Noname manuscript No.
(will be inserted by the editor)

Non-uniform Deblurring for Shaken Images

Oliver Whyte · Josef Sivic · Andrew Zisserman · Jean Ponce

the date of receipt and acceptance should be inserted later

Abstract Photographs taken in low-light conditions are often blurry as a result of camera shake, i.e. a motion of the camera while its shutter is open. 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 propose a new parametrized geometric model of the blurring process in terms of the rotational motion of the camera during exposure. This model is able to capture non-uniform blur in an image due to camera shake using a single global descriptor, and can be substituted into existing deblurring algorithms with only small modifications. To demonstrate its effectiveness, we apply this model to two deblurring problems; first, the case where a single blurry image is available, for which we examine both an approximate marginalization approach and a maximum a posteriori approach, and second, the case where a sharp but noisy image of the scene is available in addition to the blurry image. We show that our approach makes it possible to model and remove a wider class of blurs than previous approaches, including uniform blur as a special case, and demonstrate its effectiveness with experiments on synthetic and real images.

Keywords Motion blur · Blind deconvolution · Camera shake · Non-uniform / spatially-varying blur

O. Whyte · J. Sivic
INRIA, Paris, France
E-mail: Oliver.Whyte@ens.fr

A. Zisserman
Department of Engineering Science - University of Oxford
Oxford, UK

J. Ponce
Département d'Informatique - Ecole Normale Supérieure
Paris, France

All authors
Willow Project
Laboratoire d'Informatique de l'Ecole Normale Supérieure
(CNRS/ENS/INRIA UMR 8548), Paris, France

1 Introduction

Everybody is familiar with camera shake, since the resulting blur spoils many photos taken in low-light conditions. While significant progress has been made recently towards removing this blur from images, almost all current approaches to this problem model the blurred image as the convolution of a sharp image with a *spatially uniform* filter (Chan and Wong 1998; Fergus et al. 2006; Shan et al. 2007; Yuan et al. 2007a). However, real camera shake, which we show can be (mostly) attributed to the rotation of the camera during exposure, does not in general cause uniform blur, as illustrated by Figure 1.

In this paper we propose a geometrically motivated model of *non-uniform* image blur due to camera shake. By showing that such blur can be mainly attributed to the rotation (as opposed to translation) of the camera during exposure, we develop a global descriptor for such parametrically non-uniform blur, derived from the geometry of camera rotations about a fixed center. Our descriptor can be seen as a generalization of a convolution kernel, and as such our model includes uniform blur as a special case. We demonstrate the effectiveness of our model by using it to replace the uniform blur model in three existing approaches to camera shake removal, and show quantitative and qualitative improvements in the results.

Specifically, we consider the problems of "blind" deblurring, where only a single blurry image is available, and the case where an additional sharp but noisy image of the same

scene is available. To approach these two problems, we apply our model within the frameworks proposed by Miskin and MacKay (2000) and Fergus et al. (2006), and by Cho and Lee (2009) for the blind case, and Yuan et al. (2007a) for the case of a noisy / blurry image pair.

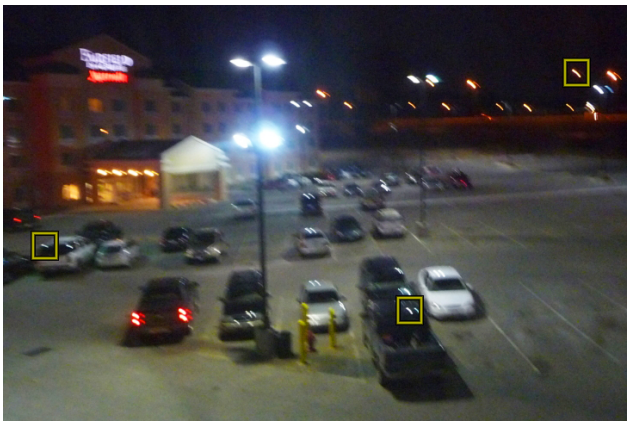
1.1 Related Work

The problem of modeling non-uniform blur is not new, and previous approaches to this problem are diverse, as are the many possible causes of such blurs. Much of the previous work has relied on the local uniformity of the blur, for example, modeling blur due to moving objects as piecewise uniform (Levin 2006; Cho et al. 2007; Chakrabarti et al. 2010), or approximating a continuously varying blur by a spatially varying combination of localized uniform blurs (Nagy and O'Leary 1998; Vio et al. 2005; Hirsch et al. 2010; Tai et al. 2010a). Models of non-uniform blur that do not rely on assumptions of local uniformity have been applied under various constrained motion models (Shan et al. 2007; Sawchuk 1974; Klein and Drummond 2005; Tai et al. 2010b), and while (as in this work) global models are used to describe these continuously varying blurs, the constraints are often restrictive.

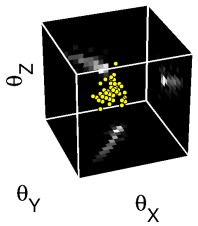
Recent work has investigated the automatic estimation of global descriptors for the non-uniform blur caused by camera shake. Joshi et al. (2010) use inertial measurement sensors to estimate the motion of the camera over the course of the exposure. This information can then be used to effectively deblur the captured image. Gupta et al. (2010) propose a model similar to ours, where the blur-causing motions are approximated using image-plane translations and rotations, as opposed to the 3D camera rotations used in this work. Levin et al. (2009) note that the assumption of uniform blur made by most algorithms is often violated, but do not address this fact.

If the point spread function (PSF) for each pixel in an image is known, the problem of recovering a sharp image is generally referred to as non-blind deconvolution, for which standard techniques such as the Wiener filter (for uniform blur) or the Richardson-Lucy algorithm (for general blur) exist (Richardson 1972; Lucy 1974; Banham and Katsaggelos 1997; Puetter et al. 2005). Sophisticated algorithms for non-blind deconvolution have recently been proposed (Dabov et al. 2008; Shan et al. 2008; Yuan et al. 2008; Couzinie-Devy et al. 2011), but their application has generally been limited to the case of uniform blur. Note however that Tai et al. (2009) propose a modified version of the Richardson-Lucy algorithm for deblurring scenes under general projective motion, where the temporal sequence of projective transformations which caused the blur is known.

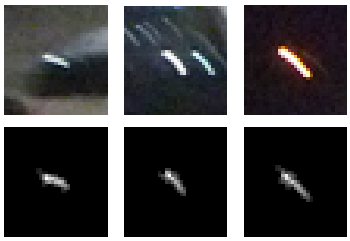
The task of recovering a sharp image when the pixels' PSFs are unknown, so-called blind deblurring, is a difficult



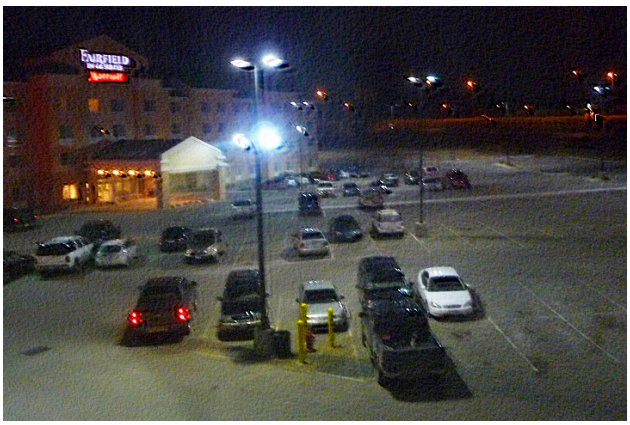
(a) An image blurred by camera shake



(b) Estimated global blur descriptor



(c) Top: Details from blurry image. Bottom: local point spread functions generated by (b)



(d) Deblurred using the estimated blur descriptor

(e) Some corresponding details from the blurry and deblurred images

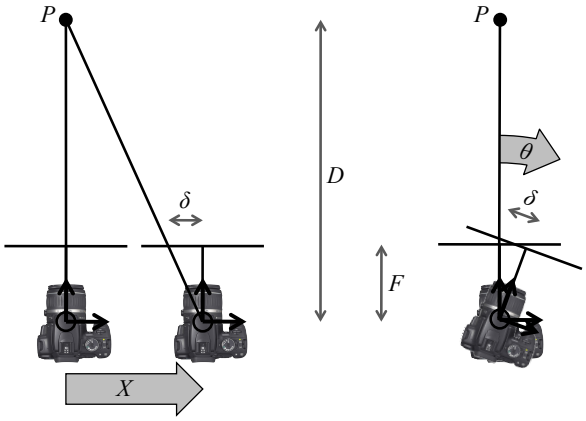
Fig. 1 Modeling non-uniform blur in a shaken image. The blurry image (a) clearly exhibits blur which is non-uniform, as highlighted at different locations in the image. Using the model proposed in this paper, we can describe this blur using a single global descriptor (b), which in this case has been estimated from the blurry image itself, simply by modifying existing algorithms for blind deblurring (see Section 3 for a complete explanation). Close-ups of different parts of the image (c) 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 (d) and the close-ups in (e), different parts of the image, blurred in different ways, can be deblurred to recover a sharp image

problem. Existing approaches for uniform blur, where a single PSF, or "blur kernel", describes the blur everywhere typically proceed by first estimating this kernel, then applying a non-blind deconvolution algorithm to estimate the sharp image. For uniform blur, Fergus et al. (2006) estimate the kernel by first applying the variational algorithm of Miskin and MacKay (2000) to approximate the posterior for the kernel and sharp image with a simpler distribution. This distribution is chosen such that it is then trivial to estimate the kernel by marginalizing over all possible sharp images. Many different maximum a posteriori (MAP) formulations have also been proposed, such as those of Shan et al. (2008), Cho and Lee (2009), Cai et al. (2009), and Xu and Jia (2010). These algorithms typically use an alternation scheme, updating the estimate of the blur kernel at one step, and of the sharp image at the next. The algorithm proposed by Gupta et al. (2010) for their non-uniform blur model also follows this paradigm. To simplify the deblurring problem, others have considered using additional information in the form of additional blurry images (Rav-Acha and Peleg 2005; Chen et al. 2008), or a sharp but noisy image of the same scene (Yuan et al. 2007a; Lim and Silverstein 2008).

