind deblurring using the exact and approximate forward models.** Given a blurry image of size $1024 \times 768$ and blur kernel (a), this figure shows the results and computation times for least-squares deconvolution with $\ell_2$ gradient regularisation (Equation (5.3)), using (b) conjugate gradient descent (CG) with the exact forward model, (c) CG with the approximate forward model, and (d) direct deconvolution using the approach described in Section 5.3.2. The results are visually similar using all three methods. Using CG with the approximate forward model is much faster than with the exact model, however the direct approach takes only a fraction of the time of either of these.

**Algorithm 1:** Fast non-blind deconvolution of spatially-varying blur.

**Input**: Blurry image $\mathbf{g}$, blur descriptor $\mathbf{w}$

**Output**: Deblurred image $\hat{\mathbf{f}}$

1 Decompose $\mathbf{g}$ into patches $\mathbf{g}^{(r)}$ using (5.14)

2 Deconvolve each patch to estimate $\hat{\mathbf{f}}^{(r)}$ by solving (5.15) in the frequency domain

3 Combine patches to estimate sharp image $\hat{\mathbf{f}}$ using (5.18)

## 5.4 Conclusion

We have described how an efficient approximation for spatially-variant blur can be used to reduce the computational cost of computing the spatially-variant forward blur model from Chapter 3. In addition, we have proposed an approximate direct method for spatially-varying non-blind deblurring, which is able to perform least-squares non-blind deblurring in a fraction of the time required for iterative methods such as conjugate gradient descent. As a result of the approximations described, we are able to obtain a substantial speed-up in both the blind PSF estimation and the non-blind deblurring steps over the exact model, reducing the time required to completely deblur an image from several hours to a few minutes, with no considerable reduction in quality.

For a $1024 \times 768$ image, our C implementation of the exact model in Equation (3.11) takes approximately 5 seconds to compute, compared to 2 seconds for our MATLAB implementation of the approximate forward model in Equation (5.6), on an Intel Xeon 2.93GHz CPU.

# Chapter 6

# Handling Saturation in Non-Blind Deblurring

## 6.1 Introduction

When the PSF for a blurry image is known, it is the goal of "non-blind" deblurring to recover an estimate of the sharp image. This problem has a long history, having been studied since the 1940s (Wiener, 1949). However, one feature of blurry images, and in particular "shaken" images, that is problematic for non-blind deblurring but that has received very little attention is the presence of saturated pixels. These are caused when the radiance of the scene exceeds the range of the camera's sensor, leaving bright highlights clipped at the maximum output value (e.g. 255 for an 8-bit image). To anyone who has attempted to take hand-held photographs at night, this effect should be familiar as the conspicuous bright streaks left by electric lights, such as in Figure 6.1 (a). These bright pixels, with their clipped values, violate the assumption made by many algorithms that the image formation process is linear, and as a result can cause obtrusive artefacts in the deblurred images. This can be seen in the deblurred images in Figures 6.1 (b) and 6.1 (c).

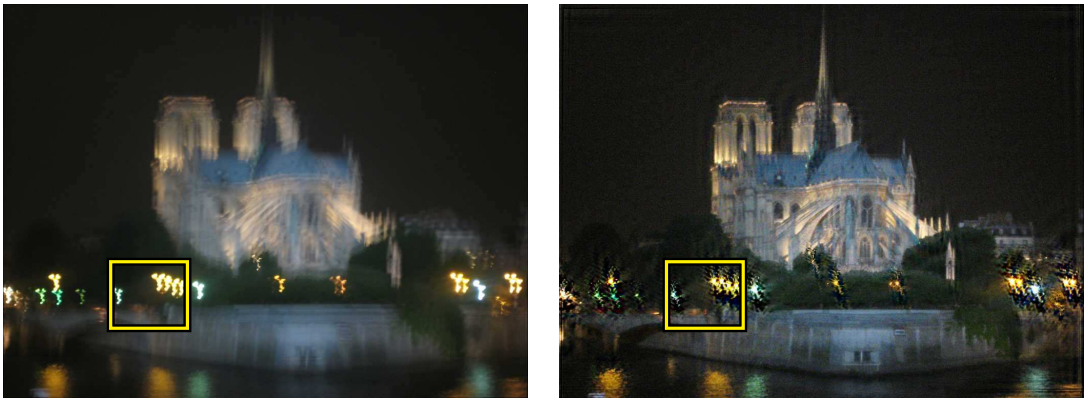
Saturation has not received wide attention in the literature, although it has been

cited as the cause of artefacts in the deblurred outputs from deconvolution algorithms. For example, Fergus et al. (2006), Cho and Lee (2009) and Tai et al. (2011) mention the fact that saturated pixels cause problems, sometimes showing their effect on the deblurred output, but leave the problem to be addressed in future work. An exception is Harmeling et al. (2010b), who address the issue in the setting of multi-frame blind deblurring by thresholding the blurry image to detect saturated pixels, and ignoring these in the deblurring process. When many blurry images are available, these pixels can be safely discarded, since there will generally remain unsaturated pixels covering the same area in other images.

In this chapter we address the problem of non-blind deblurring for images that contain saturated pixels. By handling such pixels explicitly, we are able to produce significantly better results than existing methods. Figure 6.1 (d) shows the output of the proposed algorithm, which contains far fewer artefacts than the two existing algorithms shown for comparison. We begin in Section 6.2 by considering possible ways of explicitly handling saturated pixels, and propose modified versions of the Richardson-Lucy algorithm for each. In Section 6.3 we augment the Richardson-Lucy algorithm with an additional modification, which prevents the propagation of ringing artefacts. We discuss the implementation of the algorithm in Section 6.4, and show results in Section 6.5. In Section 6.6 we look at the different causes of ringing in non-blind deblurring. We conclude in Section 6.7.

## 6.2 Explicitly Handling Saturated Pixels

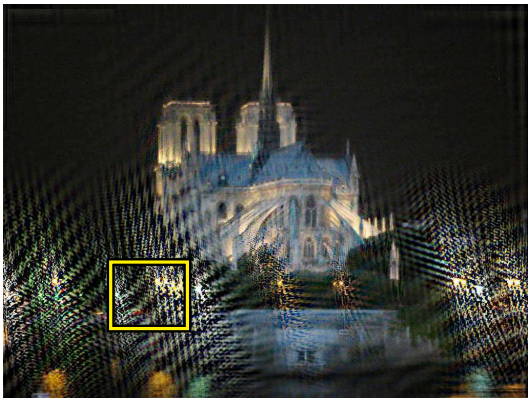
We model sensor saturation as follows: the sensor outputs pixel values which are proportional to the scene radiance, up to some limit, beyond which the pixel value is clipped at the maximum output level. This model is supported by the data in Figure 6.2, which shows the relationship between pixel intensities in three different exposures of a bright light source. The pixel values in the short exposure (with no saturation) and the longer exposures (with saturation) clearly exhibit this clipped linear relationship. As the length



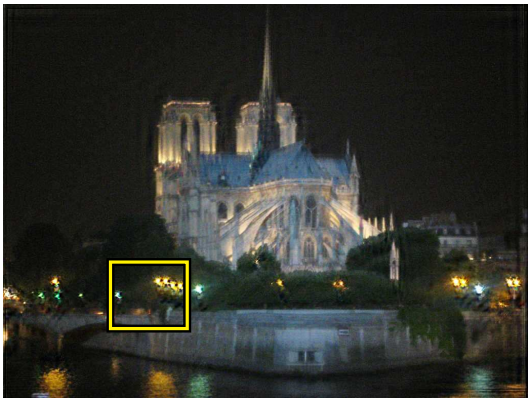
(a) Blurry image



(b) Deblurred with the Richardson-Lucy algorithm

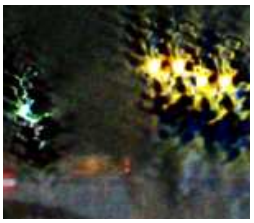


(c) Deblurred with the method of Krishnan and Fergus (2009)

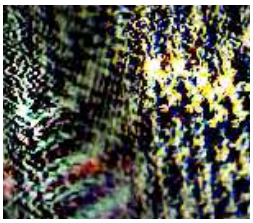


(d) Deblurred with the proposed approach

(a)



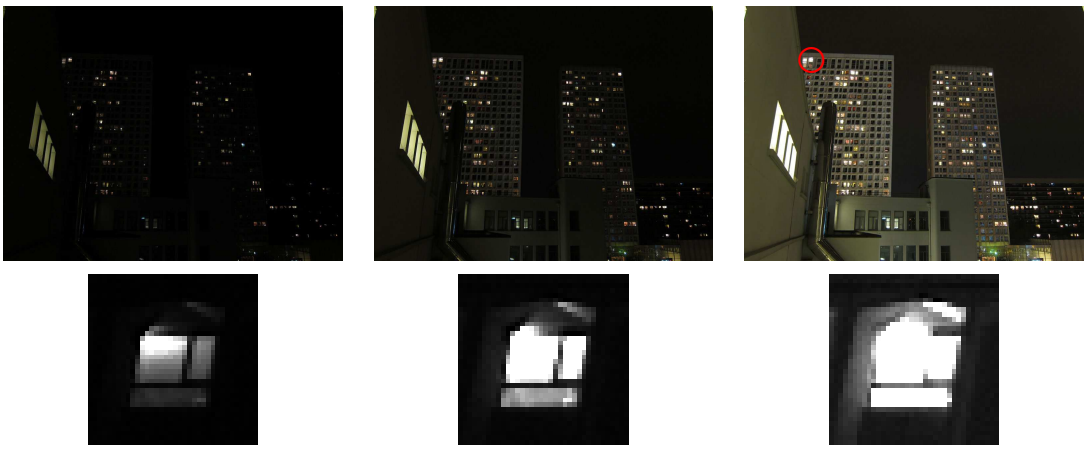
(b)



(c)

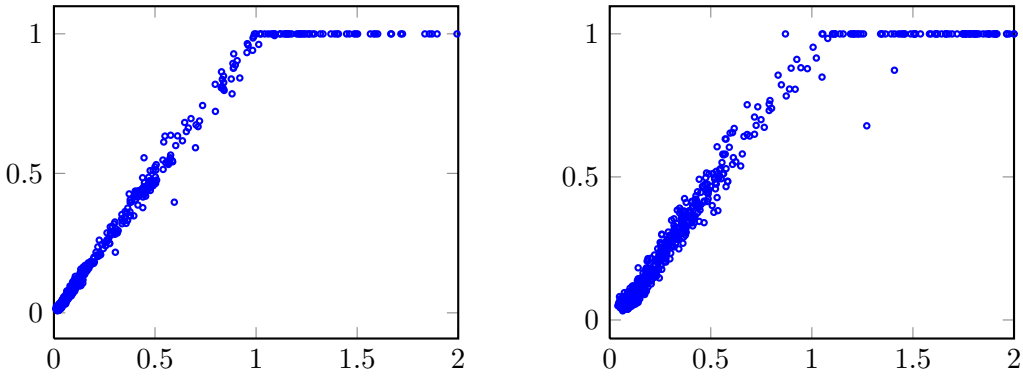
(d)

Figure 6.1. **Deblurring in the presence of saturation.** Existing deblurring methods, such as those in (b) and (c), do not take account of saturated pixels. This leads to large and unsightly artefacts in the results, such as the "ringing" around the bright lights in the zoomed section. Using the proposed method (d), the ringing is greatly reduced and the quality of the deblurring improved. The PSF for this image causes a blur about 35 pixels in width, and was estimated using the MAP-$\ell_1$ algorithm from Chapter 4.



(a) 0.05s (b) 0.2s (c) 0.8s

3 different exposures of a scene containing bright lights




(d) Scatter plot of 0.2s exposure against 0.05s exposure

(e) Scatter plot of 0.8s exposure against 0.05s exposure

Figure 6.2. **Saturated and unsaturated photos of the same scene.** (a)–(c) 3 different exposure times for the same scene, with bright regions that saturate in the longer exposures. A small window has been extracted which is unsaturated at the shortest exposure, and increasingly saturated in the longer two. (d) Scatter plot of the intensities in the small window in (b) against those in the window in (a), normalised by exposure time. (e) Scatter plot of the intensities in the window in (c) against the window in (a), normalised by exposure time. The scatter plots in (d) and (e) clearly show the clipped linear relationship expected.

of the exposure increases, more pixels saturate.

This suggests two possible ways of handling saturation when performing non-blind deblurring: (a) discard the clipped pixels, so that we only use data which follows the linear model, or (b) modify the forward model to take into account this non-linear relationship. We describe both of these approaches in the following, by modifying the

popular Richardson-Lucy algorithm, and demonstrate that the second approach is better, as it does not discard any potentially-useful data.

### 6.2.1 Discarding Saturated Pixels

It is possible to estimate which blurry pixels are saturated by defining a threshold $T$, above which a blurry pixel is considered to be saturated, and therefore an outlier to the linear model. If we discard these pixels, the problem of deblurring with saturated pixels becomes deblurring with missing data.

#### Richardson-Lucy With Missing Data

We begin by defining the set of unsaturated (inlier) blurry pixels $\mathcal{Z} = \{i | g_i \leq T\}$. We also define the corresponding binary mask $\mathbf{z}$, where each element $z_i = 1$ if blurry pixel $g_i \leq T$, and $z_i = 0$ if $g_i > T$. The image formation model for a single blurry pixel $g_i$ in $\mathcal{Z}$ is

$$g_i^* = \sum_j A_{ij} f_j \tag{6.1}$$

$$\frac{\partial g_i^*}{\partial f_j} = A_{ij}. \tag{6.2}$$

Recall from Section 2.1.4 that for a Poisson noise model, the likelihood for the inlier blurry pixels is

$$p(\mathbf{g}_\mathcal{Z} | \mathbf{w}, \mathbf{f}) = \prod_{i \in \mathcal{Z}} \frac{g_i^{* g_i} e^{-g_i^*}}{g_i!} \tag{6.3}$$

The negative log-likelihood is then:

$$\mathcal{L}_\mathcal{Z} = -\sum_{i \in \mathcal{Z}} \log \frac{g_i^{* g_i} e^{-g_i^*}}{g_i!} \tag{6.4}$$

$$= \sum_{i \in \mathcal{Z}} g_i^* - g_i \log g_i^* + \log(g_i!), \tag{6.5}$$

The Richardson-Lucy algorithm minimises $\mathcal{L}_{\mathcal{Z}}$ over the sharp image $\mathbf{f}$. The Karush-Kuhn-Tucker conditions for the optimum of this likelihood are, for all $f_j$:

$$f_j \frac{\partial}{\partial f_j} \mathcal{L}_{\mathcal{Z}} = 0, \tag{6.6}$$

$$\text{and} \qquad \frac{\partial}{\partial f_j} \mathcal{L}_{\mathcal{Z}} \geq 0 \qquad \text{if } f_j = 0. \tag{6.7}$$

The Richardson-Lucy update rule is a fixed-point iteration derived from Equation (6.6):

$$f_j \sum_{i \in \mathcal{Z}} \frac{\partial}{\partial f_j} \left( g_i \log g_i^* - g_i^* - \log(g_i!) \right) = 0 \tag{6.8}$$

$$f_j \sum_{i \in \mathcal{Z}} \left( \frac{g_i}{g_i^*} \frac{\partial g_i^*}{\partial f_j} - \frac{\partial g_i^*}{\partial f_j} \right) = 0. \tag{6.9}$$

Substituting in Equation (6.2) we get:

$$f_j \sum_{i \in \mathcal{Z}} A_{ij} = f_j \sum_{i \in \mathcal{Z}} A_{ij} \frac{g_i}{g_i^*}. \tag{6.10}$$

We can replace the sum over $\mathcal{Z}$ with a weighted sum over all blurry pixels, using the mask elements $z_i$ as the weights:

$$f_j \sum_i A_{ij} z_i = f_j \sum_i A_{ij} \frac{g_i}{g_i^*} z_i. \tag{6.11}$$

Here, in order to obtain an expression for $f_j$, we could divide both sides by $\sum_i A_{ij} z_i$, however this would be unstable in regions where $z_i \simeq 0$. Thus we add $f_j(1 - \sum_i A_{ij} z_i)$ to both sides:

$$f_j = f_j \left( \sum_i A_{ij} \frac{g_i}{g_i^*} z_i + 1 - \sum_i A_{ij} z_i \right). \tag{6.12}$$

Assuming that the blur is conservative, i.e. $\sum_i A_{ij} = 1$, we can factorise this and obtain the update rule by replacing the unknowns $f_j$ and $g_i^*$ with our estimates $\hat{f}_j$ and $(\mathbf{A}\hat{\mathbf{f}})_i = \sum_{j'} A_{ij'} \hat{f}_{j'}$, and adding iteration indices:

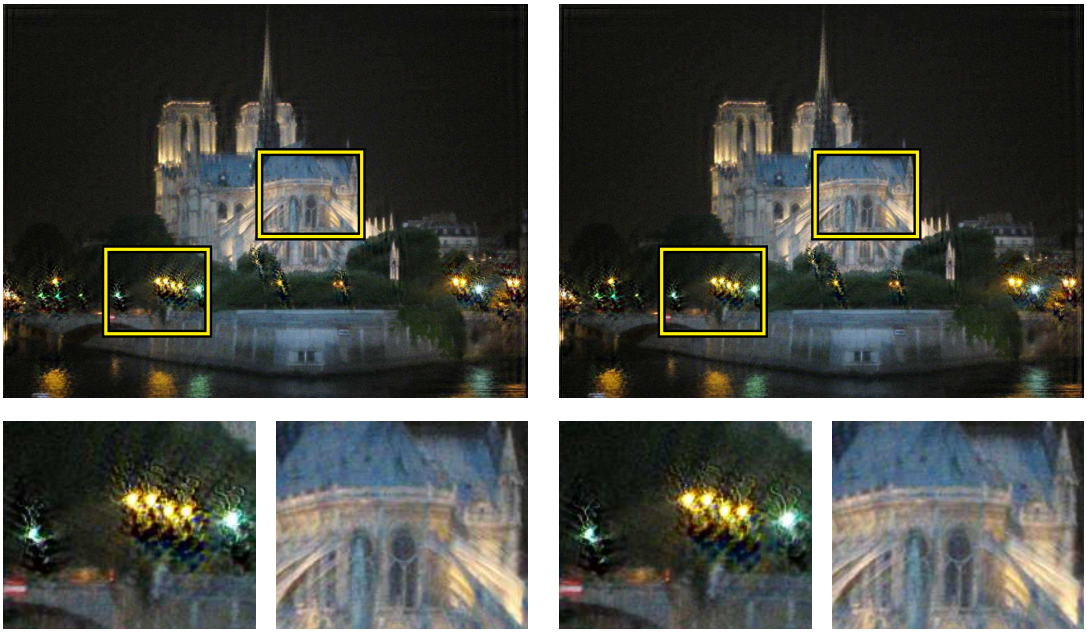
$$\hat{f}_j^{t+1} = \hat{f}_j^t \sum_i A_{ij} \left( \frac{g_i}{(\mathbf{A}\hat{\mathbf{f}}^t)_i} z_i + 1 - z_i \right), \tag{6.13}$$

In matrix-vector form, this update equation is

$$\hat{\mathbf{f}}^{t+1} = \hat{\mathbf{f}}^t \circ \mathbf{A}^\top \left( \frac{\mathbf{g} \circ \mathbf{z}}{\mathbf{A}\hat{\mathbf{f}}^t} + \mathbf{1} - \mathbf{z} \right), \tag{6.14}$$

where $\mathbf{1}$ is a vector of ones. For an unsaturated pixel $g_i$, the mask $z_i = 1$, and the term in parentheses is the same as for the standard RL update. For a saturated (outlier/missing) pixel, $z_i = 0$, so the term in parentheses is equal to unity. Since the update is multiplicative, this means that the saturated observation $g_i$ has no influence on the estimated latent image $\hat{\mathbf{f}}$.

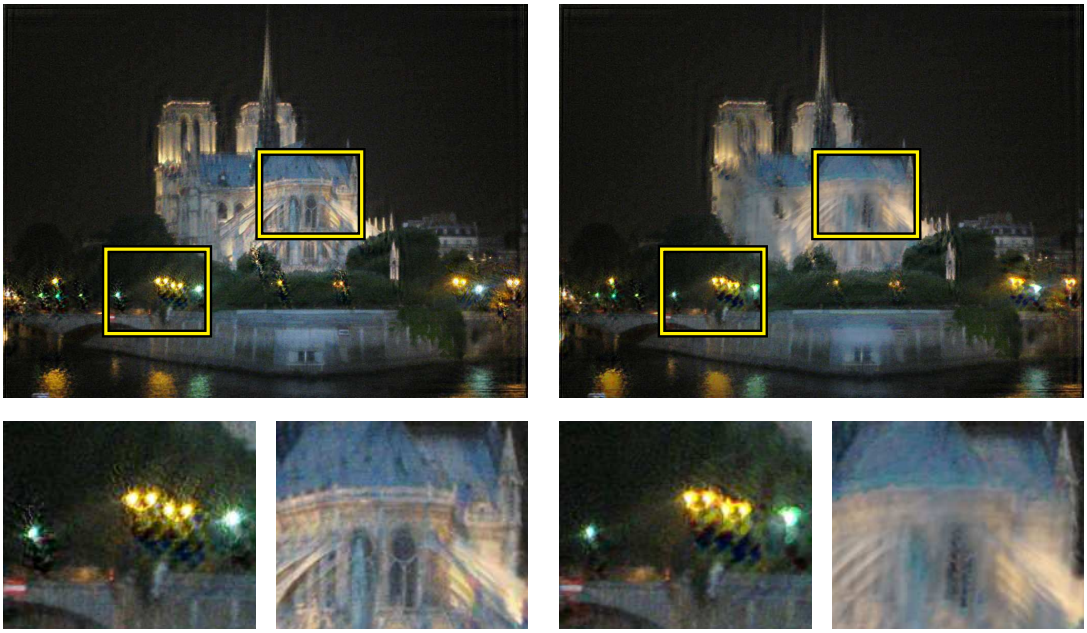
The choice of threshold $T$ can be problematic however; there may not be a single threshold that correctly separates the saturated pixels from the unsaturated ones. A low threshold may discard large numbers of inlying pixels from $\mathbf{g}$, causing some parts of $\hat{\mathbf{f}}$ to become decoupled from the data, and remain unchanged from their initialisation by the algorithm. A high threshold, on the other hand, may treat some saturated pixels as inliers, causing artefacts in the deblurred result. Figure 6.3 shows the result of deblurring using Equation (6.14) for different values of threshold $T$. As is visible in the figure, no particular threshold produces a result free of artefacts. At high values of $T$, the building is deblurred well, but artefacts appear around the lights. At the lowest value of $T$, the lights are deblurred reasonably well, but the face of the building is mistakenly discarded and thus remains blurry in the output.



