Non-uniform Deblurring for Shaken Images

Oliver Whyte[1,4]   Josef Sivic[1,4]   Andrew Zisserman[2,4]   Jean Ponce[3,4]

[1]INRIA   [2]Dept. of Engineering Science, University of Oxford   [3]Ecole Normale Supérieure

Abstract

Blur from camera shake is mostly due to the 3D rotation of the camera, resulting in a blur kernel that can be significantly non-uniform *across the image. However, most current deblurring methods model the observed image as a convolution of a sharp image with a* uniform *blur kernel. We propose a new parametrized geometric model of the blurring process in terms of the rotational velocity of the camera during exposure. We apply this model to two different algorithms for camera shake removal: the first one uses a single blurry image (blind deblurring), while the second one uses both a blurry image and a sharp but noisy image of the same scene. 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 real images.*

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 approaches model the blurred image as a convolution of a sharp image with a *spatially uniform* filter [5, 10, 24, 29]. However, real camera shake does not in general cause uniform blur [13], as illustrated by Figure 1.

In this paper we propose a geometrically consistent model of *non-uniform* image blur due to camera shake, arising from rotations of the camera. We develop a global representation of such parametrically non-uniform blur, using a single "blur kernel" analogous to (but different from) a convolution kernel, and demonstrate its ability to model a more general class of blurs than previous approaches. We demonstrate the effectiveness of our model by using it to replace the uniform blur model in two existing approaches to camera shake removal, and show quantitative and qualitative improvements in the results. In addition, we show that uniform blur is a special case of our model.

Figure 1. **Visible non-uniformity of blur in a shaken image.** Left: The blurry image. Right: Close-ups of different parts of the image. Note the differences in shape of the blur between the middle and bottom close-ups.

Specifically, we consider the problems of "blind" deblurring, where only a single blurred 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 & MacKay [16] and Fergus *et al.* [10] for the blind case, and Yuan *et al.* [29] for the case of a noisy / blurred image pair.

1.1. Related Work

Previous work on non-uniform deblurring has focused on piecewise-uniform blurs arising from multiple moving objects in the scene [7, 12], spatially varying combinations of localized uniform blurs [17, 27], or blur arising from rotations of planar objects in the scene [24]. However, apart from Sawchuk [22], who assumes a known transformation, these approaches generally rely on the assumption that the blur is locally uniform, and do not consider global models for continuously varying blur, such as those arising from arbitrary rotations of the camera about its optical center during exposure, as modelled in our work.

If the blur kernel is known, standard deblurring techniques such as the Wiener filter (for uniform blur) or the Richardson-Lucy algorithm (for general blur) can be used to recover the original sharp image [3, 8, 14, 19, 21]. However, blind image deblurring with an unknown blur kernel is a difficult problem, which is typically addressed by first

estimating the kernel, and then estimating the sharp image assuming the kernel is known. Blind camera shake removal from a single image using a uniform blur model and priors on image statistics has been addressed in [10, 13, 16, 23]. To simplify the deblurring problem, others have considered using additional information in the form of additional blurry images [6, 20], or a sharp but noisy image of the same scene [29].

2. Geometric Model

To motivate our approach, we begin with the assertion that in most cases of camera shake, the rotation of the camera during exposure has a significantly larger effect than its translation [13]: Typical consumer cameras have a field of view of about $50°$, and rotating the camera by only a small fraction of this, *e.g.* $1° = \frac{1}{50} \times 50°$, during exposure will cause an image blur whose size in the image approximately follows the same proportion, *i.e.*, a 20 pixel blur on a 1000 pixel-wide image. Translation of the camera, on the other hand, will cause a blur whose size is inversely proportional to the depth of the scene, and reaching the same 20 pixel blur for an object at a depth of 2m would require translating the camera by about 4cm. The rotation of $1°$ represents a significantly smaller motion of the camera and, in most cases, camera rotation can be assumed to be the only significant source of camera shake blur.

2.1. Motion Blur and Homographies

We assume from now on that the only significant motion of the camera is a rotation about its optical center, and that the scene being photographed is static. It is well known that in this case, under the pinhole model of a camera, all views seen by the camera are projectively equivalent, excluding boundary effects. This means that 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 in the case of a camera with known internal parameters, the homography $\mathbf{H}$ is parameterized by the 3×3 rotation matrix $\mathbf{R}$ describing the rotation of the camera [11]:

$$\mathbf{H} = \mathbf{KRK}^{-1}, \tag{1}$$

where $\mathbf{K}$ is the camera's internal calibration matrix.

