Non-Uniform Blind Deblurring by Reblurring

Yuval Bahat * Netalee Efrat * Michal Irani
Dept. of Computer Science and Applied Math
The Weizmann Institute of Science, ISRAEL
http://www.wisdom.weizmann.ac.il/~vision/DeblurringByReblurring

Abstract

We present an approach for blind image deblurring, which handles non-uniform blurs. Our algorithm has two main components: (i) A new method for recovering the unknown blur-field *directly from the blurry image, and (ii) A method for deblurring the image given the recovered non-uniform blur-field. Our blur-field estimation is based on analyzing the spectral content of blurry image patches by* Re-blurring *them. Being unrestricted by any training data, it can handle a large variety of blur sizes, yielding superior blur-field estimation results compared to training-based deep-learning methods. Our non-uniform deblurring algorithm is based on the internal* image-specific *patch-recurrence prior. It attempts to recover a sharp image which, on one hand – results in the blurry image under our estimated blur-field, and on the other hand – maximizes the internal recurrence of patches within and across scales of the recovered sharp image. The combination of these two components gives rise to a blind-deblurring algorithm, which exceeds the performance of state-of-the-art CNN-based blind-deblurring by a significant margin, without the need for any training data.*

1. Introduction

Images taken in our daily lives are often corrupted by blur. This blur may be caused by camera motion, defocus, movement of rigid or non-rigid objects in the scene, and more. Recovering the sharp image solely from the blurry image without any knowledge of the blur function, has been the focus of many studies in the past years and is termed *"blind" image deblurring*. Some methods assume a uniform image blur (e.g., [2, 22, 20, 15, 18]), while others assume a varying blur-field [7, 16, 21, 23, 3, 9, 10, 19]

When the blur is uniform in the image, the relation between the sharp and blurry images is a simple convolution:

$$I_b = k * I_s + n \tag{1}$$

where I_s is the sharp image, I_b is the blurry image, k is the spatially uniform blur kernel and n is noise. When the blur kernel k is known, recovering I_s reduces to a deconvolution with k (an ill-posed problem on its own, since k has many zeros in the frequency domain). The problem becomes much harder in the blind case, where both the sharp image I_s and the blur kernel k are unknown [2, 22, 20, 15, 18], since many different pairs of I_s and k can give rise to the same input blurry image I_b. The problem becomes even more ill-posed in the case where the blur is no longer uniform. This case can no longer be modeled as a convolution. Nevertheless, it can still be written as the linear transformation

$$I_b = K I_s + n \tag{2}$$

where the matrix K denotes a very large blur matrix, in which each row corresponds to a kernel that models the blur at a specific pixel. It is a matrix similar to a convolution matrix with the exception that each row contains a different kernel (it is no longer Toeplitz).

Non-uniform blind deblurring has attracted much attention in recent years. Some methods [7, 16, 21, 23] assume that the blur is caused by a general 3D camera shake; these do not handle dynamic scenes with blur discontinuities. Other methods [3, 1, 8, 13] detect boundaries and blur-fields of moving objects, but assume that the blur within each moving object is either uniform [1, 8, 13] or parametric [3]. The method of [10] handles general blurs, but assumes the input is a video sequence. The methods of [9, 19] allow a non-uniform non-parametric blur-field, with different blur per pixel, however assume that the blur at each pixel is linear. Our method belongs to that category. In [19] CNNs have been recruited to the task, and to the best of our knowledge, this method provides the current state-of-the-art results in non-uniform bind deblurring. Nonetheless, being a learning-based approach, CNNs are restricted by the type of data they are trained on (e.g., restricted blur sizes). In contrast, our method (which is not learning-based) is not restricted to specific motion sizes or image types. In particular, it can handle very large blurs (e.g., see Figs. 1,2).

In this paper we propose an algorithm for *non-uniform* blind image deblurring. Our algorithm has two major components: (i) A new method for recovering the *unknown blur-field* directly from the blurry image, and (ii) A method for deblurring the image given the recovered non-uniform

*equal contributors

Funded in part by the Israel Science Foundation (Grant 931/14)