(a) $T = 0.7$



(b) $T = 0.5$



(c) $T = 0.3$



(d) $T = 0.1$

Figure 6.3. **Ignoring saturated pixels using a threshold.** A simple way to handle saturation is to threshold the blurry image at some level $T$, and discard the blurry pixels above this threshold. Shown are the results of running the Richardson-Lucy algorithm for different thresholds. As the threshold decreases, the artefacts around the bright lights are reduced compared to the standard RL result in Figure 6.1 (b). At the lowest threshold (d) the fewest artefacts appear, but parts of the church are also discarded, hence remain blurred.

### 6.2.2 A Forward Model for Saturation

Instead of attempting to segment the blurry image into saturated and unsaturated regions, we may instead modify our forward model to include the saturation process. This avoids making *a priori* decisions about which data are inliers or outliers. To this end, we modify the original image degradation model in Equation (2.1) by introducing a response function $R : \mathbb{R}_+ \to \mathbb{R}_+$, which models the non-linear response of the sensor. This function is applied to the degraded image before it is output by the sensor, i.e.

$$g(\mathbf{x}) = R\big(N\big((Hf)(\mathbf{x})\big)\big), \tag{6.15}$$

where, recalling the notation of Section 2.1, $H$ is the blurring operator and $N$ represents the perturbation by random noise. This formulation however, is not particularly tractable, except for trivial response functions $R$. Since the noise lies inside $R$, the likelihood is distorted and can generally no longer be written in closed-form.

Instead, we use the following, more tractable, formulation:

$$g(\mathbf{x}) = N\big(R\big((Hf)(\mathbf{x})\big)\big), \tag{6.16}$$

which allows us to incorporate the response into the formation of the "noiseless" blurry image

$$g^*(\mathbf{x}) = R\big((Hf)(\mathbf{x})\big), \tag{6.17}$$

$$\mathbf{g}^* = R\left(\mathbf{A}\mathbf{f}\right), \tag{6.18}$$

where the function $R$ is applied element-wise to a vector. Figure 6.4 shows a diagram of this model.

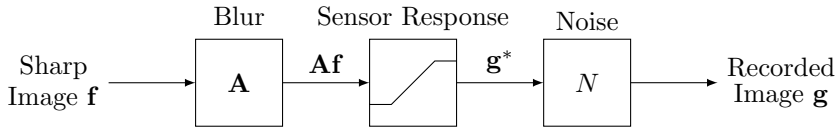



Figure 6.4. **Diagram of image formation process.** We model the response function as acting on the noiseless blurry image, before the noise is added.

### Richardson-Lucy With a Non-Linear Response

We can incorporate this non-linear image formation model into the Richardson-Lucy algorithm. From Equation (6.18) we have

$$\frac{\partial g_i^*}{\partial f_j} = A_{ij} R'\big((\mathbf{Af})_i\big). \tag{6.19}$$

Substituting this into Equation (6.9) and following a similar derivation as for the Richardson-Lucy with missing data, we arrive at the update rule

$$\hat{f}_j^{t+1} = \hat{f}_j^t \sum_i A_{ij} \left( \frac{g_i}{R\big((\mathbf{A}\hat{\mathbf{f}}^t)_i\big)} R'\big((\mathbf{A}\hat{\mathbf{f}}^t)_i\big) + 1 - R'\big((\mathbf{A}\hat{\mathbf{f}}^t)_i\big) \right). \tag{6.20}$$

In matrix-vector form, this update equation is

$$\hat{\mathbf{f}}^{t+1} = \hat{\mathbf{f}}^t \circ \mathbf{A}^\top \Big( \frac{\mathbf{g} \circ R'(\mathbf{A}\hat{\mathbf{f}}^t)}{R(\mathbf{A}\hat{\mathbf{f}}^t)} + \mathbf{1} - R'(\mathbf{A}\hat{\mathbf{f}}^t) \Big). \tag{6.21}$$

### Choosing the Response Function

One choice for $R$ would be simply to truncate the linear model at 1 (the maximum pixel value), i.e. $R(x) = \min(x, 1)$. This choice is empirically justified, as can be seen in Figure 6.2. However, this function is non-differentiable at $x = 1$, i.e. $R'(1)$ is not defined. We thus use a smooth approximation (Chen and Mangasarian, 1996), where

$$R(x) = x - \frac{1}{a} \log\Big(1 + \exp\big(a(x-1)\big)\Big) \tag{6.22}$$

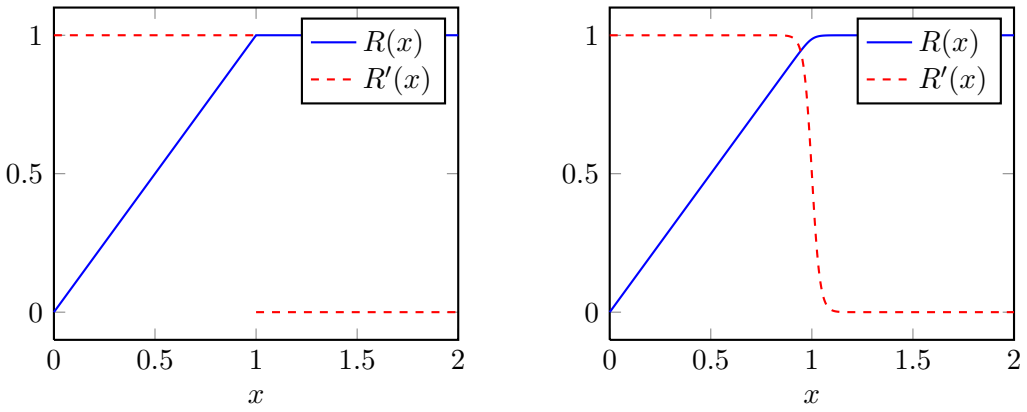
$$R'(x) = \frac{1}{1 + \exp\big(a(x-1)\big)}, \tag{6.23}$$

where $R'$ represents the first derivative of $R$. The parameter $a$ controls the smoothness of the approximation, and we have found $a = 50$ to be a suitable compromise between smoothness and accuracy. Figure 6.5 shows the shape of $R$ and $R'$ compared to the simple truncated linear model. In all our experiments $a = 50$.

Given the shape of $R$, Equation (6.21) can easily be interpreted: in the linear portion $R(x) \simeq x$ and $R'(x) \simeq 1$, so that the term in parentheses is the same as for the standard Richardson-Lucy algorithm, while in the saturated portion $R(x) \simeq 1$ and $R'(x) \simeq 0$, so that the term in parentheses is equal to unity and has no influence on $\mathbf{f}$. Considering this shape of $R'$, and by comparing the update equation in Equation (6.21) to the "masked RL" update in Equation (6.14), we can see that $R'(\mathbf{Af})$ play a role very similar to the mask $\mathbf{z}$, downweighting saturated observations. However, it is important to note that this behaviour arises naturally from the forward model, and is not detected from the blurry image. Given our current estimate of the latent image $\mathbf{f}$, the forward model tells us which blurry pixels are saturated, and automatically downweights those pixels in the update equation. We refer to the algorithm using this update rule as "saturated RL". Figure 6.6 demonstrates the advantage of this method over the standard RL algorithm on a synthetic 1D example.

## 6.3 Preventing the Propagation of Errors

It is important to note that even when we take account of saturation in our image formation model, we are not necessarily able to estimate every pixel in $\mathbf{f}$ accurately. In the blurring process, each latent pixel $f_j$ in the sharp image is blurred across multiple pixels in the blurry image $\mathbf{g}$. If some (or all) of these are saturated, we are left with an incomplete set of data concerning $f_j$, and our estimate of $f_j$ is likely to be less accurate than if we had a full set of unsaturated observations available. This mis-estimation is one source of "ringing" artefacts in the deblurred output. An over-estimate at one pixel




(a) Ideal saturation function

(b) Smooth, differentiable saturation function

Figure 6.5. **Modelling the saturated sensor response.** (a) Ideal clipped linear response function (solid blue line) and its derivative (dashed red line). The derivative is not defined at $x = 1$. (b) Smooth and differentiable approximation to (a) defined in Equation (6.22). The derivative is also smooth and defined everywhere.

must be balanced out by an under-estimate at a neighbouring pixel, which must in turn be balanced out by another over-estimate. In this way, an error at one pixel spreads outwards in waves across the image. In order to mitigate this effect, we propose a second modification to the Richardson-Lucy algorithm to stop the propagation of these errors.

Following the arguments above, we see that it is the brightest pixels in $\mathbf{f}$, and their neighbours, that we are likely to estimate poorly (since they cause saturation), and that we would like to prevent the propagation of errors from these poorly estimated pixels to their neighbours. To this end, we segment $\mathbf{f}$ into two disjoint regions: $\mathcal{S}$, which includes the bright pixels and their neighbours that we are unlikely to estimate accurately, and $\mathcal{U}$, which covers the rest of the image and which we can estimate accurately. We decompose the latent image correspondingly: $\mathbf{f} = \mathbf{f}_\mathcal{U} + \mathbf{f}_\mathcal{S}$.

Our aim is then to prevent the propagation of errors from $\mathbf{f}_\mathcal{S}$ to $\mathbf{f}_\mathcal{U}$. To achieve this, we propose to estimate $\mathbf{f}_\mathcal{U}$ using only data which is not influenced by any pixels from $\mathcal{S}$. To this end, we first define the region (denoted by $\mathcal{V}$) of the blurry image which is

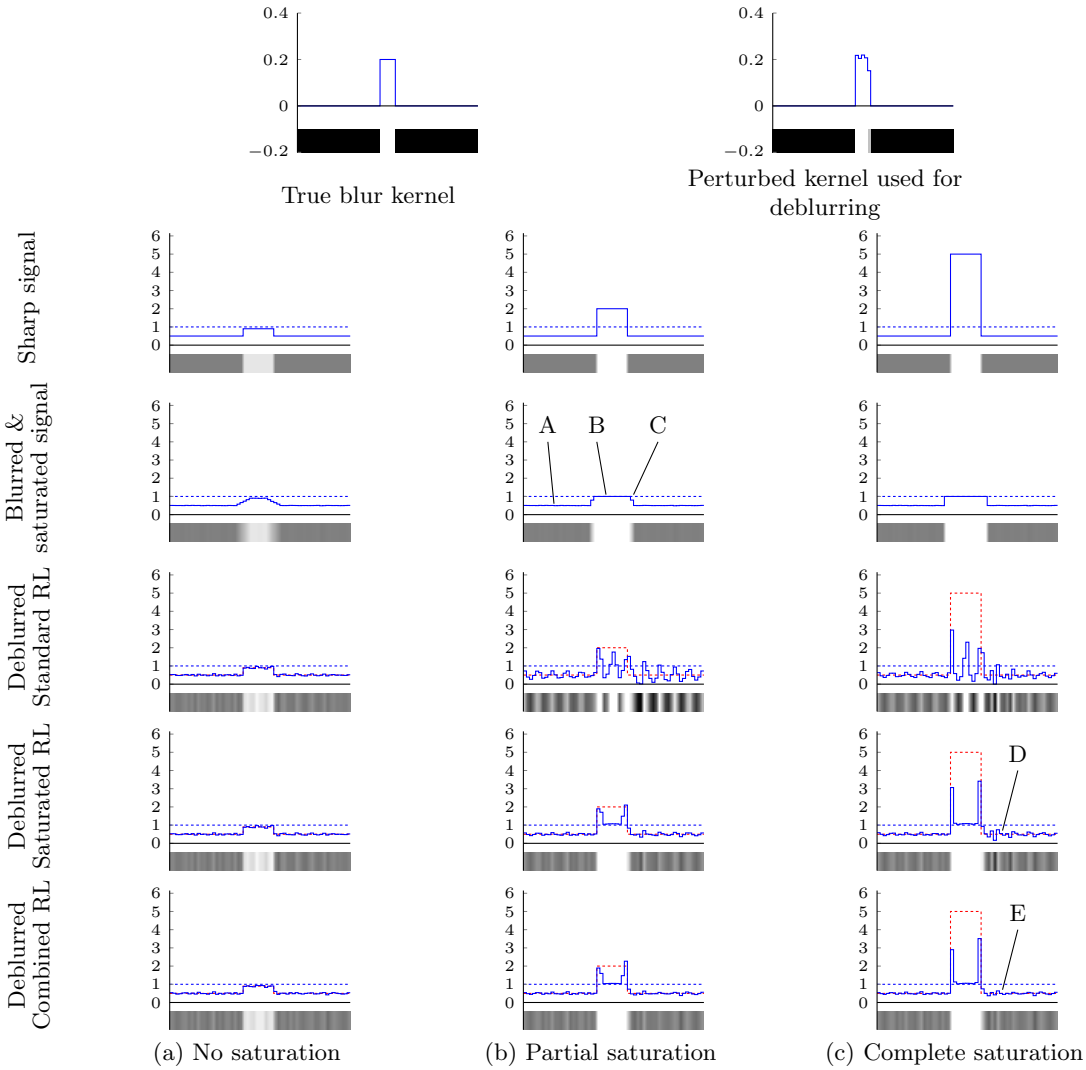



Figure 6.6. **Synthetic example of blur and saturation.** Each column shows a sharp "top-hat" signal, blurred using the box filter shown at the top left. Gaussian noise is added and the blurred signal is clipped, to model saturation. The kernel is also degraded with noise and one large error to produce a "perturbed" kernel which is used for deconvolution, to simulate errors in the kernel estimation step. The last three rows show the deblurred outputs for three algorithms discussed in Section 6.2. (a) With no saturation, all three algorithms produce similar results. (b) When some of the blurred signal is saturated (region B), the standard RL algorithm produces an output with large ringing artifacts. Although region A is not itself saturated, the ringing propagates outwards from B & C across the whole signal. The "saturated RL" algorithm reduces the ringing and correctly estimates the height of the top-hat at its edges (region C), where there are some unsaturated observations available. In region B all information about the height of the sharp signal is lost, and the output takes a sensible value close to 1. (c) When the blurred top-hat is completely saturated, it is no longer possible to estimate its true height anywhere. The saturated RL result accurately locates the top-hat, but contains ringing. The proposed method (combined RL) mitigates this by preventing the propagation of errors to the non-saturated region (compare D to E).

independent of $\mathbf{f}_\mathcal{S}$, by eroding $\mathcal{U}$ using the non-zero elements of the PSF:

$$\mathcal{V} = \bigcap_{k:w_k>0} \mathcal{U}_{\mathbf{T}^{(k)}}, \tag{6.24}$$

where $\mathcal{U}_{\mathbf{T}^{(k)}}$ denotes the set $\mathcal{U}$ transformed by $\mathbf{T}^{(k)}$. By taking the intersection of all the transformed versions of $\mathcal{U}$, we ensure that $\mathcal{V}$ contains only those blurry pixels that are completely independent of $\mathcal{S}$. We can then estimate $\mathbf{f}_\mathcal{U}$ using only the data in $\mathcal{V}$, by defining the binary mask $\mathbf{v}$ which corresponds to $\mathcal{V}$ and adapting the update equation from Equation (6.14) for Richardson-Lucy with missing data (Section 6.2.1):

$$\hat{\mathbf{f}}_\mathcal{U}^{t+1} = \hat{\mathbf{f}}_\mathcal{U}^{t} \circ \mathbf{A}^\top \left( \frac{\mathbf{g} \circ R'(\mathbf{A}\hat{\mathbf{f}}^t) \circ \mathbf{v}}{R(\mathbf{A}\hat{\mathbf{f}}^t)} + \mathbf{1} - R'(\mathbf{A}\hat{\mathbf{f}}^t) \circ \mathbf{v} \right). \tag{6.25}$$

We estimate $\mathbf{f}_\mathcal{S}$ using the previously defined "saturated RL" algorithm:

$$\hat{\mathbf{f}}_\mathcal{S}^{t+1} = \hat{\mathbf{f}}_\mathcal{S}^{t} \circ \mathbf{A}^\top \left( \frac{\mathbf{g} \circ R'(\mathbf{A}\hat{\mathbf{f}}^t)}{R(\mathbf{A}\hat{\mathbf{f}}^t)} + \mathbf{1} - R'(\mathbf{A}\hat{\mathbf{f}}^t) \right). \tag{6.26}$$

Since the Richardson-Lucy algorithm is an iterative process, we do not know beforehand which parts of $\mathbf{f}$ belong in $\mathcal{U}$ and which in $\mathcal{S}$. We thus perform the segmentation at each iteration $t$ using a threshold on the *latent image*:

$$\mathcal{U} = \left\{ j \middle| \hat{f}_j^t \leq \varphi \right\}. \tag{6.27}$$

We decompose $\hat{\mathbf{f}}^t$ according to

$$\hat{\mathbf{f}}_\mathcal{U}^{t} = \mathbf{u} \circ \hat{\mathbf{f}}^t \tag{6.28}$$

$$\hat{\mathbf{f}}_\mathcal{S}^{t} = \hat{\mathbf{f}}^t - \hat{\mathbf{f}}_\mathcal{U}^{t}, \tag{6.29}$$

where $\mathbf{u}$ is the binary mask corresponding to $\mathcal{U}$. We then compute $\mathcal{V}$, update $\hat{\mathbf{f}}_\mathcal{U}$ and $\hat{\mathbf{f}}_\mathcal{S}$ using Equations (6.25) and (6.26), and recombine them to form our new estimate of the latent image $\hat{\mathbf{f}}^{t+1} = \hat{\mathbf{f}}_\mathcal{U}^{t+1} + \hat{\mathbf{f}}_\mathcal{S}^{t+1}$. We refer to this algorithm as "combined RL",

and Figure 6.6 shows the results of applying it to a synthetic 1D example, demonstrating the advantage over the standard RL and "saturated RL" algorithms. Algorithm 2 summarises the method.

**Algorithm 2:** Combined Richardson-Lucy algorithm.

**Input**: Blurry image $\mathbf{g}$, blur descriptor $\mathbf{w}$

**Output**: Deblurred image $\hat{\mathbf{f}}$

$\hat{\mathbf{f}}^0 \leftarrow \mathbf{g}$

**for** $t = 0$ **to** *nIters* **do**

  Decompose $\hat{\mathbf{f}}^t$ into $\hat{\mathbf{f}}^t_{\mathcal{U}}$ and $\hat{\mathbf{f}}^t_{\mathcal{S}}$ using (6.27) to (6.29)

  Compute set $\mathcal{V}$ of blurry pixels independent of $\mathcal{S}$ using (6.24)

  Compute $\hat{\mathbf{f}}^{t+1}_{\mathcal{U}}$ using only data from $\mathcal{V}$ (6.25)

  Compute $\hat{\mathbf{f}}^{t+1}_{\mathcal{S}}$ using all data (6.26)

  $\hat{\mathbf{f}}^{t+1} = \hat{\mathbf{f}}^{t+1}_{\mathcal{U}} + \hat{\mathbf{f}}^{t+1}_{\mathcal{S}}$

**end**

Although this combined RL algorithm involves the use of a threshold to segment the image (Equation (6.27)), its effect is less dramatic than in Section 6.2.1. In this case, the threshold only determines whether a given pixel $f_j$ should be updated using all the available data, or a subset of it. This is in contrast to Section 6.2.1, where parts of the data are discarded and never used again. Since our aim is to ensure that no large errors are introduced in $\mathbf{f}_{\mathcal{U}}$, we set the threshold low enough that most potentially-bright pixels are assigned to $\mathcal{S}$. Empirically, we choose $\varphi = 0.9$ for the results in this paper.

## 6.4 Implementation

In this section we describe some of the implementation details of the proposed algorithm. When segmenting the current estimate of the latent image in the combined RL algorithm, we take additional steps to ensure that we make a conservative estimate of which pixels can be estimated accurately. First, after thresholding the latent image in

Equation (6.27), we perform a binary erosion on $\mathcal{U}$, such that

$$\mathcal{U} = \{j | f_j \leq \varphi\} \ominus \mathcal{M}, \tag{6.30}$$

where $\ominus$ denotes binary erosion, and the structural element $\mathcal{M}$ used for erosion is a disk of radius 3 pixels. This ensures that all poorly-estimated pixels are correctly assigned to $\mathcal{S}$ (perhaps at the expense of wrongly including some well-estimated pixels too). Performing this step improves the deblurred results, since it is not only the bright pixels whose value is likely to be inaccurate due to saturation, but their neighbours too, and fewer artefacts arise from wrongly assigning a well-estimated pixel into $\mathcal{S}$ than the other way around. Second, in order to avoid introducing visible boundaries between the two regions, we blur the mask $\mathbf{u}$ slightly using a Gaussian filter with standard deviation 3 pixels to produce a smoother set of weights when extracting $\hat{\mathbf{f}}^t_{\mathcal{U}}$ and $\hat{\mathbf{f}}^t_{\mathcal{S}}$ from the current latent image $\hat{\mathbf{f}}^t$ in Equations (6.28) and (6.29).

## 6.5 Results

Figures 6.1 and 6.7 to 6.9 show results of non-blind deblurring using the proposed "combined RL" algorithm described in Section 6.3 on real hand-held photographs. The (spatially-variant) PSFs for these images were estimated from the blurry images themselves using the MAP-$\ell_1$ algorithm described in Chapter 4. The only modification required to handle saturated images using this algorithm is to discard potentially saturated regions of the blurry image using a threshold. Since in this case the aim is only to estimate the PSF (and not a complete deblurred image), we can safely discard all of these pixels, since the number of saturated pixels in an image is typically small compared to the total number of pixels. There will typically remain sufficient unsaturated pixels from which to estimate the PSF.

Note that the standard Richardson-Lucy algorithm and the approach of Krishnan and Fergus (2009) produce large amounts of ringing around the saturated regions, while the proposed "combined RL" algorithm avoids this with no loss of quality elsewhere. In

all results in this paper we performed 50 iterations of all Richardson-Lucy variants.

### 6.5.1 Comparison to Cho et al. (2011)

Figure 6.11 shows the results of non-blind deblurring using our algorithm, alongside those of the recently-proposed algorithm of Cho et al. (2011). The blurry images and their spatially-invariant PSFs are provided by Cho et al., along with their deblurred results[1]. The results of Cho et al. (2011) are smooth from heavy sparse-gradient regularisation. Our results are produced without any regularisation, however, despite this, they exhibit fewer visible artefacts in most cases than those of Cho et al. (2011).

## 6.6 Perspective: The Causes of Ringing

In this section we offer some analysis of the causes of "ringing" artefacts in deblurred images – medium-frequency ripples spreading across the image. Ringing is common in deblurred images, and has been explained variously as due to the Gibbs phenomenon (Yuan et al., 2007b) (arising from the inability of a finite Fourier basis to reconstruct perfect step edges), or a combination of inaccurately-modelled image noise and errors in the estimated PSF (Shan et al., 2008). We have found the latter to be accurate, and provide some analysis of this here.

Ringing is caused by the interaction between zeros in the PSF and (a) outliers in the data (e.g. caused by noise, non-linearities in the camera response, non-static scene) and (b) errors in the estimated blur kernel. By zeros in the PSF, we are referring to singular values of $\mathbf{A}$ that are zero or very small. Each singular value of $\mathbf{A}$ corresponds approximately to particular spatial frequencies (they correspond exactly for spatially-invariant blur, where the singular value decomposition (SVD) of $\mathbf{A}$ is equivalent to the discrete Fourier transform of $\mathbf{a}$). Thus small or zero singular values correspond to spatial frequencies that are annihilated by the blur.