The rest of this paper is organized as follows. Section 2 presents our geometric model. Section 3 presents a discrete version of this model. In Section 4, we demonstrate its application within two existing algorithms for deblurring a single blurry image, both an approximate marginalization approach and a maximum a posteriori (MAP) approach, and in Section 5 we compare the results obtained these two algorithms. In Section 6 we examine a second deblurring problem, where a sharp but noisy image of the same scene is available, in addition to the blurry image. In Section 7 we describe some of the implementation details for the presented algorithms, and in Section 8 we conclude with a discussion of some limitations of our work and potential future research directions.

2 Geometric Model

To motivate our approach, let us begin by noting that the blur in a "shaken" image is caused by the motion of the camera during the exposure, i.e. changes in the pose of the camera. The pose of a camera can be split into two components: position and orientation, and in this section, we argue that in most cases of camera shake, the changes in orientation (rotations) of the camera during exposure have a significantly larger effect than the changes in position (translations). Consider the simplified case shown in Figure 2 of a scene point P, at a distance D from the camera, being imaged at the center of the camera's retina. During the exposure the image of the point is blurred through a distance δ pixels, either by (a) translating the camera through a distance X parallel to the image plane, or (b) rotating the camera through an angle θ about its optical center. By simple trigonometry, we can see that in the case of translation the camera must move by



(a) Camera translation (b) Camera rotation

Fig. 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 θ about its optical center. We consider the scene point P at a distance D from the camera, whose image is blurred by δ pixels as a result of either of the two motions. In most cases, for a given blur size δ the rotation θ constitutes a significantly smaller motion of the photographer's hands than the translation X (see text for details)

$$X = \frac{\delta}{F} D, \tag{1}$$

where F is the camera's focal length, while for the rotation, the camera must move through an angle

$$\theta = \tan^{-1}\left(\frac{\delta}{F}\right). \tag{2}$$

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 (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 $X = 1$ 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!

To cause the same amount of blur by rotating the camera, on the other hand, we can see from (2) that we would need to rotate the camera by $\theta = \tan^{-1}\left(\frac{1}{100}\right) \approx 0.6°$, 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 motion is at least an order of magnitude smaller than for a translation of the camera.

In reality, both the position and orientation of the camera vary simultaneously during the exposure. However, if we assume that 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. We do this from now on, and ignore the translational component of camera motion. We consider all rotations to occur about the camera's optical center, and although this may not be the case, we note that rotations about a different center can be written as rotations about the optical center, plus translations.

2.1 Motion Blur and Homographies

Assuming that the scene being photographed is static, it is well known that rotations of a camera about its optical center induce projective transformations of the image being observed, assuming a pinhole camera model. That is to say that, excluding boundary effects, the image at one camera orientation is 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

$$\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}, \tag{3}$$

where the 3×3 matrices $\mathbf{R}$ and $\mathbf{K}$ are, respectively, a rotation matrix describing the motion of the camera, and the camera's internal calibration matrix (Hartley and Zisserman 2004).

The matrix $\mathbf{R}$ has only 3 parameters. We adopt here the "angle-axis" parameterization, in which a rotation is described by the angle θ moved about an axis $\mathbf{a}$ (a unit-norm 3-vector). This can be summarized in a single vector $\boldsymbol{\theta} = \theta\mathbf{a} = (\theta_X, \theta_Y, \theta_Z)^\top$. $\mathbf{R}$ is then given by the matrix exponential

$$\mathbf{R}_{\boldsymbol{\theta}} = e^{[\boldsymbol{\theta}]_\times}, \quad \text{where} \tag{4}$$

$$[\boldsymbol{\theta}]_\times = \begin{bmatrix} 0 & -\theta_Z & \theta_Y \\ \theta_Z & 0 & -\theta_X \\ -\theta_Y & \theta_X & 0 \end{bmatrix}. \tag{5}$$

We fix our 3D coordinate frame to have its origin at the camera's optical center. 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 (a). In this configuration, θ_X describes the "pitch" of the camera, θ_Y the "yaw", and θ_Z the "roll", or in-plane rotation, of the camera.

In this work, we assume that the calibration matrix $\mathbf{K}$ is known and takes the standard form

$$\mathbf{K} = \begin{bmatrix} F & 0 & x_0 \\ 0 & F & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

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. Section 2.3 describes how we estimate $\mathbf{K}$ in practice.

Having defined the type of image transformation we expect, we now assume that when the shutter of the camera opens, there is a sharp image $f : \mathbb{R}^2 \to \mathbb{R}$ of a static scene that we would like to capture. The camera's sensor accumulates photons while the shutter is open, and outputs an observed image $g : \mathbb{R}^2 \to \mathbb{R}$. In the ideal pinhole case, each point on the sensor sees a single scene point throughout the exposure, giving us a sharp image. However if, while the shutter is open, the camera undergoes a sequence of rotations $\mathbf{R}_t$, the sensor is exposed to a sequence of projectively transformed versions of the sharp image f. For each point on the sensor (a point in the observed blurry image g), denoted by the homogeneous vector $\mathbf{x}$, we can trace the sequence of points $\mathbf{x}'_t$ in the ideal sharp image f which were visible there during the exposure:

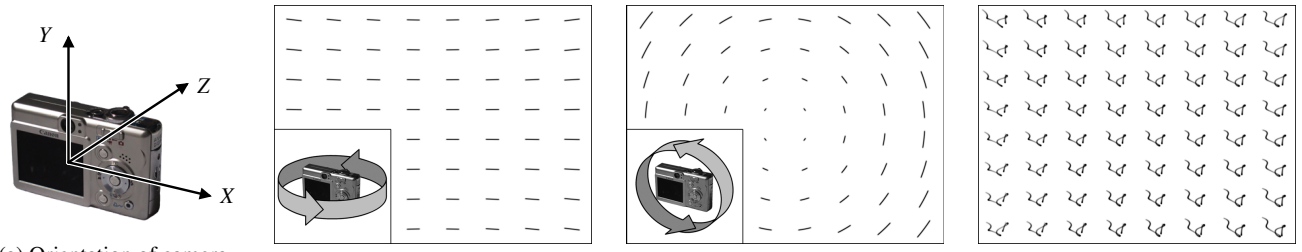
$$\mathbf{x}'_t \sim \mathbf{H}_t\mathbf{x}, \tag{7}$$

where $\mathbf{H}_t$ is the homography induced by the rotation $\mathbf{R}_t$, and $\sim$ denotes equality up to scale. The observed image g is then the integral over the exposure time T of all the projectively-transformed versions of f, plus some observation noise ε:

$$g(\mathbf{x}) = \int_0^T f\big(\mathbf{H}_t\mathbf{x}\big)\, dt + \varepsilon, \tag{8}$$

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.

According to this model, the apparent motion of scene points may vary significantly across the image. Figure 3 demonstrates this, showing the paths followed by points in an image under rotations about either the Y or Z axis 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. In addition, it is common for camera shake to include an in-plane rotational motion.



(a) Orientation of camera axes (b) Y-axis rotation of the camera (c) Z-axis rotation of the camera (d) Arbitrary sequence of rotations

Fig. 3 Our coordinate frame with respect to initial camera orientation, and the paths followed by image points under single-axis rotations. We define our coordinate frame (a) to have its origin at the camera's optical center, 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 center, and the pixels are assumed to be square

From this, it is clear that modeling camera shake as a convolution with a spatially invariant kernel is insufficient to fully describe its effects (see also Figure 1).

In general, a blurry image has no temporal information associated with it, so it is convenient to replace the temporal integral in (8) by a weighted integral over a set of camera orientations:

$$g(\mathbf{x}) = \int f\big(\mathbf{H}_{\boldsymbol{\theta}}\mathbf{x}\big) w(\boldsymbol{\theta})\, d\boldsymbol{\theta} + \varepsilon, \tag{9}$$

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, while the value of the function along that trajectory corresponds (inversely) to the camera's rotational speed, i.e. if the camera moves slowly through a certain orientation, the weight will be large there, and vice versa.

2.2 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 θ_X, θ_Y are small, Equation (4) can be approximated by discarding the 2nd and higher order terms:

$$\mathbf{R}_{\boldsymbol{\theta}} \approx \begin{bmatrix} 1 & 0 & \theta_Y \\ 0 & 1 & -\theta_X \\ -\theta_Y & \theta_X & 1 \end{bmatrix}. \tag{10}$$

Combining this with Equations (3) and (6), it can be shown that as $F \to \infty$,

$$\mathbf{H}_{\boldsymbol{\theta}} \to \begin{bmatrix} 1 & 0 & F\theta_Y \\ 0 & 1 & -F\theta_X \\ 0 & 0 & 1 \end{bmatrix}, \tag{11}$$

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

2.3 Camera Calibration

In order to compute the homography in Equation (3) that is induced by a particular rotation of the camera, we need to know the camera's calibration matrix $\mathbf{K}$, as given by Equation (6). 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 center of the image. Note that this assumes the image has not been cropped or resized beforehand, as these operations will generally invalidate the EXIF information.

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 tool[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

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

to recover the linear values, and where it is unknown, we apply a generic sRGB curve.

