Deblurring Shaken and Partially Saturated Images

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

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

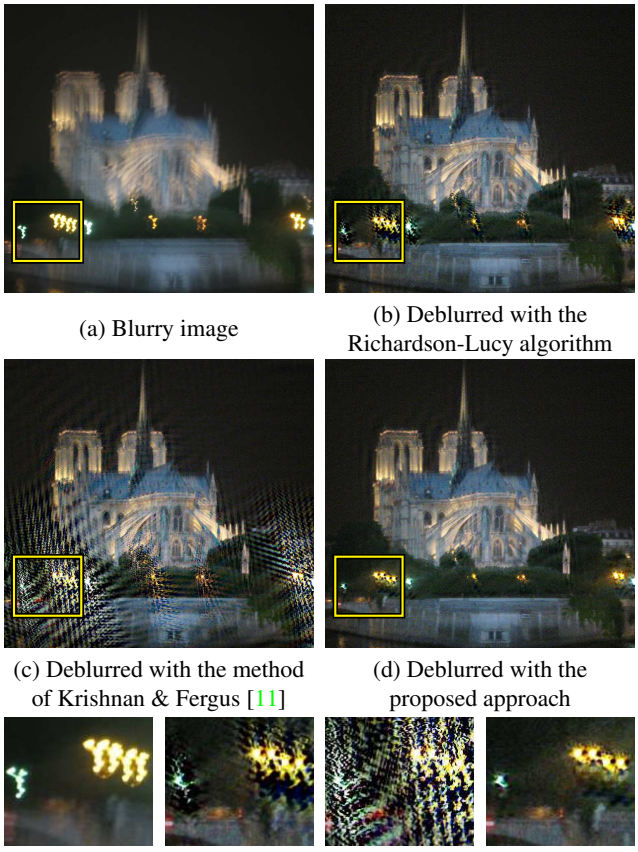
Abstract

We address the problem of deblurring images degraded by camera shake blur and *saturated or over-exposed pixels. Saturated pixels are a problem for existing non-blind deblurring algorithms because they violate the assumption that the image formation process is linear, and often cause significant artifacts in deblurred outputs. We propose a forward model that includes sensor saturation, and use it to derive a deblurring algorithm properly treating saturated pixels. By using this forward model and reasoning about the causes of artifacts in the deblurred results, we obtain significantly better results than existing deblurring algorithms. Further we propose an efficient approximation of the forward model leading to a significant speed-up.*

1. Introduction

The task of deblurring "shaken" images has received considerable attention recently [2, 4, 5, 6, 10, 18, 21, 22]. Significant progress has been made towards reliably estimating the point spread function (PSF) for a given blurry image, and towards inverting the blur process to recover a high-quality sharp image. However, one feature of "shaken" images that has received very little attention is the presence of saturated pixels. These are caused when the radiance of the scene exceeds the range of the camera's sensor, leaving bright highlights clipped at the maximum output value (*e.g.* 255 for an 8-bit image). To anyone who has attempted to take hand-held photographs at night, this effect should be familiar as the conspicuous bright streaks left by electric lights, such as in Figure 1 (a). These bright pixels, with their clipped values, violate the assumption made by many algorithms that the image formation process is linear, and as a result can cause obtrusive artifacts in the deblurred images. This can be seen in the deblurred images in Figure 1 (b) & (c).

The process of deblurring an image typically involves two steps. First, the PSF is estimated, which specifies how the image is blurred. This may be achieved using a "blind" deblurring algorithm, which estimates the PSF from the blurry image itself, or alternatively using additional hardware attached to the camera, or with the help of a sharp reference image of the same scene. Second, a "non-blind" deblurring algorithm is used to estimate the sharp image,



(a) Blurry image (b) Deblurred with the Richardson-Lucy algorithm

(c) Deblurred with the method of Krishnan & Fergus [11] (d) Deblurred with the proposed approach

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

Figure 1. **Deblurring in the presence of saturation.** Existing deblurring methods, such as those in (b) & (c), do not take account of saturated pixels. This leads to large and unsightly artifacts in the results, such as the "ringing" around the bright lights in the zoomed section. Using the proposed method (d), the ringing is greatly reduced and the quality of the deblurring improved.

[4]Willow Project, Laboratoire d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548

given the PSF. In this work we estimate the PSF in all cases using the algorithm of Cho & Lee [4], adapted to spatially-varying blur (Section 2). We can then consider the problem as non-blind deblurring (since the PSF is known) of images that contain saturated pixels. By handling such pixels explicitly, we are able to produce significantly better results than existing methods. Figure 1 (d) shows the output of the proposed algorithm, which contains far fewer artifacts than the two existing algorithms shown for comparison.

Our principal contribution is to propose a forward model for camera shake blur that includes sensor saturation (Section 3.2), and to use it to derive a modified version of the Richardson-Lucy algorithm properly treating the saturated pixels. We show that by explicitly modeling poorly-estimated pixels in the deblurred image, we are able to prevent "ringing" artifacts in the deblurred results (Section 3.3), as shown in Figure 1. We also propose an efficient piece-wise uniform approximation of spatially-varying blur in the forward model leading to a significant speed-up (Section 4.3) of both the PSF estimation and the non-blind deblurring steps.