The matrix $\mathbf{R}$ requires only 3 parameters, and we adopt here the "angle-axis" representation, in which a rotation is described by the angle θ moved about an axis $\mathbf{a}$ (a unit-norm 3-vector), summarized by the vector $\boldsymbol{\theta} = \theta\mathbf{a} = (\theta_X, \theta_Y, \theta_Z)^\top$. We fix our 3D coordinate frame to have its origin at the camera's optical center, with the XY-plane aligned with the camera sensor's coordinate frame and the Z-axis parallel to the camera's optical axis. $\mathbf{R}$ is given by

the matrix exponential

$$\mathbf{R}(\boldsymbol{\theta}) = e^{[\boldsymbol{\theta}]\times}, \quad \text{where} \tag{2}$$

$$[\boldsymbol{\theta}]_\times = \begin{bmatrix} 0 & -\theta_Z & \theta_Y \\ \theta_Z & 0 & -\theta_X \\ -\theta_Y & \theta_X & 0 \end{bmatrix}. \tag{3}$$

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 case, each point on the sensor would see 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, parametrized by $\boldsymbol{\theta}(t)$, each ray from the static scene will trace a sequence of points on the image. For each point $\mathbf{x}$ in the observed blurred image we can trace the record of rays $\mathbf{x}'(t)$ contributing to it as:

$$\mathbf{x}'(t) \sim \mathbf{H}_t\mathbf{x}, \tag{4}$$

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

$$g(\mathbf{x}) = \int_0^T f(\mathbf{H}_t\mathbf{x}) \, dt + \varepsilon, \tag{5}$$

where, with a slight abuse of notation, $\mathbf{H}_t\mathbf{x}$ denotes inhomogeneous coordinates of a point in f.

In general, a single blurry image has no temporal information associated with it, so it is convenient to replace the temporal integral in (5) by a weighted integral over the set of all possible rotations $\mathcal{R}$:

$$g(\mathbf{x}) = \int_{\mathcal{R}} f(\mathbf{H}_{\boldsymbol{\theta}}\mathbf{x})w(\boldsymbol{\theta}) \, d\boldsymbol{\theta} + \varepsilon, \tag{6}$$

where the weight function $w(\boldsymbol{\theta})$ corresponds to the time the camera spends at the orientation $\boldsymbol{\theta}$ while the shutter is open.

According to this model, the apparent motion of pixels may be significantly non-uniform across the image. Figure 2 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. As the focal length increases, this out-of-plane blur becomes more uniform, however most consumer cameras operate at focal lengths of the same order as the sensor width. In addition, as argued by Levin *et al.* [13], the assumption of zero in-plane rotation is often violated. From this, it is clear that modelling camera shake as a convolution is insufficient to fully describe its effects.
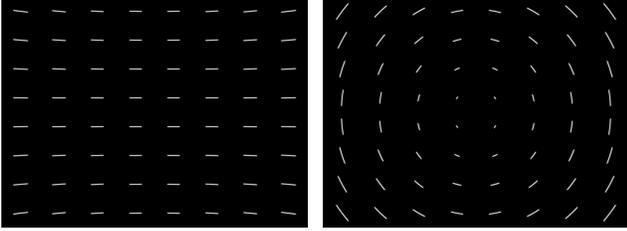
Figure 2. **The paths followed by image points under single-axis rotations.** Left: Rotation about the Y-axis. Right: Rotation about the Z-axis. Under single-axis camera rotations, the paths followed by points in the image are visibly curved and non-uniform across the image. The focal length of the camera 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.

2.2. Camera Calibration

In order to compute the homography in Equation (1), we need to know the camera's internal parameters. 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.

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 [2], which uses a database of lens and camera parameters to correct for the distortion.

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 $\mathbf{g} \in \mathbb{R}^N$, with $N = H \times W$ 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. By analogy with convolutional blur, we refer to $\mathbf{w}$ as the *blur kernel*. Each element w_k corresponds to a camera orientation $\boldsymbol{\theta}_k$, and consequently to a homography $\mathbf{H}_k$, and in general $\mathbf{w}$ will be very sparse, since the camera will have passed through only a few of these orientations during exposure. Discretizing Eqn. (6), each observed pixel g_i is modelled as:

$$g_i = \sum_k \left(\sum_j C_{ijk} f_j \right) w_k + \varepsilon, \qquad (7)$$

where i, j and k index into the observed image, the sharp image and the blur kernel, respectively. For an observed pixel g_i with coordinate vector $\mathbf{x}_i$, the sum $\sum_j C_{ijk} f_j$ interpolates the point $f(\mathbf{H}_k \mathbf{x}_i)$ in the sharp image, with C_{ijk} being the coefficients of, for example, bilinear interpolation.