3 Restoration Model

So far, our model has been defined in terms of the continuous functions f and g, and the weight function w. Real cameras are equipped with a discrete set of pixels, and output an observed blurry image $\mathbf{g} \in \mathbb{R}^N$, where $N = H \times W$ pixels for an image with H rows and W columns. We consider $\mathbf{g}$ to be generated by a sharp image $\mathbf{f} \in \mathbb{R}^N$ and a set of weights $\mathbf{w} \in \mathbb{R}^K$, whose size $K = N_X \times N_Y \times N_Z$ is controlled by the number of rotation steps about each axis that we consider. 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 1 (b) shows a visualization 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 visualization, 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 modeled as a weighted sum of a set of projectively transformed versions of $\mathbf{f}$:

$$\mathbf{g} = \sum_k w_k \mathbf{C}_k \mathbf{f} + \boldsymbol{\varepsilon}, \tag{12}$$

where $\mathbf{C}_k$ is the $N \times N$ matrix which applies homography $\mathbf{H}_k$ to the sharp image $\mathbf{f}$. The matrix $\mathbf{C}_k$ is very sparse. For example, if bilinear interpolation is used when transforming the image, each row has only 4 non-zero elements. Expanding Equation (12), we obtain the discrete analog of Equation (9):

$$g_i = \sum_k w_k \left(\sum_j C_{ijk} f_j \right) + \varepsilon, \tag{13}$$

where i and j index the pixels of the observed image and the sharp image, respectively. Appendix B describes how to calculate the coefficients C_{ijk}. For an observed pixel g_i with coordinate vector $\mathbf{x}_i$, the sum $\sum_j C_{ijk} f_j$ interpolates the point $\mathbf{H}_k \mathbf{x}_i$ in the sharp image. Figure 4 shows an example of this, where a blurry pixel g_5, with homogeneous coordinate vector $\mathbf{x}_5$, is mapped under a homography $\mathbf{H}_k$ to the point $\mathbf{H}_k \mathbf{x}_5$ in the sharp image. The value of $\mathbf{f}$ at the point $\mathbf{H}_k \mathbf{x}_5$ is then interpolated as a weighted sum of the pixels of $\mathbf{f}$.

Due to the bilinear form of Equation (13), note that when either the blur kernel or the sharp image is known, the blurry image is linear in the remaining unknowns, i.e.

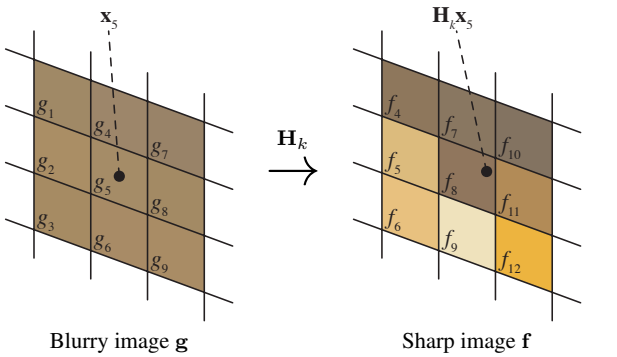


Fig. 4 Interpolation of sub-pixel locations in the sharp image. In general, a homography $\mathbf{H}_k$ will not map a pixel (e.g. $\mathbf{x}_5$) in the blurry image $\mathbf{g}$ to a single pixel in the sharp image $\mathbf{x}$. Instead, the value of $\mathbf{f}$ at the point $\mathbf{H}_k \mathbf{x}_5$ is interpolated as a weighted sum of nearby pixels. Using bilinear interpolation, the value of $\mathbf{f}$ at $\mathbf{H}_k \mathbf{x}_5$ will be interpolated from the pixels f_7, f_8, f_{10}, and f_{11}

$$\mathbf{g} = \mathbf{A}\mathbf{f} + \boldsymbol{\varepsilon}, \qquad \text{or} \tag{14}$$

$$\mathbf{g} = \mathbf{B}\mathbf{w} + \boldsymbol{\varepsilon}, \tag{15}$$

where $A_{ij} = \sum_k C_{ijk} w_k$, and $B_{ik} = \sum_j C_{ijk} f_j$. 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.

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 non-uniform blur. In common with our model, they generally model the blurry image as a sum of transformed versions of the sharp image. Tai et al. (2011) model the blur process using a temporally-ordered sequence of homographies, which is known in advance, for example by using additional hardware attached to the camera. Joshi et al. (2010) also recover a temporally ordered sequence of homographies by using inertial measurement sensors attached to the camera to recover the camera's path through 6D (rotation and translation) pose space, and assuming that the scene can be modelled as a single fronto-parallel plane (although the authors note that the blur is generally independent of depth for objects further than 1 metre from the camera). 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.

3.2 Application to Existing Deblurring Algorithms

The fact that Equation (13) 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 (13) without significant modification to the algorithm. The remainder of the paper demonstrates this, first in Sections 4 and 5 for the problem of single-image deblurring, and second in Section 6 for the case where a sharp but noisy image of the scene is also available.

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. 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.

The aim of deblurring algorithms is to recover an estimate $\hat{\mathbf{f}}$ of the true sharp image $\mathbf{f}$. Generally, the approach taken is to also estimate the blur kernel $\hat{\mathbf{w}}$ such that together, $\hat{\mathbf{f}}$ and $\hat{\mathbf{w}}$ are able to accurately reconstruct the observed blurry image $\mathbf{g}$. We denote this reconstruction as $\hat{\mathbf{g}}(\hat{\mathbf{f}}, \hat{\mathbf{w}})$, where, under our model, a blurry pixel is reconstructed as:

$$\hat{g}_i = \sum_k \hat{w}_k \left(\sum_j C_{ijk} \hat{f}_j \right). \tag{16}$$

The problem of finding the sharp image and blur kernel that best reconstruct the observed image is in general ill-posed, since we have fewer equations than parameters. In fact, for a given $\mathbf{g}$, there are an infinite number of $(\hat{\mathbf{f}}, \hat{\mathbf{w}})$ pairs that can reconstruct $\mathbf{g}$ equally well. To obtain a useful solution, it is thus necessary to add some regularization and/or constraints on both the sharp image and the kernel.

Successful algorithms for deblurring camera shake generally share the same two pieces of prior information about the blur kernel being estimated, which we mention here. First, all its elements are non-negative, since the image formation process is additive, with sensor elements accumulating photons. This constraint is equally applicable to our model, since each kernel element w_k corresponds to a camera orientation, so that if that if the camera passed through orientation $\boldsymbol{\theta}_k$ during the exposure, w_k will be positive, and if not, $w_k = 0$.

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. This sparsity prior has been a prominent feature of previous camera shake removal algorithms, and has also been leveraged for the alignment of blurred / non-blurred images (Yuan et al. 2007b). Fergus et al. (2006) encourage sparsity by placing a mixture of exponentials prior on the kernel values, while Cho and Lee (2009) and Yuan et al. (2007a) proceed by thresholding the estimated kernel values such that most of the kernel is set to zero. In a contrasting approach, Cai et al. (2009) choose to construct the blur kernel as a linear combination of a predefined set of "curvelets", and place the sparsity prior on the coefficients of the curvelets, rather than on the kernel elements directly. This 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.

Many different image priors/regularizers have been proposed for image reconstruction tasks such as deblurring, denoising, and super-resolution, often based on the statistics of natural images. The most commonly used priors for deblurring are those which encourage the image's response to a set of derivative filters to follow heavy-tailed distributions, e.g. (Fergus et al. 2006; Shan et al. 2008; Krishnan and Fergus 2009), which have been shown to be effective at suppressing noise while preserving sharp edges in the reconstructed images. When substituting our blur model into the algorithms in Sections 4 and 6, we use the image regularizers suggested in the original works. In order to compare the final results of the two methods for single-image deblurring, shown in Section 5, we use the algorithm of Krishnan and Fergus (2009), adapted for our non-uniform blur model.

4 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) and Cho and Lee (2009). As discussed in the previous section, good priors for $\mathbf{f}$ and $\mathbf{w}$ are necessary for blind deblurring to be successful, so both approaches take the posterior distribution for $\mathbf{f}$ and $\mathbf{w}$ as their starting point:

$$p(\mathbf{f}, \mathbf{w}|\mathbf{g}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{w}) p(\mathbf{f}) p(\mathbf{w}), \tag{17}$$

where the likelihood is derived from an assumption of isotropic Gaussian noise:

$$p(\mathbf{g}|\mathbf{f}, \mathbf{w}) \propto \prod_i \exp\left(-\frac{(\hat{g}_i(\mathbf{f}, \mathbf{w}) - g_i)^2}{2\sigma^2} \right), \tag{18}$$

where σ is the standard deviation of the noise, and the individual priors used by the two algorithms will be discussed in the following sections. Both of the algorithms are mainly concerned with estimating the blur kernel $\hat{\mathbf{w}}$, and after the

termination of this process, a final non-blind image reconstruction step is performed using the estimate of $\hat{\mathbf{w}}$ to produce the deblurred output $\hat{\mathbf{f}}$. To estimate the kernel, Fergus et al. (2006) use the variational inference approach of Miskin and MacKay (2000) to perform approximate marginalization of the posterior over $\mathbf{f}$, while Cho and Lee (2009) use alternating optimizations to maximize the posterior over both $\mathbf{f}$ and $\mathbf{w}$.

4.1 The Marginalization Approach

In this section we adapt the algorithm proposed by Fergus et al. (2006) for blind deconvolution of a single image. 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. 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 optimization process, and we show in Section 5 that doing so improves the deblurred results.

The algorithm proposed by Miskin and MacKay (2000) attempts to approximate the posterior distribution for both the kernel and the sharp image $p(\mathbf{f}, \mathbf{w}|\mathbf{g})$ by a simpler, factorized distribution using a variational method. The factorized form of this distribution means that it is straightforward to marginalize over the sharp image in order to produce an estimate $\hat{\mathbf{w}}$ of the kernel. 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. They use a prior on the kernel which assumes that each element w_k is independent, and follows a mixture of exponential distributions with mixture weights π_d and decay parameters λ_d:

$$p(\mathbf{w}) = \prod_k \sum_{d=1}^{D} \pi_d \exp(-\lambda_d w_k). \tag{19}$$

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. Fergus et al. use a prior on the sharp image which assumes that the derivatives for all pixels are independent and follow a mixture of zero-mean Gaussians with mixture weights π_c and variances v_c:

$$p(\mathbf{f}^x) = \prod_j \sum_{c=1}^{C} \pi_c \exp\left(-\frac{f_j^{x\,2}}{2v_c}\right), \tag{20}$$

and likewise for the y derivatives. For simplicity, within the context of this algorithm, we use $\mathbf{f}$ to denote the concatenation of the derivative images $\mathbf{f}^x$ and $\mathbf{f}^y$, and use j to index over this, i.e. $j \in \{1, \ldots, 2N\}$. Fergus et al. learn the parameters π_d, λ_d, π_c and v_c from real data, and we use the values provided by them directly. Finally, to free the user from manually tuning the noise variance σ^2, the inverse variance $\beta_\sigma = \sigma^{-2}$ is also considered as a latent variable.