When we attempt to deblur the image, there is very little information about these

[1] `http://cg.postech.ac.kr/research/deconv_outliers/` (accessed November 12, 2011).

(a) Blurry image



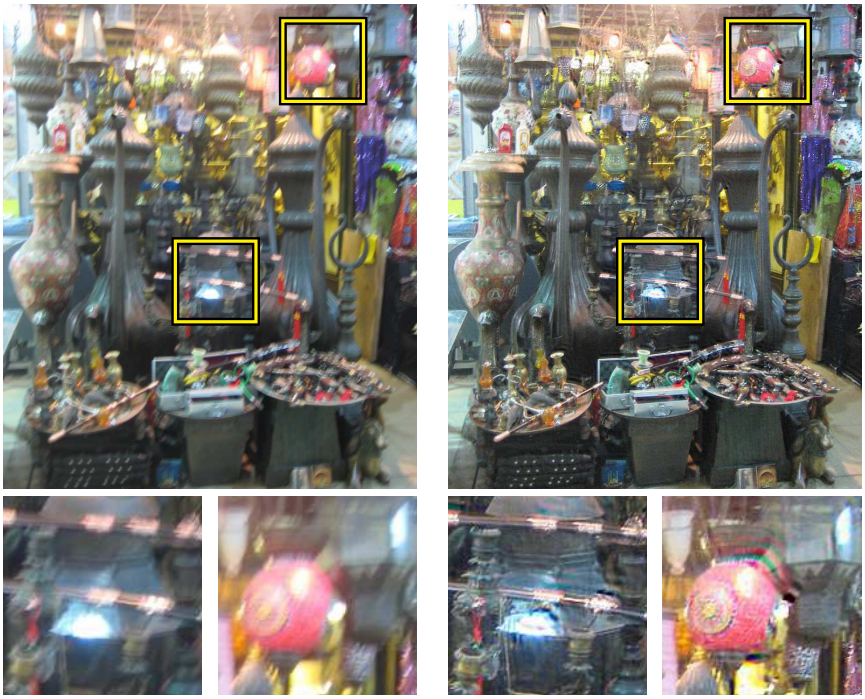
(b) Deblurred with Richardson-Lucy

(c) Deblurred with algorithm of Krishnan and Fergus (2009)



(d) Deblurred with proposed method

Figure 6.7. **Deblurring saturated images.** Note that the ringing around saturated regions, visible in columns (b)–(c) is removed by our method (d), without causing any loss in visual quality elsewhere.



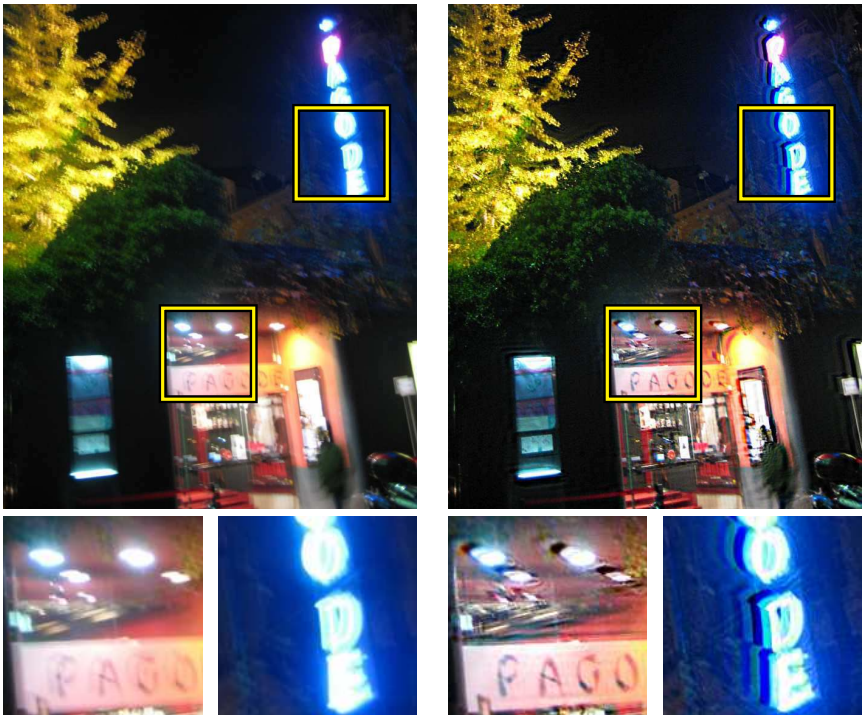
(a) Blurry image

(b) Deblurred with Richardson-Lucy

(c) Deblurred with algorithm of Krishnan and Fergus (2009)

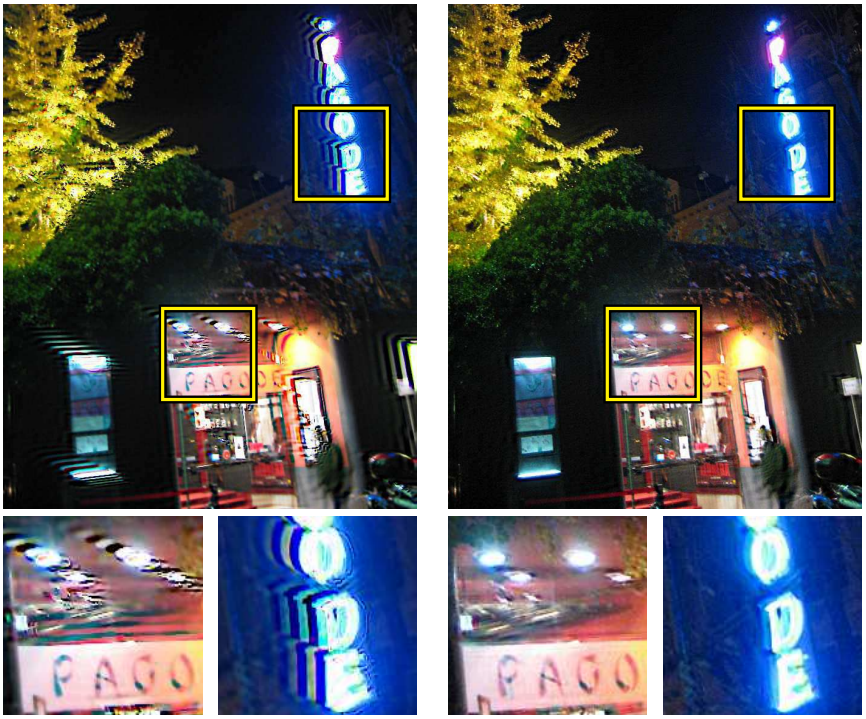
(d) Deblurred with proposed method

Figure 6.8. **Deblurring saturated images.** Note that the ringing around saturated regions, visible in columns (b)–(c) is removed by our method (d), without causing any loss in visual quality elsewhere.



(a) Blurry image

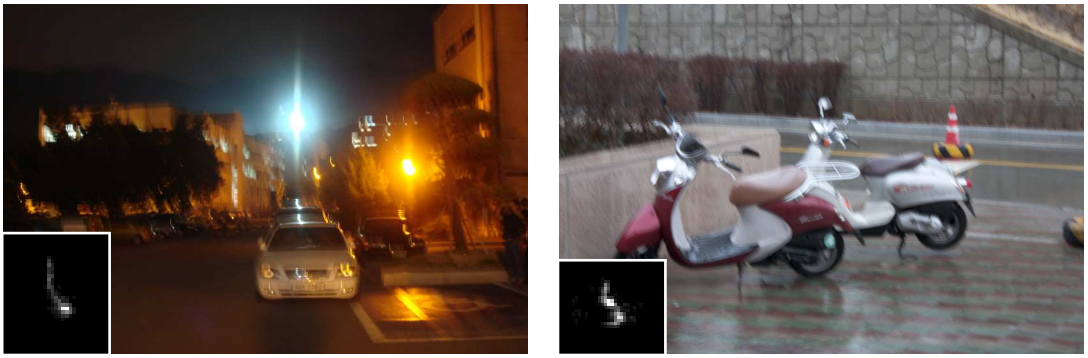
(b) Deblurred with Richardson-Lucy



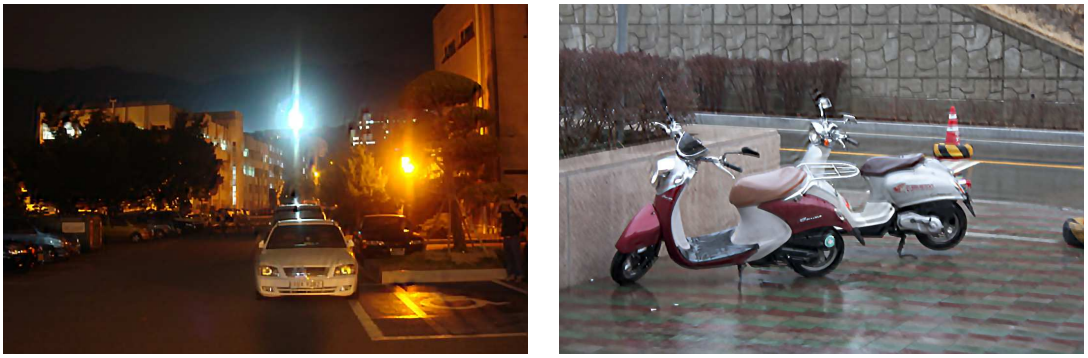
(c) Deblurred with algorithm of Krishnan and Fergus (2009)

(d) Deblurred with proposed method

Figure 6.9. **Deblurring saturated images.** Note that the ringing around saturated regions, visible in columns (b)–(c) is removed by our method (d), without causing any loss in visual quality elsewhere.



(a) Real blurry images, with estimated kernels shown

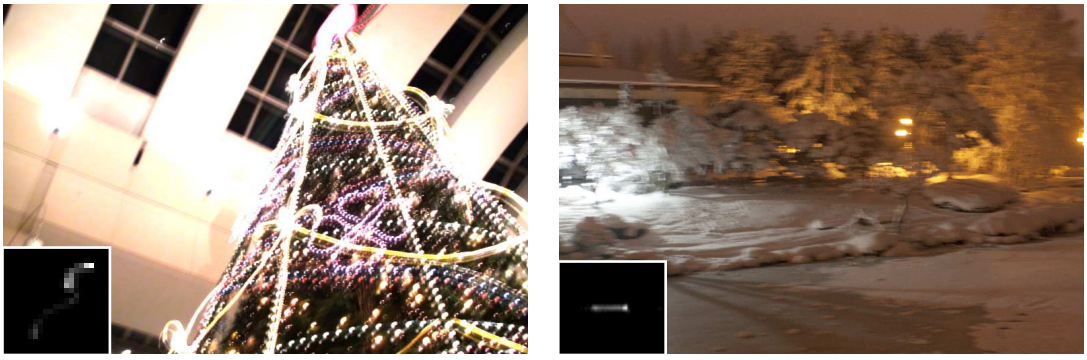


(b) Results from Cho et al. (2011)

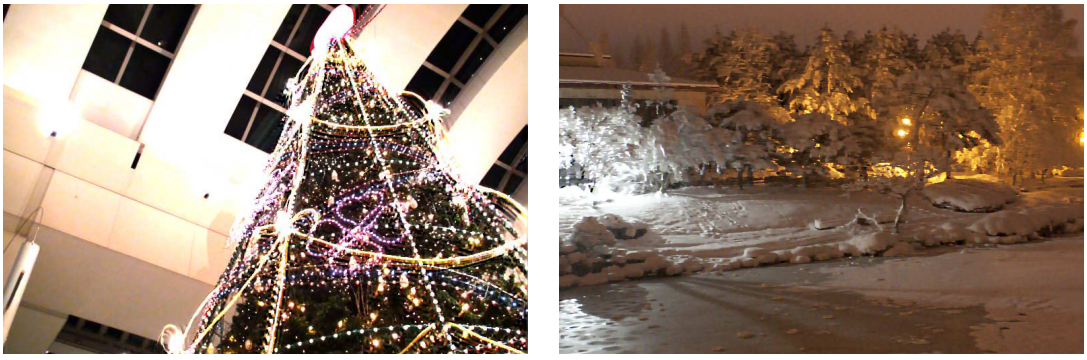
(c) Our deblurred results

(b) Cho et al. (c) Ours (b) Cho et al. (c) Ours

Figure 6.10. **Comparison to the method of Cho et al. (2011).** The blurry images, the spatially-invariant PSFs and the results of their method were provided by the authors.

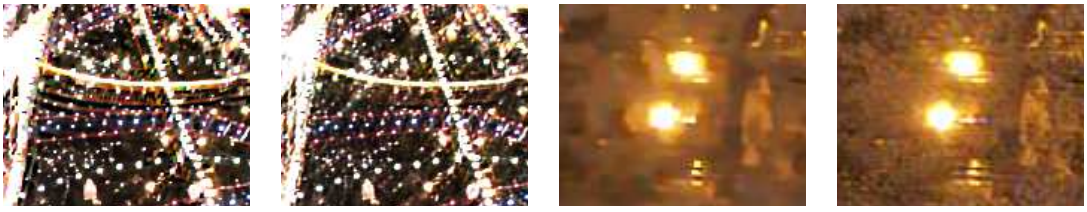


(a) Blurry images, with estimated kernels shown



(b) Results from Cho et al. (2011)

(c) Our deblurred results



(b) Cho et al. (c) Ours (b) Cho et al. (c) Ours

Figure 6.11. **Comparison to the method of Cho et al. (2011).** The blurry images, the spatially-invariant PSFs and the results of their method were provided by the authors.

spatial frequencies, and numerical round-off and noise may dominate over the true signal. This immediately gives rise to ringing, however this kind of ringing can often be suppressed with regularisation.

Regularisation in non-blind deblurring is able to alleviate ringing to some extent, however since most regularisers only concerns image derivatives, they are only capable of reducing ringing at high spatial frequencies. Yuan et al. (2008) observe this fact and propose a multi-scale non-blind deblurring algorithm capable of preventing ringing caused by noise and numerical inaccuracies. However their method does not handle ringing caused by outliers in the data, shown in Figure 6.12.

The other factor to consider is that the zero singular values represent spatial frequencies which are very poorly constrained by the data. Thus if the blurry image contains outliers which do not fit the linear model, any disruption to the deblurred image will happen in these spatial frequencies first, since they are the cheapest place to hide the errors.

### 6.6.1 Ringing Due to Outliers

We can see how outliers in the data cause ringing by decomposing the observed image $\mathbf{g}$ into an "ideal" noiseless component $\mathbf{g}^*$, a noise component $\mathbf{n}$, and an outlier component $\mathbf{e}$. For simplicity, we will consider spatially-invariant blur with a blur kernel $\mathbf{a}$, such that the noiseless blurry image $\mathbf{g}^*$ is related to the true sharp image $\mathbf{f}$ by $\mathbf{g}^* = \mathbf{a} * \mathbf{f}$. We decompose the observed blurry image as

$$\mathbf{g} = \mathbf{g}^* + \mathbf{n} + \mathbf{e}. \tag{6.31}$$

Using direct deconvolution in the frequency domain with an $\ell_2$ regularisation term $\boldsymbol{\phi}$ (e.g. $\boldsymbol{\phi} = \alpha\mathcal{F}(\mathbf{d}^x)^* \circ \mathcal{F}(\mathbf{d}^x) + \alpha\mathcal{F}(\mathbf{d}^y)^* \circ \mathcal{F}(\mathbf{d}^y)$), we can write Fourier transform of the deblurred image $\hat{\mathbf{f}}$ as

$$\mathcal{F}(\hat{\mathbf{f}}) = \frac{\mathcal{F}(\mathbf{a})^* \circ \mathcal{F}(\mathbf{g})}{\mathcal{F}(\mathbf{a})^* \circ \mathcal{F}(\mathbf{a}) + \boldsymbol{\phi}} \tag{6.32}$$

$$= \frac{\mathcal{F}(\mathbf{a})^*}{\mathcal{F}(\mathbf{a})^* \circ \mathcal{F}(\mathbf{a}) + \boldsymbol{\phi}} \circ \left(\mathcal{F}(\mathbf{g}^*) + \mathcal{F}(\mathbf{n}) + \mathcal{F}(\mathbf{e})\right) \tag{6.33}$$

$$\hat{\mathbf{f}} = \hat{\mathbf{f}}_{\mathbf{g}^*} + \hat{\mathbf{f}}_{\mathbf{n}} + \hat{\mathbf{f}}_{\mathbf{e}}. \tag{6.34}$$

The component $\hat{\mathbf{f}}_{\mathbf{g}^*}$, which is deconvolved from $\mathbf{g}^*$ will be close to the true sharp image, apart from some smoothing due to the regularisation. The magnitude of the noise $\mathbf{n}$ is typically small relative to the image, and so the component $\hat{\mathbf{f}}_{\mathbf{n}}$ will also have a fairly small amplitude in the deblurred image. The outliers $\mathbf{e}$ however, may have a large magnitude, and are also likely to have a high bandwidth – e.g. will contain impulses or step edges which have large frequency components at a wide range of frequencies. This is a significant source of ringing, since those spatial frequencies for which the power spectrum of $\mathbf{a}$ is small (in the denominator of Equation (6.33)) may be amplified. Since the outliers have such a large bandwidth, it is very likely that a large frequency component of the outliers will coincide with a small frequency of the kernel. Figure 6.13 demonstrates this visually with a synthetic example.

This also explains why ringing appears around sharp edges, since such features in an image will have a broad frequency bandwidth, and even small errors may cause large amplification of the zero frequencies of $\mathbf{a}$.

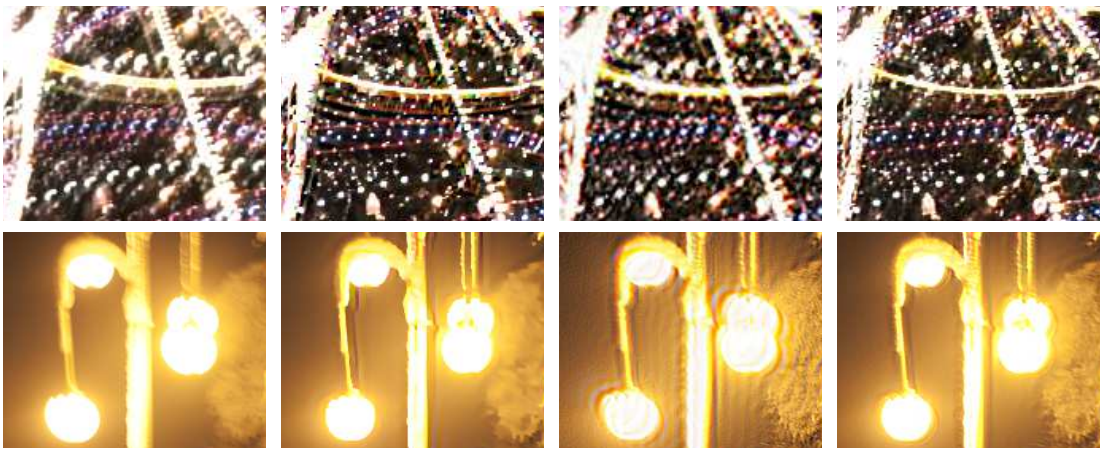
### 6.6.2 Ringing Due to Kernel Errors

When the kernel contains errors, a second source of ringing is introduced. We denote the estimated kernel as $\hat{\mathbf{a}}$, and the true kernel as $\mathbf{a}$, and again write Fourier transform of the deblurred image $\hat{\mathbf{f}}$ as

$$\mathcal{F}(\hat{\mathbf{f}}) = \frac{\mathcal{F}(\hat{\mathbf{a}})^*}{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}}) + \boldsymbol{\phi}} \circ \left(\mathcal{F}(\mathbf{g}^*) + \mathcal{F}(\mathbf{n}) + \mathcal{F}(\mathbf{e})\right) \tag{6.35}$$

$$\hat{\mathbf{f}} = \hat{\mathbf{f}}_{\mathbf{g}^*} + \hat{\mathbf{f}}_{\mathbf{n}} + \hat{\mathbf{f}}_{\mathbf{e}}. \tag{6.36}$$

In this case, we need only examine $\hat{\mathbf{f}}_{\mathbf{g}^*}$, since the effect of deconvolving $\mathbf{n}$ and $\mathbf{e}$ with the incorrect kernel is essentially the same as deconvolving them with the correct kernel.



(a) Blurry image (b) Cho et al. (2011) (c) Yuan et al. (2008) (d) Ours

Figure 6.12. **Comparison to Cho et al. (2011) and Yuan et al. (2008).** This figure compares non-blind deblurring results, on saturated images, for (b) the algorithm of Cho et al. (2011), (c) the Richardson-Lucy-based algorithm of Yuan et al. (2008), and (d) our proposed "combined RL" algorithm. While both our algorithm and the algorithm of Yuan et al. are based on the Richardson-Lucy algorithm, our results contain much less ringing, due to handling the saturated pixels explicitly. Compared to the method of Cho et al., our results contain similar or less ringing. Results in (b) and (c) provided by Cho et al.