Related Work. Saturation has not received wide attention in the literature, although it has been cited as the cause of artifacts in the deblurred outputs from deconvolution algorithms. For example, Fergus *et al*. [5], Cho & Lee [4] and Tai *et al*. [20] mention the fact that saturated pixels cause problems, sometimes showing their effect on the deblurred output, but leave the problem to be addressed in future work. An exception is Harmeling *et al*. [8], who address the issue in the setting of multi-frame blind deblurring by thresholding the blurry image to detect saturated pixels, and ignoring these in the deblurring process. When multiple blurry images of the same scene are available, these pixels can be safely discarded, since there will generally remain unsaturated pixels covering the same area in other images.

Single-image blind PSF estimation for camera shake has been widely studied [2, 4, 5, 12, 15, 18, 22], using variational and maximum a posteriori (MAP) algorithms. Levin *et al*. [14] review several approaches, as well as providing a ground-truth dataset for comparison on spatially-invariant blur. While most work has focused on spatially-invariant blur, several approaches have also been proposed for spatially-varying blur [6, 7, 10, 20, 21].

Many algorithms exist for non-blind deblurring, perhaps most famously the Richardson-Lucy algorithm [16, 17]. Recent work has revolved around the use of regularization, derived from natural image statistics [1, 10, 11, 13, 20], to suppress noise in the output while encouraging sharp edges to appear.

2. The Blur Process

In this work we consider the following model of the image formation process: that we are photographing a static scene, and there exists some sharp latent image $\mathbf{f}$ (represented as a vector) of this scene that we would like to record. However, while the shutter of the camera is open, the camera moves, capturing a sequence of different views of the scene as it does so. We will assume that each of these views can be modeled by applying some transformation $\mathbf{T}_k$ to the sharp image $\mathbf{f}$. The recorded (blurry) image $\mathbf{g}$ is then the sum of all these different views of the scene, each weighted by its duration:

$$\mathbf{g} = \sum_k w_k \mathbf{T}_k \mathbf{f}, \tag{1}$$

where the weight w_k is proportional to the time spent at view k, and $\sum_k w_k = 1$. The sharp image $\mathbf{f}$ and blurry image $\mathbf{g}$ are N-vectors, where N is the number of pixels, and each $\mathbf{T}_k$ is a sparse $N \times N$ matrix.

Often, the transformations $\mathbf{T}_k$ are assumed to be 2D translations of the image, which allows Eq. (1) to be computed using a 2D convolution. In this paper we use our recently-proposed model of spatially-varying camera shake blur [21], where the transformations $\mathbf{T}_k$ are homographies corresponding to rotations of the camera about its optical center. However, the non-blind deblurring algorithm proposed in this work is equally applicable to other models, spatially-variant or not.

In a non-blind deblurring setting, the weights w_k, which characterize the PSF, are assumed to be known, and Eq. (1) can be written as the matrix-vector product

$$\mathbf{g} = \mathbf{A}\mathbf{f} \tag{2}$$

where $\mathbf{A} = \sum_k w_k \mathbf{T}_k$. Given the PSF, non-blind deblurring algorithms typically maximize the likelihood of the observed blurry image $\mathbf{g}$ over all possible latent sharp images $\mathbf{f}$, or maximize the posterior probability of the latent image given some prior knowledge about its properties. One popular example is the Richardson-Lucy (RL) [16, 17] algorithm, which converges to the maximum likelihood estimate of the latent image under a Poisson noise model [19], using the following multiplicative update equation:

$$\mathbf{f}^{t+1} = \mathbf{f}^t \circ \mathbf{A}^\top \left(\frac{\mathbf{g}}{\mathbf{A}\mathbf{f}^t} \right), \tag{3}$$

where $\circ$ represents element-wise multiplication, the fraction represents element-wise division, and t represents the iteration number.

Unfortunately, the image produced by a digital camera does not generally follow the linear model in Eq. (1), and so naïvely applying a non-blind deblurring algorithm such as Richardson-Lucy may cause artifacts in the result, such

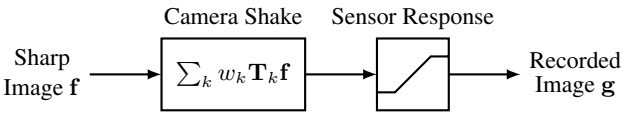


Figure 2. **Diagram of image formation process.** See text for explanation and definitions of terms.

as in Figure 1. The pixel values stored in an image file are not directly proportional to the scene radiance for two main reasons: (a) saturation in the sensor, and (b) the compression curve applied by the camera to the pixel values before writing the image to a file. To handle the latter of these, we either work directly with raw image files, which have not had any compression applied, or follow the standard approach of pre-processing the blurry image, applying a fixed curve which approximately inverts the camera's (typically unknown) compression curve. The curve is then re-applied to the deblurred image before outputting the result. This leaves saturation as the remaining source of non-linearities in the image formation model, as shown in Figure 2.