Following Miskin and MacKay (2000), we collect the latent variables $\mathbf{f}$, $\mathbf{w}$, and β_σ into an "ensemble" $\boldsymbol{\Theta}$. The aim is to find the factorized distribution

$$q(\boldsymbol{\Theta}) = q(\beta_\sigma) \prod_j q(f_j) \prod_k q(w_k) \tag{21}$$

that best approximates the true posterior $p(\boldsymbol{\Theta}|\mathbf{g})$, by minimizing the following cost function (Miskin and MacKay 2000, Eqn. (10)) over both the form and the parameters of $q(\boldsymbol{\Theta})$:

$$C_{\mathrm{KL}} = \int q(\boldsymbol{\Theta}) \left[\ln \frac{q(\boldsymbol{\Theta})}{p(\boldsymbol{\Theta})} - \ln p(\mathbf{g}|\boldsymbol{\Theta})\right] \mathrm{d}\boldsymbol{\Theta}. \tag{22}$$

Minimizing this cost function is equivalent to minimizing 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 $q(f_j)$, $q(w_k)$ and $q(\beta_\sigma)$, then iteratively optimizing their parameters. 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 the supplementary material). For our non-uniform blur model, we find the following optimal values for the parameters, *cf*. (Miskin and MacKay 2000, Eqns. 46–49):

$$w_k^{(2)} = \langle\beta_\sigma\rangle \sum_i \left\langle \left(\sum_j C_{ijk} f_j\right)^2 \right\rangle_{q(\mathbf{f})} \tag{23}$$

$$\begin{aligned} w_k^{(1)} w_k^{(2)} = \langle\beta_\sigma\rangle \sum_i \Bigg(& g_i \sum_j C_{ijk} \langle f_j\rangle_{q(f_j)} \\ & - \sum_{k'\neq k} \left\langle \left(\sum_j C_{ijk} f_j\right)\left(\sum_j C_{ijk'} f_j\right)\right\rangle_{q(\mathbf{f})} \langle w_{k'}\rangle_{q(w_{k'})} \Bigg) \end{aligned} \tag{24}$$

$$f_j^{(2)} = \langle\beta_\sigma\rangle \sum_i \left\langle \left(\sum_k C_{ijk} w_k\right)^2 \right\rangle_{q(\mathbf{w})} \tag{25}$$

$$\begin{aligned} f_j^{(1)} f_j^{(2)} = \langle\beta_\sigma\rangle \sum_i \Bigg(& g_i \sum_k C_{ijk} \langle w_k\rangle_{q(w_k)} \\ & - \sum_{j'\neq j} \langle f_{j'}\rangle_{q(f_{j'})} \left\langle \left(\sum_k C_{ij'k} w_k\right)\left(\sum_k C_{ijk} w_k\right)\right\rangle_{q(\mathbf{w})} \Bigg), \end{aligned} \tag{26}$$

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 $q(f_j)$, $q(\mathbf{f}) = \prod_j q(f_j)$, $q(\mathbf{w}) = \prod_k q(w_k)$, and $\langle\cdot\rangle_q$ represents the expectation with respect to the distribution q. Note that in this context, the latent image pixels f_j, kernel elements w_k, and noise precision β_σ are random variables. Note also that these equations cannot be

implemented directly in this form, as they involve expectations over combinations of random variables. However, they may be rewritten in terms of the mean and variance of individual variables, and we provide these expanded versions in Appendix A. Having found the optimal $q(\boldsymbol{\Theta})$, the expectation of $q(\mathbf{w})$ is taken to be the optimal blur kernel, i.e., $\hat{\mathbf{w}} = \langle \mathbf{w} \rangle_{q(\mathbf{w})}$. Fergus et al. choose to discard the latent image distribution $q(\mathbf{f})$, although as shown in Figure 8 (d), this may in fact provide a useful estimate of the sharp image.

4.2 The Maximum A Posteriori Approach

Cho and Lee (2009) proposed an effective single image deblurring algorithm, optimized for speed on uniform blurs. Again, we show that this algorithm can be readily adapted to handle non-uniform blur, substituting our model in place of convolution. The algorithm can be considered to perform alternating maximum a posteriori estimation of the blur kernel, using Gaussian priors on the kernel elements and on the latent image gradients. Simply performing an alternating optimization of $\mathbf{f}$ and $\mathbf{w}$ using these priors would almost certainly not produce any reasonable result, 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 3.2. The algorithm proceeds by iterating over three main steps, of which we give a brief overview here.

The first step takes the current estimate $\hat{\mathbf{f}}$ of the sharp image and aims to predict strong edges, which are useful for the kernel estimation step, by applying a bilateral filter (Tomasi and Manduchi 1998) followed by a shock filter (Osher and Rudin 1990). The derivatives of this filtered image are computed, then thresholded to produce sparse gradient maps $\{\mathbf{p}_x, \mathbf{p}_y\}$ which contain only the most salient edges. The threshold is chosen so as to retain only a small number of non-zero gradients, while ensuring that all orientations are well-represented.

In the second step, the gradient maps are used to estimate the blur kernel by minimizing the energy function

$$E_{\mathbf{w}}(\mathbf{w}) = \sum_{(\mathbf{p}_*, \mathbf{g}_*)} \omega_* \|\hat{\mathbf{g}}(\mathbf{p}_*, \mathbf{w}) - \mathbf{g}_*\|_2^2 + \beta \|\mathbf{w}\|_2^2, \qquad (27)$$

where the weights $\omega_* \in \{\omega_1, \omega_2\}$ weight each partial derivative, $(\mathbf{p}_*, \mathbf{g}_*)$ varies among $\big\{(\mathbf{p}_x, \mathbf{D}_x\mathbf{g}),\ (\mathbf{p}_y, \mathbf{D}_y\mathbf{g}),\ (\mathbf{D}_x\mathbf{p}_x, \mathbf{D}_{xx}\mathbf{g}),\ (\mathbf{D}_y\mathbf{p}_y, \mathbf{D}_{yy}\mathbf{g}),\ (\frac{1}{2}(\mathbf{D}_x\mathbf{p}_y + \mathbf{D}_y\mathbf{p}_x), \mathbf{D}_{xy}\mathbf{g})\big\}$, and β is the regularization weight. Since $\hat{\mathbf{g}}(\mathbf{p}_*, \mathbf{w})$ is linear in $\mathbf{w}$, this is simply a linear least squares problem, which can be solved efficiently using a conjugate gradient method. Having found the kernel that minimizes (27), 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, as discussed in Section 3.2.

In the third step, the current estimate of the blur kernel $\hat{\mathbf{w}}$ is used to deconvolve the blurry image and obtain an improved estimate of the sharp image. This is performed by minimizing the energy function

$$E_{\mathbf{f}}(\mathbf{f}) = \sum_{\mathbf{D}_*} \omega_* \|\hat{\mathbf{g}}(\mathbf{D}_*\mathbf{f}, \hat{\mathbf{w}}) - \mathbf{D}_*\mathbf{g}\|_2^2 + \alpha \|\nabla \mathbf{f}\|_2^2, \qquad (28)$$

where α is the regularization weight, and now, the partial derivatives include the zeroth order: $\mathbf{D}_* \in \{I, \mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_{xx}, \mathbf{D}_{xy}, \mathbf{D}_{yy}\}$, where I is the identity, and $\omega_* \in \{\omega_0, \omega_1, \omega_2\}$. The use of the partial derivatives $\mathbf{D}_*$ in the data terms of (27) and (28), as suggested by Shan et al. (2008), has the effect of improving the conditioning and regularizing the solutions.

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 7 iterations are typically sufficient. For the parameters ω_0, ω_1, ω_2, α, and β, we use the values given by Cho and Lee (2009). Although we are not able to take full advantage of the speed optimizations proposed by Cho and Lee, 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 marginalization algorithm of Section 4.1.

4.2.1 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 point spread function (PSF) of a single pixel does not uniquely determine the full 3D kernel, i.e. for every PSF there are many different kernels that could explain it. 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 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 3×3 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, a rotation of the camera about its Z axis causes a very small displacement for pixels towards the center 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 minimizing (27) with our model generally contain many non-zeros spread smoothly throughout, and do not produce good deblurred outputs (see Figure 9).

If instead of the ℓ_2 regularization in (27), we apply ℓ_1 regularization combined with non-negativity constraints, the optimization 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 type of ℓ_1 kernel regularization was previously applied by Shan et al. (2008) for uniform blur. In our case, the energy function becomes

$$E_{\mathbf{w}}(\mathbf{w}) = \sum_{(\mathbf{p}_*, \mathbf{g}_*)} \omega_* \|\hat{\mathbf{g}}(\mathbf{p}_*, \mathbf{w}) - \mathbf{g}_*\|_2^2 + \beta \sum_k w_k$$
$$\text{s.t.} \quad \forall k = 1, \ldots, K, \quad w_k \geq 0. \tag{29}$$

This is an instance of the lasso problem (Tibshirani 1996), for which efficient optimization algorithms exist (Efron et al. 2004; Kim et al. 2007; Mairal et al. 2010). The different results obtained using ℓ_2 and ℓ_1 regularization are discussed in Section 5. With the use of the ℓ_1 regularization, we found that the best results were obtained with a lower value of β than that given by Cho and Lee, and for the results in this paper using ℓ_1 regularization, we set $\beta = 0.1$. In the remainder of the paper, we refer to the original algorithm of Cho and Lee as MAP-ℓ_2, and our ℓ_1-regularized version as MAP-ℓ_1.