Noting that $\mathbf{g}^* = \mathbf{a} * \mathbf{f}$, and writing the estimated kernel as a perturbation of $\Delta\mathbf{a}$ from the true kernel: $\hat{\mathbf{a}} = \frac{1}{1+\sum \Delta a_k}(\mathbf{a} + \Delta\mathbf{a})$, we can write

$$\mathcal{F}(\hat{\mathbf{f}}_{\mathbf{g}^*}) = \mathcal{F}(\mathbf{f}) \circ \frac{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\mathbf{a})}{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}}) + \boldsymbol{\phi}} \tag{6.37}$$

$$= \mathcal{F}(\mathbf{f}) \circ \left( \left(1 + \sum \Delta a_k\right) \frac{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}})}{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}}) + \boldsymbol{\phi}} - \frac{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\Delta\mathbf{a})}{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}}) + \boldsymbol{\phi}} \right). \tag{6.38}$$

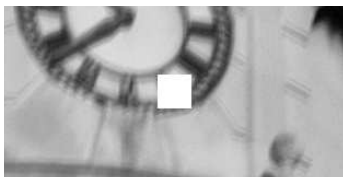
We can write the deconvolved image as the sum of two terms, from the two terms in the parentheses, i.e.

$$\mathcal{F}(\hat{\mathbf{f}}^{(1)}_{\mathbf{g}^*}) = \mathcal{F}(\mathbf{f}) \circ \left(1 + \sum \Delta a_k\right) \frac{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}})}{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}}) + \boldsymbol{\phi}} \tag{6.39}$$

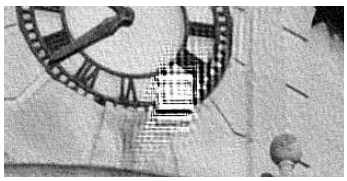
$$\mathcal{F}(\hat{\mathbf{f}}^{(2)}_{\mathbf{g}^*}) = -\mathcal{F}(\mathbf{f}) \circ \frac{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\Delta\mathbf{a})}{\mathcal{F}(\hat{\mathbf{a}})^* \circ \mathcal{F}(\hat{\mathbf{a}}) + \boldsymbol{\phi}} \tag{6.40}$$

$$\hat{\mathbf{f}}_{\mathbf{g}^*} = \hat{\mathbf{f}}^{(1)}_{\mathbf{g}^*} + \hat{\mathbf{f}}^{(2)}_{\mathbf{g}^*} \tag{6.41}$$

(a) Original image $\mathbf{f}$ and kernel $\mathbf{a}$

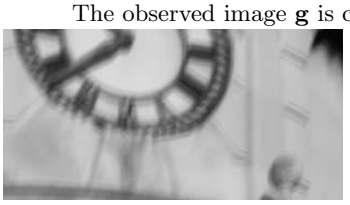


(b) Observed image $\mathbf{g}$

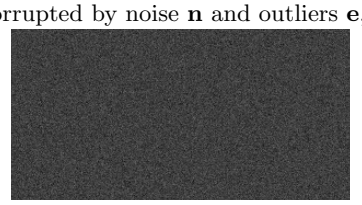


(c) Deconvolved image $\hat{\mathbf{f}}$, using (6.32)

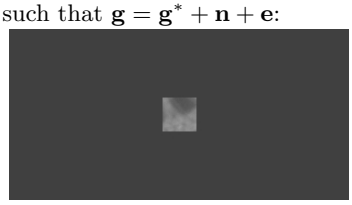
The observed image $\mathbf{g}$ is corrupted by noise $\mathbf{n}$ and outliers $\mathbf{e}$, such that $\mathbf{g} = \mathbf{g}^* + \mathbf{n} + \mathbf{e}$:



(d) Ideal blurry image $\mathbf{g}^*$



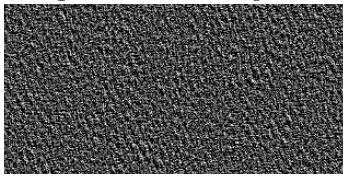
(e) Noise $\mathbf{n}$



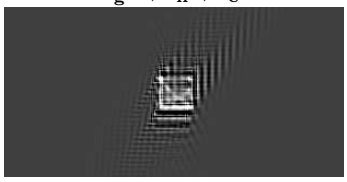
(f) Outliers $\mathbf{e}$

Likewise, the deconvolved image $\hat{\mathbf{f}}$ can be decomposed such that $\hat{\mathbf{f}} = \hat{\mathbf{f}}_{\mathbf{g}^*} + \hat{\mathbf{f}}_{\mathbf{n}} + \hat{\mathbf{f}}_{\mathbf{e}}$:

(g) $\hat{\mathbf{f}}_{\mathbf{g}^*}$

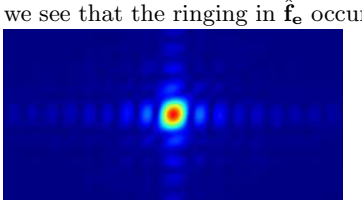


(h) $\hat{\mathbf{f}}_{\mathbf{n}}$

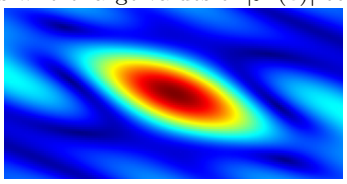


(i) $\hat{\mathbf{f}}_{\mathbf{e}}$

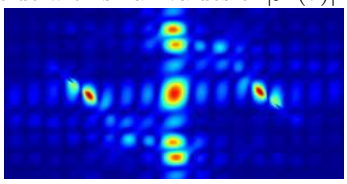
Clearly, the ringing is caused by the outliers $\mathbf{e}$. From the spectra of the outliers $\mathbf{e}$ and the kernel $\mathbf{a}$, we see that the ringing in $\hat{\mathbf{f}}_{\mathbf{e}}$ occurs where large values of $|\mathcal{F}(\mathbf{e})|$ coincide with small values of $|\mathcal{F}(\mathbf{a})|$:



(j) $|\mathcal{F}(\mathbf{e})|$

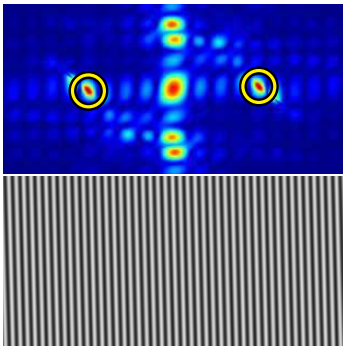


(k) $|\mathcal{F}(\mathbf{a})|$

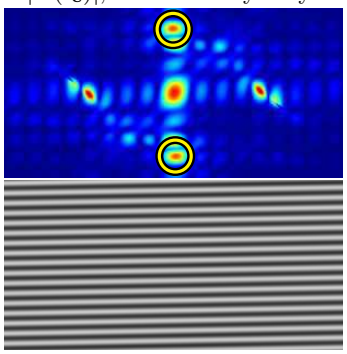


(l) $|\mathcal{F}(\hat{\mathbf{f}}_{\mathbf{e}})|$

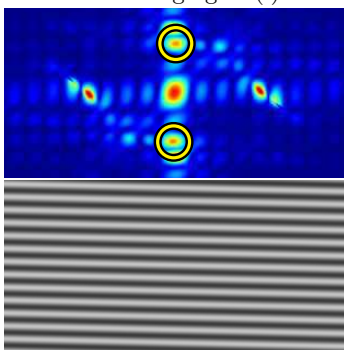
Isolating the spurious peaks of $|\mathcal{F}(\hat{\mathbf{f}}_{\mathbf{e}})|$, we can verify they match the visible ringing in (i):



(m) A pair of peaks of $|\mathcal{F}(\hat{\mathbf{f}}_{\mathbf{e}})|$ and their inverse Fourier transform



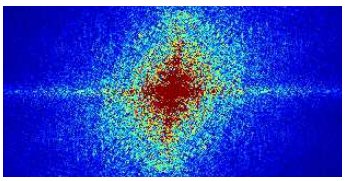
(n) A pair of peaks of $|\mathcal{F}(\hat{\mathbf{f}}_{\mathbf{e}})|$ and their inverse Fourier transform



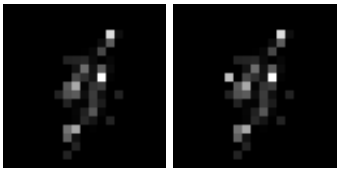
(o) A pair of peaks of $|\mathcal{F}(\hat{\mathbf{f}}_{\mathbf{e}})|$ and their inverse Fourier transform

Figure 6.13. **Synthetic example showing ringing due to outliers.** By decomposing the deconvolved image into the contributions from the "noiseless" blurry image $\mathbf{g}^*$, the noise $\mathbf{n}$, and outliers $\mathbf{e}$, we can see that the ringing is caused entirely by the outliers. By examining the spectra of the outliers and the blur kernel, we can trace the causes of the ringing to places in the frequency domain where small components of the kernel coincide with large components of the outliers.

(a) Original image $\mathbf{f}$

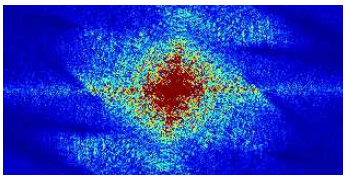


(b) $|\mathcal{F}(\mathbf{f})|$

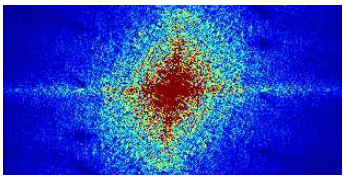


(c) True kernel $\mathbf{a}$ and corrupted kernel $\hat{\mathbf{a}}$

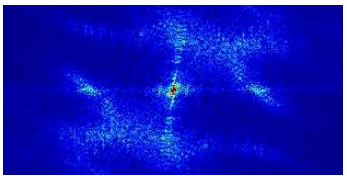
Examining the spectra of the deconvolved image $\hat{\mathbf{f}}_{\mathbf{g}^*}$, and the two components $\hat{\mathbf{f}}^{(1)}_{\mathbf{g}^*}$ and $\hat{\mathbf{f}}^{(2)}_{\mathbf{g}^*}$, we can see that $\hat{\mathbf{f}}^{(1)}_{\mathbf{g}^*}$ is very similar to the original sharp image (slightly brighter due to the $(1 + \sum \Delta a_k)$ scale factor), while the ringing is caused by $\hat{\mathbf{f}}^{(2)}_{\mathbf{g}^*}$, which amplifies small frequency components of $\hat{\mathbf{a}}$:



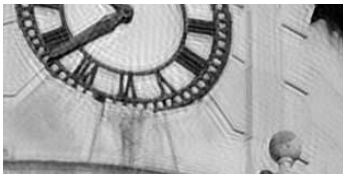
(d) $|\mathcal{F}(\hat{\mathbf{f}}_{\mathbf{g}^*})|$



(e) $|\mathcal{F}(\hat{\mathbf{f}}^{(1)}_{\mathbf{g}^*})|$

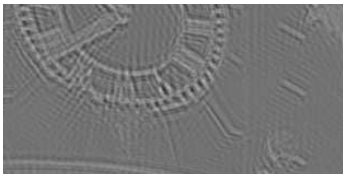


(f) $|\mathcal{F}(\hat{\mathbf{f}}^{(2)}_{\mathbf{g}^*})|$



(g) Deconvolved image $\hat{\mathbf{f}}_{\mathbf{g}^*}$

(h) $\hat{\mathbf{f}}^{(1)}_{\mathbf{g}^*}$



(i) $\hat{\mathbf{f}}^{(2)}_{\mathbf{g}^*}$

Figure 6.14. **Synthetic example showing ringing due to kernel errors.** By decomposing the deconvolved image as described in Equations (6.39) to (6.41), we can see that the ringing is caused when the kernel error $\Delta\mathbf{a}$ contains frequency components which are small in the estimated kernel $\hat{\mathbf{a}}$. .

The first component $\hat{\mathbf{f}}^{(1)}_{\mathbf{g}^*}$ is close to the true sharp image scaled by $(1 + \sum \Delta a_k)$, except with some frequency components reduced due to the regularisation $\boldsymbol{\phi}$. The second component $\hat{\mathbf{f}}^{(2)}_{\mathbf{g}^*}$ is the source of the ringing, which arises when the error $\Delta\mathbf{a}$ has a broad spectrum (e.g. delta functions) which aligns with the small frequency components of the corrupted kernel $\hat{\mathbf{a}}$. Figure 6.14 demonstrates ringing caused by kernel errors on a synthetic example.

### 6.6.3 Implications for Blind and Non-Blind Deblurring

In the preceding sections we have examined ringing caused by outliers in the blurry image, and errors in the kernel. What do these insights mean for deblurring in practice?

In terms of errors in the kernel, the good news is that unless the errors are large (i.e. the kernel is completely wrong) the ringing will have a small amplitude. If we want to improve our estimate of the kernel, the deblurred image will typically contain the correct edges and texture for us to do so. This can be seen in Figure 6.14 (g), where the step edges and ridges that are used by the blind PSF estimation algorithm of Cho and Lee (2009) are clearly visible, and the ringing does not interfere too much with this. In an iterative PSF estimation algorithm, the next estimate of the kernel would almost certainly be closer to the true kernel.

For ringing caused by outliers in the data, our conclusions must depend on the sources of these outliers. When the true forward model is non-linear, and we assume a linear forward model (as in this section and in most work on deblurring), then outliers will occur anywhere the true forward model is non-linear. Since these outliers are caused by an error in the model rather than a random noise process, the chances of the outliers having structure, and thus potentially a wide bandwidth are much greater. In order to reduce the ringing caused, we can either improve our models to take account of the non-linearities, or attempt to find and discard the non-linear data. These approaches are described in Section 6.2, and this is the approach to handling saturation proposed by Cho et al. (2011). As shown in Section 6.3 however, this is typically insufficient, since the loss of information will cause inaccuracies in the deblurred image, which then propagate as ringing. Thus in our "combined RL" algorithm we take the additional step of attempting to determine which pixels can be estimated accurately, and decoupling their estimates from the poorly-estimated pixels.

When outliers in the data are caused by noise such as "hot pixels", dirt or scratches, or deleted pixels, a non-linear model will not help to prevent ringing. In these cases, the outliers must be prevented from affecting our estimate of the sharp image, and this is the approach taken by Cho et al. (2011). In this chapter we have not addressed deblurring in the presence of such outliers, however we envision that our "combined RL" approach could be extended to handle such phenomena.

## 6.7 Conclusion

In this chapter we have developed an approach for the problem of blind deblurring for images blurred with camera shake and suffering from saturation. We have proposed a non-blind deblurring algorithm, based on the Richardson-Lucy algorithm, which is able to deblur images containing saturated regions without introducing artifacts, without sacrificing detail in the result.

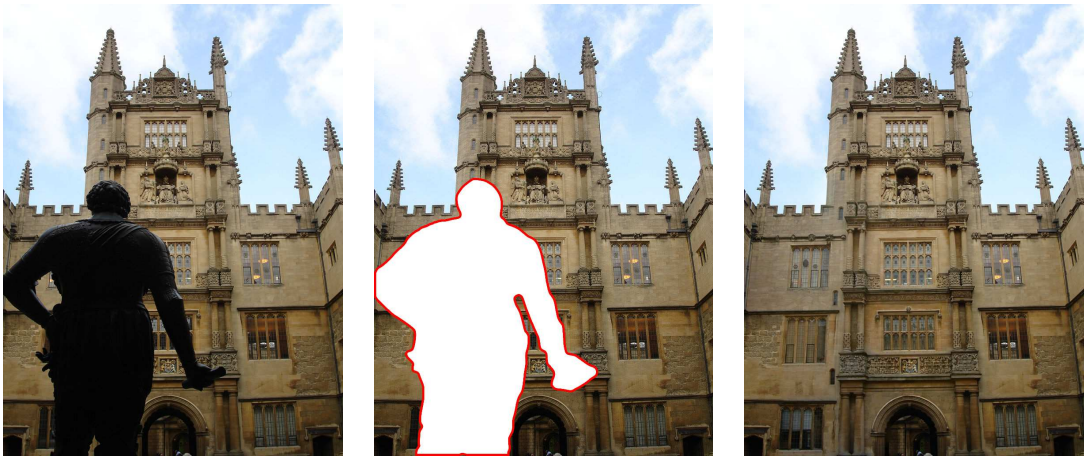
# Chapter 7

# Removing Occluders from Photos of Famous Landmarks

## 7.1 Introduction

Often when we review our holiday photos, we notice things that we wish we could have avoided, such as vehicles, construction work, or simply other tourists. To go back and retake the photograph is impossible, so what can we do if we want to remove these things from our photos? We wish to replace the offending pixels in as convincing a way as possible, a problem often referred to as image completion or inpainting, which has been widely studied recently. However, most existing methods are only applicable to small image regions, or require significant user intervention. We would ideally like to have no limit on the complexity of the image in the regions we are replacing, and would like to be able to fill pixels with whatever would actually have been observed there had the occlusions not occurred, with minimal user interaction. With the growth of online photo sharing websites, we have millions of photos of the world available to us, probably including many of the very same place as our own photo, so how can we use them to fix up our snaps?

In the same way that strong priors are essential to successful blind deblurring algo-



(a) Original query image (b) Target region (c) Result

Figure 7.1. **An example result from our algorithm.** The replaced region is consistent with the rest of the original image and the boundary between is effectively hidden, producing a convincing result. Note the complexity of image structures on the inpainted facade.

rithms, it is not possible to replace a region of an image convincingly without some prior knowledge of the replaced pixels' appearance. While previous authors have focussed on statistical approaches or intuition to model the local behaviour of images (as discussed in Section 2.5), in general we may wish to replace a region of an image that contains structures that are too complex to be reproduced using these approaches. In this chapter we leverage recent advances in viewpoint invariant image search (Chum et al., 2007; Philbin et al., 2007) to find other images of the same scene. Beginning with a *query* image $\mathbf{g}$ containing a *target region* $\Psi$ to be replaced, our aim is to replace the pixels in the target region $\Psi$ in a way that is convincing and corresponds to the true underlying scene. We first use an online image search engine to retrieve images of the same scene, which we refer to as a set of *oracles*. We use this set of oracles to propose a set of solutions $\{\mathbf{p}^{(\mathbf{q})}\}$ to the filling problem. Finally we solve a labelling problem to decide how best to combine the proposals into a single output result $\hat{\mathbf{f}}$. Figure 7.1 shows an example result obtained from our system.

Image completion using images of the same scene from the Internet was independently of our work approached by Amirshahi et al. (2007); Amirshahi et al. (2008), who

obtain relevant images using a text-based image search, then combine unoccluded blocks from them to replace the target region using a greedy algorithm. Our novel contributions are the following: (a) candidate images are obtained automatically using viewpoint invariant image search, (b) geometric registration is performed using multiple homographies, which allows registering significantly more complex scenes captured from varying camera viewpoints, and (c) occlusion reasoning and seamless combination of multiple registered oracle images is formulated as a single labelling problem efficiently solved using tree-reweighted belief propagation.

We begin by describing the viewpoint invariant image search used in Section 7.2, before describing our method of geometrically and photometrically registering each search result to the query image in Section 7.3. In Section 7.4 we discuss how to combine the proposals generated from all the retrieved images into a single output image. We present some results and failure cases for our system in Section 7.5, and conclude in Section 7.6

## 7.2 Retrieving Oracle Images

Recent works (Chum et al., 2007; Jégou et al., 2008; Philbin et al., 2007) have demonstrated the feasibility of large-scale image-based search engines, allowing the user to input an image region as a search query, and be presented with other images depicting the same object or scene. Such viewpoint invariant image search is now becoming commercially available, e.g. Google goggles[1]. In this work we use the "Oxford Buildings Search"[2] online demonstration provided by Philbin et al. (2007). The system uses a set of 100,000 images collected from Flickr[3], for which an indexing structure has already been computed. Both the query images and oracle images from which we construct the solution come from this set. Figure 7.2 shows a typical query image and the top 30 results returned by the search engine. These search results are the *oracles* which are used to replace the target region in the query image $\mathbf{g}$.

[1] `http://www.google.com/mobile/goggles`
[2] `http://www.robots.ox.ac.uk/~vgg/research/oxbuildings/`
[3] `http://www.flickr.com`

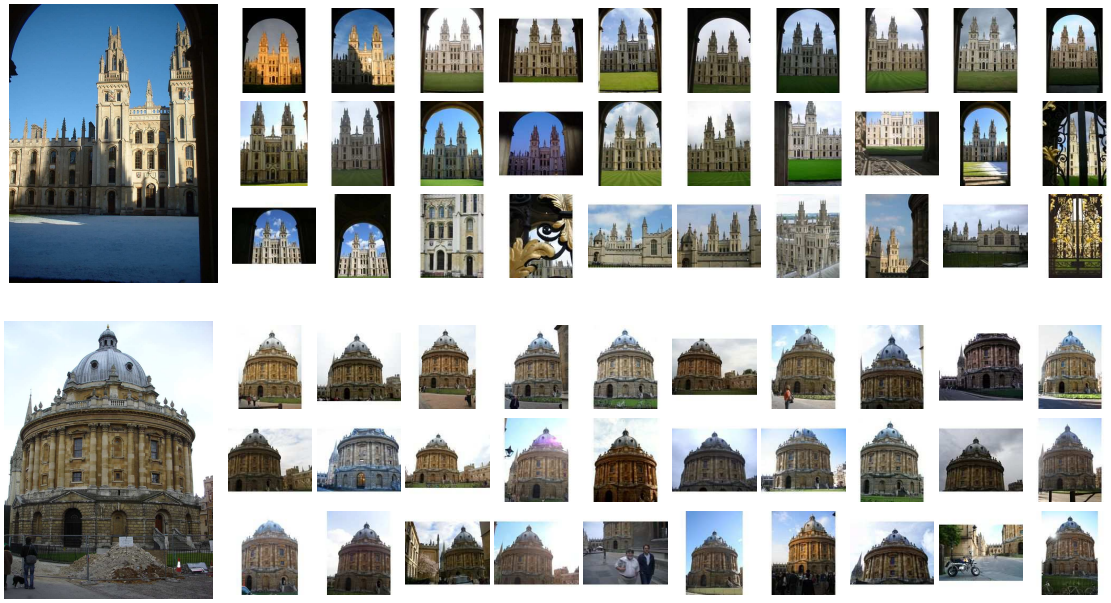


Figure 7.2. **Two example queries and the first 30 results returned by the viewpoint invariant image search engine.** For popular landmarks it is reasonable to expect the top retrieved results to contain the building in the query image, and for some to have been taken from a similar viewpoint. These search results will be used to replace the target region of our query image.

## 7.3 Geometric and Photometric Registration

Having retrieved a set of oracle images depicting the same place as the query image, we first need to deduce their geometrical and photometric relationship to the query image. We would also like to reason jointly about their registration, in order to be able to combine them into a single, coherent, occlusion-free image. In this section, we first explain the use of homographies to register the oracle images to a query image, and the extension to multiple homographies that is necessary to register scenes with multiple planes. We then find regions which have been registered well by each homography, and use these regions to estimate a global transformation on the colour channels of each oracle. Finally, we group together homographies which are likely to have registered the same scene plane, and for each group compute an unoccluded "median" image, which will be used to guide the region selection process in Section 7.4. The output of this part

of the system is a set of geometrically and photometrically registered oracles $\{\mathbf{o}^{(\mathbf{q})}\}$, and a set of unoccluded "median" images which are used as guides in the final part of the system. Note that since we allow multiple homographies, each retrieved oracle may appear multiple times in the set of registered oracles $\{\mathbf{o}^{(\mathbf{q})}\}$.

