Handling Noise in Single Image Deblurring using Directional Filters

Lin Zhong[1] Sunghyun Cho[2] Dimitris Metaxas[1] Sylvain Paris[2] Jue Wang[2]
[1]Rutgers University [2]Adobe Research

Abstract

State-of-the-art single image deblurring techniques are sensitive to image noise. Even a small amount of noise, which is inevitable in low-light conditions, can degrade the quality of blur kernel estimation dramatically. The recent approach of Tai and Lin [17] tries to iteratively denoise and deblur a blurry and noisy image. However, as we show in this work, directly applying image denoising methods often partially damages the blur information that is extracted from the input image, leading to biased kernel estimation.

We propose a new method for handling noise in blind image deconvolution based on new theoretical and practical insights. Our key observation is that applying a directional low-pass filter to the input image greatly reduces the noise level, while preserving the blur information in the orthogonal direction to the filter. Based on this observation, our method applies a series of directional filters at different orientations to the input image, and estimates an accurate Radon transform of the blur kernel from each filtered image. Finally, we reconstruct the blur kernel using inverse Radon transform. Experimental results on synthetic and real data show that our algorithm achieves higher quality results than previous approaches on blurry and noisy images. [1]

1. Introduction

Taking handheld photos in low-light conditions is challenging. Since less light is available, longer exposure times are needed – and without a tripod, camera shake is likely to happen and produce blurry pictures. Increasing the camera light sensitivity, i.e., using a higher ISO setting, can reduce the exposure time, which helps. But it comes at the cost of higher noise levels. Further, this is often not enough, and exposure time remains too long for handheld photography, and many photos end up being blurry *and* noisy. Although many techniques have been proposed recently to deal with camera shake, most of them assume low noise levels. In this work, we do not make this assumption and aim to restore a sharp image from a blurry and noisy input.

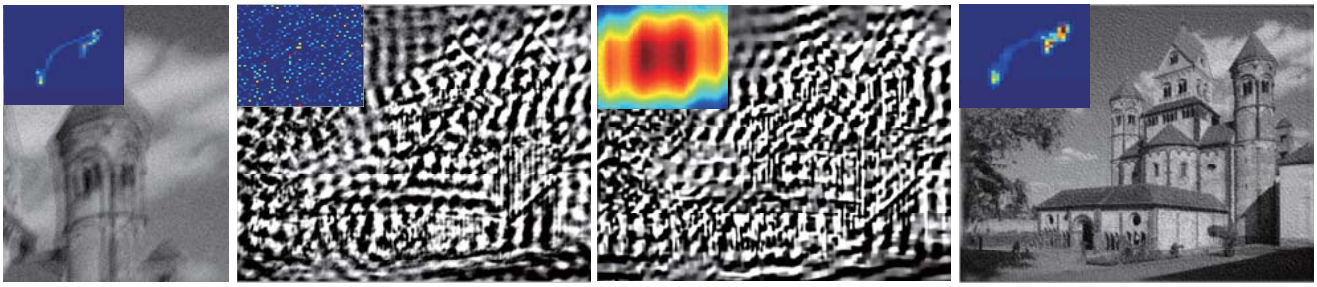
[1]This work was performed when the first author interned at Adobe Research.

Many single image blind deconvolution methods have been recently proposed [4, 6, 8–10, 13, 14, 16, 20]. Although they generally work well when the input image is noise-free, their performance degrades rapidly when the noise level increases. Specifically, the blur kernel estimation step in previous deblurring approaches is often too fragile to reliably estimate the blur kernel when the image is contaminated with noise, as shown in Fig. 1. Even assuming that an accurate blur kernel can be estimated, the amplified image noise and ringing artifacts generated from the non-blind deconvolution also significantly degrade the results [5, 11, 21, 22].

To handle noisy inputs in single image deblurring, Tai and Lin [17] first apply an existing denoising package [1] as preprocessing, and then estimate the blur kernel and the latent image from the denoised result. This process iterates a few times to produce the final result. However, applying existing denoising methods is likely to damage, at least partially, the detailed blur information that one can extract from the input image, thereby leading to a biased kernel estimation. In Sec. 2, we illustrate that standard denoising methods, from bilateral filtering to more advanced approaches such as Non-Local Means [3] and BM3D [7], have negative impacts on the accuracy of kernel estimation.

In this paper, we propose a new approach for estimating an accurate blur kernel from a noisy blurry image. Our approach still involves denoising and deblurring steps. However, we carefully design the denoising filters and deblurring procedures in such a way that the estimated kernel is not affected by the denoising filters. That is, we shall see that, unlike existing approaches, we can theoretically guarantee that our approach does not introduce any bias in the estimated kernel.