4.3 Image Reconstruction

Having estimated the blur kernel $\hat{\mathbf{w}}$ for the blurry image, we wish to invert Equation (14) in order to estimate the sharp image $\hat{\mathbf{f}}$. This process is often referred to as deconvolution, and while many algorithms exist for this process (Banham and Katsaggelos 1997; Puetter et al. 2005; Dabov et al. 2008), they are often applicable only to uniform blur, since they typically rely on convolutions or the ability to work in the Fourier domain.

For the results of single-image deblurring in Section 5, we have adapted the deconvolution algorithm of Krishnan and Fergus (2009), which performs MAP estimation of the sharp image using a hyper-Laplacian prior on the image gradients. Specifically, it attempts to maximize the following posterior over $\mathbf{f}$:

$$p(\mathbf{f}|\mathbf{g}, \mathbf{w}) \propto p(\mathbf{g}|\mathbf{f}, \mathbf{w}) p(\mathbf{f}), \tag{30}$$

with the prior

$$p(\mathbf{f}) = \prod_j \exp(-\lambda |f_j^x|^p) \exp(-\lambda |f_j^y|^p), \tag{31}$$

where the exponent p is chosen to be less than one, to encourage sparsity on the sharp image gradients. In this work we use $p = 0.5$. We refer the reader to the original work for full details, but note that the algorithm is easily adapted to non-uniform blur since it involves repeated minimizations of quadratic cost functions of the form

$$E(\mathbf{f}) = \|\mathbf{A}\mathbf{f} - \mathbf{g}\|_2^2 + \alpha \|\mathbf{D}_x \mathbf{f} - \mathbf{v}^x\|_2^2 + \alpha \|\mathbf{D}_y \mathbf{f} - \mathbf{v}^y\|_2^2, \tag{32}$$

where $\mathbf{v}^x$ and $\mathbf{v}^y$ are intermediate variables of the optimization scheme, used to decouple the exact form of the prior from the main image reconstruction step. For our non-uniform blur model, we use the conjugate gradient algorithm to minimize this cost function.

Another method frequently used for deconvolution is the Richardson-Lucy algorithm (Richardson 1972; Lucy 1974). Although originally proposed for convolutional blur, this algorithm can equally be used to invert general linear systems (Lee and Seung 2001). Using the notation of Equation (14) for a known blur, the algorithm iteratively improves the estimate $\hat{\mathbf{f}}$ using the following update equation:

$$\hat{\mathbf{f}} \leftarrow \hat{\mathbf{f}} \odot \left(\mathbf{A}^\top \left(\mathbf{g} \oslash \mathbf{A}\hat{\mathbf{f}} \right) \right), \tag{33}$$

where $\mathbf{g}$ is the observed blurry image, and the matrix $\mathbf{A}$ depends on the estimated non-uniform blur. Here, $\odot$ represents the element-wise product and $\oslash$ the element-wise division of two vectors. We have found that for images containing saturated regions (pixels where the signal is clipped and the linear model is no longer valid), such as in Figure 1, the Richardson-Lucy algorithm gives better results, with fewer artifacts around saturated regions such as the bright streetlights.

5 Single-Image Deblurring Results

We show in this section results of single-image deblurring using the algorithms described in Section 4, 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. Implementation details are discussed in Section 7.

Figure 1 shows a result on a real camera shake blur, using the MAP-ℓ_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

True sharp image

	Marginalization				MAP			
	10px		20px		10px		20px	
Y-axis	Non-uniform	Uniform	Non-uniform	Uniform	Non-uniform	Uniform	Non-uniform	Uniform
$\sigma = 0$	**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)

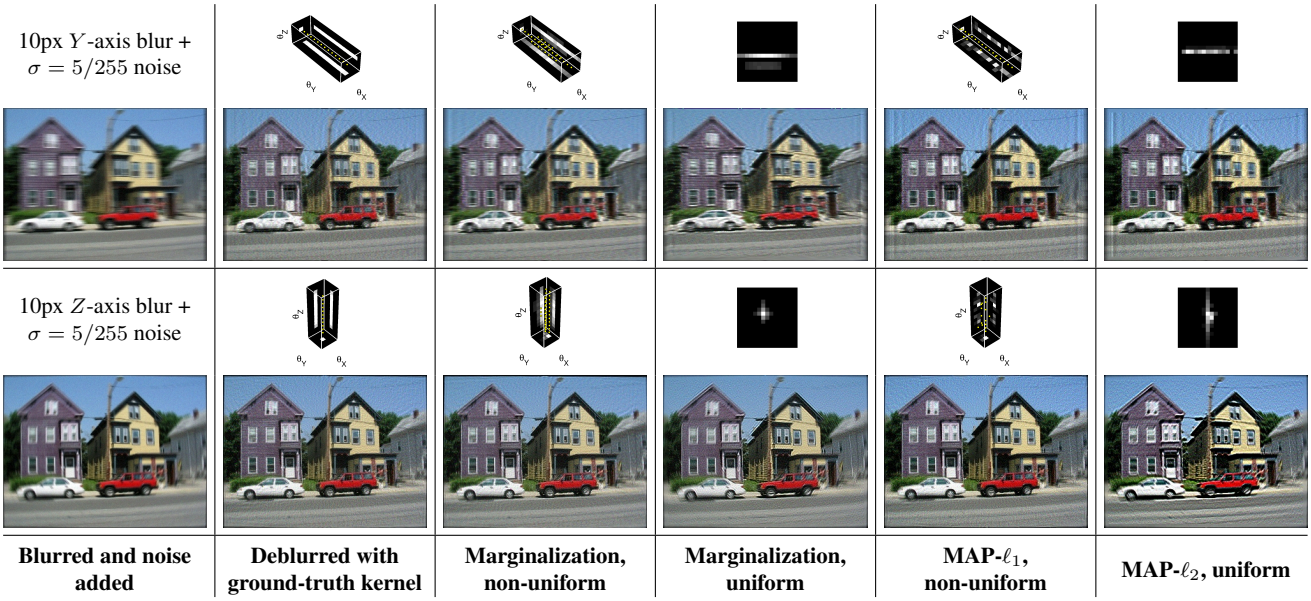


Fig. 7 Blind deblurring of synthetic single-axis blurs. A sharp image (top left) with examples of synthetic blur by rotation of the camera about its Y and Z-axis, and the kernels and deblurred results for different cases. We compare the results of blind deblurring for two sizes of blur and three noise levels, and the reconstruction errors are summarized 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

any least-squares based algorithms, such as that of Krishnan and Fergus (2009).

Figures 5 and 8 show blind deblurring results on images blurred by real camera shake. Our model, used in both the marginalization 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 8 (d), we also demonstrate the use of the variational marginalization algorithm of Fergus et al. to produce the deblurred output, as opposed to the Richardson-Lucy algorithm used in Figure 8 (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 6 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 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 7 shows results for blind deblurring of synthetically blurred images using the two methods, 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

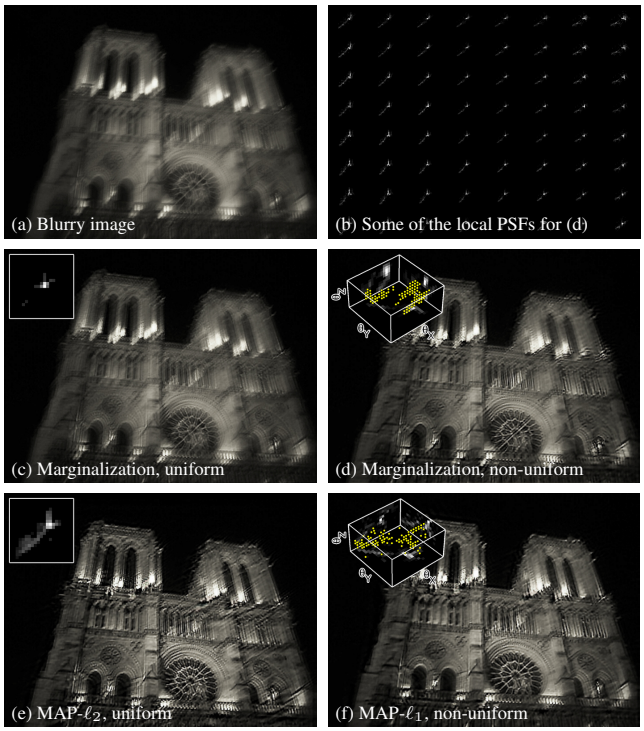


Fig. 5 Blind deblurring of real camera shake, example 1. The result of blind deblurring on a real camera shake image (a), captured with a shutter speed of $\frac{1}{2}$ second, using both the marginalization algorithm of Fergus et al. and the MAP approach of Cho and Lee with both the uniform and non-uniform 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 marginalization 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 $\pm 1.3°$ along each dimension. We encourage the reader to examine the figures in more detail in the digital version of the paper