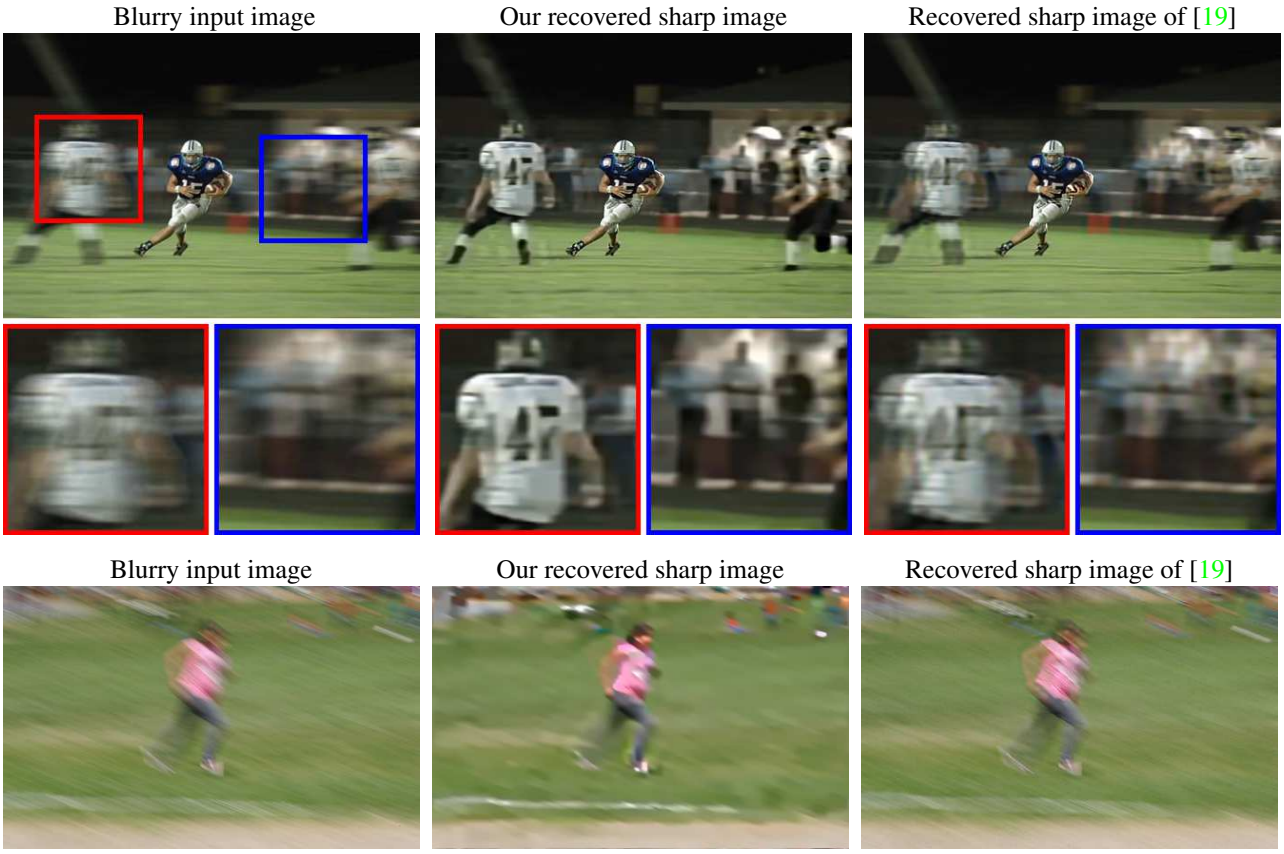

Figure 1: ***Blind-deblurring with very large real blur.*** *Note also the significant differences between us and [19] in the running woman, as well as the recovered background (e.g., the children and white ball in the top-right corner). (Please zoom-in on screen to see fine details.)*

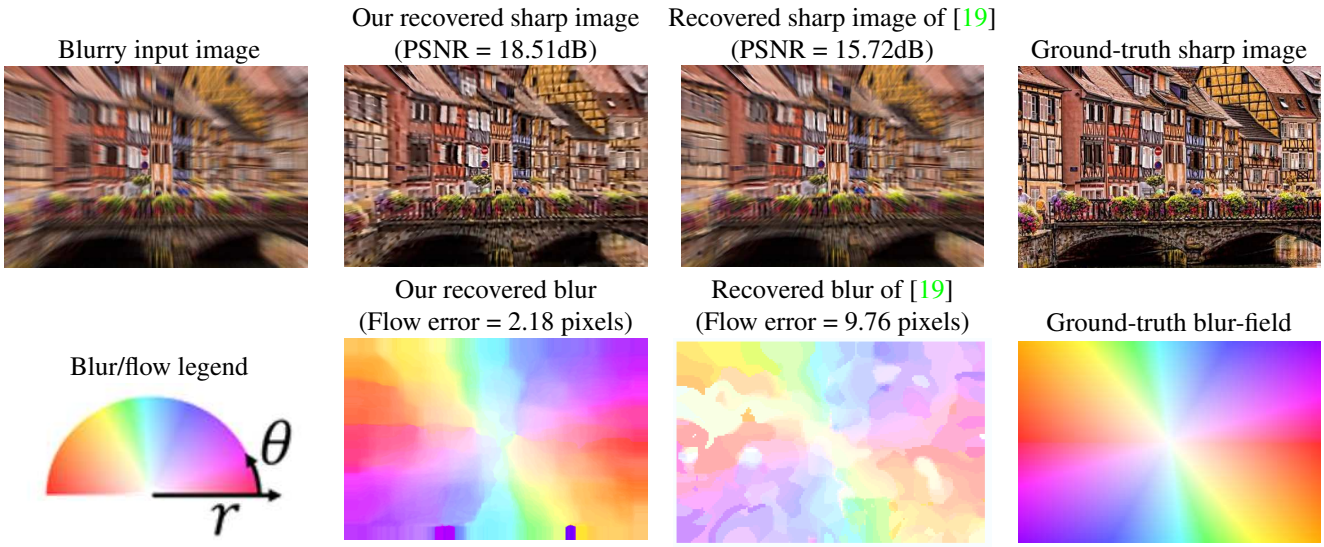

Figure 2: ***Blind-deblurring with very large synthetic blur.*** Top: *blurry and sharp Images.* Bottom: *Recovered blur-fields (chroma denotes direction, intensity denotes magnitude). Please zoom-in on screen to see fine details.*

blur-field. The combination of these two gives rise to a blind-deblurring algorithm, which exceeds the performance of state-of-the-art CNN-based blind-deblurring by a significant margin, without the need for any training data.

Our first major component – the *recovery of the unknown flow-field* (Sec. 3) – is based on spectral analysis of blurry

image patches, and is quite simple: We assume that the non-uniform blur-field is *locally* linear (a 1D blur kernel), but may vary across pixels and may have spatial discontinuities (i.e., has the standard flow-field assumptions, except that it is *undirected*). Have we had the sharp image I_s, we could have simply blurred (convolved) the entire image with various 1D blur kernels $b(r,\theta)$ of different lengths r and orientations θ, and then choose for each pixel (x,y) the blur $b(r,\theta)$ for which its surrounding patch in $I_s * b(r,\theta)$ is closest to the one in the blurry input I_b (in terms of MSE). However, we do not have the sharp image. We show (Sec. 3) that the same effect can be achieved by applying the blur kernels *directly on the blurry input image* I_b. We refer to this approach as the *Re-blurring method*. Unlike the global spectral analysis of [6], our analysis applies to local image patches and to non-uniform blur.

Our second major component – the *non-uniform deblurring algorithm* (Sec. 4) – is based on the internal patch-recurrence property, which forms an *image-specific prior*: small image patches (e.g. 5×5, 7×7) tend to repeat 'as is' many times inside the same scale, as well as across different scales, of a single natural image. This recurrence property was shown to be true for almost any small patch in almost any natural image [5, 24]. While this recurrence is very strong in sharp images, it was shown to *diminish significantly in blurry images* [15] (see Fig. 3). It hence forms a very strong prior on the unknown sharp image I_s. This prior was used by [15] in the context of *uniform blurs* (the case of Eq. 1). We extend its applicability to *non-uniform blurs*. Our deblurring component thus attempts to recover a sharp image I_s which, on one hand – results in the blurry image I_b under our estimated non-uniform blur-field, and on the other hand – maximizes the internal recurrence of patches both within and across scales of the recovered sharp I_s.