3. Explicitly Handling Saturated Pixels

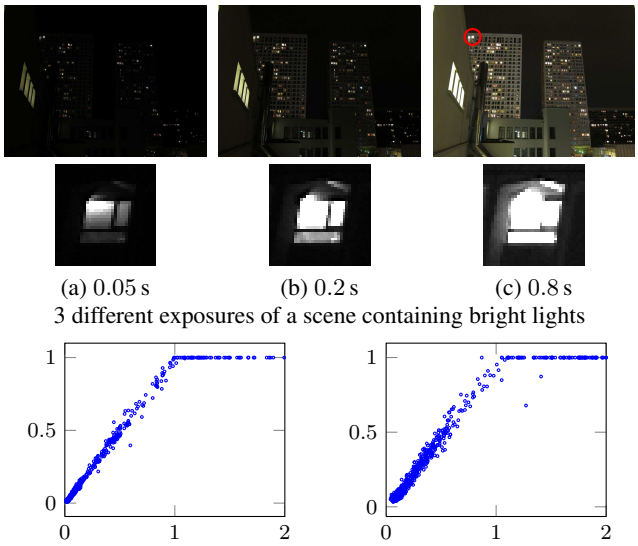
We model sensor saturation as follows: the sensor outputs pixel values which are proportional to the scene radiance, up to some limit, beyond which the pixel value is clipped at the maximum output level. This model is supported by the data in Figure 3, which shows the relationship between pixel intensities in three different exposures of a bright light source. The pixel values in the short exposure (with no saturation) and the longer exposures (with saturation) clearly exhibit this clipped linear relationship. As the length of the exposure increases, more pixels saturate.

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

3.1. Discarding Saturated Pixels

It is possible to estimate which blurry pixels are saturated by defining a threshold T, above which a blurry pixel is considered to be saturated, and therefore an outlier to the linear model. If we discard these pixels, the problem of deblurring with saturated pixels becomes deblurring with missing data. It is possible to re-derive the Richardson-Lucy algorithm to take account of missing data, by defining a binary mask of unsaturated (inlier) pixels $\mathbf{z}$, where each element $z_i = 1$ if $g_i < T$, and 0 otherwise. The new Richardson-Lucy update equation is then

$$\mathbf{f}^{t+1} = \mathbf{f}^t \circ \mathbf{A}^\top \left(\frac{\mathbf{g} \circ \mathbf{z}}{\mathbf{A}\mathbf{f}^t} + \mathbf{1} - \mathbf{z} \right), \qquad (4)$$



(a) 0.05 s (b) 0.2 s (c) 0.8 s
3 different exposures of a scene containing bright lights

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

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

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

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

The choice of threshold T can be problematic however; a low threshold may discard large numbers of inlying pixels from $\mathbf{g}$, causing some parts of $\mathbf{f}$ to become decoupled from the data. A high threshold, on the other hand, may treat some saturated pixels as inliers, causing artifacts in the deblurred result. Figure 4 shows the result of deblurring using Eq. (4) for different values of threshold T. As is visible in the figure, no particular threshold produces a result free of artifacts. At high values of T, the building is deblurred well, but artifacts appear around the lights. At the lowest value of T, the lights are deblurred reasonably well, but the face of the building is mistakenly discarded and thus remains blurry in the output.

Ideally, we would like to utilise all the data that we have available, whilst taking account of the fact that some pixels are more useful that others. We describe this approach in the following sections.



(a) $T = 0.7$ (b) $T = 0.5$ (c) $T = 0.3$ (d) $T = 0.1$

Figure 4. **Ignoring saturated pixels using a threshold.** A simple way to handle saturation is to threshold the blurry image at some level T, and discard the blurry pixels above this threshold. Shown are the results of running the Richardson-Lucy algorithm for different thresholds. As the threshold decreases, the artifacts around the bright lights at the bottom of the image are reduced compared to the standard RL result in Figure 1 (b). At the lowest threshold (d) the fewest artifacts appear, but parts of the church are also discarded, hence remain blurred.

3.2. A Forward Model for Saturation

Instead of attempting to segment the blurry image into saturated and unsaturated regions, we may instead modify our forward model to include the saturation process. This avoids making *a priori* decisions about which data are inliers or outliers, and allows us to use all the data in the blurry image. To this end, we introduce a response function $R(\cdot)$ into Eq. (2) so that the forward model becomes

$$\mathbf{g} = R\left(\mathbf{A}\mathbf{f}\right), \tag{5}$$

where the function R is applied element-wise. Re-deriving the Richardson-Lucy algorithm using this model leads to the new update equation:

$$\mathbf{f}^{t+1} = \mathbf{f}^{t} \circ \mathbf{A}^{\top}\left(\frac{\mathbf{g} \circ R'(\mathbf{A}\mathbf{f}^{t})}{R(\mathbf{A}\mathbf{f}^{t})} + \mathbf{1} - R'(\mathbf{A}\mathbf{f}^{t})\right), \tag{6}$$

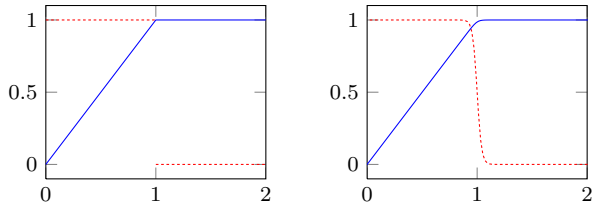
where R' is the derivative of R.