Thanks to the bilinear form of Eqn. (7), note that when either the blur kernel or the sharp image is known, the blurry image is linear in the remaining unknowns, *i.e.*

$$\mathbf{g} = \mathbf{A}\mathbf{f} + \varepsilon, \qquad (8)$$

$$\text{or} \qquad \mathbf{g} = \mathbf{B}\mathbf{w} + \varepsilon, \qquad (9)$$

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.

4. Applications

In this section, we outline two applications of our blur model, where the aim is to recover an estimate $\hat{\mathbf{f}}$ of the true sharp image $\mathbf{f}$. Generally, blind deblurring algorithms approach this by also attempting to estimate a 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 $\hat{g}_i = \sum_k (\sum_j C_{ijk} \hat{f}_j) \hat{w}_k$.

Blind deblurring. In Section 5, we examine the case of blind deblurring, where we have only a single blurred image $\mathbf{g}$ from which to estimate $\hat{\mathbf{f}}$. In this work, we modify the algorithms of Miskin & MacKay [16] and Fergus *et al.* [10], which attempt to estimate the blur kernel $\hat{\mathbf{w}}$ using a variational inference approach. The estimated kernel $\hat{\mathbf{w}}$ is then used to deblur $\mathbf{g}$ directly, using the Richardson-Lucy (RL) algorithm, to give $\hat{\mathbf{f}}$. In this algorithm, the kernel estimation step uses the model in Eqn. (7), then, assuming the kernel to be known, the image reconstruction step attempts to invert Eqn. (8).

Deblurring with noisy / blurry image pairs. In Section 6, 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.* [29]. The noisy image is first used as a proxy for the sharp image in order to estimate the blur kernel $\hat{\mathbf{w}}$, using the form in Eqn. (9). In the second step, the kernel is again assumed to be known, and used to deblur $\mathbf{g}$, inverting (8). However, in this case we also modify the RL algorithm (as proposed by Yuan *et al.*) using $\mathbf{f}_N$ to suppress ringing.

4.1. Constraints and Priors for Blur Kernels

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. To obtain a useful solution, it is thus necessary to add some regularization and/or constraints on the quantities being estimated.

The first thing to note is that the elements of $\mathbf{w}$ and $\mathbf{f}$ must be non-negative, since each coefficient w_k records an elapsed time, and each pixel value f_j records a number of incident photons. Furthermore, we may only recover $\mathbf{f}$ and $\mathbf{w}$ up to scale, since $\hat{\mathbf{g}}(\hat{\mathbf{f}}, \hat{\mathbf{w}}) = \hat{\mathbf{g}}(\frac{1}{\alpha}\hat{\mathbf{f}}, \alpha\hat{\mathbf{w}})$, so we may fix the norm of either one. A natural choice is to constrain the ℓ_1 norm of $\hat{\mathbf{w}}$ to be unity, so that $\hat{\mathbf{f}}$ will occupy the same range of values as $\mathbf{g}$.

A final useful observation about the kernel is that it is caused by the camera following a path through the set of rotations. Thus a natural prior is that it should be sparse, since the camera will have only passed through a small subset of all possible rotations. This sparsity prior has been a feature of previous camera shake removal algorithms, and has also been leveraged for the alignment of blurred / non-blurred images [28]. Fergus *et al*. [10] encourage sparsity by placing a mixture-of-exponentials prior on the kernel values, while Yuan *et al*. [29] proceed by hysteresis thresholding in scale-space.

5. Blind Deblurring

One of the most successful [13] algorithms for blind deblurring is the variational inference approach of Miskin & MacKay [16], designed for simultaneous deblurring and source separation, which has been adapted by Fergus *et al*. [10] to the removal of camera shake blur. Fergus *et al*. use this algorithm to estimate the blur kernel, and obtain the final sharp image by "deconvolving" the blurry image with this kernel, using the Richardson-Lucy algorithm. In this section, 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 that doing so improves the deblurred results.

Kernel Estimation. The algorithm proposed by Miskin & MacKay [16] attempts to marginalize the posterior distribution for both the kernel and the sharp image $p(\mathbf{f}, \mathbf{w}|\mathbf{g})$ over the sharp image $\mathbf{f}$ to obtain an estimate of the blur kernel $\hat{\mathbf{w}}$, using a variational method to approximate the true posterior by a simpler, factorized distribution. Fergus *et al*. [10] successfully adapted this algorithm to the removal of uniform camera shake blur from photographs by applying it within a multiscale framework and in the gradient domain, using priors on the kernel and sharp image learnt from real data.