Finally, these two components are combined into an iterative *coarse-to-fine blind-deblurring algorithm* (Sec. 5), which alternates between recovering the sharp image and refining the computed blur-field. Our experimental results (Sec. 6) show that our algorithm exceeds the performance of state-of-the-art CNN-based blind-deblurring [19], as well as other methods, especially in the presence of very large blurs (e.g., Figs. 1,2). This is shown both qualitatively as well as quantitatively (Table 1).

2. Problem Formulation and Approach

Our goal is to "blindly" remove non-uniform blur from still images. To solve this ill-posed problem, we define the following objective function:

$$E(\hat{I}_s,\hat{K}) = E_{data}(\hat{I}_s,\hat{K}) + E_{image}(\hat{I}_s) + E_{blur}(\hat{K}) \quad (3)$$

where E_{data}, the data fidelity term, enforces the forward model (Eq. 2), meaning that the recovered sharp image $\hat{I}_s$, when blurred, will be similar to the blurry input image I_b:

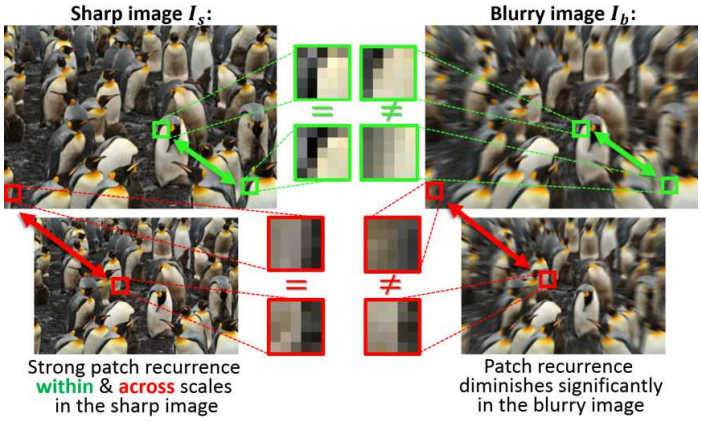

Figure 3: ***Patch recurrence diminishes under blur.*** *Patches which are similar in the sharp image (either within or across scales), are no longer similar in the blurry image.*

$$E_{data}(\hat{I}_s,\hat{K}) = \|\hat{K}\hat{I}_s - I_b\|^2 \quad (4)$$

As in [9, 19], we restrict our blur matrix $K = K(\mathbf{u})$ to *blur-fields* $\mathbf{u}(\mathbf{x},\mathbf{y}) = (u(x,y), v(x,y))^T$. Namely, while the blur may be non-rigid and different at every pixel, we do assume that each pixel undergoes a *locally* linear blur. We later show (Sec. 6) that this is not a very restrictive assumption. Note, however, that unlike standard optical flow, the blur flow in this case is *undirected* (since (u,v) and $(-u,-v)$ induce the same image blur). A row in K which corresponds to the blur of the i^{th} pixel in I_b, is a discrete blur kernel $h_i(x,y)$ induced by the flow vector (u_i,v_i) at that pixel:

$$h_i(x,y) = \begin{cases} \frac{1}{\|\mathbf{u_i}\|}\delta(v_i x - u_i y), & \text{if } x \le \frac{|u_i|}{2}, y \le \frac{|v_i|}{2} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

The additional terms $E_{image}(\hat{I}_s)$ and $E_{blur}(\hat{K})$ in Eq. 3 are prior terms on the unknown sharp image $\hat{I}_s$ and the unknown blur matrix $\hat{K}$, respectively. The prior term on the latent sharp image $\hat{I}_s$ is based on a combination of *gradient sparsity prior* [14, 12] and the *internal patch recurrence prior* [5, 24]. While the patch recurrence property is very strong in sharp images, it *diminishes significantly in blurry images* [15] (see Fig. 3). It hence forms a strong *image-specific prior* on the sharp image $\hat{I}_s$ (see Sec. 4). The second regularization term, E_{blur}, enforces smoothness of the blur-field underlying the blur matrix $\hat{K}$.

The objective in Eq. 3 is not convex (due to the nature of the patch-recurrence prior, as well as the matrix K), and has no closed-form solution. Hence, we minimize it using an alternating iterative minimization procedure. Our initial estimate of the non-uniform flow-field $\hat{\mathbf{u}}$ is computed *directly* from the blurry image I_b (see Sec. 3). This constitutes our initial blur matrix $\hat{K}$. The algorithm then proceeds to alternate between the following two steps: (i) Fix $\hat{K}$ and solve for a sharper image $\hat{I}_s$ that maximizes the image prior

(Sec. 4), and (ii) Fix $\hat{I}_s$ and refine the motion flow estimate $\hat{\mathbf{u}}$ that constitutes $\hat{K}$ (Sec. 3). This iterative alternating optimization is performed coarse-to-fine, and is summarized in Sec. 5.

3. Motion Blur Estimation by Reblurring

In this section we elaborate on estimating the underlying blur field of the blurry input image I_b. Note that unlike standard optical flow, the blur field in this case is undirected (since (u, v) and $(-u, -v)$ induce the same blur). Moreover, regular flow-field estimation is performed between two images, while here we have a *single* (blurry) image I_b.

Our approach is based on spectral analysis of image patches, and is conceptually simple: Have we had the sharp image I_s, we could have simply blurred (convolved) the entire image with various 1D blur kernels $b(r, \theta)$ of different lengths r and orientations θ, and then choose for each pixel (x, y) the blur $b(r, \theta)$ for which its surrounding patch in $I_s * b(r, \theta)$ is closest to the one in the input blurry image I_b (e.g., in terms of MSE). However, we do not have the sharp image. We next show that the same effect can be achieved by applying the blur kernels *directly on the blurry input image* I_b. We refer to this approach as the *Re-blurring method*.