One choice for R would be simply to truncate the linear model in Eq. (1) at 1 (the maximum pixel value), using the function $R(x) = \min(x, 1)$. This choice is empirically justified, as can be seen in Figure 3. However, this function is non-differentiable at $x = 1$, *i.e.* $R'(1)$ is not defined. We thus use a smooth approximation [3], where

$$R(x) = x - \frac{1}{a}\log\Big(1 + \exp\big(a(x-1)\big)\Big). \tag{7}$$

The parameter a controls the smoothness of the approximation, and in all our experiments we set $a = 50$. Figure 5 shows the shape of R and R' compared to the simple truncated linear model.

Given the shape of R, Eq. (6) can easily be interpreted: in the linear portion $R(x) \simeq x$ and $R'(x) \simeq 1$, so that the term in parentheses is the same as for the standard RL algorithm, while in the saturated portion $R(x) \simeq 1$ and $R'(x) \simeq 0$, so that the term in parentheses is equal to unity and has no influence on $\mathbf{f}$. It is important to note that these two regimes are not detected from the blurry image using a threshold, but arise naturally from our current estimate of the latent image, and thus no explicit segmentation of the blurry image into unsaturated and saturated regions is necessary. We refer to the algorithm using this update rule as "saturated RL". Figure 6 demonstrates the advantage of this method over the standard RL algorithm on a synthetic 1D example.



(a) Ideal saturation function (b) Smooth, differentiable saturation function

Figure 5. **Modeling the saturated sensor response.** (a) Ideal clipped linear response function (solid blue line) and its derivative (dashed red line). The derivative is not defined at $x = 1$. (b) Smooth and differentiable approximation to (a) defined in Eq. (7). The derivative is also smooth and defined everywhere.

3.3. Preventing the Propagation of Errors

It is important to note that even using the correct forward model, we are not necessarily able to estimate every latent pixel in $\mathbf{f}$ accurately. In the blurring process, each pixel f_j in the latent image is blurred across multiple pixels in the blurry image $\mathbf{g}$. If some (or all) of these are saturated, we are left with an incomplete set of data concerning f_j, and our estimate of f_j is likely to be less accurate than if we had a full set of unsaturated observations available. This mis-estimation is one source of "ringing" artifacts in the deblurred output; an over-estimate at one pixel must be balanced by an under-estimate at a neighboring pixel, which must in turn be balanced by another over-estimate. In this way, an error at one pixel spreads outwards in waves across the image. In order to mitigate this effect, we propose a second modification to the Richardson-Lucy algorithm to prevent the propagation of these errors.

We first segment $\mathbf{f}$ into two disjoint regions: $\mathcal{S}$, which includes the bright pixels that we are unlikely to estimate accurately, and $\mathcal{U}$, which covers the rest of the image and which we can estimate accurately. We decompose the latent image correspondingly: $\mathbf{f} = \mathbf{f}_{\mathcal{U}} + \mathbf{f}_{\mathcal{S}}$. Our aim is then to prevent the propagation of errors from $\mathbf{f}_{\mathcal{S}}$ to $\mathbf{f}_{\mathcal{U}}$. To achieve this, we propose to estimate $\mathbf{f}_{\mathcal{U}}$ using only data which is not influenced by any pixels from $\mathcal{S}$. To this end, we first define the region (denoted by $\mathcal{V}$) of the blurry image which is in-