We apply the priors learnt by Fergus *et al*. directly in our own implementation. The observation noise ε is assumed to be Gaussian, and 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.

We follow [16] and collect the latent variables $\mathbf{f}$, $\mathbf{w}$, and β_σ into an "ensemble" $\boldsymbol{\Theta}$. The aim is to find the factorized distribution $q(\boldsymbol{\Theta}) = q(\beta_\sigma)\prod_j q(f_j)\prod_k q(w_k)$ that best approximates the true posterior $p(\boldsymbol{\Theta}|\mathbf{g})$, by minimizing the following cost function ([16, Eqn. (10)]) over both the form and the parameters of $q(\boldsymbol{\Theta})$:

$$C_{\mathrm{KL}} = \int q(\boldsymbol{\Theta}) \left[\ln \frac{q(\boldsymbol{\Theta})}{p(\boldsymbol{\Theta})} - \ln p(\mathbf{g}|\boldsymbol{\Theta})\right] \mathrm{d}\boldsymbol{\Theta}. \quad (10)$$

Minimizing this cost function is equivalent to minimizing the Kullback-Leibler (KL) divergence between the posterior and the approximating distribution [4], 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 [16]. However the parameter update equations differ significantly and we have calculated these afresh (the equations and their derivation are given at [1]). Having found the optimal $q(\boldsymbol{\Theta})$, the expectation of $q(\mathbf{w})$ is taken to be the optimal blur kernel, *i.e.*, $\hat{\mathbf{w}} = \langle\mathbf{w}\rangle_{q(\mathbf{w})}$, where $\langle\cdot\rangle_q$ represents the expectation with respect to the distribution q, while the latent image distribution $q(\mathbf{f})$ is discarded.

Image Reconstruction. Having estimated the blur kernel for the blurry image, we wish to invert Eqn. (8) in order to estimate the sharp image $\hat{\mathbf{f}}$. This process is often referred to as deconvolution, and while classical algorithms exist for this process [3, 8, 19], they are generally applicable only to uniform blur, relying on convolutions or the ability to work in the Fourier domain. One method frequently used for deconvolution is the Richardson-Lucy algorithm [14, 21]. This algorithm can be applied to general linear systems as well as to convolutional blurs, using the notation of Eqn. (8) for a known blur [25]. The algorithm iteratively improves the estimate $\hat{\mathbf{f}}$ using the following update equation:

$$\hat{\mathbf{f}} \leftarrow \hat{\mathbf{f}} \odot \left(\mathbf{A}^\top \left(\mathbf{g} \oslash \mathbf{A}\hat{\mathbf{f}}\right)\right), \quad (11)$$

where $\mathbf{g}$ is the observed blurry image, and the matrix $\mathbf{A}$ depends on the estimated non-uniform blur. Here, $\mathbf{u} \odot \mathbf{v}$ represents the element-wise product and $\mathbf{u} \oslash \mathbf{v}$ the element-wise division of two vectors $\mathbf{u}$ and $\mathbf{v}$.

5.1. Results

We compare in this section our results to those obtained with the code provided by Fergus *et al*. [10] on both synthetic and real data. Implementation details are discussed in Section 7.

Figures 3 and 4 show blind deblurring results on images blurred by real camera shake. Our approach is able to capture and remove the blur, while the uniform algorithm of Fergus *et al*. fails to find meaningful kernels or good deblurred results. This is explained by both the short focal length (typical of compact cameras), and the fact that the kernels estimated using our algorithm exhibit significant in-plane components.

Figure 5 shows results for blind deblurring of synthetic images using the two methods, and demonstrates two im-
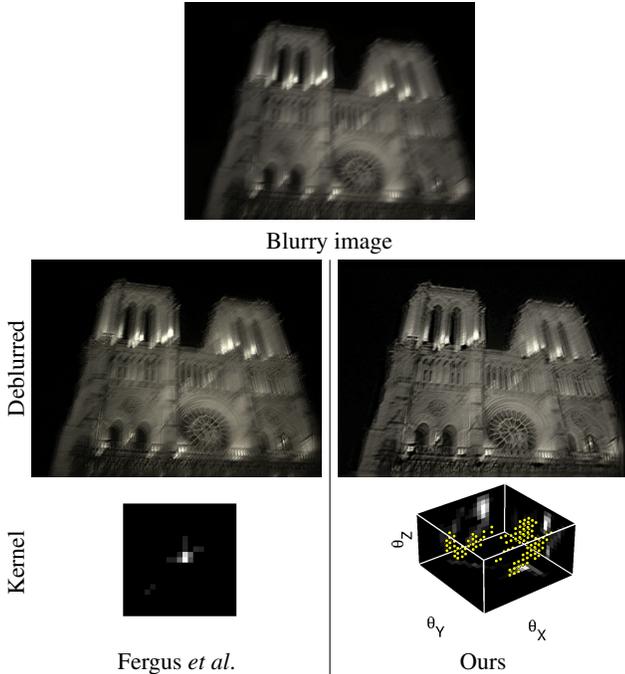
Figure 3. **Blind deblurring of real camera shake, example 1.** The result of blind deblurring on a real camera shake image, captured with a shutter speed of $\frac{1}{2}$ second, using the algorithm of Fergus *et al*. and our non-uniform approach. Our approach is able to recover a useful kernel and a good deblurred image, while the uniform algorithm of Fergus *et al*. fails to find a meaningful kernel. The rotational kernel visualized in the right-hand column shows the non-zero kernel elements plotted as points in the 3D rotational parameter space $(\theta_X, \theta_Y, \theta_Z)$. Each of the cuboid's faces shows the projection of the kernel onto that face. Note that our estimated rotational kernel has a significant in-plane component (non-zeros over many values of θ_Z).