Ideal uniform blur: For the sake of simplicity, let us first assume that: (i) the entire (unknown) sharp image I_s was blurred by a single uniform 1D blur kernel $b(r^*, \theta^*)$, and (ii) that it is an *ideal* Low Pass Filter (ideal LPF), namely, a rotated 1D sinc function in the spatial domain with orientation θ^*. This corresponds to multiplication by a rect function in the frequency domain in the same orientation. We will first prove our method works for this simple case, and later relax these two assumptions.

Claim 1 *If $b(r^*, \theta^*)$ is an ideal LPF that caused the blur in I_b (i.e., $I_b = I_s * b$), then Re-blurring the image I_b with $b(r^*, \theta^*)$ will not change I_b. Namely: $I_b = I_b * b(r^*, \theta^*)$.*

Proof: $I_b * b(r^*, \theta^*) = I_s * b(r^*, \theta^*) * b(r^*, \theta^*)$. Shifting to the Fourier domain, this becomes:

$$\mathcal{I}_b \cdot B(r^*, \theta^*) = \mathcal{I}_s \cdot B^2(r^*, \theta^*) = \mathcal{I}_s \cdot B(r^*, \theta^*) = \mathcal{I}_b \quad (6)$$

where $\mathcal{I}_s, \mathcal{I}_b, B(r^*, \theta^*)$ denote the Fourier transforms of $I_s, I_b, b(r^*, \theta^*)$, respectively. Eq. 6 is true since $b(r^*, \theta^*)$ is an ideal LPF, $B(r^*, \theta^*)$ is a *rect* in the direction θ^*, hence $B^2(r^*, \theta^*) = B(r^*, \theta^*)$. Thus, in the spatial domain:

$$I_b * b(r^*, \theta^*) = I_b \quad (7)$$

■

Note, however, that Eq. 7 is satisfied also by *smaller* ideal blurs $b(r, \theta^*)$, $r \leq r^*$, with the *same orientation* θ^* (since $B(r^*, \theta^*) \cdot B(r, \theta^*) = B^2(r^*, \theta^*)$ for $r \leq r^*$). This is not true for blurs in other orientations $\theta \neq \theta^*$. In other words, $b(r^*, \theta^*)$ is the *largest* blur that satisfies Eq. 7. Nonetheless, if we now add noise n to the blurry input I_b and then Re-blur it, $b(r^*, \theta^*)$ will now become the kernel which provides *minimal MSE* $\|(I_b+n)*b(r,\theta)-I_b\|$ among all blur kernels: $\|(I_b+n)*b(r,\theta)-I_b\| = \|Ib*b(r,\theta)+n*b(r,\theta)-I_b\|$. According to Claim 1, for $\theta = \theta^*$ and $r \leq r^*$, this error reduces to $\|n * b(r, \theta^*)\|$. The latter term is minimized by the largest blur among all those with orientation θ^*, namely $r = r^*$.

To recap, in the case of uniform global ideal blur, the way to recover the unknown blur kernel $b(r^*, \theta^*)$ underlying the blurry image I_b, is to add noise n to I_b, Re-blur (convolve) it with a variety of 1D kernels $b(r, \theta)$, and choose the one that minimizes the MSE w.r.t. I_b.

Generalizing to non-uniform blurs: We next show that the same procedure generalizes to non-uniform blur, by applying it to smaller image patches (windows) rather than the entire image. We assume a *local* 1D uniform blur within each local patch in I_b. Let P be the patch of size $s \times s$ centered at pixel $\{x_0, y_0\}$: $P[x_0, y_0, s] = I_b \cdot \text{win}_p$, where

$$\text{win}_p \triangleq \begin{cases} 1 & |x - x_0| \leq \frac{s}{2},\ |y - y_0| \leq \frac{s}{2} \\ 0 & \text{otherwise} \end{cases}$$

Claim 2 *Claim 1 holds for the case of smaller image patches too. Namely, the true blur kernel $b(r_p, \theta_p)$ in patch P minimizes the difference between the blurry patch P and the corresponding patch in the Re-blurred image $I_b * b(r_p, \theta_p)$ (i.e., within win_p).*

Proof: Let I_{b_p} denote a uniformly blurred image resulting from globally blurring the (unknown) sharp image I_s with the local blur of patch P :

$$I_{b_p} = I_s * b(r_p, \theta_p). \quad (8)$$

Since I_{b_p} is uniformly blurred, then according to Claim 1: $I_{b_p} = I_{b_p} * b(r_p, \theta_p)$. Multiplying both sides by win_p yields:

$$I_{b_p} \cdot \text{win}_p = I_{b_p} * b(r_p, \theta_p) \cdot \text{win}_p. \quad (9)$$

While I_b and I_{b_p} are globally different, they are identical on the patch P:

$$I_b \cdot \text{win}_p = I_{b_p} \cdot \text{win}_p. \quad (10)$$

Similarly, $I_b * b(r_p, \theta_p)$ and $I_{b_p} * b(r_p, \theta_p)$ will also be the same in P, except at the patch boundaries. If, however, we assume the local blur is uniform within a slightly larger region than the window size, then the two blurred images will be identical for the entire window win_p, hence:

$$I_b * b(r_p, \theta_p) \cdot \text{win}_p = I_{b_p} * b(r_p, \theta_p) \cdot \text{win}_p = I_{b_p} \cdot \text{win}_p = I_b \cdot \text{win}_p. \quad (11)$$

■

To recap, in the case of *non-uniform blur*, we globally Re-blur (convolve) the noised blurry image $I_b + n$ with