Our approach is derived from the key observation that if a directional low-pass linear filter is applied to the input image, it can reduce the noise level greatly, while the frequency content, including essential blur information, along the orthogonal direction is not affected. We use this property to estimate 1D projections of the desired blur kernel to the orthogonal directions of these filters. These projections, also known as the *Radon transform*, will not be affected by applying directional low-pass filters to the input image, except for the noise reduction. Based on this observation,



(a) Input (b) Cho and Lee [4] (c) Levin *et al.* [15] (d) Our method

Figure 1. Previous deblurring methods are sensitive to image noise. (a) Synthetic input image with 5% noise and the ground truth kernel (overlayed). It is cropped to better show blur and noise. (b) Estimated kernel and latent image by Cho and Lee [4]. (c) Results by Levin *et al.* [15]. (d) Results of our method.

we apply a series of directional low-pass filters at different orientations, and estimate a slice of kernel projection from each image. This yields an accurate estimate of the Radon transform. Finally, we reconstruct the blur kernel using the *inverse Radon transform*. Once a good kernel is obtained, we incorporate denoising filtering into the final deconvolution process to suppress noise and obtain a high-quality latent image. Results on synthetic and real noisy data show that our method is more robust and achieves better results than previous approaches.

2. Side effects of denoising as preprocessing

Before introducing our approach, we first analyze the negative impact of employing denoising as preprocessing on kernel estimation. In single image deblurring, a blurry and noisy input image b is usually modeled as:

$$b \quad = \quad \ell * k + n, \tag{1}$$

where ℓ, k and n represent the latent sharp image, blur kernel, and additive noise, respectively, $*$ is the convolution operator. Solving ℓ and k from input b is a severely ill-posed problem, and the additional noise n makes this problem even more challenging.

Assuming that ℓ is known, a common approach to solve for k is:

$$k \quad = \quad \text{argmin}_k \left\{ \|b - k * \ell\|^2 + \rho(k) \right\}, \tag{2}$$

where $\rho(k)$ is the additional regularization term that imposes smoothness and/or sparsity prior on k. Without considering the regularization term, this becomes a least-squares problem and the optimal k can be found by solving the following linear system:

$$\mathbf{L}^T\mathbf{L}\mathbf{k} = \mathbf{L}^T\mathbf{b} = \mathbf{L}^T(\mathbf{b}' + \mathbf{n}), \tag{3}$$

where $\mathbf{k}$ and $\mathbf{b}$ are the corresponding vector forms of k and b, respectively, and $\mathbf{L}$ is the matrix form of ℓ. We also introduce the noise-free blurry image $\mathbf{b}' = \mathbf{b} - \mathbf{n}$. We estimate the relative error of $\mathbf{k}$ with respect to the noise in $\mathbf{b}$ using the condition number of the linear system, that is:

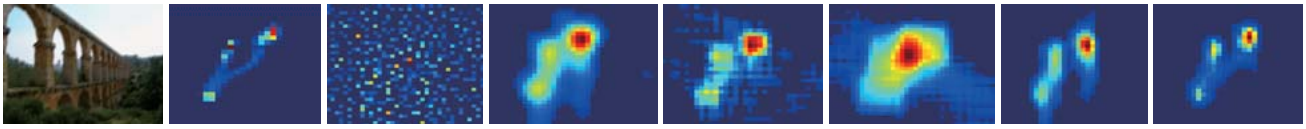
$$\begin{aligned} \frac{e(\mathbf{k})}{e(\mathbf{b})} &= \frac{\|(\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{n}\| / \|(\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{b}'\|}{\|\mathbf{L}^T\mathbf{n}\| / \|\mathbf{L}^T\mathbf{b}'\|} \\ &\leq \|(\mathbf{L}^T\mathbf{L})\| \cdot \|(\mathbf{L}^T\mathbf{L})^{-1}\| = \kappa(\mathbf{L}^T\mathbf{L}), \end{aligned} \tag{4}$$

where $e(\mathbf{k})$ and $e(\mathbf{b})$ are relative errors in $\mathbf{k}$ and $\mathbf{b}$, respectively. Thus, the noise $\mathbf{n}$ in the input image will be amplified at most by the condition number $\kappa(\mathbf{L}^T\mathbf{L})$ for kernel estimation, where $\mathbf{L}^T\mathbf{L}$ is often called the deconvolution matrix and has a block-circulant-with-circulant-block (BCCB) structure [12]. Eq. 4 shows that the upper bound on the error in the estimated kernel is proportional to the amplitude of the noise in input image. Building on this result, one can attempt to apply sophisticated denoising filter to the blurry image to reduce the noise amplitude, hoping that this will improve the kernel estimate. However, denoising filters also alter the profile of edges, e.g., [2]. This information is critical to accurate kernel estimation, and as we shall see, the benefits of the noise reduction are often outweighed by the artifacts caused by the profile alteration.