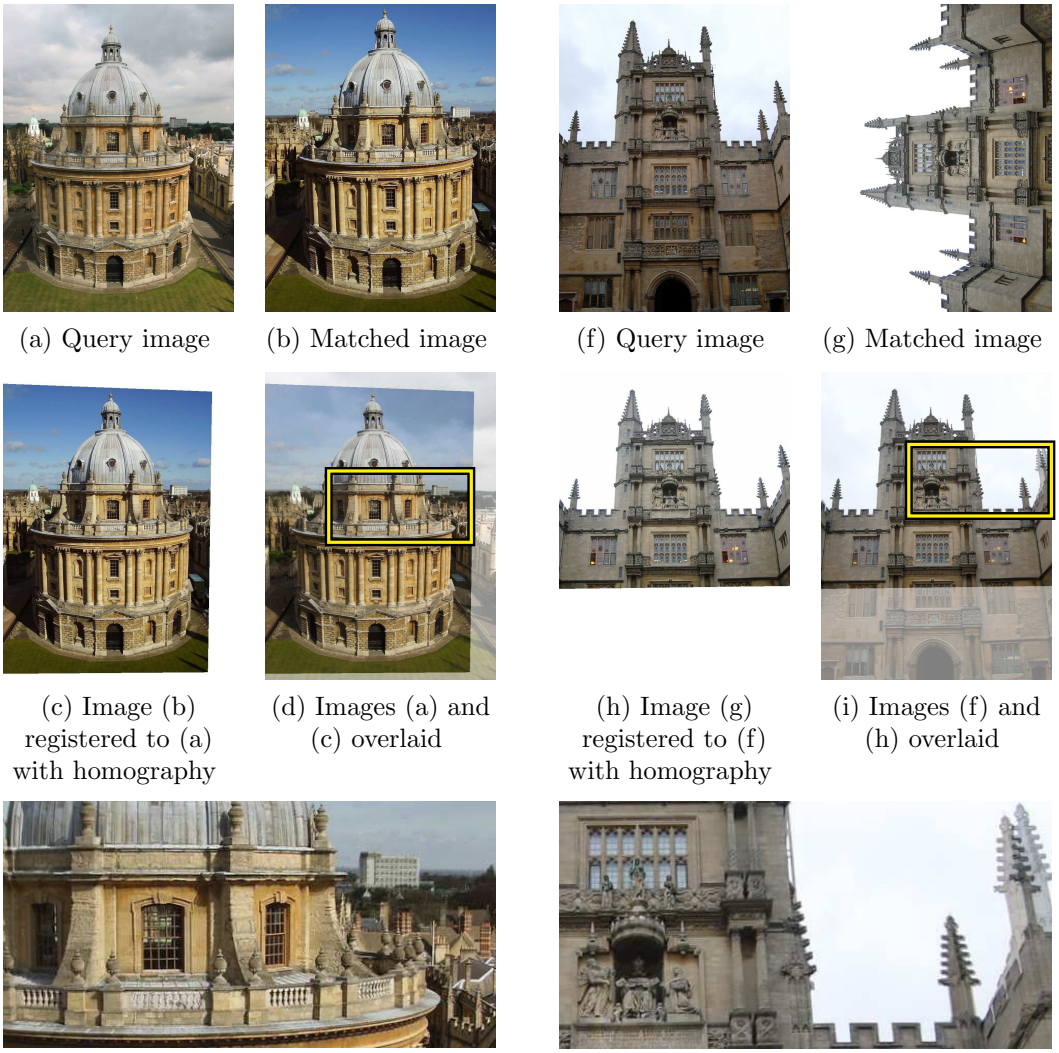
Registering two images with a homography (Hartley and Zisserman, 2004) is valid when the optical centres of the two cameras coincide, or when the scene being imaged is planar, as shown in Figure 7.3. In outdoor urban scenes, the second condition is approximately satisfied very frequently. Additionally, when we have a large number of photographs of a scene taken by many photographers, we find that the first condition can also sometimes be approximately satisfied. Although we are unlikely to find a photo taken from exactly the same location, we are likely to find many for which the effects of parallax will be small.

### 7.3.1 Homography estimation

We use the standard method (Hartley and Zisserman, 2004) of determining putative point matches between the two images (query and oracle) and estimating the inliers and homography simultaneously using RANSAC (Fischler and Bolles, 1981). Here we use Harris-Affine (Mikolajczyk and Schmid, 2004) and SIFT (Lowe, 2004) interest points (typically 5000-15000 per image) and SIFT descriptors.[4]

Having estimated the homography using the inlying interest points, we check that the number of inliers is sufficient for it to be reliable, discarding any homographies with fewer than 50 inliers. We also check whether the line at infinity in the oracle becomes visible under the homography, and discard it if so, since we deem the transformation to be too extreme in this case. Finally, we retain the oracle only if it covers at least 25% of the target region after warping, since an oracle covering less than this is unlikely to be much use for replacing the region's contents.

[4]Binaries for the Harris-Affine detector / SIFT descriptor obtained from `http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html`, binaries for the SIFT detector / descriptor obtained from `http://www.cs.ubc.ca/~lowe/keypoints/`, both accessed April 2008.



(a) Query image (b) Matched image (f) Query image (g) Matched image

(c) Image (b) registered to (a) with homography (d) Images (a) and (c) overlaid (h) Image (g) registered to (f) with homography (i) Images (f) and (h) overlaid

(e) (j)

Figure 7.3. **Pairs of images related by homographies.** In each column, the first row shows a query photograph and another photograph of the same scene returned by the image search engine. Below, we show the second image registered to the first using a homography, and the two images overlaid. The last row shows a close-up of the overlaid images. Left column: A pair of images with approximately the same camera centre – the transformation applies almost equally well to all parts of the image. Right column: A pair of images of a piecewise planar scene – the homography registers the dominant wall, where the variation in depth is small, but is no longer valid for points not on that plane, such as the protruding wall on the right.

### 7.3.2 Multiple homographies

For a scene containing multiple planes, a single homography will in general be insufficient to register an oracle to the whole query image. Thus we allow multiple homographies to be detected for each oracle. After estimating a homography between the two images, we remove all the inlying interest points from consideration, and run RANSAC again on the remaining putative matches. We repeat this process until the last estimated homography is rejected by the criteria above.

### 7.3.3 Ground plane registration

While simply removing inliers to previous homographies is sufficient for RANSAC to register the different planes of buildings, it often fails to register the ground plane, where there are generally very few interest points, which can get lost amongst the large number of putative matches. However, given that many occlusions are caused by people or objects standing on the ground, it may be important for the ground plane to be well registered. In the situation where the ground plane is not registered and causes significant problems, we allow the user to interactively segment the ground region in the query and oracle images, and run RANSAC again using only the interest points in that region. This process is illustrated in Figure 7.4, where removing the putative matches from other parts of the image enables RANSAC to find the homography which registers the ground plane.

### 7.3.4 Photometric registration

Due to the diverse origins of the images retrieved from the Internet, they may have very different lighting conditions from the query image. To reduce the effect of lighting variations when combining oracle images with the query, we work in the gradient domain, and furthermore estimate a global linear correction on the gradients of each colour channel of each oracle image. The correction is robustly estimated by first estimating which regions are well-registered in each oracle, in a lighting-invariant way. To do this we compute

the normalised cross-correlation (NCC) between the square $(15 \times 15)$ patch around each pixel in the query image with the corresponding square patch in the transformed oracle image. To convert the resulting NCC scores into a mask of well-registered regions, we use a binary Markov random field (MRF) with unary costs which are linear in the NCC scores, and a Potts potential on adjacent pixels, and minimise the cost function using graph-cuts. Figure 7.5 shows typical binary masks indicating the well-registered regions for each oracle.

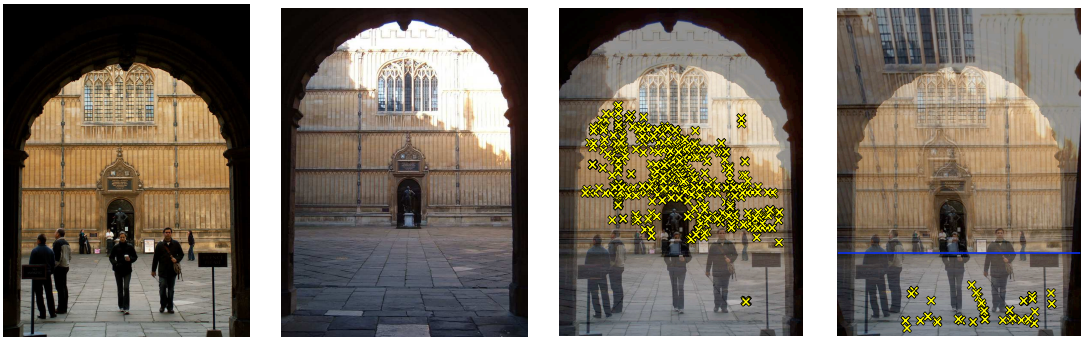
Given the mask of well-registered pixels, we estimate the linear photometric correction by taking the median ratio of gradient magnitudes between the query image $\mathbf{g}$ and the registered oracle $\mathbf{o}$, in the *well-registered* region, i.e. $\text{median}_i \left( \|(\nabla \mathbf{g})_i\| \,/\, \|(\nabla \mathbf{o})_i\| \right)$, where $\|(\nabla \mathbf{g})_i\|$ is the magnitude of the gradient of image $\mathbf{g}$ at pixel $i$. The correction is performed separately in each colour channel.

### 7.3.5 Grouping homographies

The goal here is to group homographies across multiple oracles which correspond to a particular scene plane, and subsequently obtain an unoccluded "median" image for each group, which will be used to guide selection of unoccluded and well-registered oracles for inpainting the target region (Section 7.4).

In order to group together homographies which correspond to a particular scene plane, we consider each oracle-homography combination as a node of a graph, and for two homographies having as inliers the sets of interest points $S$ and $Q$, place an edge between the two nodes if the overlap of these sets is above a threshold: $\frac{|S \cap Q|}{|S \cup Q|} > 0.1$. We consider each connected component of the graph as a group of homographies likely to register the same scene plane.

To compute the median image for a homography group, we follow the approach proposed by Weiss (2001) for computing occlusion-free "intrinsic images". For a group of homographies $G$, we first compute the $x$ and $y$ derivatives of all the registered oracles in that group. At each pixel we take the median $x$ derivative and the median $y$ derivative over all the registered oracles, before using Poisson blending (Pérez et al., 2003)



(a) Query image (b) Retrieved oracle (c) 1st homography (d) 2nd homography

Figure 7.4. **Semi-automatic ground plane registration.** This figure shows (a) a query image, (b) an oracle image, (c) the first homography extracted with inliers shown, and (d) the second homography extracted after the user manually indicates ground plane region (below horizontal line). The first homography extracts the dominant plane, and by manually indicating the ground plane region RANSAC is able to register the ground plane in the second homography.

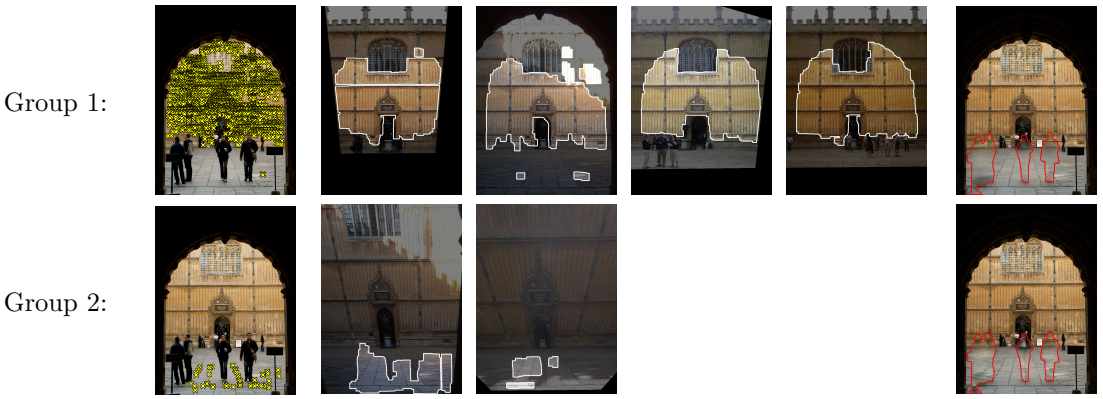



Figure 7.5. **Grouping homographies and finding well-registered regions.** From left to right on each row: The query image with each homography group's inlying interest points marked, some of the registered oracles from each group, with the regions considered to be well-registered highlighted, and (far right) the "median" image for each group within the target region. Note that for each group, the median image provides a sharp, unoccluded estimate of the relevant plane, while it is blurry elsewhere. Thus, the difference between a registered oracle and this image will be small where the oracle is well-registered and unoccluded, but large elsewhere .

to reconstruct the "median" proposal $\mathbf{p}_i^{\text{median}(G)}$ for that group. Figure 7.5 shows an example of two homography groups extracted for a scene with two dominant planes, and the median proposals for each.

## 7.4 Generating and Combining Proposals

Once we have an oracle geometrically and photometrically registered, we would like to use each registered oracle $\mathbf{o}^{(q)}$ to generate a proposal $\mathbf{p}^{(q)}$ for how the target region should be filled. The most direct way of doing this would be to simply copy the pixels from the oracle into the region, i.e. $\mathbf{p}^{(q)} = \mathbf{m} \circ \mathbf{o}^{(q)} + \overline{\mathbf{m}} \circ \mathbf{g}$, where $\mathbf{m}$ is a binary mask corresponding to the target region $\Psi$. However in practice the variations in lighting mean that this approach will produce poor quality results, with clear boundaries at the edge of the region. The problem of how best to combine two images whose properties do not necessarily match has been approached in many ways, from methods which aim to conceal boundaries between regions, such as Burt and Adelson's multiresolution spline (Burt and Adelson, 1983) and Poisson blending (Pérez et al., 2003), to methods that try and find the best place to locate the boundary itself, such as the dynamic programming of Efros and Freeman (2001) and the graph-cut technique of Kwatra et al. (2003).

In this work we use Poisson blending to combine the images, whereby instead of combining pixels from the two images, their gradient fields are combined to form a composite gradient field $\nabla\mathbf{p}^{(q)} = \mathbf{m} \circ \nabla\mathbf{o}^{(q)} + \overline{\mathbf{m}} \circ \nabla\mathbf{g}$. The composite gradient field $\nabla\mathbf{p}^{(q)}$ can then be reconstructed into an image $\mathbf{p}^{(q)}$ by solving Poisson's equation. The query image provides Dirichlet boundary conditions (which constrain the colour of the solution) for the equation around the target region. If the transformed oracle does not span the entire target region, pixels bordering the remaining unfilled region (where we have no colour information) take Neumann boundary conditions (which constrain the gradient of the solution), in order to reduce colour artefacts. Figure 7.6 (f) shows some of the individual proposals generated for a query image using Poisson blending.

### 7.4.1 Combining Multiple Proposals

Following the steps described in the previous sections, we have a set of proposals $\{\mathbf{p}^{(\mathbf{q})}\}$, where each proposal is generated using a single registered oracle. However, it may be that individual oracles cannot provide the best solution when taken alone, but may be

combined into a single result which could not have been achieved using any single oracle. Reasons for this might be occlusions in the oracles themselves, or the fact that a single homography may not be able to register the whole target region. Figure 7.6 shows the advantage of combining multiple proposals; a single oracle provides most of the result but requires other oracles to provide some small parts, partly due to the mis-registration of the ground plane.

In order to decide which proposal should be used at which pixel, we want to design and optimise an energy function which encourages each pixel to choose well, but to regularise this with the idea that pixels should agree with their neighbours about what their neighbourhood should look like. We can consider this a labelling problem, where the label $l_i$ for a pixel $i$ corresponds to which proposal is used there. This can be formulated as a multi-label conditional random field (CRF), where we wish to find the optimal label configuration $\mathbf{l}$ by solving a problem of the form

$$\min_{\mathbf{l}} \sum_{i \in \Psi} E_1\big(i, l_i\big) + \sum_{(i,i') \in \mathcal{E}} E_2\big(i, i', l_i, l_{i'}\big), \tag{7.1}$$

where $\Psi$ indicates the set of pixels in the region being solved, $(i, i')$ indicates a pair of neighbouring pixels (4-neighbours), with $\mathcal{E}$ being the set of all such pairs in the region being solved. $E_1\big(i, l\big)$ is the "cost" of using the proposal $\mathbf{p}^{(l)}$ at pixel $i$, encoding our wishes for individual pixels, while $E_2\big(i, i', l_i, l_{i'}\big)$ is the cost of using proposals $\mathbf{p}^{(l_i)}$ and $\mathbf{p}^{(l_{i'})}$ at neighbouring pixels $i$ and $i'$, encoding the way we wish neighbouring pixels to agree with each other.

Pixels outside the target region should look similar to the original query image, since they lie outside the region originally specified for replacement. Pixels on the inside of the target region however should be similar to some robust estimate of the unoccluded scene, to avoid inserting new occlusions into the image. To achieve these two goals we choose $E_1$ to have the form

$$E_1\big(i, l_i\big) = k_{\text{query}} \overline{m}_i \| p_i^{(l_i)} - g_i \| + k_{\text{median}} m_i \| p_i^{(l_i)} - p_i^{\text{median}(G(l_i))} \|, \tag{7.2}$$

where $\mathbf{m}$ is a binary mask indicating the target region, and $\overline{\mathbf{m}}$ is its logical negation. Outside the target region, where $m_i = 0$, the cost depends on the difference between $p_i^{(l_i)}$ (the colour of proposal $l_i$ at pixel $i$) and $g_i$ (the colour of the query image $\mathbf{g}$ at the pixel). This term penalises any differences between the result and the input query image, while allowing the optimisation to choose the best location for the boundary of the replaced region. Inside the target region, where $m_i = 1$, the cost depends on the difference between $p_i^{(l_i)}$ and $p_i^{\text{median}(G(l_i))}$, which is the "median" image for that oracle's homography group $G(l_i)$ as described in Section 7.3.5. This term serves a dual purpose, helping both to avoid inserting new occlusions into the result, and avoid using any proposals outside the region where they are registered, since in both these cases, the deviation $\|p_i^{(l_i)} - p_i^{\text{median}(G(l_i))}\|$ should be large. Further to this we set $E_1\big(i, l_i\big)$ to a large number if proposal $l_i$ does not cover pixel $i$. This is effectively a hard constraint which prevents a registered oracle being used for regions outside its bounds. The parameters $k_{\text{query}}$ and $k_{\text{median}}$ weight the terms according to their relative importance. The norm $\|p_i - g_i\|$ is simply the Euclidean distance in RGB space.

The purpose of $E_2$ is to encourage a few large regions to be combined instead of many small ones, and to ensure that boundaries between regions from different proposals occur at places where they will be least obvious. For this we use the "gradient" cost function suggested by Agarwala et al. (2004), where $E_2 = 0$ if $l_i = l_{i'}$, and otherwise

$$E_2\big(i, i', l_i, l_{i'}\big) = k_{\text{grad}} \left( \left\| (\nabla \mathbf{p}^{(l_i)})_i - (\nabla \mathbf{p}^{(l_{i'})})_i \right\| + \left\| (\nabla \mathbf{p}^{(l_i)})_{i'} - (\nabla \mathbf{p}^{(l_{i'})})_{i'} \right\| \right), \tag{7.3}$$

where $(\nabla \mathbf{p})_i$ is the concatenation of the image gradients at pixel $i$ in all colour channels, i.e. a 6D vector. The first term penalises the difference between the two proposals' gradients at pixel $i$, and the second term penalises the difference between the two proposals' gradients at pixel $i'$. This cost helps prevent boundaries between regions using different proposals, if the two proposals' image gradients differ at that location. By encouraging these boundaries to fall in places where the image gradients in the two proposals match well, the transitions between regions are hidden. For the results in this paper we used

$k_{\text{query}} = 1$, $k_{\text{median}} = 1$, and $k_{\text{grad}} = 10$, and for computational purposes we used at most 10 proposals from each homography group (obtained as in Section 7.3.5), ranked by the number of homography inliers.

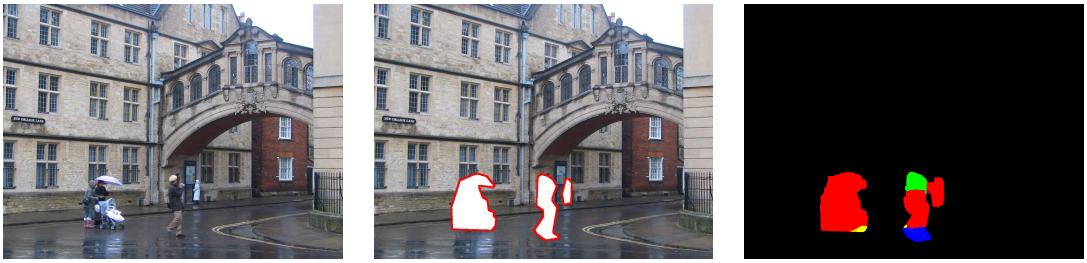
Finally, in order to achieve a good transition between the region that has been filled and the query image surrounding it, the query image itself is included as a proposal, and the CRF is solved over a larger region than the original target region. By generating the proposals such that they extend outside the target region, the optimisation may choose the best place in this boundary region at which to switch from the query image to the proposals. To optimise the CRF described above, we use tree-reweighted belief propagation (Kolmogorov, 2006; Szeliski et al., 2006; Wainwright et al., 2005), using the software made available online[5] by Szeliski et al. (2006).

## 7.5 Results and Discussion

In this section, we demonstrate our method on several query images from the Oxford Buildings image set, with various difficulties which our method is able to overcome. In Figure 7.1, the region to be replaced is spanned almost entirely by a single scene plane. In this case, the image search returned 49 photos of the same scene, of which 15 were accurately geometrically registered to the query image using single homographies. The result comes mostly from a single oracle, which was automatically selected by our CRF formulation since it provided a consistent, occlusion free replacement for the target region.

Figure 7.6 shows the advantage of our method's ability to combine multiple proposals. The search returned 48 photos of the same scene, of which 9 were registered to the query image, providing 16 proposals due to multiple homographies. Most of the result comes from a single proposal, but the other proposals are used to fill some regions where this fails demonstrating the benefit of using multiple oracles. The final composite obtained by our method is significantly better than results obtained by the approaches of Criminisi

[5] `http://vision.middlebury.edu/MRF/code/`



(a) Query Image (b) Target Regions (c) Labels

(d) Oracles

(e) Registered Oracles

(f) Proposals

(g) Our Result (h) Criminisi et al. (2003) (i) Hays and Efros (2007)

Figure 7.6. **Combining multiple proposals.** The inputs to the system (a) & (b) and the output labels (c) showing the combination of proposals in the final result (colours correspond to the borders of the images below). (d) The top 5 original (unregistered) automatically retrieved oracle images used in the result. (e) The geometrically registered oracles. (f) The proposals generated by each oracle. Note that none of the individual proposals covers the entire target region and provides a satisfactory result. Last row: (g) Our final result, (h) the result using the algorithm of Criminisi et al. (2003) and (i) the result using the method of Hays and Efros (2007). In our result, we obtain a much better image by combining regions from several proposals, and choosing the boundaries to be as inconspicuous as possible. The result of Criminisi et al. propagates strong edges into the target region, but cannot reason about where best to connect them or insert details that are hidden (e.g. the grate at the base of the wall). The method of Hays and Efros produces a result that fits well with the surrounding image, but does not correspond to the actual underlying scene. We are grateful to Hays and Efros for running their system on our input image to produce the result in (i).