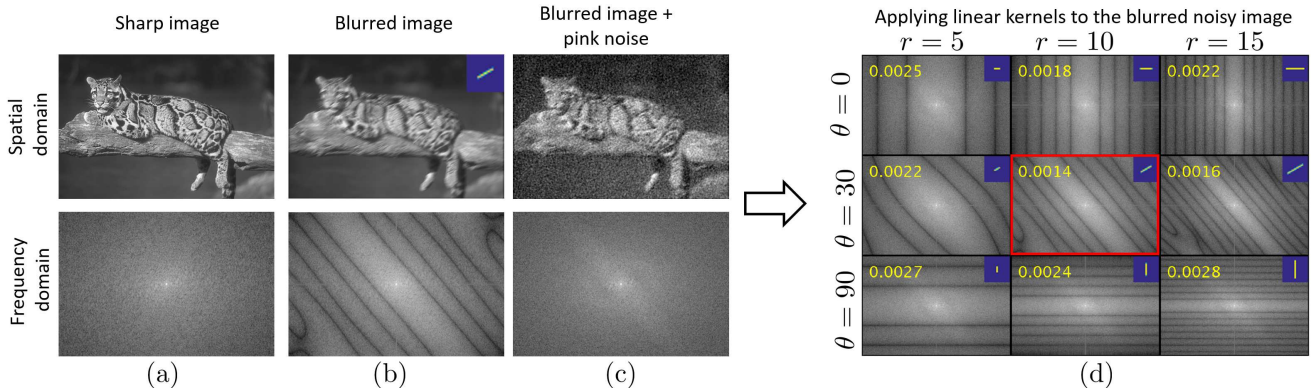

Figure 4: *Blur estimation by Re-blurring (see text in Sec. 3 for explanation of the figure).*

a variety of 1D kernels $b(r,\theta)$, and assign to each pixel (x,y) the blur which minimizes the MSE w.r.t. I_b within its surrounding window (e.g., patches P of size 30×30).

Realistic blurs: So far we discussed ideal LPF blurs. In reality however, the blur is caused by camera motion or scene objects motion, during exposure time. In the spatial domain, this corresponds to a convolution with a rect function, rather than with an ideal LPF. In the frequency domain, this corresponds to a multiplication by a sinc function, rather than by the ideal LPF rect function.

For non-ideal blurs $b(r,\theta)$, Eqs. 6 and 7 no longer hold, because $B^2(r,\theta) \neq B(r,\theta)$ (unlike a rect function, $\text{sinc}^2(r,\theta) \neq \text{sinc}(r,\theta)$). Note, however, that $\text{sinc}^2(r,\theta)$ does maintain the same zeros in the Fourier domain as $\text{sinc}(r,\theta)$, whereas multiplying by a different sinc kernel (with different r and θ) will generate new zeros in the Fourier domain (see Fig. 4.d).

For non-ideal blurs, the blur kernel minimizing $\|I_b * b(r,\theta) - I_b\|^2$ will always be the trivial delta function (no blur), rather than the correct blur kernel. As before, however, this changes once we add *noise* to I_b before Re-blurring. The kernel minimizing the MSE $\|(I_b + n) * b(r,\theta) - I_b\|$ (namely, $\|(I_b * b(r,\theta) - I_b) + n * b(r,\theta)\|$) is the one that best balances between (i) removing as much of the added noise n, and (ii) harming as little the spectral content of I_b. This balance is best reached when the noise added is "*pink-noise*", whose *Power Spectral Density* (PSD) function $S(f) \propto \frac{1}{f^\beta}$ resembles the PSD of natural images (we use $\beta = 2$) [17]. This is explained next and illustrated through Fig. 4. For simplicity, we illustrate it for uniform blur, but it generalizes to non-uniform blur using the same logic as in Claim 2.

Fig. 4(a-c) shows images along with their Fourier magnitudes. The sharp image I_s (Fig. 4.a) is blurred by a realistic (non-ideal) 1D blur kernel $b(r^*,\theta^*)$ (a *spatial* rect of length $r = 10$ and angle $\theta = 30°$, shown in the upper-right corner of 4.b). This generates the blurry image I_b (Fig. 4.b). As expected, this blur induces a multiplication by a sinc function in the Fourier domain in the direction $\theta = 30°$. Adding *pink-noise* n to the blurry image results in image $I_b + n$ (Fig. 4.c), whose spectral magnitude resembles that of a sharp natural image (see similarity to spectrum of 4.a).

We now Re-blur $I_b + n$ with various 1D blur kernels $b(r,\theta)$ of different lengths and orientations. The Fourier spectra of the resulting Re-blurs are shown in Fig. 4.d, along with their blur kernels at the upper-right corner. Notice that Re-blurring with the correct kernel (center image in (4.d)) yields the most similar result to the blurry image I_b (4.b), with zeros at the same frequencies. In contrast, other kernels with different θ and r yield zeros in different frequencies. While the differences and similarities are easier to illustrate in the fourier domain, recall that according to Parseval's theorem, the MSE between the blurry image (4.b) and its Re-blurred noisy versions in (4.d) are equal in the spatial and in the frequency domains. Indeed we see that the correct blur kernel yields the lowest MSE (computed in the spatial domain and displayed in yellow for each kernel).

The above Re-blurring method results in a dense blur-field $\hat{\mathbf{u}}$ (an *undirected* flow-field). This constitutes the initial estimate for the blur matrix $\hat{K}$ of Eq. 3, which is used for estimating an initial sharp image $\hat{I}_s$ (see Sec. 4).

As sharper images are recovered, $\hat{I}_s$ approaches the underlying sharp image I_s. Applying the Re-blurring method on $\hat{I}_s$ (instead of I_b) gradually converges towards our 'ideal' scenario mentioned earlier this section: namely, when both I_s and I_b are given, we can seek within each local window, the blur kernel that locally satisfies $I_s * b(r,\theta) = I_b$. Thus our flow estimation is gradually refined.

However, estimation of the flow with the Re-blurring method alone is not accurate enough to obtain good flow estimation, as it suffers from the *aperture problem* in small image patches (just like in regular flow estimation). For example, if a local patch contains only a blurry 1D edge, analysis of this patch alone can only estimate the component of the local blur in the direction perpendicular to the edge, but not in the direction parallel to the edge. This results in ambiguous local blur recovery. Increasing the window size

may resolve the aperture problem, but may no longer satisfy the locally-uniform motion assumption, especially at motion discontinuities. In our current implementation we usually work with patches of size around 30×30. Nonetheless, this often results in noisy blur-fields.

To alleviate this problem, we add a prior term to our blur estimation process, in the form of a smoothness constraint, which is imposed on the estimated blur-field

$$E_{blur}(\hat{K}(\mathbf{u})) = \sum_i w_i \|\nabla \mathbf{u}_i\|^2 \tag{12}$$

where w_i (the i^{th} pixel weight) penalizes large flow (blur) discontinuities that do not coincide with sharp image edges. In our current implementation, w_i is a decreasing sigmoid function of ∇I (initially estimated using the blurry image I_b, and later using the gradually recovered sharp image $\hat{I}_s$).