To illustrate it, we first look at a simple noise reduction method, Gaussian smoothing. Convolving with a Gaussian G_g decreases the noise level. However, the kernel estimation then becomes:

$$\begin{aligned} k_g \quad &= \quad \text{argmin}_{k_g} \|b * G_g - \ell * k_g\|^2 \\ &= \quad \text{argmin}_{k_g} \|(\ell * k + n) * G_g - \ell * k_g\|^2 \\ &\approx \quad \text{argmin}_{k_g} \|\ell * (k * G_g - k_g)\|^2 = k * G_g, \end{aligned} \tag{5}$$

where k is the blur kernel for the original input image and k_g is the optimal solution after Gaussian denoising. Eq. 5 shows that the estimated kernel k_g is a blurred version of the actual kernel k. Further, since G_g is a low-pass filter, the high frequencies of k are lost and recovering them from k_g would be very difficult, if possible at all. This result comes from the initial noise reduction and is independent of the kernel estimation method.



(a) Input (b) True kernel (c) No denoising (d) Gaussian filter (e) Bilateral filter (f) Non-local means (g) BM3D (h) Our method

Figure 2. The side effects of employing different denoising methods as preprocessing step in single image deblurring. (a) the synthetic input image with 5% noise. (b) the ground truth kernel. (c) the blur kernel estimated without applying any denoising method to the input image (a). (e)-(g) the estimated blur kernels after applying different denoising filters. (h) the kernel estimated by our method.

Although more sophisticated denoising methods are better at preserving high frequencies, denoising remains an open problem for which no perfect solution exists. Since no information about the blur kernel can be observed in uniform regions of the blurry image, edges are the main source of information that drives deblurring algorithms either implicitly or explicitly, e.g., [4, 6, 10, 20]. Even small degradations introduced by state-of-the-art denoising techniques can have a strong impact on deblurring results as shown in Fig. 2. In this experiment, we apply bilateral filtering [19], non-local means [3] and BM3D [7] to a test image with 5% noise, i.e., noise of standard deviation 0.05 when the intensity range is $[0, 1]$, and then use Cho and Lee's method [4] to estimate the blur kernel. The estimated kernels are not accurate due to the side effects of denoising.

The recent approach of Tai and Lin [17] first applies an existing commercial denoising package (NeatImage [1]) to the input image, then iteratively applies a motion-aware non-local mean filtering and deblurring to refine the results. Although special treatment has been added into the process, both the commercial denoising package and the non-local means filter have the same negative impacts on kernel estimation as we will show in Sec. 4.

3. Our approach

In the previous section, we have shown that there is a tension between noise reduction and edge preservation. The former helps to estimate a more accurate kernel, but the latter hinders it. Our experiments showed that even state-of-the-art denoising filters do have negative impacts on kernel estimation. In this section, we resolve this problem by using directional blur and the Radon transform to estimate the kernel. Our approach reduces the noise without degrading blur information, thereby producing better kernels.

3.1. Applying directional filters

We now show that directional low-pass filters can be applied to an image without affecting its Radon transform, while decreasing its noise level. We consider the directional low-pass filter f_θ:

$$I(\mathbf{p}) * f_\theta = \frac{1}{c}\int_{-\infty}^{\infty} w(t) I(\mathbf{p} + t\mathbf{u}_\theta) dt, \tag{6}$$

where I is an image, $\mathbf{p}$ is a pixel location, t is the spatial distance from one pixel to $\mathbf{p}$, c is the normalization factor defined as $c = \int_{-\infty}^{\infty} w(t)dt$, and $\mathbf{u}_\theta = (\cos\theta, \sin\theta)^T$ is a unit vector of direction θ. The profile of the filter is determined by $w(t)$, for which we use a Gaussian function: $w(t) = \exp(-t^2/2\sigma_f^2)$, where σ_f controls the strength of the filter.