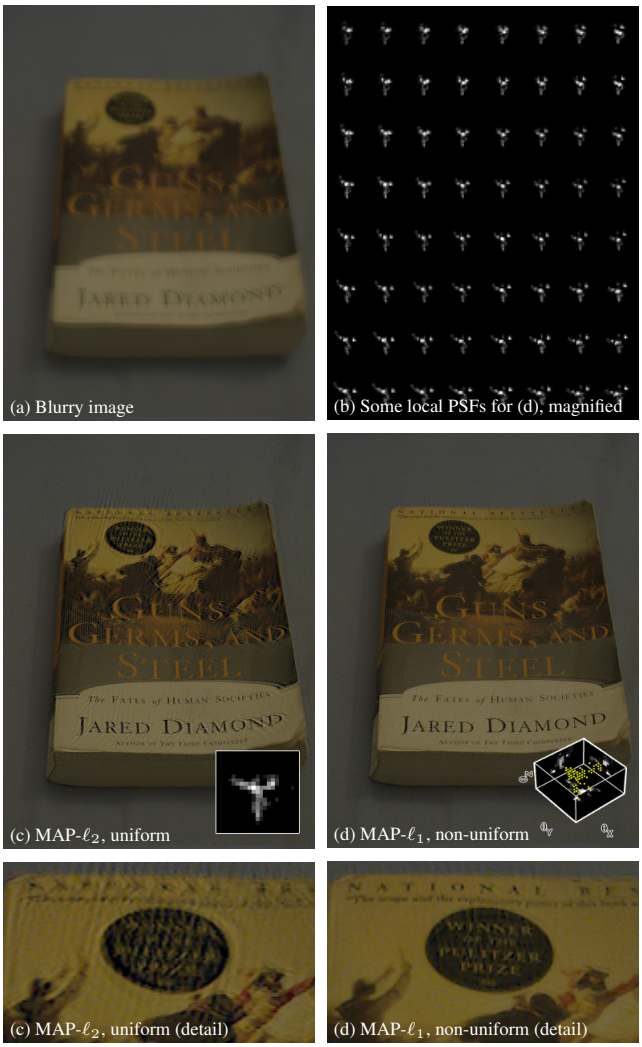


Fig. 6 Blind deblurring of real camera shake, example 2. The result of blind deblurring on a real camera shake image (a), captured with a shutter speed of 1 second, using the MAP approach of Cho and Lee with both the uniform and non-uniform blur models. Also shown in (b) are some of the local PSFs generated 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 blur kernel in (d) covers $\pm 0.4°$ in θ_X and θ_Y, and $\pm 0.9°$ in θ_Z

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 marginalization algorithms exhibit similar performance across the different blur sizes and noise levels, although as demonstrated by the displayed kernels, the MAP-ℓ_1 approach tends to find sparser, less contiguous kernels than the marginalization approach.

Figure 9 shows the failure of the MAP algorithm to produce a good result (using the blurry image from Figure 8 (a)) when using the original ℓ_2 regularization proposed by Cho and Lee (2009) with our non-uniform blur model. As discussed in Section 4.2.1, the kernel produced is highly non-sparse despite the thresholding step, and the deconvolved output correspondingly exhibits many artifacts compared to the MAP-ℓ_1 result in Figure 8 (f).

In Figure 10, we compare our approach to that of Fergus et al. (2006) on a real, 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

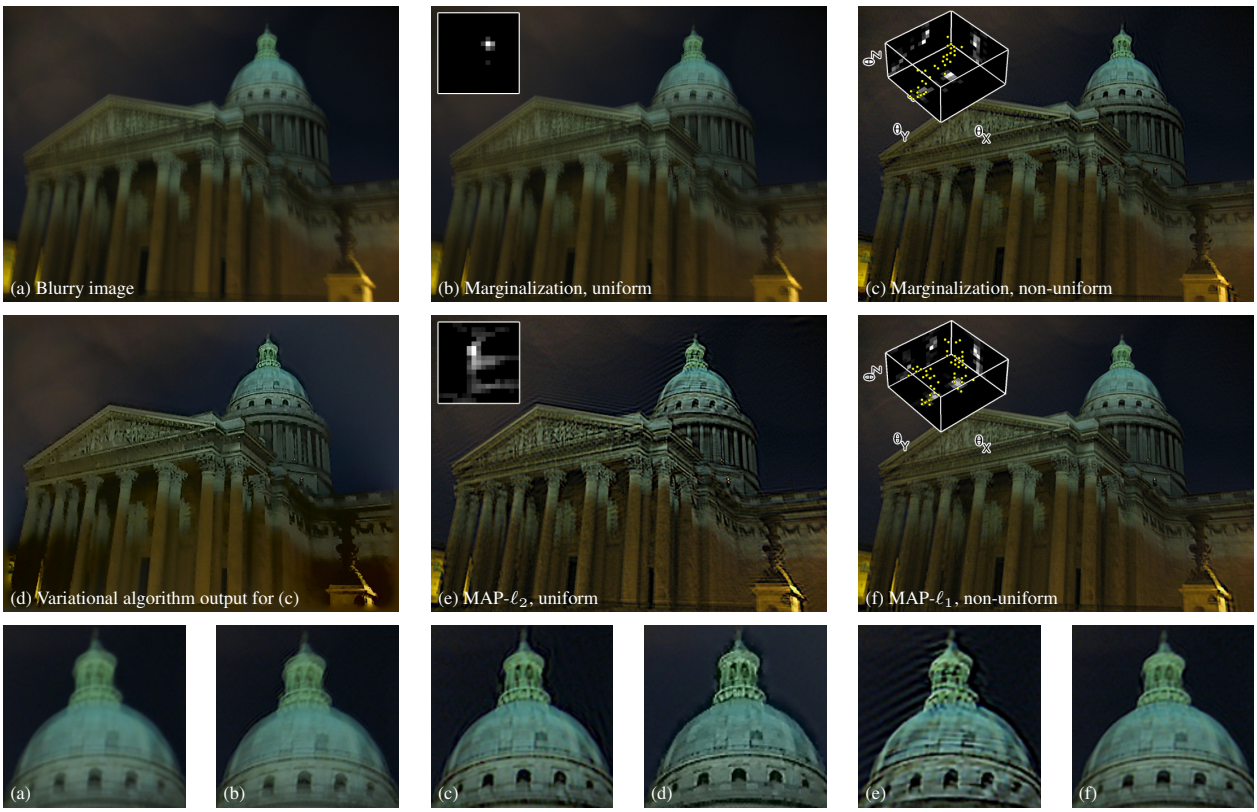


Fig. 8 Blind deblurring of real camera shake, example 3. A hand-held image with camera shake (a), captured with a shutter speed of 1 second, with the results of blind deblurring using the marginalization algorithm of Fergus et al. under both a uniform (b) and non-uniform (c–d) blur model, and the MAP algorithm of Cho and Lee with a uniform (e) and non-uniform (f) blur model. The variational output (d) is estimated using the marginalization 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 artifacts 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 $\pm 0.7^\circ$ in θ_X and θ_Y and $\pm 1.4^\circ$ in θ_Z

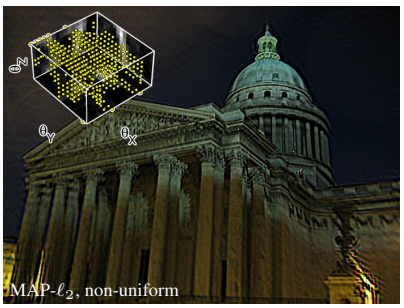


Fig. 9 Poor performance of MAP-ℓ_2 with non-uniform blur model. The corresponding blurry image can be seen in Figure 8. Shown is the estimated kernel and deblurred result when using our non-uniform blur model in the algorithm of Cho and Lee with ℓ_2 regularization on the kernel. As can be seen, the ℓ_2 regularization is not sufficient to produce a good estimate of the kernel, and results in a deblurred output containing many artifacts

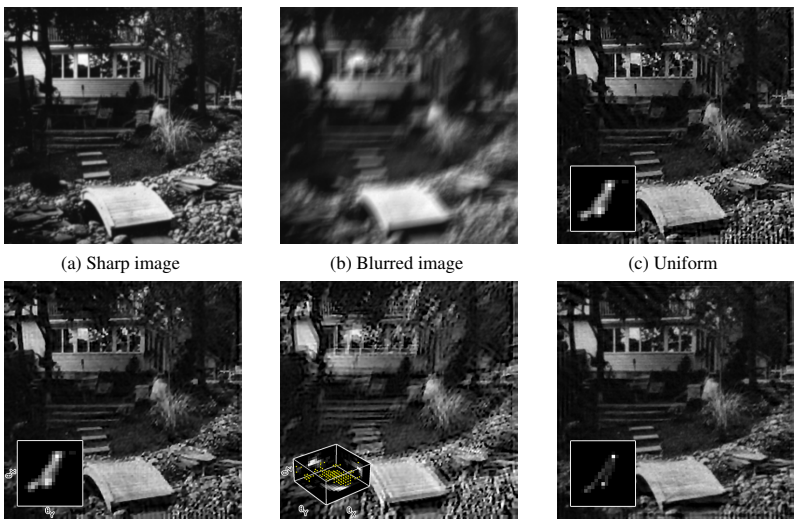


Fig. 10 Blind deblurring of a real uniform blur. A real camera shake blur (a–b) from the dataset of Levin et al. (2009), deblurred using kernels estimated with the marginalization algorithm. We show deblurred results and kernels for four cases; (c) uniform blur using original algorithm of Fergus et al., (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 θ_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 (d), while not perfect, does have the same diagonal shape as the true blur, with the non-zeros centered around a single value of θ_Z

the constraint on θ_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 11 shows the result of the non-uniform MAP-ℓ_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 our decision to ignore the camera's translation is reasonable in practice.

Besides comparing the results of a given algorithm with either a uniform or non-uniform blur model, we can also compare the marginalization 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 image of a street scene, so the application of this prior to an image with a very different distribution of pixels 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 1, the marginalization approach (using the street scene prior) fails to find a useful kernel, while the MAP approach finds a good kernel, as demonstrated by the comparison in Figure 1 (c). 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 α and β, which must be manually specified, while the marginalization approach has almost no parameters to tune.

Convergence. Neither of the algorithms can guarantee the ability to arrive at a globally optimal solution. However, in practice we have found them both to perform reliably. 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 minimizations 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 gradient prediction step helps direct the optimizations process towards a desirable minimum by encouraging the sharp image to contain step edges. In the marginalization algorithm, we have found that the algorithm converges equally reliably for both the uniform model and our model in a similar number of iterations.