To recap, our motion flow estimation boils down to the following: Given the current estimate of the sharp image $\hat{I}_s$, Eq. 3 reduces to

$$E(\hat{K}) = E_{data}(\hat{I}_s, \hat{K}) + E_{blur}(\hat{K}). \tag{13}$$

At each iteration, we first minimize E_{data} using our Reblurring method, yielding $\hat{K}(\mathbf{u})$. We then refine it to further satisfy the smoothness constraint, by optimizing Eq. 13 (using Matlab's *lsqnonlin* function), using the estimated $\hat{K}$ as an initial guess.

4. Non-Uniform Deblurring Given $\hat{K}$

We next explain how we deblur an image corrupted by non-uniform blur given the blur field $\hat{K}$. Optimizing Eq. 3 in this case reduces to

$$E(\hat{I}_s) = E_{data}(\hat{I}_s, \hat{K}) + E_{image}(\hat{I}_s) \tag{14}$$

The image prior E_{image} is a combination of the internal patch recurrence prior [15] and the gradient sparsity prior [14, 12]. The former is an *image-specific* prior, whereas the latter helps reduce ringing artifacts.

The patch recurrence property is very strong in sharp natural images [5, 24]. However, it was shown to *diminish significantly in blurry images* [15] (see illustration in Fig. 3). It hence forms a very strong *image-specific prior* for image deblurring. Michaeli and Irani [15] showed that in the *uniform blur* case, the patch recurrence property degrades only across scales, therefore, they maximize the patch recurrence property across scales in order to reconstruct a sharp image. In this work we deal with non-uniform blur, in which the patch recurrence property degrades both within and across scales (as illustrated in Fig. 3). We use both deviations from ideal recurrence to reconstruct the sharp image.

When an image is downscaled by a factor of α it becomes α-times sharper. For example, an edge smeared over 10 pixels in the original image, would be smeared over only 5 pixels in an image which is half the scale ($\alpha = 2$). Because small patches recur across scales in sharp natural images, this implies that patches of the *unknown* sharp image *surface out in coarser scales of the blurry image*. Michaeli and Irani [15] used this observation in the case of *uniform* blur. They gradually constructed the sharp image $\hat{I}_s$ from a pool of α-times sharper patches extracted from an α-times downscaled version of the image $\hat{I}_s{}^{\alpha}$. Their iterative process minimizes the degree of dissimilarity between patches in $\hat{I}_s$ and their Nearest Neighbor patches (NNs) in $\hat{I}_s{}^{\alpha}$. Since in the downscaled version of the image the patches are only α times sharper, they serve as an *evolving* image specific prior for the gradually reconstructed sharp image, which is better than a fixed prior learned externally on sharp patches [25, 20]. We next show how to extend this idea to the case of non uniform blurs.

Maximizing the patch-recurrence property within and across scales (non-uniform blur case): The non-uniform blur case is different from the uniform one, because even though patches in the downscaled version are sharper than the original scale patches, each patch will not necessarily find its Nearest Neighbor (NN) in its sharper version. For example, consider two identical sharp patches in I_s undergoing different blurs, one of length r (denoted by P_1), and the other of length $\alpha \cdot r$ (denoted by P_2). In the α-times downscaled image, the blurry patch P_1 will be able to find a blurry patch *identical* to itself in the down scaled version of the blurry patch P_2. This would be a better NN than its own sharper downsacled version. This complicates things and makes the construction algorithm of [15] inapplicable to the non uniform blur case. On the other hand, in the case of non-uniform blur, a patch might find a sharper version of itself inside the same scale. For example, the blurry patch P_1 is an α-times sharper version of the blurry patch P_2 within the same image scale. This means that in the non-uniform case, unlike [15], we do not want to maximize the similarity between patches in $\hat{I}_s$ and their NNs in $\hat{I}_s{}^{\alpha}$.

Instead, we want to maximize the similarity between patches in $\hat{I}_s$ and an *internal* database of patches $DB(\hat{I}_s, \alpha)$ extracted from the same scale and coarser scales, which are guarantied to be α times sharper. Because the current estimate $\hat{K}$ is given, for each patch we know where to look for within and across scales of the image for potentially sharper NNs. Therefore, the recurrence prior, $\rho(\hat{I}_s, DB(\hat{I}_s, \alpha))$, measures the degree of dissimilarity between patches in $\hat{I}_s$ and the internal database of at least α-times sharper patches (as defined by $\hat{K}$).

Direct optimization of Eq. 14 is intractable since ρ is not convex. As in [25, 15], we adopt the "Half Quadratic Splitting" method [4], which solves the problem iteratively. In each iteration the current estimate $\hat{I}_s$ is constructed from patches in $DB(\hat{I}_s, \alpha)$ to produce an intermediate *sharper*

image z. The data term is then enforced using the constraint that the recovered image should be *close to* z. This gives a new estimate $\hat{I}_s$, used to construct an updated database $DB(\hat{I}_s, \alpha)$. This procedure is described in detail in [15]. In the latter internal step (enforcing the data term), we add the sparse gradient prior [14, 12], which minimizes $\sum_i |\nabla \hat{I}_s|^\gamma$ where $\gamma < 1$. This helps reduce ringing artifacts, especially in cases of overestimation errors in $\hat{K}$.

5. The Blind Deblurring Algorithm

Eq. 3 is not convex (due to the nature of the patch-recurrence prior and the matrix K), hence has no closed-form solution. We minimize it using an alternating iterative minimization procedure. Our initial estimate of the non-uniform blur-field $\hat{\mathbf{u}}$ is computed *directly* from the blurry image I_b, using the Re-blurring method (see Sec. 3). This is used to generate our initial blur matrix $\hat{K}(\mathbf{u})$ (see Eq. 5). The algorithm then proceeds to alternate between estimating a sharper image $\hat{I}_s$ and refining the blur field estimate $\hat{\mathbf{u}}$, as described in Algorithm 1.