portant points: first, small out-of-plane (*e.g*. Y-axis) components of a blur are sufficiently uniform that the two models both perform well, although the rotational model performs better. Second, only our approach is able to remove 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 non-uniformity of the blur.

In Figure 6, we compare our approach to that of Fergus *et al*. [10] on a real blurred image, taken from the dataset of Levin *et al*. [13], 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 [10]. When we do not apply the constraint on θ_Z, our algorithm still produces a good result, but unsurprisingly does not perform as well, since there is a much larger number of parameters to estimate (K is increased by a factor of 8).
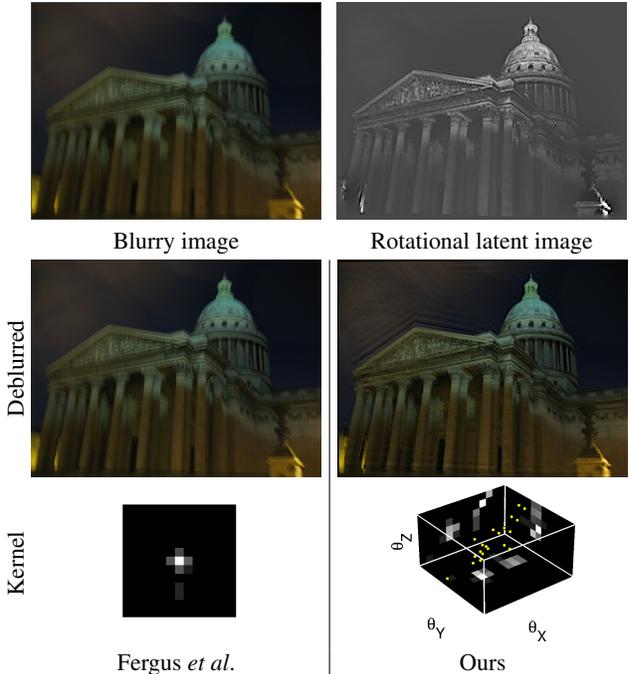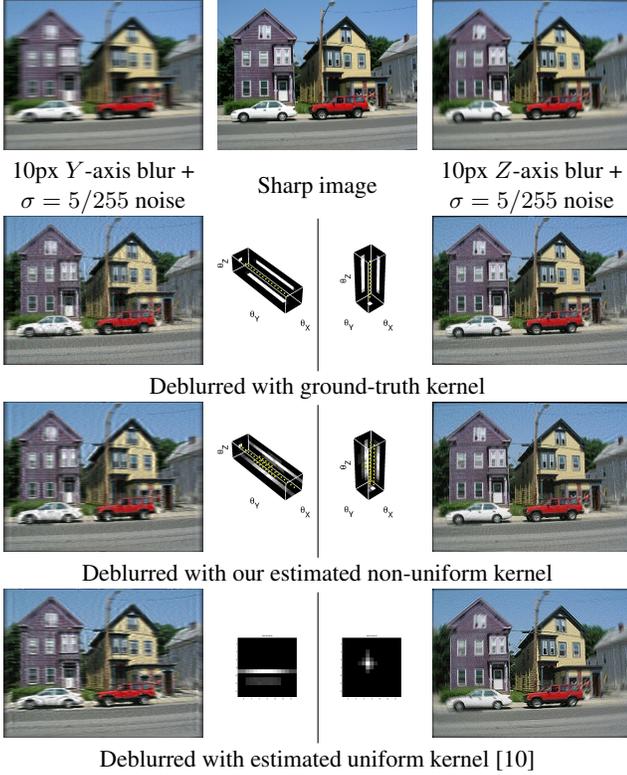