Filtering the image affects the estimated kernel. With the same argument as for Eq. 5, the kernel that we estimate from the filtered image $b_\theta = b * f_\theta$ is:

$$k_\theta \quad = \quad k * f_\theta. \tag{7}$$

Similarly to filtering with a 2D Gaussian G_g, applying f_θ averages pixels and reduces the noise level. Since f_θ filters only along the direction θ, it has nearly no influence on the blur information in the orthogonal direction. We exploit this property to estimate the projection of the *original* kernel k along the direction θ. The projection can be formulated as Radon transform [6, 18], which is the collection of integrals of a signal (i.e., k) along projection lines. The particular value on Radon transform corresponding to one projection line $\rho = x\sin(\theta) + y\cos(\theta)$ is:

$$R_{\theta'}(\rho) = \int\int k(x, y)\delta(\rho - x\sin(\theta) - y\cos(\theta))dxdy, \tag{8}$$

where $k(x, y)$ indicates the value at the coordinate (x, y) on kernel k. θ and ρ are the angle and offset of the projection line, respectively. Thus, the projection of kernel k_θ along the projection direction θ is:

$$R_{\theta'}(k_\theta) = R_{\theta'}(k * f_\theta) = R_{\theta'}(k) * R_{\theta'}(f_\theta) = R_{\theta'}(k), \tag{9}$$

where $R_{\theta'}(\cdot)$ is the Radon transform operator to the direction θ', and $\theta' = \theta + \pi/2$. It is a linear operator, and one can verify that $R_{\theta'}(f_\theta)$ is a 1D delta function, given the definition of f_θ (Eq. 6). Eq. 9 shows f_θ has no impact on the Radon transform of the blur kernel to the orthogonal direction of the filter. This is the foundation of the proposed approach. An example is shown in Fig. 3.

3.2. The algorithm

We now explain how we recover the sharp image, with the kernel estimation first, and then the deconvolution step.

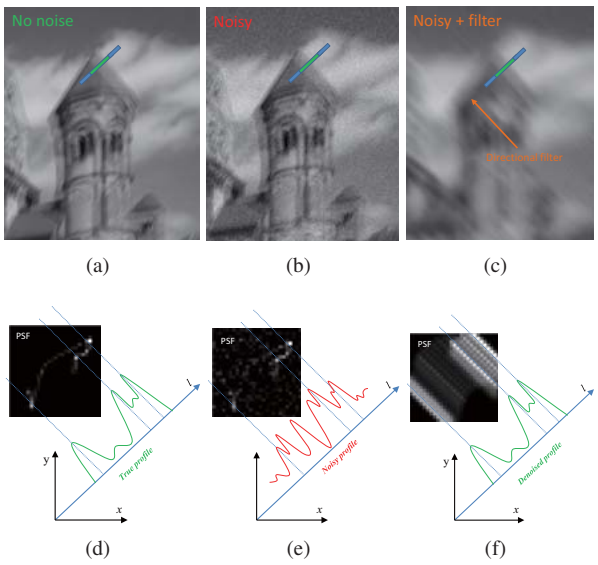


Figure 3. Directional denoising mechanism in single image deblurring. (a)(b)(c) are the synthetic image before adding noise, after adding noise, and after applying a directional filter($\theta = 3\pi/4$), respectively. (d)(e)(f) are the corresponding estimated blur kernels and their Radon transforms in the same direction. Note that the estimated kernel in (f) is largely damaged by the directional filter, but its Radon transform is the same as the one in (d).

3.2.1 Noise-aware kernel estimation

Based on the above analysis, we apply a directional blur f_θ, estimate the combined blur kernel k_θ, and then project it along the same direction of the filter to get the corresponding Radon transform. We repeat this process to get a set of projections. Finally, we compute the 2D kernel using the inverse Radon transform [18]. The advantage of this strategy is that it greatly reduces noise when applying f_θ, while keeping the computed Radon transform intact. However, so far, we have assumed that the latent image ℓ is known when estimating the blur kernels. This is not the case in practice, and even with state-of-the-art kernel estimation techniques, recovering k_θ from b_θ, which is a blurry image convolved with an additional directional blur, has proven to be challenging. The additional filter tends to make nearby edges "collide" with each other, which in turn introduces errors in the estimated kernel.

For a more reliable kernel estimation, we adopt the multiscale blind deconvolution framework commonly used in previous approaches [4, 20]. We create an image pyramid of the input image b as $\{b_0, b_1, ..., b_n\}$, where b_0 is the original resolution, and estimate the blur kernel in a bottom-up fashion from b_n to b_0. Since noise is largely removed by image downsizing, we apply an existing approach by Cho and Lee [4] to estimate the blur kernels k_i and latent images ℓ_i from layer b_n to b_1. Only for the full resolution layer b_0, we apply the directional filter f_θ and then estimate the kernel using the robust deconvolution technique described later in this section. The process is described in Algorithm 1. Steps 4 to 7 are also illustrated in Fig. 4. Specifically, in Step 5, although each filtered image b_θ is severely blurred with the additional filtering, the latent image ℓ_0, initialized from the multiscale process, is relatively sharp and clean, which allows us to estimate k_θ as:

$$k_\theta = \text{argmin}_{k_\theta} \left\{ \|\nabla b_\theta - k_\theta * \nabla \ell_0\|^2 + \rho(k_\theta) \right\}, \quad (10)$$

where ∇ is the gradient operator. This process is robust to noise because ∇b_θ is a low-pass filtered image. In Step 8, nonblind deconvolution is employed to update ℓ_0 based on the new k_0. However, existing methods do not work well in this case since we need to estimate a clean ℓ_0 from a noisy image b_0, and the results of previous methods are prone to inaccuracy. To generate a noise-free ℓ_0, we minimize the following energy function that aims for limiting the impact of noise on the result:

$$\|\nabla \ell_0 * k_0 - \nabla b_0\|^2 + w_1 \|\nabla \ell_0 - u(\nabla \ell_1)\|^2 + w_2 \|\nabla \ell_0\|^2, \quad (11)$$

where $u(\cdot)$ is the upsampling function, and w_1 and w_2 are pre-defined weights. The second term encourages the gradient of ℓ_0 to be similar to the upsampled gradient field of

Algorithm 1 *Multiscale noise-aware blind deconvolution*

Input: The pyramid $\{b_0, b_1, ..., b_n\}$ by down-sampling the input blurry and noisy image b, where $b_0 = b$.

Output: blur kernel k_0 and latent image ℓ_0.

1: Apply an existing nonblind approach ([4] in our implementation) to estimate k_i and ℓ_i for b_i,$i = n, ..., 1$.
2: Upsample ℓ_1 to generate initial ℓ_0.
3: **repeat**
4: Apply N_f directional filters to the input image b_0, each filter has a direction of $i \cdot \pi/N_f$, $i = 1, ..., N_f$, where N_f is the number of directional filters.
5: For each filtered image b_θ, use ℓ_0 as the latent image to estimate k_θ.
6: For each optimal kernel k_θ, compute its Radon transform $R_{\theta'}(k_\theta)$ as in Eq. 9, along the direction $\theta' = \theta + \pi/2$.
7: Reconstruct k_0 from the series of $R_{\theta'}(k_\theta)$ using inverse Radon transform.
8: Update ℓ_0 based on the new k_0 using a noise-aware nonblind deconvolution approach.
9: **until** k_0 converges.
10: With the final estimated kernel k_0, use the final deconvolution method described in Sec. 3.2.2 to generate the final output ℓ_0.

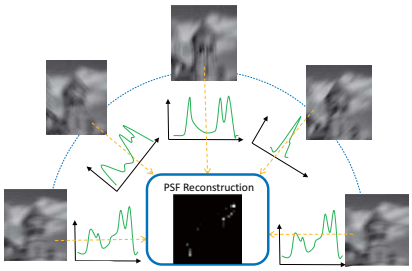


Figure 4. Illustration of applying directional filters for blur kernel estimation from a noisy input image. We apply directional filters in different orientations to the input image. From each filtered image a corresponding kernel is computed first, then projected along the same direction to generate the correct radon transform of the true kernel. The final blur kernel k_0 is reconstructed using inverse Radon transform [6].

ℓ_1, which is from the previous level in the pyramid. Since ℓ_1 contains much less noise due to image downsizing, incorporating this term can effectively reduce the noise level in ℓ_0. This non-blind deconvolution step is an intermediate step in blur kernel estimation that produces sufficiently accurate images at a limited computational cost. In the next section, we describe a more sophisticated non-blind deconvolution algorithm for generating high-quality final latent image given the estimated kernel.

It is worth mentioning that for simplicity, in the above discussion we assume b_1 is almost noise-free after downsizing the image by half. However, this will not be true, if severe noise presents in b_0. To deal with severe noise, we will only use previous methods to estimate blur kernels from b_n to b_2 in Step 1 of the algorithm, and then apply noise-aware kernel estimation from Step 2 to 9 to the last two layers b_1 and b_0. We applied this modified version of the algorithm to examples with 10% noise (Gaussian noise with standard deviation of 0.1) in Sec. 4.

Discussion Cho *et al.* [6] also use the Radon transform to recover the blur kernel. However, their approach to compute the kernel projection is different from ours. They rely on heuristics to identify straight edges in the images, and extract the projections from these edges. Because this process relies on a few arbitrary thresholds to locate and analyze the edges, it is sensitive to noise. We also show that it performs poorly on noisy inputs in the experimental section. In comparison, our approach does not rely on such arbitrary thresholds and performs well on noisy images.