When estimating the blur field, we Re-blur the input image with blurs of various magnitudes. The maximal blur magnitude we use is a parameter of our algorithm, set to 75 pixels in our current simulations. Setting this maximal radius induces a (half) circle of possible blurs, which is sampled every half a pixel. We then calculate and compare MSEs on images Re-blurred with all these blur kernels, following the description in Sec. 3.

In order to cope with large motion blurs, the algorithm is performed coarse-to-fine, starting from a downscaled version of I_b (~10 times smaller than the input). We typically use a patch of size 30×30 for estimating the blur-field. We typically perform 7 iterations at each scale, and then upscale the sharpened image $\hat{I}_s$ by a factor of $\frac{4}{3}$ to the next finer level, where we use it to initialize the process.

6. Experiments

We evaluated our blind deblurring algorithm on both synthetically blurred images and real blurry images. We used the database created by [19], which comprises 15 images synthetically blurred by various non-uniform motion blur fields with ground-truth (**GT**) data. We further added to it 30 more synthetically blurred images (uniform and non-uniform blurs); half of the images underwent *moderate blur* (motion of up to 16 pixels), while the other images were blurred by *larger motion blur* (up to 38 pixels) – see `www.wisdom.weizmann.ac.il/~vision/DeblurringByReblurring`. We also ran our algorithm on the real images used by [19], as well as on additional real images with severe (unknown) blurs.

Results on real and synthetic images are compared to the state-of-the-art CNN-based method [19]. We further compared our results on the synthetic database also to [23], since the synthetic non-uniform blurs comply with their 3D camera motion model assumption (the real images do not).

Input: Blurry image I_b
Output: Sharp image $\hat{I}_s$ and motion blur field $\hat{\mathbf{u}}$
Initialize: $\hat{I}_s = I_b$
for *t = 1, . . . ,T* **do**

1. **Estimate motion blur field** (Sec. 3):
   - **(a) Add pink noise** to $\hat{I}_s$, yielding $\hat{I}_s^{noised}$
   - **(b) Re-blur** $\hat{I}_s^{noised}$ to estimate $\hat{\mathbf{u}}$, yielding $\hat{K}(\mathbf{u})$.
   - **(c) Impose smoothness on $\hat{\mathbf{u}}$:** Fix $\hat{I}_s$ and minimize 13 w.r.t $\hat{K}(\mathbf{u})$, using $\hat{K}(\mathbf{u})$ from step **(b)** as an initial guess.
2. **Deblur image** (Sec. 4):
   - **(a) Construct** $DB(\hat{I}_s, \alpha)$ from patches of $\hat{I}_s$ and its α-times downscaled version $\hat{I}_s^{\alpha}$.
   - **(b) Deblur image**: Fix $\hat{K}$ and $DB(\hat{I}_s, \alpha)$, and minimize 14 w.r.t $\hat{I}_s$.

end

Algorithm 1: *The blind deblurring algorithm*

		PSNR (db)	SSIM	Flow Error (pixels)
Moderate blur	Ours	**24.87**	**0.743**	3.19
	[19]	24.14	0.714	**2.27**
	[23]	22.88	0.68	(no flow output)
Large blur	Ours	**22.01**	**0.624**	**5.74**
	[19]	20.84	0.56	8.64
	[23]	20.47	0.54	(no flow output)

Table 1: *Quantitative comparison to [19, 23]. Flow error is presented in average End Point Error (measured in pixels).*

Figs. 1,2 and 5 show visual comparisons; Table 1 shows quantitative comparisons. We quantify deblurring quality by comparing the recovered sharp images with the GT sharp images, using both the PSNR and SSIM measures. We also compare the performance of blur field estimation by comparing the recovered blur field with the GT blur field, using the *Average End Point Error* (EPE), measured in image pixels. It is defined as $\text{EPE} = \sqrt{(u - u_{\text{gt}})^2 + (v - v_{\text{gt}})^2}$, where $(u_{\text{gt}}, v_{\text{gt}})$ denote the ground truth flow.

Our method performs better than [19, 23] on moderate blurred images (+0.73dB, +2dB respectively), and does significantly better on severely blurred ones (+1.2dB, +1.54dB respectively). Our significant advantage over [19] on large blurs is also clearly visible in Figs. 1, 2. Our ability to handle a large range of non-uniform blurs stems from the fact that our method does not rely on any training data, hence not