Figure 4. **Blind deblurring of real camera shake, example 2.** A hand-held image with camera shake, captured with a shutter speed of 1 second, with the results of blind deblurring using the algorithm of Fergus *et al*. and our approach. Also shown for illustration is the estimated latent image from the variational inference in the non-uniform case (calculated as $\langle \mathbf{f} \rangle_{q(\mathbf{f})}$ then converted from gradients to intensities using Poisson reconstruction [18]). Our result shows much more detail than that of the uniform model, and while our deblurred result exhibits some "ringing", these artifacts are not present in the latent image, suggesting that they are largely a product of the Richardson-Lucy image reconstruction.

6. Deblurring with Noisy / Blurry Image Pairs

Another successful method for removing camera shake, proposed by Yuan *et al*. [29], takes an additional input in the form of a noisy image $\mathbf{f}_N$ of the scene. 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 direct recovery of a good sharp image, it can be used as a proxy for the sharp image in order to accurately estimate the blur kernel, and can also be used to improve the subsequent image reconstruction process.

Kernel Estimation. As discussed in Section 4.1, some prior knowledge must be applied to recover a good kernel estimate. In their algorithm, Yuan *et al*. [29] constrain the kernel to have unit ℓ_1 norm, however they simultaneously penalize the ℓ_2 norm of the kernel, reducing the sparsity-inducing effect of the constraint, and giving rise to the need for thresholding. In our approach, we opt to use the ℓ_1 and positivity constraints alone, since they lead naturally to a sparse kernel [26], a fact also exploited by Shan *et al*. [24]

10px Y-axis blur + $\sigma = 5/255$ noise

Sharp image

10px Z-axis blur + $\sigma = 5/255$ noise

Deblurred with ground-truth kernel

Deblurred with our estimated non-uniform kernel

Deblurred with estimated uniform kernel [10]

Y-axis	10px		20px		30px	
	R	U	R	U	R	U
$\sigma = 0$	**23.1** (1.4)	23.2 (1.4)	**27.2** (1.1)	58.1 (2.4)	**32.2** (1.1)	129.3 (4.4)
$\sigma = 5$	**24.9** (1.3)	25.8 (1.3)	**29.0** (1.1)	56.8 (2.2)	**33.4** (1.1)	62.9 (2.1)
$\sigma = 10$	**27.0** (1.2)	30.1 (1.3)	**30.7** (1.1)	48.7 (1.8)	**41.9** (1.3)	57.8 (1.8)

Z-axis	R	U	R	U	R	U
$\sigma = 0$	**14.4** (1.3)	21.8 (2.0)	**18.1** (1.0)	26.1 (1.6)	**25.4** (1.2)	57.6 (2.7)
$\sigma = 5$	**17.4** (1.2)	24.8 (1.7)	**23.2** (1.2)	54.5 (2.8)	**30.6** (1.3)	58.6 (2.5)
$\sigma = 10$	**22.0** (1.1)	50.9 (2.7)	**26.5** (1.1)	55.8 (2.4)	**30.0** (1.2)	57.5 (2.7)

RMS errors between deblurred results and true sharp image

Figure 5. **Blind deblurring of synthetic single-axis blurs.** A sharp image (center) 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 on a range of blur sizes and noise levels, and the reconstruction errors are summarized in the tables at the bottom. 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 10, 20, and 30 pixel blurs, using our model (R) and the uniform model (U). 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.

for blur kernel estimation.

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

$$\min_{\hat{\mathbf{w}}} \|\mathbf{g} - \hat{\mathbf{g}}(\mathbf{f}_N, \hat{\mathbf{w}})\|_2^2, \text{ s.t. } \|\hat{\mathbf{w}}\|_1 = 1, \ \hat{w}_k \geq 0 \ \forall k \quad (12)$$

where, by analogy with Eqn. (9), $\hat{\mathbf{g}}(\mathbf{f}_N, \hat{\mathbf{w}}) = \mathbf{B}_N \hat{\mathbf{w}}$, where $\mathbf{B}_N$ is the matrix whose columns contain all the projectively transformed copies of $\mathbf{f}_N$. This least-squares formulation with non-negative ℓ_1 constraints is an instance of the Lasso problem [26], for which efficient optimization algorithms