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 marginalization algorithm, given in Equations (23-26), are computationally expensive. Due to this expense, and the larger number of iterations required for our model compared to the uniform model, the marginalization algorithm with our non-uniform model can take several hours to deblur an image of several hundred pixels across on a modern workstation. Deblurring larger images with this method is not currently practical, whereas the MAP algorithm can deblur the same images in under an hour.

Limitations. Both of these algorithms are capable of removing large blurs – up to around 50 pixels across, for both uniform and non-uniform blur. For our model, this corresponds to around 3°-5° of rotation around each axis for a photograph whose width and focal length are both 1000px. Since we have assumed that camera translation has a negligible blurring effect, our model (and in general 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. Another typical failure case for the MAP algorithm comes from the fact that it relies on the ability to predict sharp step edges from blurry ones, which may not be the case on images which contain only fine-scale texture, or where the blur is too large to allow this.

6 Deblurring with Noisy / Blurry Image Pairs

In this section, we apply our model to the case where, in addition to $\mathbf{g}$, we have a sharp but noisy image $\mathbf{f}_N$ of the same scene, as proposed by Yuan et al. (2007a). 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, it can initially be used as a proxy for the sharp image, allowing us to estimate the blur kernel $\hat{\mathbf{w}}$ by solving Equation (15). Following this, the kernel is assumed to be known, and used to deblur $\mathbf{g}$, solving (14). The noisy image can also be used to improve this deconvolution step, and Yuan et al. propose a modified version of the Richardson-Lucy algorithm, which uses $\mathbf{f}_N$ to suppress artifacts in the result.

6.1 Kernel Estimation

As discussed in Section 3.2, some prior knowledge must be applied to recover a good kernel estimate. In their algorithm, Yuan et al. (2007a) constrain the kernel to have non-negative elements and unit ℓ_1 norm, however they simultaneously

Fig. 11 Blind deblurring of an image from (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-ℓ_1 method with our blur model (c). 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.

penalize the ℓ_2 norm of the kernel, reducing the sparsity-inducing effect of the constraint. To help find a sparse kernel, they propose a thresholding scheme which sets some kernel elements to zero at each iteration. In our approach, we opt to use the ℓ_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 minimize the following energy function:

$$E_N(\mathbf{w}) = \|\hat{\mathbf{g}}(\mathbf{f}_N, \mathbf{w}) - \mathbf{g}\|_2^2$$
$$\text{s.t.} \quad \forall\, k = 1, \ldots, K, \quad w_k \geq 0 \quad \text{and} \quad \sum_k w_k = 1, \quad (34)$$

where, by analogy with Equation (15), $\hat{\mathbf{g}}(\mathbf{f}_N, \mathbf{w}) = \mathbf{B}_N \mathbf{w}$, and $\mathbf{B}_N$ is the matrix whose columns each contain a projectively transformed copy of $\mathbf{f}_N$. Similar to Equation (29), this least-squares formulation with non-negative ℓ_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 a matrix $\mathbf{B}_N$ in Equation (34) whose columns contained translated versions of $\mathbf{f}_N$, rather than projectively transformed versions.

6.2 Image Reconstruction

Having estimated the blur kernel, Yuan et al. (2007a) propose several modifications to the Richardson-Lucy (RL) algorithm, 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" artifacts – low-frequency ripples spreading across the image, such as in Figure 9 – but using the denoised image it is possible to disambiguate the true low frequencies from these artifacts, 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. 2007a) for full details of the augmented RL algorithm, omitted here for brevity. We have adapted the algorithm for our non-uniform blur model, along the same lines as for the standard RL algorithm in Section 4.3.

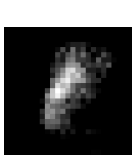
6.3 Results

In this section, we present results with noisy / blurry image pairs, and refer the reader to Section 7 for implementation details. Figures 12 and 13 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. (2007a). As can be seen from the deblurred images obtained with the two models, our results exhibit more detail and fewer artifacts than those using the uniform blur model.

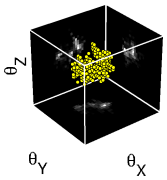
7 Implementation

The implementation of the variational kernel estimation method presented in Section 4.1 is based on the code made available by Miskin and MacKay (2000) and by Fergus et al. (2006)[2]. 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 (23–26). A package containing our code is available online[3]. The implementation of the image reconstruction algorithm of Krishnan and Fergus (2009) is also based on MATLAB code made available online by the authors[4] The implementations of the Richardson-Lucy algorithm, the algorithm of Cho and Lee (2009), and the augmented RL algorithm of Yuan et al. (2007a) 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

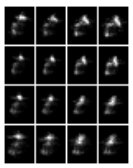
[2] http://cs.nyu.edu/~fergus/research/deblur.html
[3] http://www.di.ens.fr/willow/research/deblurring/
[4] http://cs.nyu.edu/~dilip/research/fast-deconvolution/



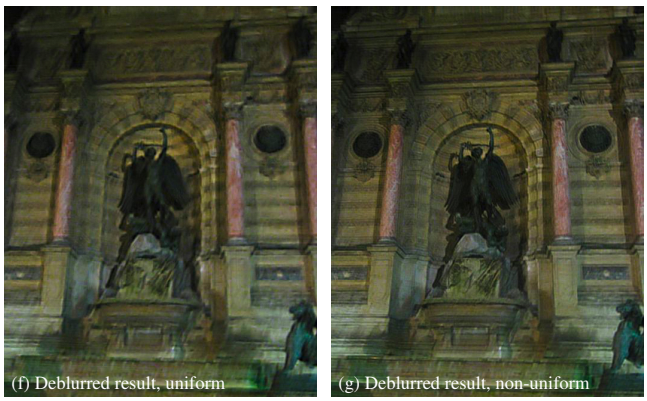
(c) Uniform kernel



(d) Non-uniform kernel

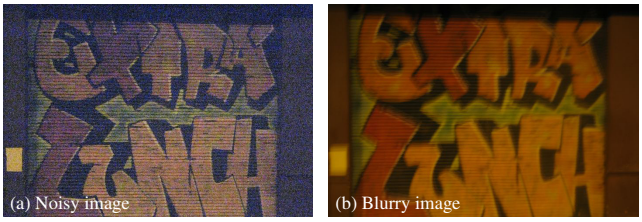


(e) Some local PSFs for (d), magnified

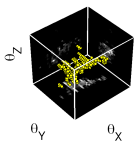


(h) Noisy (i) Blurry (j) Uniform result (k) Non-uniform result

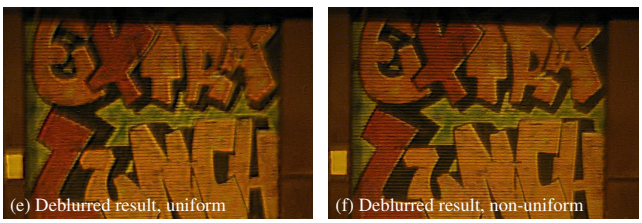
Fig. 12 Deblurring real camera shake blur using a noisy / blurry image pair. A noisy image (a) and a blurry image (b) captured with a hand-held camera, with the estimated kernels (c–d) and deblurred results (f–g) for the uniform and non-uniform blur models. Also shown for illustration are a selection of the local PSFs generated by the rotational kernel (e). As can be seen in the close-ups (h–k), our result (k) contains more details and fewer artifacts than when using the uniform blur model (j), 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. We encourage the reader to examine the images in the digital version of the paper



(c) Uniform kernel



(d) Non-uniform kernel



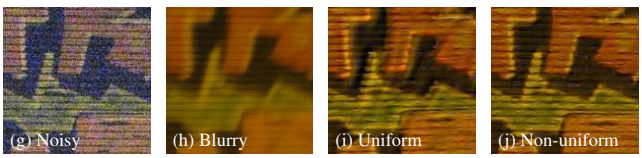


Fig. 13 Deblurring real camera shake blur using a noisy / blurry image pair. A noisy image (a) and blurry image (b) captured with a hand-held camera, shown with the estimated kernels (c–d) and deblurred images (e–f) for the uniform and non-uniform blur models. Note in the close-up that the result using our model (j) has sharper edges and fewer artifacts than that using the uniform model (i). The non-uniform kernel in (d) covers $\pm 3^\circ$ along each dimension

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.

7.1 Sampling the Set of Rotations

One important detail to consider is how finely to discretize 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 θ_X, θ_Y and θ_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 neighbors. Setting the grid spacing in terms of pixels also has the advantage that our 3D blur kernels are de-

fined 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. $[-5°, 5°]$.

7.2 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 (34) directly at full resolution would involve transforming $\mathbf{f}_N$ for every possible rotation under consideration and storing all the copies simultaneously in $\mathbf{B}_N$. 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 paper, we use the solution $\hat{\mathbf{w}}_s$ at each scale s to constrain the solution at the next scale $\hat{\mathbf{w}}_{s+1}$, by defining an "active region" where $\hat{\mathbf{w}}_s$ is non-zero, and constraining the non-zeros at the next scale to lie within this region. In the example above, this corresponds to discarding many columns of $\mathbf{B}_N$, reducing both the computation and memory demands of the algorithm. 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 scale s, we find the optimal kernel $\hat{\mathbf{w}}_s$ for that scale. We then upsample $\hat{\mathbf{w}}_s$ to the next scale $(s+1)$ using bilinear interpolation, find the non-zero elements of this upsampled kernel, and dilate this region using a $3 \times 3 \times 3$ cube. When finding the optimal kernel $\hat{\mathbf{w}}_{s+1}$, we fix all elements outside the active region to zero. We repeat this process at each scale, until we have found the optimal kernel at the finest scale.

7.3 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 $\pm 10°$ 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. 2007a).

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. $\mathbf{f}_N \leftarrow a\mathbf{f}_N$.