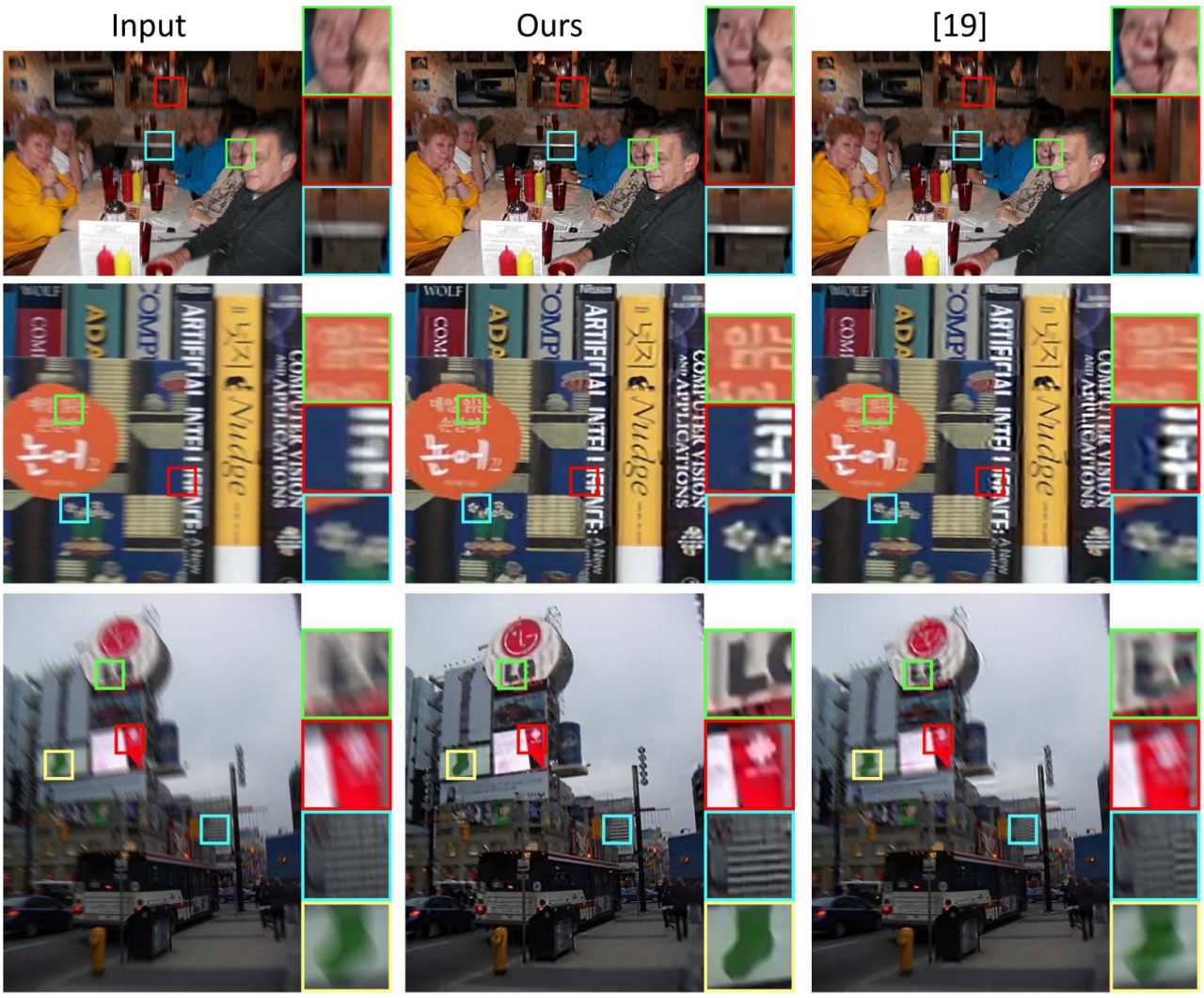

Figure 5: *Visual comparison of our method to [19] on images with moderate blurs (see www.wisdom.weizmann.ac.il/~vision/DeblurringByReblurring for full sized images and more results).*

restricted by a limited set of 'familiar' (trained) blur types.

Note however that on moderately blurred images, our flow estimation yields slightly worse results than [19]. Nevertheless, for these same estimated flows, our recovered sharp images have improved PSNR and SSIM compared to [19]. We attribute this to the fact that our flow estimation is very inaccurate in uniform image regions (due to the aperture problem). While this affects the flow error, it does not affect the deblurring quality.

To show that the locally linear blur assumption is not too restrictive, we ran our algorithm on the dataset of [11], containing *highly non-linear* blurs obtained by general camera motion. Our method yielded mean PSNR of 26.33dB, which ranks comparably to 8 non-linear deblurring methods on this dataset (based on quantitative results reported in [23]). This is despite the fact that all these methods are non-linear, whereas ours is locally linear, and some (the better ones) are inherently restricted to a pure 3D camera motion model, whereas our method can also handle non-rigid dynamic scenes. This shows that our local linear model does not pose a strong restriction.

7. Conclusion

We presented a method for single-image blind deblurring of non-uniform blurs. Our method iterates between estimating the non-uniform blur-field (using the Re-blurring method), and recovering the sharp image (using the image-specific patch recurrence prior). Our method does not rely on any training data, thus can handle a large variety of images and motions. Indeed we showed that it outperforms current training-based methods, such as CNNs, especially on large motions.

References

[1] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur, 2010. 1

[2] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics (SIGGRAPH ASIA 2009)*, 28(5):article no. 145, 2009. 1

[3] J. Gast, A. Sellent, and S. Roth. Parametric object motion from blur. *CoRR*, abs/1604.05933, 2016. 1

[4] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, Jul 1995. 6

[5] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 3, 6

[6] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision (ECCV)*, pages 622–635. Springer, 2012. 3

[7] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schlkopf. Fast removal of non-uniform camera shake. In *2011 International Conference on Computer Vision*, pages 463–470, Nov 2011. 1

[8] T. Hyun Kim, B. Ahn, and K. Mu Lee. Dynamic scene deblurring. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 1

[9] T. H. Kim and K. M. Lee. Segmentation-free dynamic scene deblurring. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2766–2773, June 2014. 1, 3

[10] T. H. Kim and K. M. Lee. Generalized video deblurring for dynamic scenes. In *Conference on Computer Vision and Pattern Recognition*, 2015. 1

[11] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 27–40, Berlin, Heidelberg, 2012. Springer-Verlag. 8

[12] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*, NIPS'09, pages 1033–1041, USA, 2009. Curran Associates Inc. 3, 6, 7

[13] A. Levin. Blind motion deblurring using image statistics. In P. B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 841–848. MIT Press, 2007. 1

[14] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971. IEEE Computer Society, 2009. 3, 6, 7

[15] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014. 1, 3, 6, 7

[16] J. Pan, Z. Hu, Z. Su, and M. H. Yang. l_0 -regularized intensity and gradient prior for deblurring text images and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):342–355, Feb 2017. 1

[17] T. Pouli, D. W. Cunningham, and E. Reinhard. Image Statistics and their Applications in Computer Graphics. In H. Hauser and E. Reinhard, editors, *Eurographics 2010 - State of the Art Reports*. The Eurographics Association, 2010. 5

[18] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schlkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1439–1451, July 2016. 1

[19] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 769–777, 2015. 1, 2, 3, 7, 8

[20] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *Proc. IEEE International Conference on Computational Photography*, 2013. 1, 6

[21] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *Int. J. Comput. Vision*, 98(2):168–186, June 2012. 1

[22] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of the 11th European Conference on Computer Vision: Part I*, ECCV'10, pages 157–170, Berlin, Heidelberg, 2010. Springer-Verlag. 1

[23] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, June 2013. 1, 7, 8

[24] M. Zontak and M. Irani. Internal statistics of a single natural image. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 977–984, Washington, DC, USA, 2011. IEEE Computer Society. 3, 6

[25] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 479–486, Washington, DC, USA, 2011. IEEE Computer Society. 6