Sharp image

Blurred image

Deblurred

Kernel

Fergus *et al.*

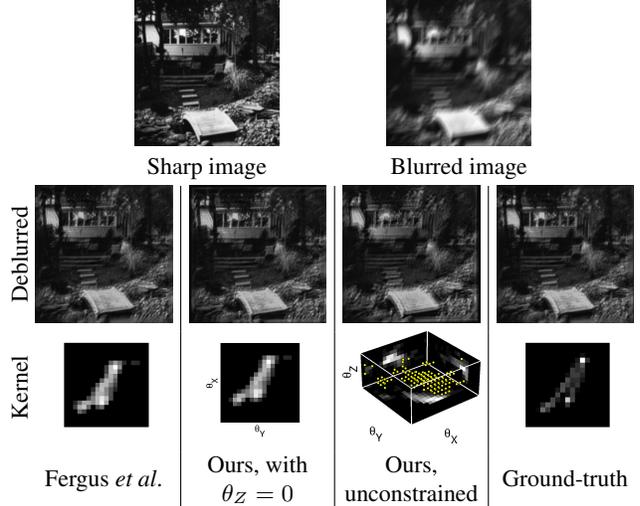Ours, with $\theta_Z = 0$

Ours, unconstrained

Ground-truth

Figure 6. **Blind deblurring of a real uniform blur.** A real camera shake blur from the dataset of [13], with the deblurred results and kernels for four cases; the original algorithm of Fergus *et al.*, our approach with a large focal length and no in-plane rotation ($\theta_Z = 0$), our approach with θ_Z unconstrained, and the ground-truth (uniform) kernel. Notice that this unconstrained kernel has the same diagonal shape as the true blur, and that the non-zeros are centered around a single value of θ_Z.

exist [9, 15], and we use these algorithms here.

For comparison, we have also implemented this algorithm for uniform blurs, using a matrix $\mathbf{B}_N$ in Eqn. (12) whose columns contained translated versions of $\mathbf{f}_N$, rather than projectively transformed versions.

Image Reconstruction. Having estimated the blur kernel, Yuan *et al.* [29] 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 4 – but using the denoised image it is possible to disambiguate the true low frequencies from these artifacts, and largely remove them from the result. We refer the reader to [29] for full details of the modified 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 5.

6.1. Results

In this section, we present results with noisy / blurry image pairs, and refer the reader to Section 7 for implementation details. Figure 7 shows a comparison between the uniform model and our rotational one, using the algorithm described above to estimate the blur kernels. Having estimated the kernel, we deblur the blurred images using the modified RL algorithm of Yuan *et al.* [29]. 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 blind kernel estimation method presented in Section 5 is based on the code made available by Miskin & MacKay [16] and by Fergus *et al*. [10]. We have modified the algorithm to use a kernel defined over rotations of the camera rather than a convolution kernel, and replaced the parameter update equations with the corresponding versions derived for our bilinear blur model (see [1]). The implementations of the Richardson-Lucy algorithm, and the modified RL algorithm of Yuan *et al*. [29] are our own, and we use these implementations for both blur models when comparing results.

Sampling the set of rotations. One important detail to consider is how finely to discretize the rotational parameter θ. Undersampling the set of rotations 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 rotational parameters θ_X, θ_Y and θ_Z, doubling the sampling resolution increases the number of kernel elements by a factor of 8, so the choice is an important one. In practice, we have found that a good choice of sample spacing is one which corresponds approximately to a displacement of 1 pixel at the edge of the image. Since we are fundamentally limited by the resolution of our images, setting the resolution higher leads to redundant rotations, that are indistinguishable from their neighbours. 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 θ, *e.g.* $[-5°, 5°]^3$.

Multiscale implementation. Both 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 blind deblurring, the reason for this is that the variational algorithm is susceptible to poor local minima, and performing the optimization at increasingly fine scales can help to avoid this. When deblurring with noisy / blurry image pairs, the problem is that the kernel at the original resolution may have thousands or tens of thousands of elements. However, very few of these should have non-zero values. To solve Eqn. (12) 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 this may furthermore cause $\mathbf{B}_N$ to become impractically large.

In both 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"