3.2.2 Final noise-aware nonblind deconvolution

Once an accurate k_0 is estimated, we use it to estimate a good latent image ℓ_0 from the noisy input b_0. This is not a trivial task when b_0 contains severe noise [21]. However, since k_0 is fixed at this stage, it is safe to apply existing denoising methods in the process. This is in sharp contrast to Tai and Lin's method [17] where denoising and kernel estimation interfere with each other.

(a) Input image (b) Estimated kernel

(c) Zoran&Weiss [23] (d) Cho *et al.* [5] (e) Our method

Figure 5. Comparison results of our final noise-aware nonblind deconvolution with other recent nonblind deconvolution methods. The results are obtained using the same input image and the estimated kernel. (c),(d),(e) show the zoom-in results.

In our approach, we minimize the following energy function to estimate the final ℓ_0:

$$\|\ell_0 * k_0 - b_0\|^2 + w_3\|\ell_0 - \mathrm{NLM}(\ell_0)\|^2, \qquad (12)$$

where $\mathrm{NLM}(\cdot)$ is the non-local means denoising operation [3], and w_3 is a balancing weight. Minimizing this energy function will ensure that the deblurred result is noise-free, and can best fit with k_0 and b_0 as well.

Directly minimizing this energy is hard because $\mathrm{NLM}(\ell_0)$ is highly nonlinear. We found that iterating the following two steps yields a good result in practice:

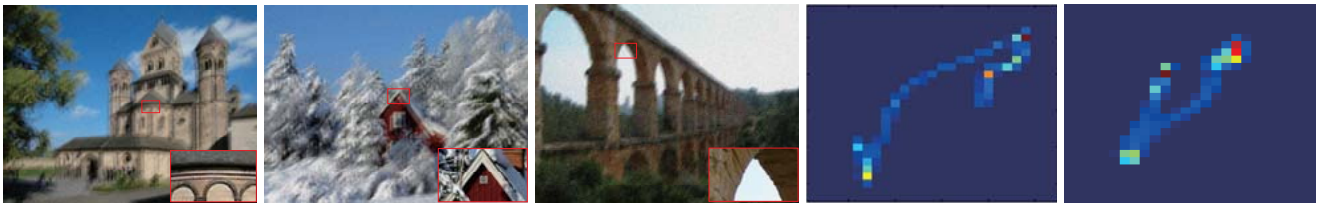
$$\ell_0' = \mathrm{NLM}(\ell_0), \qquad (13a)$$

$$\ell_0 = \mathrm{argmin}_{\ell_0} \left\{ \|\ell_0 * k_0 - b_0\|^2 + w_3\|\ell_0 - \ell_0'\|^2 \right\}. \qquad (13b)$$

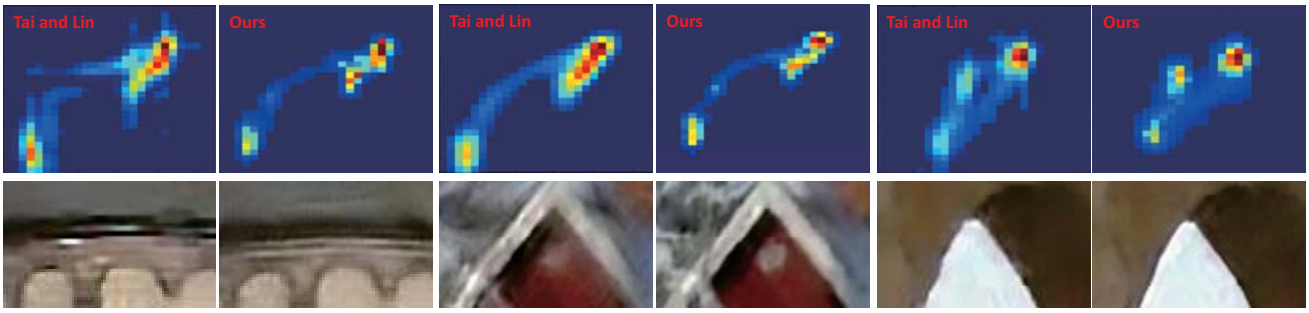
For initialization, we set ℓ_0' to be zero (a black image). Solving Eq. 13b yields a noisy ℓ_0 that also contains useful high-frequency image structures. In the alternating minimization process, the noise in ℓ_0 is gradually reduced, while the high-frequency image details are preserved. To show the effectiveness of our method, we compare it with other two recent non-blind deconvolution methods, i.e., Zoran and Weiss [23] and Cho *et al.* [5] in Fig. 5.