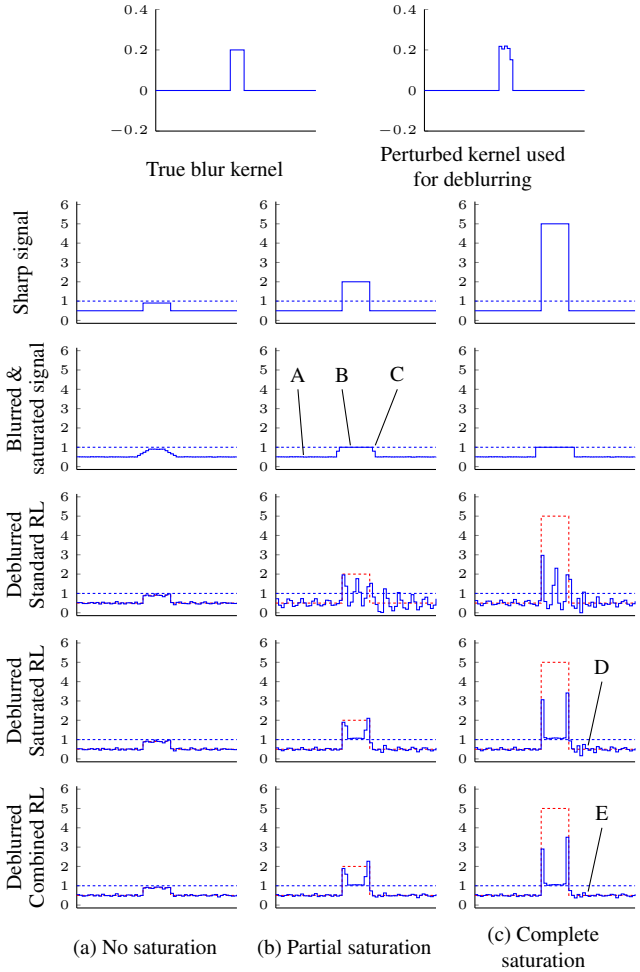


Figure 6. **Synthetic example of blur and saturation.** Each column shows a sharp "top-hat" signal, blurred using the box filter shown at the top left. Gaussian noise is added and the blurred signal is clipped, to model saturation. The kernel is also degraded with noise and one large error to produce a "perturbed" kernel which is used for deconvolution, to simulate errors in the kernel estimation step. The last three rows show the deblurred outputs for three algorithms discussed in Section 3. (a) With no saturation, all three algorithms produce similar results. (b) When some of the blurred signal is saturated (region B), the standard RL algorithm produces an output with large ringing artifacts. Although region A is not itself saturated, the ringing propagates outwards from B & C across the whole signal. The "saturated RL" algorithm reduces the ringing and correctly estimates the height of the top-hat at its edges (region C), where there are some unsaturated observations available. In region B all information about the height of the sharp signal is lost, and the output takes a sensible value close to 1. (c) When the blurred top-hat is completely saturated, it is no longer possible to estimate its true height anywhere. The saturated RL result accurately locates the top-hat, but contains ringing. The proposed method (combined RL) mitigates this by preventing the propagation of errors to the non-saturated region (compare D to E).

dependent of $\mathbf{f}_\mathcal{S}$, by eroding $\mathcal{U}$ using the non-zero elements of the PSF: $\mathcal{V} = \bigcap_{k:w_k>0} \mathcal{U}_{\mathbf{T}_k}$, where $\mathcal{U}_{\mathbf{T}_k}$ denotes the set $\mathcal{U}$ transformed by $\mathbf{T}_k$. By taking the intersection of all the transformed versions of $\mathcal{U}$, we ensure that $\mathcal{V}$ contains only those blurry pixels that are completely independent of $\mathcal{S}$. We can then estimate $\mathbf{f}_\mathcal{U}$ using only the data in $\mathcal{V}$, by defining the binary mask $\mathbf{v}$ which corresponds to $\mathcal{V}$ and adapting the update equation from Eq. (4) for Richardson-Lucy with missing data:

$$\mathbf{f}_\mathcal{U}^{t+1} = \mathbf{f}_\mathcal{U}^t \circ \mathbf{A}^\top \left(\frac{\mathbf{g} \circ R'(\mathbf{A}\mathbf{f}^t) \circ \mathbf{v}}{R(\mathbf{A}\mathbf{f}^t)} + \mathbf{1} - R'(\mathbf{A}\mathbf{f}^t) \circ \mathbf{v} \right). \tag{8}$$

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

$$\mathbf{f}_\mathcal{S}^{t+1} = \mathbf{f}_\mathcal{S}^t \circ \mathbf{A}^\top \left(\frac{\mathbf{g} \circ R'(\mathbf{A}\mathbf{f}^t)}{R(\mathbf{A}\mathbf{f}^t)} + \mathbf{1} - R'(\mathbf{A}\mathbf{f}^t) \right). \tag{9}$$

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

$$\mathcal{U} = \{ j | f_j^t \leq \phi \}. \tag{10}$$

We decompose $\mathbf{f}$ according to

$$\mathbf{f}_\mathcal{U}^t = \mathbf{u} \circ \mathbf{f}^t, \qquad \mathbf{f}_\mathcal{S}^t = \mathbf{f}^t - \mathbf{f}_\mathcal{U}^t, \tag{11}$$

where $\mathbf{u}$ is the binary mask corresponding to $\mathcal{U}$. We then compute $\mathcal{V}$, update $\mathbf{f}_\mathcal{U}$ and $\mathbf{f}_\mathcal{S}$ using Eqs. (8) and (9), and recombine them to form our new estimate of the latent image $\mathbf{f}^{t+1} = \mathbf{f}_\mathcal{U}^{t+1} + \mathbf{f}_\mathcal{S}^{t+1}$. We refer to this algorithm as "combined RL", and Figure 6 shows the results of applying it to a synthetic 1D example, demonstrating the advantage over the standard RL and "saturated RL" algorithms.