(a) Query Image (b) Target Regions (c) Our Result (d) Labels

(e) Oracles

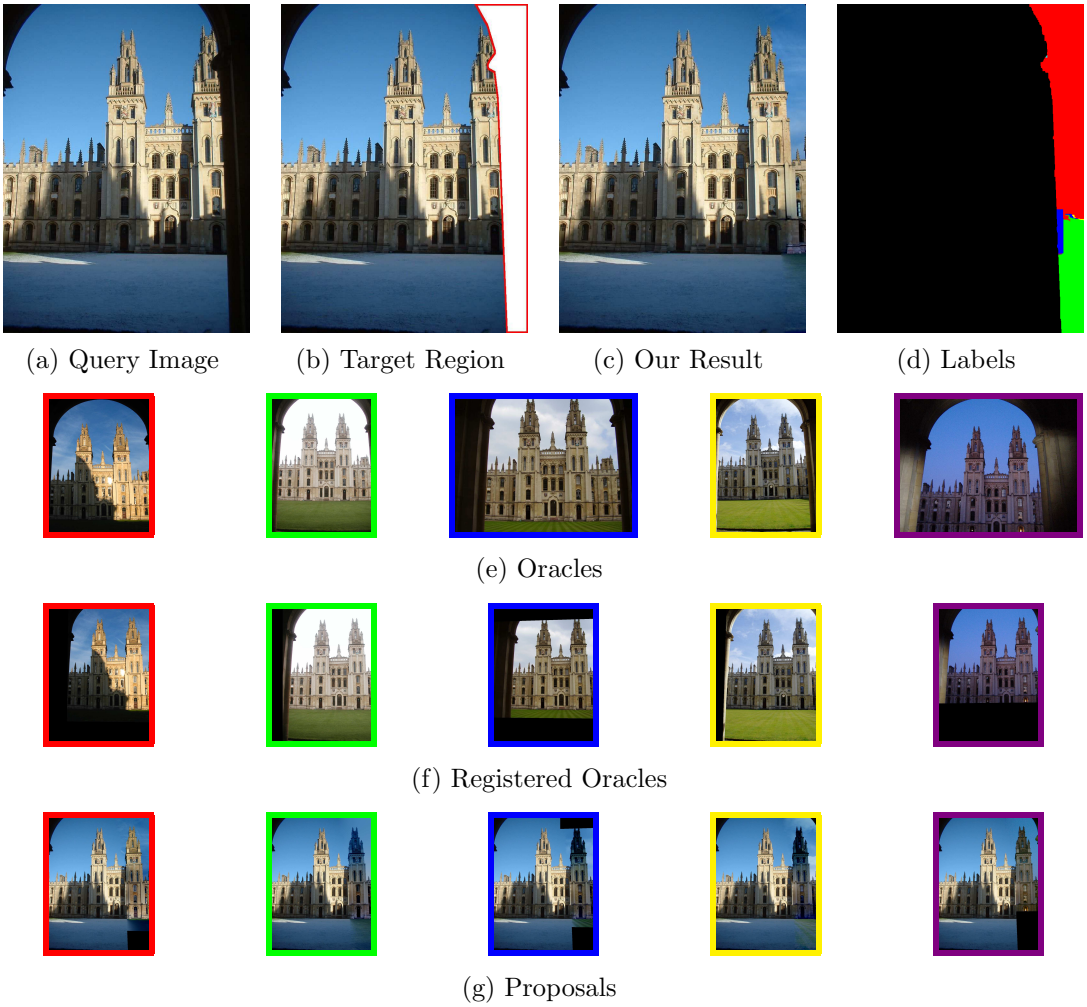
(f) Registered Oracles

(g) Proposals

Figure 7.7. **Example Result 1.** An example query and the corresponding result from our system. The CRF optimisation correctly chooses oracles registered to the ground plane where relevant (red, blue), and combines other oracles registered to the wall (green, yellow, magenta) in order to complete the result .

et al. (2003) and Hays and Efros (2007) (see the last row of Figure 7.6).

Figures 7.7 and 7.8 show two additional results. In the first example, the search returned 36 results, of which 18 were registered automatically and 2 homographies registering the ground plane were obtained semi-automatically (see Section 7.3). Thanks to the homography grouping combined with the CRF optimisation most of the occluders were convincingly removed. In the second result, the search returned 47 images of the



(a) Query Image (b) Target Region (c) Our Result (d) Labels

(e) Oracles

(f) Registered Oracles

(g) Proposals

Figure 7.8. **Example Result 2.** An example query and the corresponding result from our system. Our method successfully completes the image with the unoccluded building facade, however it cannot reproduce the environmental effects particular to the query image, such as the snow and strong shadows.

same scene, 17 of which were accurately registered. Note the successful removal of the occluding arch.

The result shown in Figure 7.9 demonstrates the effect of the unary term $E_1$ defined in Equation (7.2). When the weight $k_{\text{median}}$ is low, the pairwise smoothness term has a greater effect, and the optimisation chooses to combine fewer proposals, while following the median proposal less closely. When the weight $k_{\text{median}}$ is high, the optimisation finds

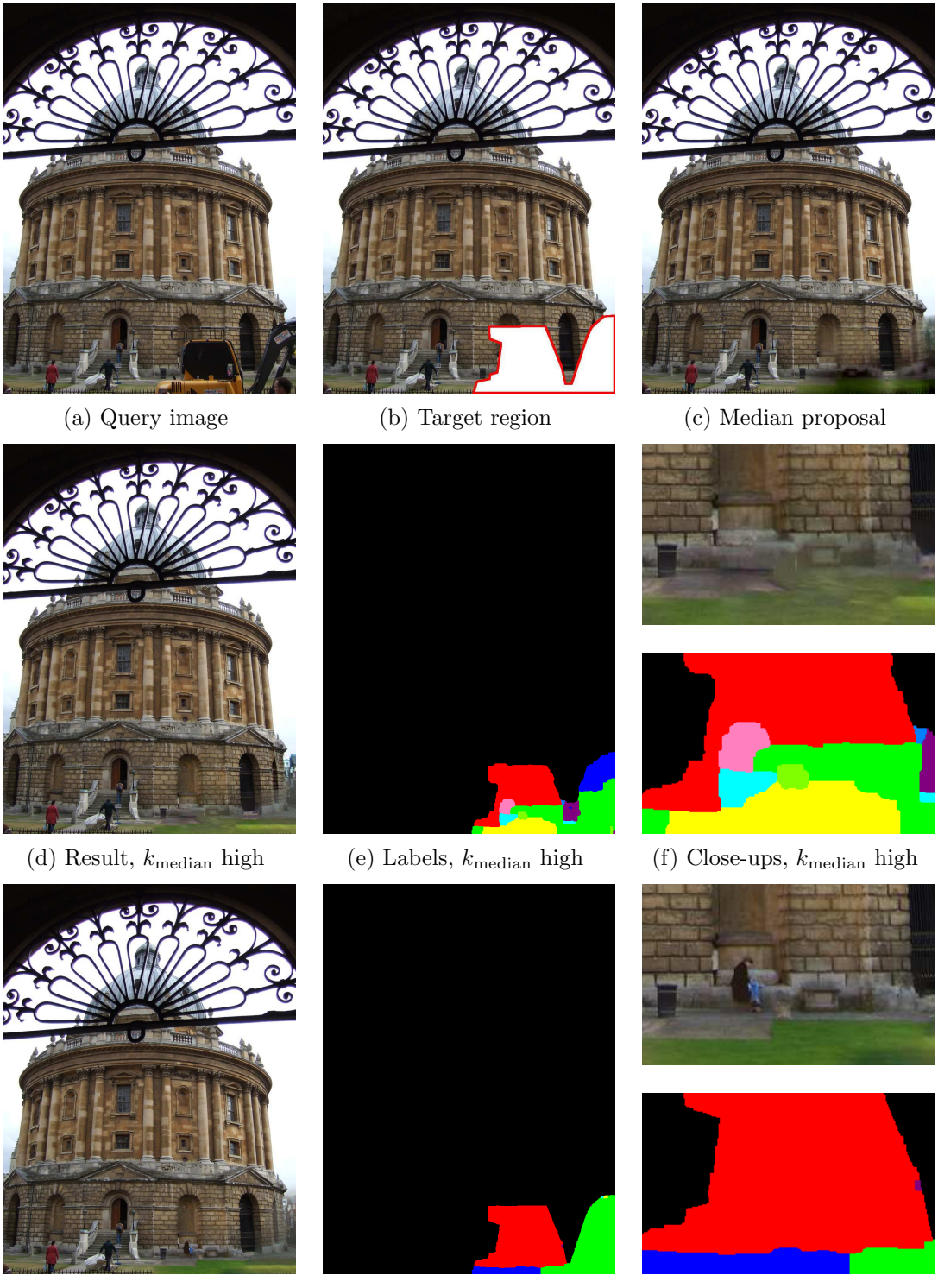
a solution using more proposals, and avoiding introducing new occlusions into the result.

Figure 7.10 shows a typical failure case for our system, which tends to occur when we have only a limited number of oracles available. In this case, there may not be sufficient images to construct an occlusion-free result. Even if there are enough oracles, the median image used to guide the final solution may be misleading, since it will have been computed from relatively few samples, which may include new occlusions of misregistered planes.

## 7.6 Conclusion

We have demonstrated an inpainting method which is able to combine images taken from different viewpoints and under different lighting conditions to convincingly replace large regions of the query photograph containing complex image structures, thereby significantly extending the capabilities of local patch-based inpainting methods (Criminisi et al., 2003; Efros and Leung, 1999).

The approach is mainly applicable to tourist snapshots of frequently visited landmarks, which have many pictures taken by other people available on the Internet. Although results are visually pleasing, in some cases under a close inspection subtle artefacts remain in the final composites. These are mainly due to photometric and resolution issues such as differences in the length of shadows and different image resolution/focus.

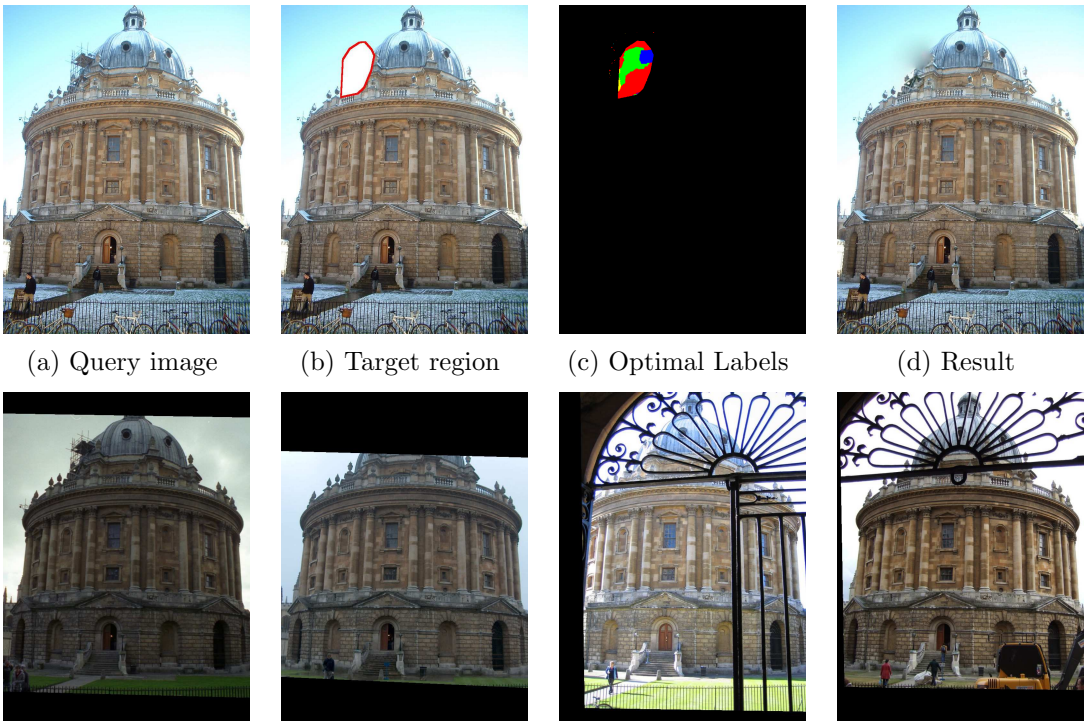


(a) Query image (b) Target region (c) Median proposal

(d) Result, $k_{\text{median}}$ high (e) Labels, $k_{\text{median}}$ high (f) Close-ups, $k_{\text{median}}$ high

(g) Result, $k_{\text{median}}$ low (h) Labels, $k_{\text{median}}$ low (i) Close-ups, $k_{\text{median}}$ low

Figure 7.9. **Example result demonstrating the effect of unary term.** In this example, it is possible to see the effect of the unary term, which encourages the solution to look similar to the median proposal. With the unary term at its default value (d)–(f) many proposals are combined and the alcove shown in the close-up is empty. When we lower the weight $k_{\text{median}}$ (g)–(i), the pairwise smoothness term has a greater effect, and the optimisation prefers to use fewer proposals, leading to a person appearing in the alcove, transferred from one of the proposals.



(a) Query image (b) Target region (c) Optimal Labels (d) Result

(e) Oracle 1 (f) Oracle 2 (g) Oracle 3 (h) Oracle 4

Figure 7.10. **A failure case of the system.** In this example, there are only 4 oracles (e)–(h) registered correctly in the target region. As a result, the solution cannot contain a good unoccluded view of the region, and instead contains a combination of the scaffolding from (e) and some mis-registered oracles (not shown).

# Chapter 8

# Perspectives

In this chapter we review the contributions of this thesis, and discuss possible directions for future research.

## 8.1 Contributions

In Chapter 3 we presented a new model for spatially-variant blur caused by camera shake. We showed that camera shake blur is predominantly caused by rotations of the camera during the exposure, as opposed to translations, and that such blur is generally spatially-variant. Starting from the geometry of camera rotations, we modelled the blurry image as a weighted combination of projectively-transformed versions of the sharp image. We demonstrated a practical, discrete version of this model, which parameterises the point-spread function (PSF) of a "shaken" image using a single set of weights.

In Chapter 4 we demonstrated that the model of spatially-variant blur proposed in Chapter 3 can be applied within existing deblurring algorithms, originally designed to estimate only spatially-invariant PSFs. We show that the resulting algorithms are able to estimate spatially-variant PSFs, and that by applying our model in non-blind deblurring algorithms, we can deblur a wider range of images than using a spatially-invariant blur model. We successfully applied our model to both a marginalisation and

a MAP algorithm for blind PSF estimation, as well as in the case where an additional noisy (but sharp) image of the scene is available.

In Chapter 5 we discussed the increase in computational cost incurred by using a spatially-variant blur model instead of a spatially-invariant one. While spatially-invariant blur can be computed quickly in the frequency domain using the fast Fourier transform (FFT), there is no equivalent for our proposed blur model. We proposed an approximation scheme for our spatially-variant blur model that divides the blurry image into a small number of overlapping sub-images, each of which is treated as having spatially-invariant blur. Using this scheme we provide the equations for quickly approximating the most intensive computations in the blind deblurring process. We demonstrated an order of magnitude speed-up of blind deblurring compared to the exact model, while retaining the accuracy and consistency of our global model from Chapter 3.

In Chapter 6 we extended our forward model for camera shake blur to include sensor saturation, which introduces a non-linearity into the image formation process. We derived a modified version of the Richardson-Lucy algorithm which incorporates this non-linear forward model and is able to recover intensities that lie outside the camera's dynamic range. We proposed a second modification to the Richardson-Lucy algorithm which explicitly handles saturated pixels separately from unsaturated ones, and prevents ringing artefacts by decoupling the update equations for the two sets of pixels. Using the proposed algorithm, we demonstrated non-blind deblurring of images with saturated pixels without introducing large ringing artefacts.

Finally in Chapter 7 we tackled the problem of removing large occluders from photographs of popular landmarks. We proposed a system that for popular locations, is able to realistically hallucinate the occluded scene content, without any user input and without an explicit 3D representation of the scene. Our system processes "oracle" images retrieved from a visual search engine by geometrically and photometrically registering them to the target image, before collecting a set of proposals for how to fill the occluded region. In the final step, the proposals are combined using a CRF formulation, which encourages a result free from occlusions, and without any visible region boundaries.

## 8.2 Future Work

One important avenue for future research is to improve the robustness and reliability of blind deblurring algorithms. The automatic algorithms used in Chapter 4 for blind PSF estimation sometimes fail, for example if the blurry image contains too much noise, or if the sharp image does not match the prior that is used. One potential line of work is to investigate the use of more powerful image priors, for example that proposed by Zoran and Weiss (2011), which models the joint distributions of pixels in image patches, rather than just pairs of pixels. Another interesting possibility is to consider the role of user interaction in blind deblurring. For many people, it is natural to interact with an image restoration process, and such a system would have the potential to be more robust to difficult images. Sometimes in these images, the shape of the PSF or the location of strong edges can be perceived visually, and it may be possible for a user to provide some assistance, allowing the PSF to be estimated. The best way of integrating user interaction is an open question, however some potentially-natural modes of interaction might be to (a) draw the approximate shape of the PSF at one or more locations in the image, (b) draw the locations of strong step edges, or (c) draw a rectangle around locations where the shape of the PSF is visible (e.g. trails left by bright spots).

Another possibility for increasing the robustness of the blind deblurring process is to take advantage of the trails left by bright lights in the image. Many images taken at night include bright electric lights, and in Chapter 6 we addressed non-blind deblurring of images that contain such saturated regions. However, the trails left by bright lights in blurry images often give a clear outline of the PSF (although the intensity is lost due to saturation), and could provide a powerful constraint for blind PSF estimation.

In this thesis, we have considered camera shake blur using a model with 3 degrees of freedom, leading to 3-dimensional blur kernels. However, in some cases it may be useful to consider additional degrees of freedom, such as camera translation (for example when photographing from a fast-moving vehicle) or varying focal length (for example if the photographer accidentally nudged a manual zoom lens while taking a photograph). In

these cases the dimensionality of the kernel would grow, and it may not be practical to estimate the high-dimensional kernel explicitly. Once the kernel dimension reaches a certain level, it may no longer provide the most compact parameterisation of the PSF. Instead, it might be easier to parameterise the PSF using a set of local filters, such as in Chapter 5. The challenge to making such a parameterisation effective is the need to enforce global consistency between the filters, without knowing the global kernel itself. This would be done with pairwise constraints between all pairs of filters, and could be approached in a purely numerical way, simply exploiting the fact that each filter is related to the kernel as $\mathbf{a}^{(r)} = \mathbf{J}_r\mathbf{w}$. Alternatively, it may be possible to utilise the underlying geometry, for example to find a sort of epipolar geometry between pairs of filters.

# Appendix A

# Parameter Update Derivation for Marginalisation Algorithm

In this appendix we derive the optimal forms and parameters of the approximating distributions $q(\mathbf{f})$, $q(\mathbf{w})$ and $q(\beta_\sigma)$, used in the marginalisation algorithm for spatially-variant blind deblurring in Chapter 4. For our spatially-variant blur model, the optimal distributions for the latent variables are the same as for spatially-invariant blur (*cf*. Equations (42, 43, 17) of (Miskin and MacKay, 2000)):

$$q(w_k) \propto p(w_k) \exp\left(-\frac{1}{2} w_k^{(2)} \left(w_k - w_k^{(1)}\right)^2\right) \tag{A.1}$$

$$q(f_j) \propto p(f_j) \exp\left(-\frac{1}{2} f_j^{(2)} \left(f_j - f_j^{(1)}\right)^2\right) \tag{A.2}$$

$$q(\beta_\sigma) = \Gamma\left(\beta_\sigma; \frac{1}{2} \sum_i \left\langle (g_i - g_i^*)^2 \right\rangle_{q(\mathbf{f},\mathbf{w})}, \frac{N}{2}\right), \tag{A.3}$$

where $w_k^{(1)}$, $w_k^{(2)}$, $f_j^{(1)}$, $f_j^{(2)}$ are parameters of the distributions, $g_i^*$ is the "noiseless" value of blurry pixel $i$, related to the unknown latent image $\mathbf{f}$ and kernel $\mathbf{w}$ through the forward model in Equation (3.11) (p. 66). Note that $\mathbf{f}$ and $\mathbf{w}$ are random variables, so in this context $g_i^*$ is also a random variable. $N$ is the number of observed blurry pixels, and $\langle\cdot\rangle_q$ represents the expectation with respect to the the distribution $q$. For each latent

variable, the parameters of its distribution depend on the distributions of all the other latent variables, e.g. $w_k^{(1)}$ and $w_k^{(2)}$ depend on $q(\beta_\sigma)$, $q(f_j)$ for all $j$ and $q(w_{k'})$ for all $k' \neq k$. For our non-uniform blur model, we find the following optimal values for the parameters, as given earlier in Equations (4.2) to (4.5) (*cf*. Equations (46–49) of (Miskin and MacKay, 2000)):

$$w_k^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big\langle \Big( \sum_j T_{ij}^{(k)} f_j \Big)^2 \Big\rangle_{q(\mathbf{f})} \tag{A.4}$$

$$w_k^{(1)} w_k^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big( g_i \sum_j T_{ij}^{(k)} \langle f_j \rangle_{q(f_j)} - \sum_{k' \neq k} \Big\langle \Big( \sum_j T_{ij}^{(k)} f_j \Big) \Big( \sum_j T_{ij}^{(k')} f_j \Big) \Big\rangle_{q(\mathbf{f})} \langle w_{k'} \rangle_{q(w_{k'})} \Big) \tag{A.5}$$

$$f_j^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big\langle \Big( \sum_k T_{ij}^{(k)} w_k \Big)^2 \Big\rangle_{q(\mathbf{w})} \tag{A.6}$$

$$f_j^{(1)} f_j^{(2)} = \langle \beta_\sigma \rangle \sum_i \Big( g_i \sum_k T_{ij}^{(k)} \langle w_k \rangle_{q(w_k)} - \sum_{j' \neq j} \langle f_{j'} \rangle_{q(f_{j'})} \Big\langle \Big( \sum_k T_{ij'}^{(k)} w_k \Big) \Big( \sum_k T_{ij}^{(k)} w_k \Big) \Big\rangle_{q(\mathbf{w})} \Big). \tag{A.7}$$

The details of the derivation are given next.