4. Experimental results

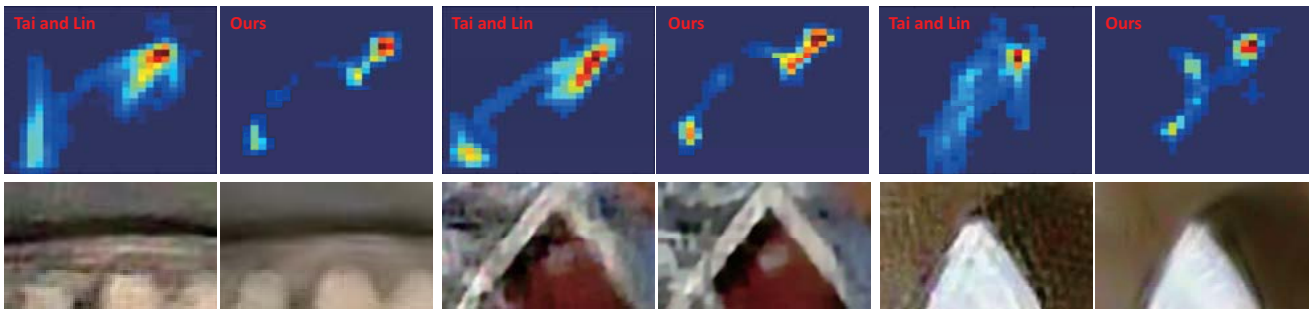
We implemented our method in Matlab on an Intel Core i5 CPU with 8GB of RAM. We apply directional filters along 36 regularly sampled orientations, that is, one sample every 5°. The computation time is a few minutes for a one-megapixel image. For all the experiments, we set the



(a) Abbey(input, 5% noise) (b) Chalet(input, 5% noise) (c) Aque(input,10% noise) (d) Kernel 1 (e) Kernel 2



(f) Abbey (result, 5% noise) (g) Chalet (result, 5% noise) (h) Aque (result, 5% noise)



(i) Abbey (result, 10% noise) (j) Chalet (result, 10% noise) (k) Aque (result, 10% noise)

Figure 6. Comparing Tai and Lin's method [17] and our method on synthetic data. Three input blurry image examples with different levels of noise are shown in (a),(b),(c). (d) and (e) are the ground truth blur kernels from Levin et. al. [14]. (d) is used for the examples "Abbey" and "Chalet", and (e) is used for the example "Aque". (f-k) show the estimated kernels and the latent images of Tai and Lin's method and our method with 5% noise and 10% noise. Due to the space limit only the areas highlighted by the bounding boxes in (a-c) are shown. Full size images for comparison are in the supplementary material.

extent σ_f of the directional filter to 30 pixels. We also set $w_1 = 0.05$ and $w_2 = 1$ (Eq. 11), and $w_3 = 0.05$ (Eq. 12).

4.1. Synthetic data

We first conducted experiments on images that we convolved with a known blur kernel and to which we added noise in a controlled fashion. This allows us to report quantitative measures in addition to visual results.

Comparisons with Tai and Lin's method Tai and Lin's method [17] is the most related work to ours since it also seeks to handle noisy images. This section focuses on comparing this method with our approach. We first ran comparisons on synthetic images (Fig. 6), where the latent sharp images were blurred using two blur kernels provided by Levin *et al.* [14]. We then added Gaussian noise with zero mean and standard deviations of 0.05 and 0.1 for a [0,1] intensity range. Tai and Lin kindly provided the results for their method. The comparison shows that visually our estimated blur kernels are closer to the ground truth, and our estimated latent images contain more details and less ringing artifacts. We also evaluate the results quantitatively by computing the Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) (Table 1).

Comparisons with other methods We also conducted experiments to explore how noise affects the performance of other state-of-the-art single-image blind deconvolution methods. Using the "Aque" image and the blur kernel shown in Fig. 6(e), we generated 10 input images with noise from 1% to 10%. We then applied different blind deconvolution methods to these test images, and measure the PSNR

		PSNR		SSIM	
	Noise	5%	10%	5%	10%
Abbey	Tai	22.43	21.05	.8122	.7242
	Ours	**22.73**	**21.61**	**.8150**	**.7270**
Chalet	Tai	19.79	18.95	.8244	.7162
	Ours	**22.80**	**19.35**	**.8273**	**.7200**
Aque	Tai	26.58	24.53	.8206	.7415
	Ours	**28.46**	**25.58**	**.8512**	**.7469**

Table 1. The comparison experiments of our method and Tai and Lin [17] on synthetic blurry images with different amount of noises. The performances are evaluated by PSNR and SSIM, comparing the generated latent images with the ground truth.