Although this combined RL algorithm involves the use of a threshold to segment the image, its effect is less dramatic than in Section 3.1. In this case, the threshold only determines whether a given pixel f_j should be updated using all the available data, or a subset of it. This is in contrast to Section 3.1, where parts of the data are discarded and never used again. Since our aim is to ensure that no large errors are introduced in $\mathbf{f}_\mathcal{U}$, we set the threshold low enough that most potentially-bright pixels are assigned to $\mathcal{S}$. Empirically, we choose $\phi = 0.9$ for the results in this paper.

4. Implementation

In this section we describe some of the implementation details of the proposed algorithm, the PSF estimation for the results shown, and an efficient approximation for the forward model that leads to a significant speed-up.

4.1. PSF Estimation

For all the results shown in this work, we estimate the PSFs using the blind deblurring algorithm proposed by Cho & Lee [4], adapted to our spatially-varying blur model [21]. Due to space considerations, we refer the reader to [4] for details of the algorithm. The only modification required to handle saturated images using this algorithm is to discard potentially saturated regions of the blurry image using a threshold. Since in this case the aim is only to estimate the PSF (and not a complete deblurred image), we can safely discard all of these pixels, since the number of saturated pixels in an image is typically small compared to the total number of pixels. There will typically remain sufficient unsaturated pixels from which to estimate the PSF.

4.2. Segmenting the Latent Image

When segmenting the current estimate of the latent image in the combined RL algorithm, we take additional steps to ensure that we make a conservative estimate of which pixels can be estimated accurately. First, after thresholding the latent image in Eq. (10), we perform a binary erosion on $\mathcal{U}$, using a disk with radius 3 pixels. This ensures that all poorly-estimated pixels are correctly assigned to $\mathcal{S}$ (perhaps at the expense of wrongly including some well-estimated pixels too). Fewer artifacts arise from wrongly assigning a well-estimated pixel into $\mathcal{S}$ than the other way around. Second, in order to avoid introducing visible boundaries between the two regions, we blur the mask $\mathbf{u}$ slightly using a Gaussian filter with standard deviation 3 pixels to produce a smoother set of weights when extracting $\mathbf{f}^t_{\mathcal{U}}$ and $\mathbf{f}^t_{\mathcal{S}}$ from the current latent image $\mathbf{f}^t$ in Eq. (11).

4.3. Efficient Approximation for Forward Model

Due to the additional computational expense incurred by using a spatially-varying blur model instead of a spatially-invariant one, both the blind and non-blind deblurring steps can be very time consuming. The number of homographies $\mathbf{T}_k$ in the forward model in Eq. (1) can be large, even for a moderately-sized blur: for a blur 30 pixels in size, up to $30^3 = 27,000$ homographies may need to be computed. To reduce the running time of both the PSF estimation and the non-blind deblurring, we extend the locally-uniform "Efficient Filter Flow" approximation proposed by Hirsch *et al*. [9] to handle blur models of the form in Eq. (1).

Locally-uniform approximation. The idea is that for a smoothly varying blur, such as camera shake blur, nearby pixels have very similar point spread functions. Thus it is reasonable to approximate the blur as being locally-uniform. In the approximation proposed by Hirsch *et al*., the sharp image $\mathbf{f}$ is covered with a coarse grid of p overlapping patches, each of which is modeled as having a spatially-invariant blur. The overlap between patches ensures that the blur varies smoothly across the image, rather than changing abruptly at the boundary between two patches. The fact that each patch has a spatially-invariant blur allows the forward model to be computed using p small convolutions. Hirsch *et al*. [9] assign each patch r a spatially-invariant blur filter $\mathbf{a}^{(r)}$, and the forward model is approximated by:

$$\mathbf{g} = \sum_{r=0}^{p-1} \mathbf{C}_r^\top \mathbf{F}^\mathsf{H} \operatorname{Diag}(\mathbf{F}\mathbf{a}^{(r)}) \mathbf{F} \operatorname{Diag}(\mathbf{m}) \mathbf{C}_r \mathbf{f}, \qquad (12)$$

where the matrix $\mathbf{F}$ takes the discrete Fourier transform, $\mathbf{F}^\mathsf{H}$ takes the inverse Fourier transform (both performed using the FFT), and $\mathbf{C}_r$ is a matrix that crops out the r^{th} patch from the image $\mathbf{f}$ (and thus $\mathbf{C}_r^\top$ reinserts it at its correct location). The vector $\mathbf{m}$ is a windowing function, *e.g*. the Bartlett-Hann window, which produces the smooth transition between neighboring patches.