8 Conclusion

We have proposed a new model for camera shake, derived from the geometric properties of cameras, and applied it to two deblurring problems within the frameworks of 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. The model assumes that the motion of the camera during exposure is limited to rotations about its optical center, and is temporally-agnostic to the distribution over camera orientations. Note, however, that camera rotations that are off the optical center can be modeled by camera rotations about the optical center together with translation; these translations should generally be small for rotation centers that are not far from the optical center. The model is not applicable for non-static scenes, or nearby scenes with large camera translations where parallax effects may become significant.

In the future, we plan to investigate the use of our general bilinear model to other non-uniform blurs. We also plan to investigate means of reducing the computational overhead of the model, for example with the use of a suitable approximation strategy, such as (Hirsch et al. 2010).

8.1 Acknowledgements

We are grateful for discussions with Bryan Russell, and comments from Fredo Durand and the reviewers. Thank you to James Miskin and Rob Fergus for making their code available. Financial support was provided by ONR MURI N00014-07-1-0182, ERC grant VisRec no. 228180, the MSR-INRIA laboratory, ANR grant HFIBMR (ANR-07-BLAN-0331-01), the EIT-ICT labs (activity 10863), and the ERC grant VideoWorld.

References

M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, 1997.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006. ISBN 0387310738.

J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *Proc. CVPR*, 2009.

A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *Proc. CVPR*, 2010.

T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Trans. Image Processing*, 7(3), 1998.

J. Chen, L. Yuan, C.-K. Tang, and L. Quan. Robust dual motion deblurring. In *Proc. CVPR*, 2008.

S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graphics (Proc. SIGGRAPH Asia 2009)*, 28(5):145:1–145:8, 2009.

S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *Proc. ICCV*, 2007.

F. Couzinie-Devy, J. Mairal, F. Bach, and J. Ponce. Dictionary learning for deblurring and digital zoom. (Submitted) Preprint HAL: inria-00627402, 2011.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image restoration by sparse 3D transform-domain collaborative filtering. In *SPIE Electronic Imaging*, 2008.

B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graphics (Proc. SIGGRAPH 2006)*, 25(3):787–794, 2006.

A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proc. ECCV*, 2010.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge: CUP, second edition, 2004. ISBN 0521540518.

M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Proc. CVPR*, 2010.

N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graphics (Proc. SIGGRAPH 2010)*, 29(4):30:1–30:9, 2010.

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE J. Selected Topics in Signal Processing*, 1(4):606–617, 2007.

G. Klein and T. Drummond. A single-frame visual gyroscope. In *Proc. BMVC*, 2005.

D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 2009.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

A. Levin. Blind motion deblurring using image statistics. In *NIPS*, 2006.

A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. CVPR*, 2009.

S. H. Lim and A. Silverstein. Estimation and removal of motion blur by capturing two images with different exposures. Technical Report HPL-2008-170, HP Laboratories, 2008.

L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79(6):745–754, 1974.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

J. W. Miskin and D. J. C. MacKay. Ensemble learning for blind image separation and deconvolution. In M. Girolani, editor, *Advances in Independent Component Analysis*. Springer-Verlag, 2000.

J. G. Nagy and D. P. O'Leary. Restoring images degraded by spatially variant blur. *SIAM J. Sci. Comput.*, 19(4):1063–1082, 1998.

S. Osher and L. I. Rudin. Feature oriented image enhancement using shock filters. *SIAM J. Numer. Anal.*, 27(4):919–940, 1990.

P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graphics (Proc. SIGGRAPH 2003)*, 22(3):313–318, 2003.

R. C. Puetter, T. R. Gosnell, and A. Yahil. Digital image reconstruction: Deblurring and denoising. *Annu. Rev. Astron. Astrophys.*, 43: 139—94, 2005.

A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3), 2005.

W. H. Richardson. Bayesian-based iterative method of image restoration. *J. of the Optical Society of America*, 62(1):55–59, 1972.

A. A. Sawchuk. Space-variant image restoration by coordinate transformations. *J. of the Optical Society of America*, 64(2):138–144, 1974.

Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *Proc. ICCV*, 2007.

Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graphics (Proc. SIGGRAPH 2008)*, 27(3), 2008.

Y.-W. Tai, P. Tan, L. Gao, and M. S. Brown. Richardson-Lucy deblurring for scenes under projective motion path. Technical report, KAIST, 2009.

Y.-W. Tai, H. Du, M. S. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE PAMI*, 32(6):1012–1028, 2010a.

Y.-W. Tai, N. Kong, S. Lin, and S. Y. Shin. Coded exposure imaging for projective motion deblurring. In *Proc. CVPR*, 2010b.

Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-Lucy deblurring for scenes under a projective motion path. *IEEE PAMI*, 33(8):1603–1618, 2011.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Royal Stat. Soc. B*, 58(1):267–288, 1996.

C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. ICCV*, 1998.

R. Vio, J. Nagy, L. Tenorio, and W. Wamsteker. Multiple image deblurring with spatially variant PSFs. *Astronomy & Astrophysics*, 434:795–800, 2005.

L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proc. ECCV*, 2010.

L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graphics (Proc. SIGGRAPH 2007)*, 26(3), 2007a.

L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Blurred/non-blurred image alignment using sparseness prior. In *Proc. ICCV*, 2007b.

L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graphics (Proc. SIGGRAPH 2008)*, 27(3), 2008.

A Parameter update equations for marginalization algorithm

Equations (23–26) cannot be evaluated directly, since they involve expectations over combinations of variables. For implementation they must be expanded and written in terms of the mean and variance of individual variables. Here we give these expansions, which are exactly the parameter updates computed in our implementation. First, let

$$v_k^w = \left\langle w_k^2 \right\rangle - \left\langle w_k \right\rangle^2 \tag{35}$$

$$v_j^f = \left\langle f_j^2 \right\rangle - \left\langle f_j \right\rangle^2 \tag{36}$$

$$\left\langle A_{ij} \right\rangle = \sum_k C_{ijk} \left\langle w_k \right\rangle \tag{37}$$

$$\langle B_{ik} \rangle = \sum_j C_{ijk} \langle f_j \rangle \tag{38}$$

$$\langle \hat{g}_i \rangle = \sum_k \left(\sum_j C_{ijk} \langle f_j \rangle \right) \langle w_k \rangle . \tag{39}$$

Then,

$$w_k^{(2)} = \langle \beta_\sigma \rangle \sum_{i,j} C_{ijk}^2 v_j^f + \langle \beta_\sigma \rangle \sum_i \langle B_{ik} \rangle^2 \tag{40}$$

$$\begin{aligned} w_k^{(1)} w_k^{(2)} = {} & \langle \beta_\sigma \rangle \sum_i \langle B_{ik} \rangle \left(g_i - \langle \hat{g}_i \rangle \right) \\ & - \langle \beta_\sigma \rangle \sum_{i,j} C_{ijk} \langle A_{ij} \rangle v_j^f + \langle w_k \rangle w_k^{(2)} \end{aligned} \tag{41}$$

$$f_j^{(2)} = \langle \beta_\sigma \rangle \sum_{i,k} C_{ijk}^2 v_k^w + \langle \beta_\sigma \rangle \sum_i \langle A_{ij} \rangle^2 \tag{42}$$

$$\begin{aligned} f_j^{(1)} f_j^{(2)} = {} & \langle \beta_\sigma \rangle \sum_i \langle A_{ij} \rangle \left(g_i - \langle \hat{g}_i \rangle \right) \\ & - \langle \beta_\sigma \rangle \sum_{i,k} C_{ijk} \langle B_{ik} \rangle v_k^w + \langle f_j \rangle f_j^{(2)}. \end{aligned} \tag{43}$$

Finally, in evaluating the parameters of the distribution $q(\beta_\sigma)$ for the variance of the noise, (Miskin and MacKay 2000, Eqn. 40), it is necessary to evaluate the following quantity:

$$\begin{aligned} \left\langle (g_i - \hat{g}_i)^2 \right\rangle = {} & (g_i - \langle \hat{g}_i \rangle)^2 + \sum_{j,k} C_{ijk}^2 v_j^f v_k^w \\ & + \sum_j \langle A_{ij} \rangle^2 v_j^f + \sum_k \langle B_{ik} \rangle^2 v_k^w. \end{aligned} \tag{44}$$

B Computation of interpolation coefficients

Here we give details of how the values of C_{ijk} can be calculated if bilinear interpolation is used. Note that these are standard interpolation weights, and are not specific to this deblurring application. We consider a pixel g_i in the blurry image which is mapped under a homography $\mathbf{H}_k$ to a point $\mathbf{x}'$ in the sharp image, i.e. $\mathbf{H}_k \mathbf{x}_i \sim (x', y', 1)^\top$. The point $\mathbf{x}'$ will, in general, be a sub-pixel location, and the value of the sharp image $\mathbf{f}$ at $\mathbf{x}'$ is interpolated from the 4 pixels surrounding $\mathbf{x}'$, using the following weights:

(x_j, y_j)	C_{ijk}
$(\lfloor x' \rfloor, \lfloor y' \rfloor)$	$(\lfloor x' \rfloor + 1 - x')(\lfloor y' \rfloor + 1 - y')$
$(\lfloor x' \rfloor, \lfloor y' \rfloor + 1)$	$(\lfloor x' \rfloor + 1 - x')(y' - \lfloor y' \rfloor)$
$(\lfloor x' \rfloor + 1, \lfloor y' \rfloor)$	$(x' - \lfloor x' \rfloor)(\lfloor y' \rfloor + 1 - y')$
$(\lfloor x' \rfloor + 1, \lfloor y' \rfloor + 1)$	$(x' - \lfloor x' \rfloor)(y' - \lfloor y' \rfloor)$

where $\lfloor \cdot \rfloor$ takes the integer part of a positive scalar. The correspondence between a pixel f_j's index j and coordinates (x_j, y_j) can be obtained using, for example, the MATLAB functions `ind2sub` / `sub2ind`.