Noisy   Blurred

Uniform kernel   Rotational kernel   Non-uniform filters

Uniform deblurred result   Our rotational deblurred result

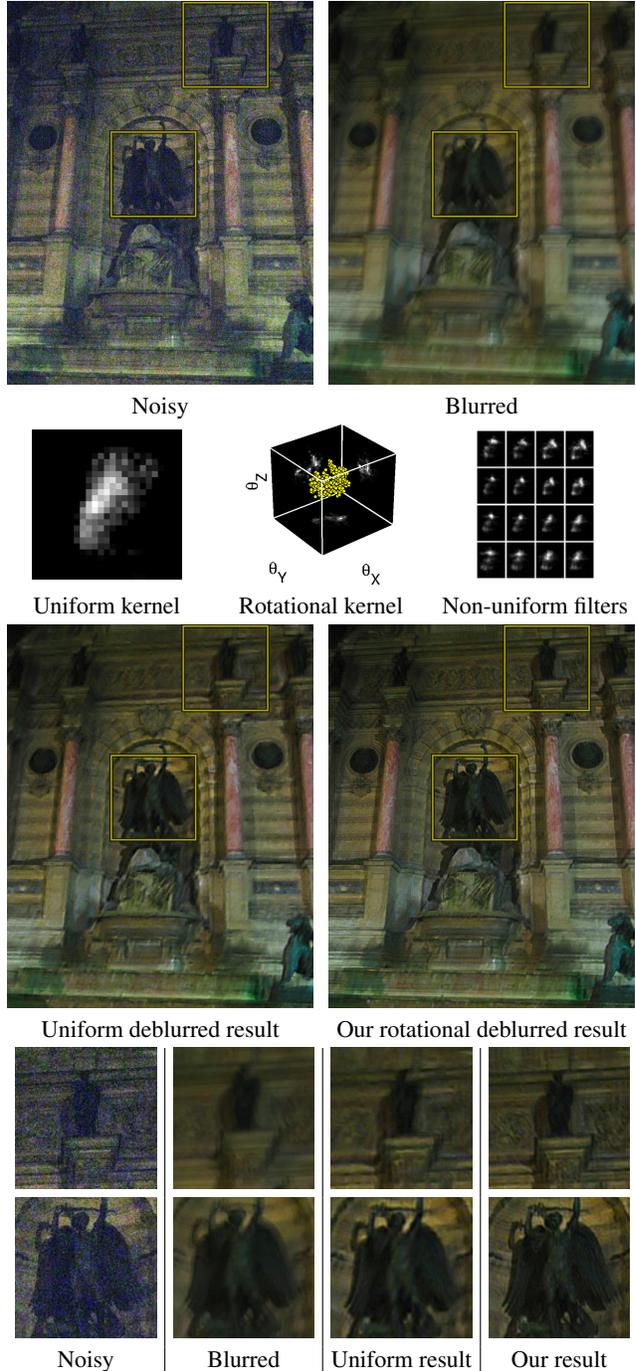Noisy   Blurred   Uniform result   Our result

Figure 7. **Deblurring real camera shake blur using a noisy / blurred image pair.** A noisy / blurred pair of images captured with a hand-held camera, with the estimated kernels, and deblurred images obtained using the modified Richardson-Lucy algorithm proposed by Yuan *et al*. [29]. Also shown for illustration are a selection of the local filters generated by the rotational kernel. As can be seen in the close-ups, our result contains more details and fewer artifacts from the deblurring than when using the uniform blur model.

where $\hat{\mathbf{w}}_s$ is non-zero, and constraining the non-zeros at the next scale to lie within this region. We first build Gaussian pyramids for the blurred image (and noisy image, if applicable), and at the finest 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.

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, and remove this mis-registration from the noisy image. 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. The model assumes rotation of the camera about its optical center during exposure, and is temporally-agnostic on 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. Also, since our model is valid over the whole image, it may be possible to estimate the sharp image and blur kernel simultaneously, as suggested by the result in Figure 4.

References

[1] http://www.di.ens.fr/willow/research/deblurring/.

[2] PTLens software. http://epaperpress.com/ptlens/.

[3] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2), 1997.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

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

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

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

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

[9] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32, 2004.

[10] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *SIGGRAPH*, 2006.

[11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. CUP, second edition, 2004.

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

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

[14] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astron. Journal*, 79(6), 1974.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11:19–60, 2010. http://www.di.ens.fr/willow/SPAMS/.

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

[17] J. G. Nagy and D. P. O'Leary. Restoring images degraded by spatially variant blur. *SIAM J. Sci. Comput.*, 19(4), 1998.

[18] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *SIGGRAPH*, 2003.

[19] R. C. Puetter, T. R. Gosnell, and A. Yahil. Digital image reconstruction: Deblurring and denoising. *Annu. Rev. Astron. Astrophys.*, 43, 2005.

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

[21] W. H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1), 1972.

[22] A. A. Sawchuk. Space-variant image restoration by coordinate transformations. *JOSA*, 64:138–144, 1974.

[23] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *SIGGRAPH*, 2008.

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

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

[26] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Royal Stat. Soc. B*, 58(1), 1996.

[27] R. Vio, J. Nagy, L. Tenorio, and W. Wamsteker. Multiple image deblurring with spatially variant PSFs. *Astronomy & Astrophysics*, 434, 2005.

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

[29] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. In *SIGGRAPH*, 2007.