Applying the approximation to the forward model. In their original work, Hirsch *et al*. [9] store a separate filter $\mathbf{a}^{(r)}$ for each patch r. However, given the blur model in Eq. (1), which is parameterized by a single set of weights $\mathbf{w}$, we can write each $\mathbf{a}^{(r)}$ in terms of $\mathbf{w}$. For each patch r, we choose $\mathbf{a}^{(r)}$ to be the point spread function for the central pixel i_r, which is given by the i_r^{th} row of $\mathbf{A}$. Since $\mathbf{A}$ is linear in $\mathbf{w}$, we can construct a matrix $\mathbf{J}_r$ such that $\mathbf{a}^{(r)} = \mathbf{C}_r \mathbf{J}_r \mathbf{w}$. The elements of each $\mathbf{J}_r$ are simply a rearrangement of the elements of the matrices $\mathbf{T}_k$: element (j,k) of $\mathbf{J}_r$ is equal to element (i_r, j) of $\mathbf{T}_k$. Figure 7 shows how the quality of the approximation varies with the number of patches being used. In all our experiments, we use a grid of 6×8 patches.

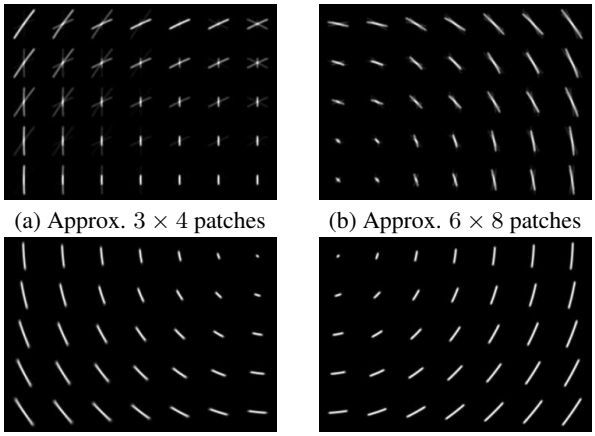
Having written each filter $\mathbf{a}^{(r)}$ in terms of $\mathbf{w}$, we can then substitute this into Eq. (12), and obtain the following approximation for the forward model of Eq. (1):

$$\mathbf{g} = \sum_{r=0}^{p-1} \mathbf{C}_r^\top \mathbf{F}^\mathsf{H} \operatorname{Diag}(\mathbf{F}\mathbf{C}_r\mathbf{J}_r\mathbf{w}) \mathbf{F} \operatorname{Diag}(\mathbf{m}) \mathbf{C}_r \mathbf{f}. \qquad (13)$$

This allows the forward model to be computed quickly using only a handful of frequency-domain convolutions. Furthermore, the derivatives of $\sum_k w_k \mathbf{T}_k \mathbf{f}$ with respect to $\mathbf{f}$ and $\mathbf{w}$ can also be computed using a small number of frequency-domain convolutions and correlations. These three operations are the computational bottleneck in both the blind PSF estimation algorithm of Cho & Lee [4], and in the Richardson-Lucy algorithm.

5. Results

Figures 1 and 8 show results of non-blind deblurring using the proposed "combined RL" algorithm described in Section 3.3 on real hand-held photographs. The PSFs for



(a) Approx. 3×4 patches (b) Approx. 6×8 patches

(c) Approx. 12×16 patches (d) Exact

Figure 7. **Approximating spatially-varying blur by combining uniformly-blurred, overlapping patches.** Using the model described in Section 4.3, we can efficiently compute approximations to the spatially-varying blur model in Eq. (1). With a small number of patches (a), the PSF at each pixel is visibly the sum of different blurs from overlapping patches. As more patches are used (b–c), the approximation becomes increasingly close to the exact model (d) – at 12×16 patches it is almost indistinguishable.

these images were estimated from the blurry images themselves using the algorithm of Cho & Lee [4] (as described in Section 4.1). Note that the standard Richardson-Lucy algorithm and the approach of Krishnan & Fergus [11] produce large amounts of ringing around the saturated regions, while the proposed algorithm avoids this with no loss of quality elsewhere. In all results in this paper we performed 50 iterations of the Richardson-Lucy algorithm.

As a result of the approximation described in Section 4.3, we are able to obtain a speed-up in both the blind PSF estimation and the non-blind deblurring steps over the exact model, with no visible reduction in quality. For a typical 1024×768 image, the exact model in Eq. (1) takes approximately 20 seconds to compute in our MATLAB implementation and 5 seconds to compute in our C implementation, compared to 2 seconds for our MATLAB implementation of the approximation, on an Intel Xeon 2.93GHz CPU.

6. Conclusion

In this work we have developed an approach for deblurring images blurred by camera shake and suffering from saturation. The proposed algorithm is able to effectively deblur saturated images without introducing ringing or sacrificing detail, and is applicable to any blur model, whether spatially-varying or not. We have also demonstrated an efficient approximation for computing spatially-varying blur, applicable to any model of blur with the form of Eq. (1).

Acknowledgments. This work was partly supported by the MSR-INRIA laboratory, the EIT ICT labs (activity 10863) and ONR MURI N00014-07-1-0182.

References

[1] M. Afonso, J. Bioucas-Dias, and M. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Processing*, 19(9), 2010.

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

[3] C. Chen and O. L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2), 1996.

[4] S. Cho and S. Lee. Fast motion deblurring. In *Proc. SIGGRAPH Asia*, 2009.

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

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

[7] S. Harmeling, M. Hirsch, and B. Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *NIPS*, 2010.