## A.1 Variational method

For convenience, we will collect the latent variables $\mathbf{f}$, $\mathbf{w}$, and $\beta_\sigma$ into the "ensemble" $\boldsymbol{\Theta}$. The aim is to approximate the true posterior $p(\boldsymbol{\Theta}|\mathbf{g})$ with a simpler factorized distribution $q(\boldsymbol{\Theta}|\mathbf{g})$, denoted for simplicity as $q(\boldsymbol{\Theta}) = q(\beta_\sigma) \prod_j q(f_j) \prod_k q(w_k)$. Our model, from Equation (3.12), provides the likelihood $p(\mathbf{g}|\boldsymbol{\Theta})$:

$$g_i^* = \sum_{j,k} w_k T_{ij}^{(k)} f_j, \tag{A.8}$$

$$p(\mathbf{g}|\boldsymbol{\Theta}) = \prod_i \mathcal{G}\big(g_i; g_i^*, \beta_\sigma^{-1}\big), \qquad \text{see Equation (7) of (Miskin and MacKay, 2000)} \tag{A.9}$$

where $\mathcal{G}(\,\cdot\,;\mu,\sigma^2)$ is a Gaussian with mean $\mu$ and variance $\sigma^2$. In order to get to the posterior, we also need a prior $p(\mathbf{\Theta})$ for our latent variables. The latent variables are assumed to be independent, so that the prior factorizes:

$$p(\mathbf{\Theta}) = p(\mathbf{f})p(\mathbf{w})p(\beta_\sigma), \tag{A.10}$$

and furthermore the elements of both $\mathbf{f}$ and $\mathbf{w}$ are assumed to be independent and identically-distributed, i.e.

$$p(\mathbf{f}) = \prod_j p(f_j) \tag{A.11}$$

$$p(\mathbf{w}) = \prod_k p(w_k). \tag{A.12}$$

From Equation (10) of (Miskin and MacKay, 2000), we wish to minimize the following cost function, first using the calculus of variations to find the optimal form of the approximate distributions, then iteratively optimizing their parameters, which is equivalent to minimizing the Kullback-Leibler (KL) divergence between the posterior and the approximating distribution (see Bishop, 2006, Equation. (10.3), page 463):

$$C_{\mathrm{KL}} = \int q(\mathbf{\Theta}) \left[ \ln \frac{q(\mathbf{\Theta})}{p(\mathbf{\Theta})} - \ln p(\mathbf{g}|\mathbf{\Theta}) \right] \mathrm{d}\mathbf{\Theta}. \tag{A.13}$$

## A.2 Inside the Cost Function

Since $q(\mathbf{\Theta}) = q(\mathbf{f})q(\mathbf{w})q(\beta_\sigma)$ and $p(\mathbf{\Theta}) = p(\mathbf{f})p(\mathbf{w})p(\beta_\sigma)$,

$$C_{\mathrm{KL}} = \int q(\mathbf{\Theta}) \left[ \ln \frac{q(\mathbf{f})}{p(\mathbf{f})} + \ln \frac{q(\mathbf{w})}{p(\mathbf{w})} + \ln \frac{q(\beta_\sigma)}{p(\beta_\sigma)} - \ln p(\mathbf{g}|\mathbf{\Theta}) \right] \mathrm{d}\mathbf{\Theta}. \tag{A.14}$$

$$C_{\mathrm{KL}} = \int q(\mathbf{f}) \ln \frac{q(\mathbf{f})}{p(\mathbf{f})} \,\mathrm{d}\mathbf{f} + \int q(\mathbf{w}) \ln \frac{q(\mathbf{w})}{p(\mathbf{w})} \,\mathrm{d}\mathbf{w} + \int q(\beta_\sigma) \ln \frac{q(\beta_\sigma)}{p(\beta_\sigma)} \,\mathrm{d}\beta_\sigma - \int q(\mathbf{\Theta}) \ln p(\mathbf{g}|\mathbf{\Theta}) \,\mathrm{d}\mathbf{\Theta}. \tag{A.15}$$

Similarly, since $q(\mathbf{f}) = \prod_j q(f_j)$, $q(\mathbf{w}) = \prod_k p(w_k)$, $p(\mathbf{f}) = \prod_j p(f_j)$, and $p(\mathbf{w}) = \prod_k p(w_k)$,

$$C_{\mathrm{KL}} = \sum_j \int q(f_j) \ln \frac{q(f_j)}{p(f_j)} \,\mathrm{d}f_j + \sum_k \int q(w_k) \ln \frac{q(w_k)}{p(w_k)} \,\mathrm{d}w_k + \int q(\beta_\sigma) \ln \frac{q(\beta_\sigma)}{p(\beta_\sigma)} \,\mathrm{d}\beta_\sigma - \int q(\boldsymbol{\Theta}) \ln p(\mathbf{g}|\boldsymbol{\Theta}) \,\mathrm{d}\boldsymbol{\Theta}. \tag{A.16}$$

Finally, we expand the last term:

$$p(\mathbf{g}|\boldsymbol{\Theta}) = \prod_i \mathcal{G}(g_i; g_i^*, \beta_\sigma^{-1}), \tag{A.17}$$

$$\ln p(\mathbf{g}|\boldsymbol{\Theta}) = \sum_i \ln \mathcal{G}(g_i; g_i^*, \beta_\sigma^{-1}) \tag{A.18}$$

$$= \frac{1}{2} \sum_i \left( \ln \beta_\sigma - \beta_\sigma (g_i - g_i^*)^2 - \ln 2\pi \right) \tag{A.19}$$

$$\int q(\boldsymbol{\Theta}) \ln p(\mathbf{g}|\boldsymbol{\Theta}) \,\mathrm{d}\boldsymbol{\Theta} = \frac{1}{2} \int q(\boldsymbol{\Theta}) \sum_i \left( \ln \beta_\sigma - \beta_\sigma (g_i - g_i^*)^2 - \ln 2\pi \right) \mathrm{d}\boldsymbol{\Theta} \tag{A.20}$$

$$= \frac{1}{2} \int q(\beta_\sigma) \sum_i \left( \ln \beta_\sigma - \beta_\sigma \int q(\mathbf{f}) q(\mathbf{w}) (g_i - g_i^*)^2 \,\mathrm{d}\mathbf{f} \,\mathrm{d}\mathbf{w} \right) \mathrm{d}\beta_\sigma - \frac{1}{2} \sum_i \ln 2\pi. \tag{A.21}$$

Putting (A.21) into (A.16) and ignoring terms independent of $\boldsymbol{\Theta}$,

$$C_{\mathrm{KL}} = \sum_j \int q(f_j) \ln \frac{q(f_j)}{p(f_j)} \,\mathrm{d}f_j + \sum_k \int q(w_k) \ln \frac{q(w_k)}{p(w_k)} \,\mathrm{d}w_k + \int q(\beta_\sigma) \ln \frac{q(\beta_\sigma)}{p(\beta_\sigma)} \,\mathrm{d}\beta_\sigma - \frac{1}{2} \int q(\beta_\sigma) \sum_i \left( \ln \beta_\sigma - \beta_\sigma \int q(\mathbf{f}) q(\mathbf{w}) (g_i - g_i^*)^2 \,\mathrm{d}\mathbf{f} \,\mathrm{d}\mathbf{w} \right) \mathrm{d}\beta_\sigma \tag{A.22}$$

## A.3 Optimal Distributions

### A.3.1 Optimal $q(\beta_\sigma)$

To derive the optimal form of $q(\beta_\sigma)$, we ignore terms in $C_{\mathrm{KL}}$ independent of $\beta_\sigma$, add a Lagrange multiplier for the constraint that $\int q(\beta_\sigma) \,\mathrm{d}\beta_\sigma = 1$, and differentiate with

respect to $q(\beta_\sigma)$ :

$$C_{\mathrm{KL}}\big(q(\beta_\sigma)\big) = \int q(\beta_\sigma)\left[\ln\frac{q(\beta_\sigma)}{p(\beta_\sigma)} - \frac{1}{2}\sum_i\big(\ln\beta_\sigma - \beta_\sigma\big\langle(g_i - g_i^*)^2\big\rangle\big)\right]\mathrm{d}\beta_\sigma + \lambda_\sigma\left(\int q(\beta_\sigma)\,\mathrm{d}\beta_\sigma - 1\right), \tag{A.23}$$

where $\langle\cdot\rangle$ denotes the expectation under the approximating distribution $q(\boldsymbol{\Theta})$.

$$\begin{aligned} C_{\mathrm{KL}}\big(q(\beta_\sigma)+\delta q(\beta_\sigma)\big) &= \int q(\beta_\sigma)\left[\ln\frac{q(\beta_\sigma)+\delta q(\beta_\sigma)}{p(\beta_\sigma)} - \frac{1}{2}\sum_i\big(\ln\beta_\sigma - \beta_\sigma\big\langle(g_i - g_i^*)^2\big\rangle\big)\right]\mathrm{d}\beta_\sigma \\ &\quad + \int \delta q(\beta_\sigma)\left[\ln\frac{q(\beta_\sigma)+\delta q(\beta_\sigma)}{p(\beta_\sigma)} - \frac{1}{2}\sum_i\big(\ln\beta_\sigma - \beta_\sigma\big\langle(g_i - g_i^*)^2\big\rangle\big)\right]\mathrm{d}\beta_\sigma \\ &\quad + \lambda_\sigma\left(\int q(\beta_\sigma)\,\mathrm{d}\beta_\sigma + \int \delta q(\beta_\sigma)\,\mathrm{d}\beta_\sigma - 1\right) \end{aligned} \tag{A.24}$$

$$\ln\big(q(\beta_\sigma)+\delta q(\beta_\sigma)\big) \simeq \ln q(\beta_\sigma) + \frac{\delta q(\beta_\sigma)}{q(\beta_\sigma)} \qquad \text{to first order, so} \tag{A.25}$$

$$\begin{aligned} C_{\mathrm{KL}}\big(q(\beta_\sigma)+\delta q(\beta_\sigma)\big) &= C_{\mathrm{KL}}\big(q(\beta_\sigma)\big) + \int \delta q(\beta_\sigma)\,\mathrm{d}\beta_\sigma \\ &\quad + \int \delta q(\beta_\sigma)\left[\ln\frac{q(\beta_\sigma)}{p(\beta_\sigma)} + \frac{\delta q(\beta_\sigma)}{q(\beta_\sigma)} - \frac{1}{2}\sum_i\big(\ln\beta_\sigma - \beta_\sigma\big\langle(g_i - g_i^*)^2\big\rangle\big)\right]\mathrm{d}\beta_\sigma \\ &\quad + \lambda_\sigma\int \delta q(\beta_\sigma)\,\mathrm{d}\beta_\sigma. \end{aligned} \tag{A.26}$$

Discarding higher order terms in $\delta q$,

$$\delta C_{\mathrm{KL}} = \int \delta q(\beta_\sigma)\left[1 + \ln\frac{q(\beta_\sigma)}{p(\beta_\sigma)} - \frac{1}{2}\sum_i\big(\ln\beta_\sigma - \beta_\sigma\big\langle(g_i - g_i^*)^2\big\rangle\big) + \lambda_\sigma\right]\mathrm{d}\beta_\sigma \tag{A.27}$$

$$\frac{\partial C_{\mathrm{KL}}}{\partial q(\beta_\sigma)} = 1 + \ln\frac{q(\beta_\sigma)}{p(\beta_\sigma)} - \frac{1}{2}\sum_i\big(\ln\beta_\sigma - \beta_\sigma\big\langle(g_i - g_i^*)^2\big\rangle\big) + \lambda_\sigma. \tag{A.28}$$

Setting this derivative to zero, we obtain an relation similar to Equation (14) in (Miskin and MacKay, 2000):

$$\ln q(\beta_\sigma) = \ln p(\beta_\sigma) + \frac{1}{2}\sum_i \left(\ln \beta_\sigma - \beta_\sigma \left\langle (g_i - g_i^*)^2 \right\rangle\right) - 1 - \lambda_\sigma. \tag{A.29}$$

Thus the optimal distribution is

$$q(\beta_\sigma) \propto p(\beta_\sigma)\beta_\sigma^{\frac{N}{2}} \exp\left(-\frac{1}{2}\beta_\sigma \sum_i \left\langle (g_i - g_i^*)^2 \right\rangle\right), \tag{A.30}$$

which, given that $p(\ln \beta_\sigma) = 1$, which implies that $p(\beta_\sigma) = \Gamma(\beta_\sigma; \epsilon, \epsilon)$ with $\epsilon \to 0$ (Equation (8) of (Miskin and MacKay, 2000)), gives Equation (17) of (Miskin and MacKay, 2000):

$$q(\beta_\sigma) = \Gamma\left(\beta_\sigma; \frac{1}{2}\sum_i \left\langle (g_i - g_i^*)^2 \right\rangle, \frac{N}{2}\right), \tag{A.31}$$

where the $\Gamma$ distribution is given by Equation (15) of (Miskin and MacKay, 2000):

$$\Gamma(x; a, b) = \frac{1}{\Gamma(b)} a^b x^{(b-1)} \exp(-ax). \tag{A.32}$$

### A.3.2 Optimal $q(f_j)$

Starting from Equation (A.22), and isolating the relevant terms,

$$\begin{aligned} C_{\mathrm{KL}}\big(q(f_j)\big) = \int q(f_j) \ln \frac{q(f_j)}{p(f_j)} \,\mathrm{d}f_j + \frac{1}{2}\langle \beta_\sigma \rangle \sum_i \int q(\mathbf{f}) q(\mathbf{w}) (g_i - g_i^*)^2 \,\mathrm{d}\mathbf{f}\,\mathrm{d}\mathbf{w} \\ + \lambda_j \left(\int q(f_j)\,\mathrm{d}f_j - 1\right). \end{aligned} \tag{A.33}$$

For convenience, we partition $\mathbf{f}$ into $f_j$, the pixel of interest, and $\mathbf{f}_{j^*}$, the remaining pixels:

$$C_{\mathrm{KL}}\big(q(f_j)\big) = \int q(f_j)\left[\ln\frac{q(f_j)}{p(f_j)} + \frac{1}{2}\langle\beta_\sigma\rangle\sum_i \int q(\mathbf{f}_{j^*})q(\mathbf{w})(g_i - g_i^*)^2\,\mathrm{d}\mathbf{f}_{j^*}\,\mathrm{d}\mathbf{w}\right]\mathrm{d}f_j + \lambda_j\left(\int q(f_j)\,\mathrm{d}f_j - 1\right) \tag{A.34}$$

$$\begin{aligned} C_{\mathrm{KL}}\big(q(f_j)+\delta q(f_j)\big) &= \int q(f_j)\left[\ln\frac{q(f_j)}{p(f_j)} + \frac{\delta q(f_j)}{q(f_j)} + \frac{1}{2}\langle\beta_\sigma\rangle\sum_i \int q(\mathbf{f}_{j^*})q(\mathbf{w})(g_i - g_i^*)^2\,\mathrm{d}\mathbf{f}_{j^*}\,\mathrm{d}\mathbf{w}\right]\mathrm{d}f_j \\ &\quad + \int \delta q(f_j)\left[\ln\frac{q(f_j)}{p(f_j)} + \frac{\delta q(f_j)}{q(f_j)} + \frac{1}{2}\langle\beta_\sigma\rangle\sum_i \int q(\mathbf{f}_{j^*})q(\mathbf{w})(g_i - g_i^*)^2\,\mathrm{d}\mathbf{f}_{j^*}\,\mathrm{d}\mathbf{w}\right]\mathrm{d}f_j \\ &\quad + \lambda_j\left(\int q(f_j)\,\mathrm{d}f_j + \int \delta q(f_j)\,\mathrm{d}f_j - 1\right) \end{aligned} \tag{A.35}$$

$$\begin{aligned} &= C_{\mathrm{KL}}\big(q(f_j)\big) + (1+\lambda_j)\int \delta q(f_j)\,\mathrm{d}f_j \\ &\quad + \int \delta q(f_j)\left[\ln\frac{q(f_j)}{p(f_j)} + \frac{1}{2}\langle\beta_\sigma\rangle\sum_i \int q(\mathbf{f}_{j^*})q(\mathbf{w})(g_i - g_i^*)^2\,\mathrm{d}\mathbf{f}_{j^*}\,\mathrm{d}\mathbf{w}\right]\mathrm{d}f_j \end{aligned} \tag{A.36}$$

$$\frac{\partial C_{\mathrm{KL}}}{\partial q(f_j)} = \ln\frac{q(f_j)}{p(f_j)} + \frac{1}{2}\langle\beta_\sigma\rangle\sum_i \int q(\mathbf{f}_{j^*})q(\mathbf{w})(g_i - g_i^*)^2\,\mathrm{d}\mathbf{f}_{j^*}\,\mathrm{d}\mathbf{w} + 1 + \lambda_j. \tag{A.37}$$

Setting this equal to zero, we obtain the optimal form

$$\ln q(f_j) = \ln p(f_j) - \frac{1}{2}\langle\beta_\sigma\rangle\sum_i \int q(\mathbf{f}_{j^*})q(\mathbf{w})(g_i - g_i^*)^2\,\mathrm{d}\mathbf{f}_{j^*}\,\mathrm{d}\mathbf{w} - 1 - \lambda_j. \tag{A.38}$$

Here we need to make some simplifications to obtain a function of $f_j$. For convenience we re-write the forward model in Equation (3.11) (p. 66) as

$$g_i^* = \mathbf{f}^\top \mathbf{C}_i \mathbf{w}, \tag{A.39}$$

where the $N \times K$ matrix $\mathbf{C}_i$ is obtained by re-arranging the elements of the transformation matrices $\mathbf{T}^{(k)}$. We split $g_i^*$ into the contribution from $j$ and $j^*$:

$$g_i^* = \mathbf{f}_{j^*}^\top \mathbf{C}_{ij^*} \mathbf{w} + f_j \mathbf{c}_{ij}^\top \mathbf{w} \tag{A.40}$$

where $\mathbf{c}_{ij}^\top$ is the $j^{\text{th}}$ row of $\mathbf{C}_i$, and $\mathbf{C}_{ij^*}$ is $\mathbf{C}_i$ with this row removed.

$$\begin{aligned}
(g_i - g_i^*)^2 &= g_i^2 - 2 g_i g_i^* + g_i^{*2} && \text{(A.41)} \\
&= g_i^2 - 2 g_i (\mathbf{f}^\top \mathbf{C}_i \mathbf{w}) + (\mathbf{f}^\top \mathbf{C}_i \mathbf{w})^2 && \text{(A.42)} \\
&= g_i^2 - 2 g_i (\mathbf{f}_{j^*}^\top \mathbf{C}_{ij^*} \mathbf{w}) - 2 g_i (f_j \mathbf{c}_{ij}^\top \mathbf{w}) + (\mathbf{f}_{j^*}^\top \mathbf{C}_{ij^*} \mathbf{w})^2 \\
&\quad + 2 (\mathbf{f}_{j^*}^\top \mathbf{C}_{ij^*} \mathbf{w})(f_j \mathbf{c}_{ij}^\top \mathbf{w}) + (f_j \mathbf{c}_{ij}^\top \mathbf{w})^2 && \text{(A.43)} \\
&= -2 g_i (f_j \mathbf{c}_{ij}^\top \mathbf{w}) + 2 (\mathbf{f}_{j^*}^\top \mathbf{C}_{ij^*} \mathbf{w})(f_j \mathbf{c}_{ij}^\top \mathbf{w}) + (f_j \mathbf{c}_{ij}^\top \mathbf{w})^2 + \text{const.} && \text{(A.44)} \\
\left\langle (g_i - g_i^*)^2 \right\rangle_{q(\mathbf{f}_{j^*}, \mathbf{w})} &= \int q(\mathbf{f}_{j^*}) q(\mathbf{w}) (g_i - g_i^*)^2 \, \mathrm{d}\mathbf{f}_{j^*} \, \mathrm{d}\mathbf{w} && \text{(A.45)} \\
&= \left\langle -2 g_i (f_j \mathbf{c}_{ij}^\top \mathbf{w}) + 2 (\mathbf{f}_{j^*}^\top \mathbf{C}_{ij^*} \mathbf{w})(f_j \mathbf{c}_{ij}^\top \mathbf{w}) + (f_j \mathbf{c}_{ij}^\top \mathbf{w})^2 + \text{const.} \right\rangle_{q(\mathbf{f}_{j^*}, \mathbf{w})} && \text{(A.46)} \\
&= -2 f_j \left( g_i \mathbf{c}_{ij}^\top \left\langle \mathbf{w} \right\rangle_{q(\mathbf{w})} - \left\langle \mathbf{f}_{j^*} \right\rangle_{q(\mathbf{f}_{j^*})}^\top \left\langle (\mathbf{C}_{ij^*} \mathbf{w})(\mathbf{c}_{ij}^\top \mathbf{w}) \right\rangle_{q(\mathbf{w})} \right) \\
&\quad + f_j^2 \left\langle (\mathbf{c}_{ij}^\top \mathbf{w})^2 \right\rangle_{q(\mathbf{w})} + \text{const.} && \text{(A.47)} \\
\sum_i \left\langle (g_i - g_i^*)^2 \right\rangle_{q(\mathbf{f}_{j^*}, \mathbf{w})} &= -2 f_j \sum_i \left( g_i \mathbf{c}_{ij}^\top \left\langle \mathbf{w} \right\rangle_{q(\mathbf{w})} - \left\langle \mathbf{f}_{j^*} \right\rangle_{q(\mathbf{f}_{j^*})}^\top \left\langle (\mathbf{C}_{ij^*} \mathbf{w})(\mathbf{c}_{ij}^\top \mathbf{w}) \right\rangle_{q(\mathbf{w})} \right) \\
&\quad + f_j^2 \sum_i \left\langle (\mathbf{c}_{ij}^\top \mathbf{w})^2 \right\rangle_{q(\mathbf{w})} + \text{const.} && \text{(A.48)}
\end{aligned}$$

which is just a quadratic in $f_j$. Replacing the coefficients with $a_j$ and $b_j$,

$$\begin{aligned}
\sum_i \left\langle (g_i - g_i^*)^2 \right\rangle_{q(\mathbf{f}_{j^*}, \mathbf{w})} &= a_j f_j^2 - b_j f_j + \text{const.} && \text{(A.49)} \\
&= a_j \left( f_j - \frac{b_j}{2 a_j} \right)^2 + \text{const.} && \text{(A.50)}
\end{aligned}$$

$$\ln q(f_j) = \ln p(f_j) - \frac{1}{2}\langle\beta_\sigma\rangle a_j \left(f_j - \frac{b_j}{2a_j}\right)^2 + \text{const.} \tag{A.51}$$

$$q(f_j) \propto p(f_j) \exp\left(-\frac{1}{2}\langle\beta_\sigma\rangle a_j \left(f_j - \frac{b_j}{2a_j}\right)^2\right) \tag{A.52}$$

$$\propto p(f_j) \exp\left(-\frac{1}{2} f_j^{(2)} \left(f_j - f_j^{(1)}\right)^2\right) \tag{A.53}$$

*cf*. Equation (43) of (Miskin and MacKay, 2000)

where

$$f_j^{(2)} = \langle\beta_\sigma\rangle a_j \tag{A.54}$$

$$= \langle\beta_\sigma\rangle \sum_i \left\langle (\mathbf{c}_{ij}^\top \mathbf{w})^2 \right\rangle_{q(\mathbf{w})} \tag{A.55}$$

*cf*. Equation (48) of (Miskin and MacKay, 2000)

$$= \langle\beta_\sigma\rangle \sum_i \left\langle \left(\sum_k T_{ij}^{(k)} w_k\right)^2 \right\rangle_{q(\mathbf{w})} \tag{A.56}$$

$$f_j^{(1)} f_j^{(2)} = \frac{1}{2}\langle\beta_\sigma\rangle b_j \tag{A.57}$$

$$= \langle\beta_\sigma\rangle \sum_i \left( g_i \mathbf{c}_{ij}^\top \left\langle \mathbf{w} \right\rangle_{q(\mathbf{w})} - \left\langle \mathbf{f}_{j^*} \right\rangle_{q(\mathbf{f}_{j^*})}^\top \left\langle (\mathbf{C}_{ij^*}\mathbf{w})(\mathbf{c}_{ij}^\top \mathbf{w}) \right\rangle_{q(\mathbf{w})} \right) \tag{A.58}$$

*cf*. Equation (49) of (Miskin and MacKay, 2000)

$$= \langle\beta_\sigma\rangle \sum_i \left( g_i \sum_k T_{ij}^{(k)} \langle w_k \rangle_{q(w_k)} - \sum_{j' \neq j} \langle f_{j'} \rangle_{q(f_{j'})} \left\langle \left(\sum_k T_{ij'}^{(k)} w_k\right) \left(\sum_k T_{ij}^{(k)} w_k\right) \right\rangle_{q(\mathbf{w})} \right). \tag{A.59}$$