[8] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf. Multi-frame blind deconvolution, super-resolution, and saturation correction via incremental EM. In *Proc. ICIP*, 2010.

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

[10] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *Proc. SIGGRAPH*, 2010.

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

[12] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proc. CVPR*, 2011.

[13] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *Proc. SIGGRAPH*, 2007.

[14] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. Technical report, MIT, 2009.

[15] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proc. CVPR*, 2011.

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

[17] W. H. Richardson. Bayesian-based iterative method of image restoration. *J. of the Optical Society of America*, 62(1), 1972.

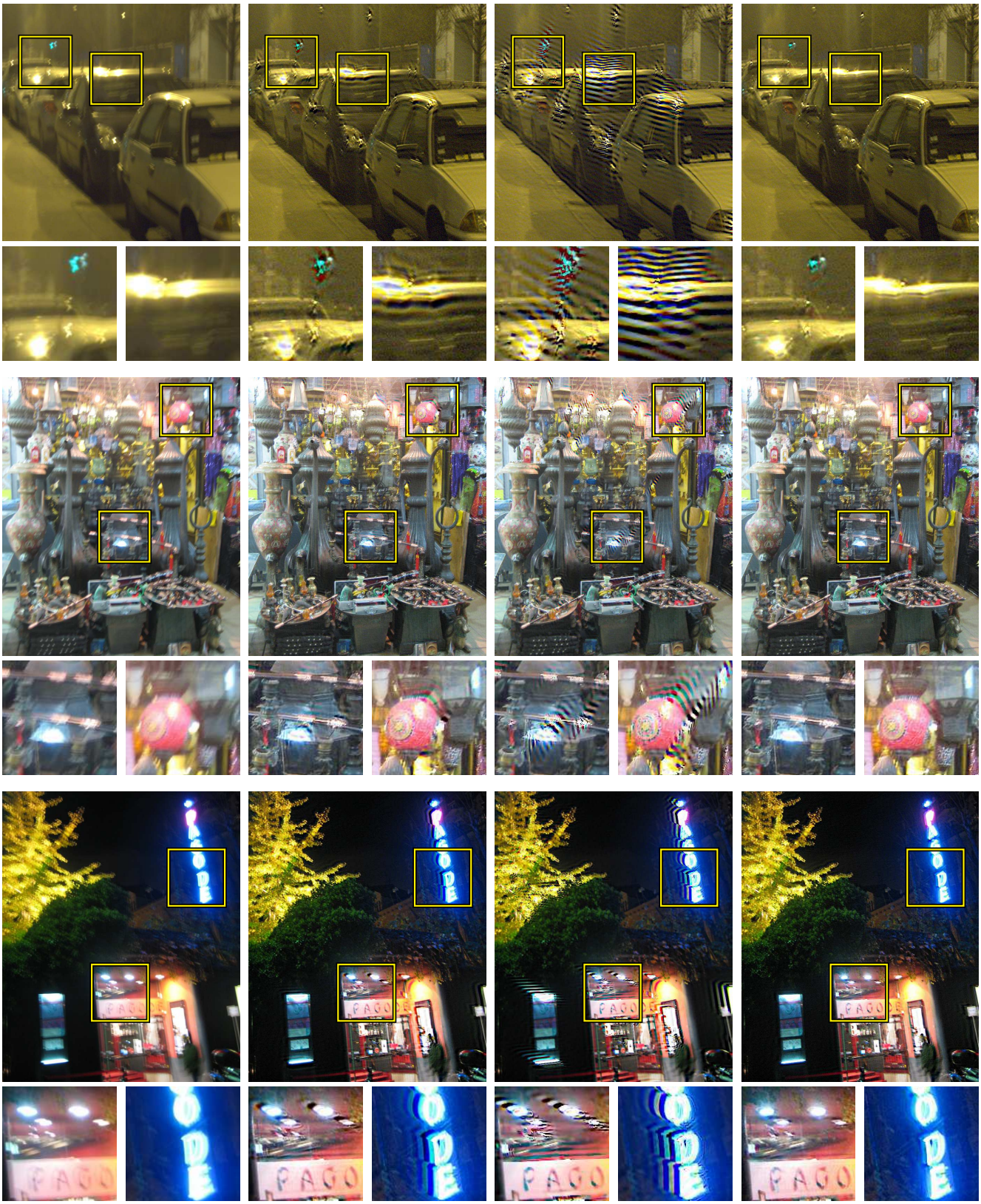
[18] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *Proc. SIGGRAPH*, 2008.

[19] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Transactions on Medical Imaging*, 1(2), 1982.

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

[21] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proc. CVPR*, 2010.

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



(a) Blurry image

(b) Deblurred with Richardson-Lucy

(c) Deblurred with algorithm of Krishnan & Fergus [11]

(d) Deblurred with proposed method

Figure 8. **Deblurring saturated images.** Note that the ringing around saturated regions, visible in columns (b) and (c) is removed by our method (d), without causing any loss in visual quality elsewhere.