### A.3.3 Optimal $q(w_k)$

We proceed much the same as for $q(f_j)$, starting from Equation (A.22), and isolating the relevant terms,

$$C_{\mathrm{KL}}\big(q(w_k)\big) = \int q(w_k) \ln \frac{q(w_k)}{p(w_k)} \, \mathrm{d}w_k + \frac{1}{2}\langle\beta_\sigma\rangle \sum_i \int q(\mathbf{f}) q(\mathbf{w}) (g_i - g_i^*)^2 \, \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{w}$$

$$+ \lambda_k \left( \int q(w_k)\, \mathrm{d}w_k - 1 \right). \tag{A.60}$$

similarly, we partition $\mathbf{w}$ into $w_k$, the element of interest, and $\mathbf{w}_{k^*}$, the remaining elements:

$$C_{\mathrm{KL}}\big(q(w_k)\big) = \int q(w_k) \left[ \ln \frac{q(w_k)}{p(w_k)} + \frac{1}{2}\langle \beta_\sigma \rangle \sum_i \int q(\mathbf{f}) q(\mathbf{w}_{k^*}) (g_i - g_i^*)^2 \, \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{w}_{k^*} \right] \mathrm{d}w_k + \lambda_k \left( \int q(w_k)\, \mathrm{d}w_k - 1 \right) \tag{A.61}$$

and obtain the optimal form

$$\ln q(w_k) = \ln p(w_k) - \frac{1}{2}\langle \beta_\sigma \rangle \sum_i \int q(\mathbf{f}) q(\mathbf{w}_{k^*}) (g_i - g_i^*)^2 \, \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{w}_{k^*} - 1 - \lambda_k. \tag{A.62}$$

Here we need to make some simplifications to obtain a function of $w_k$, as in Appendix A.3.2.

$$g_i^* = \mathbf{f}^\top \mathbf{C}_i \mathbf{w}, \tag{A.63}$$

$$= \mathbf{f}^\top \mathbf{c}_{ik} w_k + \mathbf{f}^\top \mathbf{C}_{ik^*} \mathbf{w}_{k^*} \tag{A.64}$$

where $\mathbf{c}_{ik}$ is the $k^{\text{th}}$ column of $\mathbf{C}_i$, and $\mathbf{C}_{ik^*}$ is $\mathbf{C}_i$ with this column removed.

$$(g_i - g_i^*)^2 = g_i^2 - 2g_i(\mathbf{f}^\top \mathbf{C}_i \mathbf{w}) + (\mathbf{f}^\top \mathbf{C}_i \mathbf{w})^2 \tag{A.65}$$

$$= g_i^2 - 2g_i(\mathbf{f}^\top \mathbf{c}_{ik} w_k) - 2g_i(\mathbf{f}^\top \mathbf{C}_{ik^*} \mathbf{w}_{k^*}) + (\mathbf{f}^\top \mathbf{c}_{ik} w_k)^2 + 2(\mathbf{f}^\top \mathbf{c}_{ik} w_k)(\mathbf{f}^\top \mathbf{C}_{ik^*} \mathbf{w}_{k^*}) + (\mathbf{f}^\top \mathbf{C}_{ik^*} \mathbf{w}_{k^*})^2 \tag{A.66}$$

$$= -2g_i(\mathbf{f}^\top \mathbf{c}_{ik} w_k) + 2(\mathbf{f}^\top \mathbf{c}_{ik} w_k)(\mathbf{f}^\top \mathbf{C}_{ik^*} \mathbf{w}_{k^*}) + (\mathbf{f}^\top \mathbf{c}_{ik} w_k)^2 + \text{const.} \tag{A.67}$$

$$\sum_i \big\langle (g_i - g_i^*)^2 \big\rangle_{q(\mathbf{f}, \mathbf{w}_{k^*})} = -2w_k \sum_i \Big( g_i \big\langle \mathbf{f} \big\rangle_{q(\mathbf{f})}^\top \mathbf{c}_{ik} - \big\langle (\mathbf{f}^\top \mathbf{c}_{ik})(\mathbf{f}^\top \mathbf{C}_{ik^*}) \big\rangle_{q(\mathbf{f})} \big\langle \mathbf{w}_{k^*} \big\rangle_{q(\mathbf{w}_{k^*})} \Big)$$

$$+ w_k^2 \sum_i \left\langle (\mathbf{f}^\top \mathbf{c}_{ik})^2 \right\rangle_{q(\mathbf{f})} + \text{const.} \tag{A.68}$$

$$q(w_k) \propto p(w_k) \exp\left( -\frac{1}{2} w_k^{(2)} \left( w_k - w_k^{(1)} \right)^2 \right) \tag{A.69}$$

*cf*. Equation (42) of (Miskin and MacKay, 2000)

where

$$w_k^{(2)} = \langle \beta_\sigma \rangle \sum_i \left\langle (\mathbf{f}^\top \mathbf{c}_{ik})^2 \right\rangle_{q(\mathbf{f})} \tag{A.70}$$

*cf*. Equation (46) of (Miskin and MacKay, 2000)

$$= \langle \beta_\sigma \rangle \sum_i \left\langle \left( \sum_j T_{ij}^{(k)} f_j \right)^2 \right\rangle_{q(\mathbf{f})} \tag{A.71}$$

$$w_k^{(1)} w_k^{(2)} = \langle \beta_\sigma \rangle \sum_i \left( g_i \left\langle \mathbf{f} \right\rangle_{q(\mathbf{f})}^\top \mathbf{c}_{ik} - \left\langle (\mathbf{f}^\top \mathbf{c}_{ik})(\mathbf{f}^\top \mathbf{C}_{ik^*}) \right\rangle_{q(\mathbf{f})} \left\langle \mathbf{w}_{k^*} \right\rangle_{q(\mathbf{w}_{k^*})} \right) \tag{A.72}$$

*cf*. Equation (47) of (Miskin and MacKay, 2000).

$$= \langle \beta_\sigma \rangle \sum_i \left( g_i \sum_j T_{ij}^{(k)} \langle f_j \rangle_{q(f_j)} - \sum_{k' \neq k} \left\langle \left( \sum_j T_{ij}^{(k)} f_j \right) \left( \sum_j T_{ij}^{(k')} f_j \right) \right\rangle_{q(\mathbf{f})} \langle w_{k'} \rangle_{q(w_{k'})} \right) \tag{A.73}$$

# Bibliography

M. Afonso, J. Bioucas-Dias, and M. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Transactions on Image Processing* 19(9), pp. 2345–2356, 2010. (Cit. on p. 34).

A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)* 23(3), pp. 294–302, 2004. (Cit. on pp. 53, 152).

M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* 54(11), pp. 4311–4322, Nov. 2006. (Cit. on p. 23).

H. Amirshahi, S. Kondo, and T. Aoki. Photo completion using images from internet photo sharing sites. In *Proceedings of the Meeting on Image Recognition and Understanding (MIRU)*, 2007. (Cit. on p. 142).

H. Amirshahi, S. Kondo, K. Ito, and T. Aoki. An image completion algorithm using occlusion-free images from internet photo sharing sites. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E91-A(10), pp. 2918–2927, Oct. 2008. (Cit. on p. 142).

G. R. Ayers and J. C. Dainty. Iterative blind deconvolution method and its applications. *Optics Letters* 13(7), 1988. (Cit. on p. 34).

M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424, 2000. (Cit. on pp. 50, 51).

M. Bertalmío, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing* 12(8), pp. 882–889, Aug. 2003. (Cit. on p. 52).

D. S. C. Biggs and M. Andrews. Acceleration of iterative image restoration algorithms. *Applied Optics* 36(8), pp. 1766–1775, Mar. 1997. (Cit. on p. 28).

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, Aug. 2006. (Cit. on pp. 28, 38, 106, 167).

C. Boncelet. Image noise models. In *Handbook of Image and Video Processing*. Ed. by A. C. Bovik. Second ed. Elsevier Academic Press, 2005. (Cit. on p. 18).

P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics* 2(4), pp. 217–236, 1983. (Cit. on p. 150).

J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, 2009. (Cit. on pp. 25, 35, 39).

A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 2010. (Cit. on p. 45).

T. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing* 7(3), Mar. 1998. (Cit. on p. 55).

C. Chen and O. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications* 5(2), pp. 97–138, Mar. 1996. (Cit. on p. 120).

S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)* 28(5), 145:1–145:8, Dec. 2009. (Cit. on pp. 25, 34, 37, 39–42, 71, 72, 75–81, 83, 85, 91, 99, 102, 104, 112, 138).

S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *Proceedings of the 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil, 2007. (Cit. on p. 45).

S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. In *Proceedings of the 13th International Conference on Computer Vision*. Barcelona, Spain, 2011. (Cit. on pp. 21, 127, 131, 132, 135, 138).

O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil, 2007. (Cit. on pp. 53, 142, 143).

A. Criminisi, P. Pérez, and K. Toyama. Object removal by exemplar-based inpainting. In *Proceedings of the 16th IEEE Conference on Computer Vision and Pattern Recognition*. Madison, WI, pp. 721–728, 2003. (Cit. on pp. 52, 153, 154, 157).

B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics* 32(2), pp. 407–499, 2004. (Cit. on pp. 77, 99).

A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 341–346, Aug. 2001. (Cit. on p. 150).

A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the 7th International Conference on Computer Vision*. Kerkyra, Greece, pp. 1033–1038, Sept. 1999. (Cit. on pp. 52, 157).

R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)* 25(3), pp. 787–794, 2006. (Cit. on pp. 22, 23, 25, 34, 36–38, 42, 43, 55, 71–73, 75, 77–80, 83–85, 91, 112).

P. J. S. Ferreira and A. J. Pinho. Errorless restoration algorithms for band-limited images. In *Proceedings of the 1st IEEE International Conference on Image Processing*. Austin, TX, USA, 1994. (Cit. on p. 51).

D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A* 4(12), pp. 2379–2394, 1987. (Cit. on p. 21).

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6), pp. 381–395, 1981. (Cit. on p. 145).

D. Fish, A. Brinicombe, E. Pike, and J. Walker. Blind deconvolution by means of the Richardson-Lucy algorithm. *Journal of the Optical Society of America A* 12(1), pp. 58–65, 1995. (Cit. on p. 34).

Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 2010. (Cit. on p. 53).

T. W. Gamelin. *Complex Analysis*. New York: Springer-Verlag, 2001. (Cit. on p. 30).

R. Garg, H. Du, S. M. Seitz, and N. Snavely. The dimensionality of scene appearance. In *Proceedings of the 12th International Conference on Computer Vision*. Kyoto, Japan, 2009. (Cit. on p. 53).

D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing* (4), pp. 932–946, 1995. (Cit. on p. 32).

P. Getreuer. *tvreg v2: Variational Imaging Methods for Denoising, Deconvolution, Inpainting, and Segmentation*, 2010. (Cit. on p. 29).

D. N. Godard. Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *IEEE Transactions on Communications* 28(11), pp. 1867–1875, 1980. (Cit. on p. 42).

M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-view stereo for community photo collections. In *Proceedings of the 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil, 2007. (Cit. on p. 53).

R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., 1992. (Cit. on pp. 9, 12).

S. Gull and J. Skilling. Maximum entropy method in image processing. *Communications, Radar and Signal Processing, IEE Proceedings F* 131(6), pp. 646–659, Oct. 1984. (Cit. on pp. 21, 34).

A. Gupta, N. Joshi, C. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proceedings of the 11th European Conference on Computer Vision*. Crete, Greece, 2010. (Cit. on pp. 35, 45, 47, 67, 68).

S. Harmeling, M. Hirsch, and B. Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Advances in Neural Information Processing Systems*. Vancouver, Canada, 2010a. (Cit. on p. 102).

S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf. Multiframe blind deconvolution, super-resolution, and saturation correction via incremental EM. In *Proceedings of the IEEE International Conference on Image Processing*, 2010b. (Cit. on p. 112).

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second ed. Cambridge University Press, 2004. (Cit. on pp. 61, 145).

J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)* 26(3), 2007. (Cit. on pp. 53, 154, 155).

A. N. Hirani and T. Totsuka. Combining frequency and spatial domain information for fast interactive image noise removal. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 269–276, 1996. (Cit. on pp. 51, 53).

M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 2010. (Cit. on pp. 47, 48, 97, 101, 102).

H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proceedings of the 10th European Conference on Computer Vision*. Marseille, France, Oct. 2008. (Cit. on pp. 53, 143).

J. Jia and C.-K. Tang. Image repairing: robust image synthesis by adaptive $N$D tensor voting. In *Proceedings of the 16th IEEE Conference on Computer Vision and Pattern Recognition*. Madison, WI, pp. 643–650, 2003. (Cit. on p. 52).

N. Joshi, S. B. Kang, C. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)* 29(4), 30:1–30:9, 2010. (Cit. on pp. 45–47, 59, 60, 67, 85, 86).

N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *Proceedings of the 21st IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, AK, 2008. (Cit. on p. 42).

E. Kee, S. Paris, S. Chen, and J. Wang. Modeling and removing spatially-varying optical blur. In *Proceedings of the IEEE International Conference on Computational Photography*. Pittsburgh, PA, 2011. (Cit. on p. 45).

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale $\ell_1$-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing* 1(4), pp. 606–617, Dec. 2007. (Cit. on pp. 40, 77, 90).

G. Klein and T. Drummond. A single-frame visual gyroscope. In *Proceedings of the 16th British Machine Vision Conference*. Oxford, 2005. (Cit. on p. 47).

V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(10), pp. 1568–1583, Oct. 2006. (Cit. on p. 153).

D. Krishnan and R. Fergus. Fast image deconvolution using hyper-Laplacian priors. In *Advances in Neural Information Processing Systems*. Vancouver, Canada, 2009. (Cit. on pp. 22, 23, 32, 33, 77, 78, 91, 113, 126, 128–130).

D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO, 2011. (Cit. on pp. 22, 35, 42).

V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)* 22(3), pp. 277–286, July 2003. (Cit. on p. 150).

H. Lappalainen and J. W. Miskin. Ensemble learning. In *Advances in Independent Component Analysis*. Ed. by M. Girolani. Springer-Verlag, 2000. (Cit. on p. 17).

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*. Vancouver, Canada, 2001. (Cit. on p. 26).

A. Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*. Vancouver, Canada, 2006. (Cit. on p. 45).

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 (Proceedings of SIGGRAPH 2007)*, 2007. (Cit. on pp. 22, 23, 31, 33).

A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, 2009. (Cit. on pp. 16, 24, 42, 43, 55, 84).

A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*. Colorado Springs, CO, 2011. (Cit. on pp. 35, 43).

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, pp. 91–110, 2004. (Cit. on p. 145).

L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal* 79(6), pp. 745–754, 1974. (Cit. on p. 26).

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11, pp. 19–60, 2010. (Cit. on pp. 77, 90).

J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing* 17(1), pp. 53–69, Jan. 2008. (Cit. on pp. 23, 51).

S. Mallat. *A Wavelet Tour of Signal Processing*. 2nd ed. New York: Academic Press, 1999. (Cit. on p. 23).

K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision* 60(1), pp. 63–86, 2004. (Cit. on p. 145).

J. W. Miskin and D. J. C. MacKay. Ensemble learning for blind image separation and deconvolution. In *Advances in Independent Component Analysis*. Ed. by M. Girolani. Springer-Verlag, 2000. (Cit. on pp. 34, 36–38, 73–75, 91, 165–167, 170, 173, 175).

R. Molina, J. Mateos, and A. K. Katsaggelos. Blind deconvolution using a variational approach to parameter, image, and blur estimation. *IEEE Transactions on Image Processing* 15(12), pp. 3715–3727, Dec. 2006. (Cit. on p. 34).

R. Molina and B. Ripley. Using spatial models as priors in astronomical image analysis. *Journal of Applied Statistics* 16, pp. 193–206, 1989. (Cit. on p. 22).

J. G. Nagy and D. P. O'Leary. Restoring images degraded by spatially variant blur. *SIAM Journal on Scientific Computing* 19(4), pp. 1063–1082, 1998. (Cit. on pp. 47, 48).

R. M. Neal. *Probabilistic inference using Markov chain Monte Carlo methods*. Tech. rep. CRG-TR-93-1. University of Toronto, 1993. (Cit. on p. 17).

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381, pp. 607–609, 1996. (Cit. on p. 23).

S. Osher, Y. Mao, B. Dong, and W. Yin. Fast linearized Bregman iteration for compressive sensing and sparse denoising. arXiv:1104.0262, 2011. (Cit. on p. 34).

S. Osher and L. I. Rudin. Feature oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis* 27(4), pp. 919–940, 1990. (Cit. on p. 39).

P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)* 22(3), pp. 313–318, 2003. (Cit. on pp. 53, 78, 80, 148, 150).

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the 20th IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, MN, 2007. (Cit. on pp. 53, 142, 143).

C. Rasmussen and T. Korah. Spatiotemporal inpainting for recovering texture maps of partially occluded building facades. In *Proceedings of the IEEE International Conference on Image Processing*. Vol. 3, pp. 125–128, 2005. (Cit. on p. 53).

W. H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America* 62(1), pp. 55–59, 1972. (Cit. on p. 26).

S. Roth and M. J. Black. Fields of experts: a framework for learning image priors. *Proceedings of the 18th IEEE Conference on Computer Vision and Pattern Recognition* 2, pp. 860–867, 2005. (Cit. on pp. 24, 51).

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, pp. 259–268, 1992. (Cit. on p. 22).

A. A. Sawchuk. Space-variant image restoration by coordinate transformations. *Journal of the Optical Society of America* 64(2), pp. 138–144, 1974. (Cit. on p. 47).

C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf. Non-stationary correction of optical aberrations. In *Proceedings of the 13th International Conference on Computer Vision*. Barcelona, Spain, 2011. (Cit. on p. 45).

S. M. Seitz and S. Baker. Filter flow. In *Proceedings of the 12th International Conference on Computer Vision*. Kyoto, Japan, 2009. (Cit. on p. 45).

A. Shahrokni, C. Mei, P. H. Torr, and I. D. Reid. From visual query to visual portrayal. In *Proceedings of the 19th British Machine Vision Conference*. Leeds, 2008. (Cit. on p. 53).

O. Shalvi and E. Weinstein. New criteria for blind deconvolution of nonminimum phase systems (channels). *IEEE Transactions on Information Theory* 36(2), pp. 312–321, 1990. (Cit. on p. 42).

Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)* 27(3), Aug. 2008. (Cit. on pp. 22, 23, 25, 33, 34, 39–42, 127).

Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *Proceedings of the 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil, 2007. (Cit. on pp. 47, 55, 89).

L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging* 1(2), pp. 113–122, Oct. 1982. (Cit. on p. 26).

J. R. Shewchuk. *An introduction to the conjugate gradient method without the agonizing pain*. Tech. rep. Carnegie Mellon University, Aug. 1994. (Cit. on pp. 26, 99).

N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)* 25(3), pp. 835–846, 2006. (Cit. on p. 53).

M. Šorel and J. Flusser. Space-variant restoration of images degraded by camera motion blur. *IEEE Transactions on Image Processing* 17(2), pp. 105–116, Feb. 2008. (Cit. on p. 47).

T. G. Stockham, Jr. High-speed convolution and correlation. In *Proceedings of the April 26-28, 1966, Spring joint computer conference*. ACM, pp. 229–233, 1966. (Cit. on p. 49).

J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005)* 24(3), pp. 861–868, 2005. (Cit. on p. 52).

R. Szeliski. *Image alignment and stitching: a tutorial*. Tech. rep. MSR-TR-2004-92. Microsoft Research, Dec. 2004. (Cit. on p. 64).

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov random fields. In *Proceedings of the 9th European Conference on Computer Vision*. Vol. 2. Graz, Austria, pp. 16–29, 2006. (Cit. on p. 153).

W. Tai, H. Du, M. S. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(6), pp. 1012–1028, 2010a. (Cit. on pp. 47, 48).

W. Tai, N. Kong, S. Lin, and S. Y. Shin. Coded exposure imaging for projective motion deblurring. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 2010b. (Cit. on p. 47).

W. Tai, P. Tan, and M. S. Brown. Richardson-Lucy deblurring for scenes under a projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(8), pp. 1603–1618, Aug. 2011. (Cit. on pp. 28, 45–47, 67, 112).

M. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Proceedings of the 3rd Intl. Workshop on Statistical and Computational Theories of Vision, with ICCV 2003*, 2003. (Cit. on pp. 22, 23).

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B (Methodological)* 58(1), pp. 267–288, 1996. (Cit. on pp. 77, 89).

C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the 6th International Conference on Computer Vision*. Bombay, India, 1998. (Cit. on p. 39).

R. Vio, J. Nagy, L. Tenorio, and W. Wamsteker. Multiple image deblurring with spatially variant PSFs. *Astronomy & Astrophysics* 434, pp. 795–800, 2005. (Cit. on p. 47).

M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on trees: message-passing and linear-programming approaches. *IEEE Transactions on Information Theory* 51(11), pp. 3697–3717, Nov. 2005. (Cit. on p. 153).

Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences* 1(3), pp. 248–272, 2008. (Cit. on p. 33).

Y. Weiss. Deriving intrinsic images from image sequences. In *Proceedings of the 8th International Conference on Computer Vision*. Vancouver, Canada, 2001. (Cit. on p. 148).

Y. Weiss. Old and new algorithms for blind deconvolution. Talk at Machine Learning meets Computational Photography, NIPS Workshop. (Cit. on p. 42).

M. Welk. *Robust variational approaches to positivity-constrained image deconvolution*. Tech. rep. 261. Saarbrücken, Germany: Saarland University, Mar. 2010. (Cit. on pp. 28, 34).

O. Whyte, J. Sivic, and A. Zisserman. Get out of my picture! Internet-based inpainting. In *Proceedings of the 20th British Machine Vision Conference*. London, 2009. (Cit. on p. 7).

O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. In *Proceedings of the IEEE Workshop on Color and Photometry in Computer Vision (CPCV 2011), with ICCV 2011*. Barcelona, Spain, 2011. (Cit. on p. 7).

O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*. San Francisco, CA, 2010. (Cit. on p. 7).

O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision* 98(2), pp. 168–186, 2012. (Cit. on p. 7).

N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. MIT Press, 1949. (Cit. on pp. 26, 72, 111).

M. Wilczkowiak, G. J. Brostow, B. Tordoff, and R. Cipolla. Hole filling through photomontage. In *Proceedings of the 16th British Machine Vision Conference*. Oxford, pp. 492–501, July 2005. (Cit. on p. 53).

L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of the 11th European Conference on Computer Vision*. Crete, Greece, 2010. (Cit. on pp. 34, 35, 39–41).

L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Blurred/non-blurred image alignment using sparseness prior. In *Proceedings of the 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil, 2007a. (Cit. on p. 25).

L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)* 26(3), 2007b. (Cit. on pp. 25, 43, 44, 55, 71, 88, 90, 91, 127).

L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)* 27(3), 2008. (Cit. on pp. 29, 133, 135).

D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proceedings of the 13th International Conference on Computer Vision*. Barcelona, Spain, 2011. (Cit. on pp. 24, 33, 51, 163).