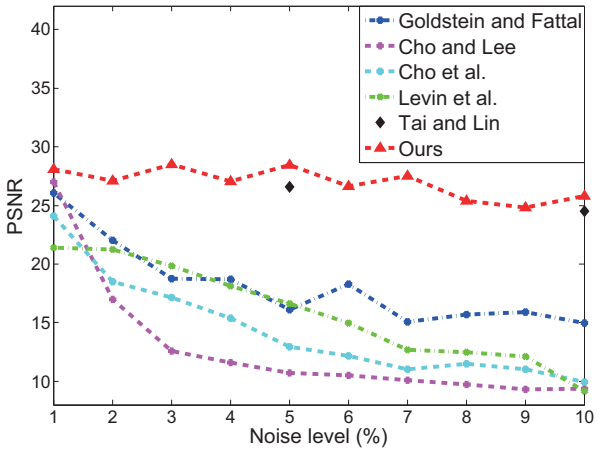


Figure 7. The PSNR curves of various blind deconvolution algorithms, including Goldstein and Fattal [9], Cho and Lee [4], Cho *et al.* [6], Levin *et al.* [15] and our method, on the 10 synthetic test images with noise level from 1% to 10%, generated by the "Aque" image and the kernel shown in Fig. 6(e). The two data points of Tai and Lin's method [17] are shown as black diamonds, which are provided by the authors. While the PSNR values are closer to ours, the visual difference is still significant; our approach produces cleaner images (Fig. 8). All images are included in the supplementary material.

curve of each method (Fig. 7). The accuracies of previous methods degrade rapidly when the noise level increases. On the contrary, our method is more robust, i.e., it works more reliably in the presence of noise, and achieves satisfactory results even when the input noise level is high. This figure also includes two data points of the Tai and Lin's method [17] provided by the authors themselves.

4.2. Results on real examples

We first compared our method and Tai and Lin's method on real-world images shown in their original paper [17], and the results are shown in Fig. 8. The results of other state-of-the-art methods can be found in [17]. Our estimated kernels are sharper than Tai and Lin's. The close-ups show that our method recovers more high-frequency details. For the boundaries of objects, our results have less noticeable ringing artifacts. Overall, our approach produces visually more satisfying results.

We further show our results on real-world photographs that were captured under common low-light conditions with a Nikon D90 DLSR camera and a $18 - 105$mm lens. We compare our results with those of other state-of-the-art methods, including Goldstein and Fattal [9], Cho and Lee [4], Cho *et al.* [6], Levin *et al.* [15]. The results (Fig. 9) show that our recovered latent images exhibit less artifacts, such as noise and ringing, and contain more high-frequency details at the same time. These observations are consistent across all test images. We provide additional examples in supplemental material.

5. Conclusion

We have shown that most state-of-the-art image deblurring techniques are sensitive to image noise. In this paper, we propose a new single image blind deconvolution method that is more robust to noise than previous approaches. Our method uses directional filters to reduce the noise while keeping the blur information in their orthogonal direction intact. By applying a series of such directional filters, we showed how to recover correct 1D projections of the kernel in all directions, which we use to estimate an accurate blur kernel using the inverse Radon transform. We also introduced a noise-tolerant non-blind deconvolution technique that generates high-quality final results. The effectiveness of the proposed approach is demonstrated on several comparisons on synthetic and real data.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback. This research is partially supported by the following grants to Dimitris Metaxasa: NSF-1069258 and Adobe Systems.

References

[1] Neatimage. http://www.neatimage.com/. 1, 3

[2] A. Buades, C. B., and J.-M. Morel. The staircasing effect in neighborhood filters and its solution. *IEEE Transaction on Image Processing*, 15(6), 2006. 2

[3] A. Buades, B. coll, and J. Morel. A non-local algorithm for image denoising. *CVPR*, 2005. 1, 3, 5

[4] S. Cho and S. Lee. Fast motion deblurring. *SIGGRAPH ASIA*, 2009. 1, 2, 3, 4, 7, 8

[5] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. *ICCV*, 2011. 1, 5

[6] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman. Blur kernel estimation using the radon transform. *CVPR*, 2011. 1, 3, 5, 7, 8

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *TIP*, 2007. 1, 3

(a) Santorini (b) Books

Figure 8. Comparisons of Tai and Lin's method and our method on real-world images from [17]. Our results contain more high-frequency details and less ringing artifacts. Zoom-in regions are shown in bounding boxes.

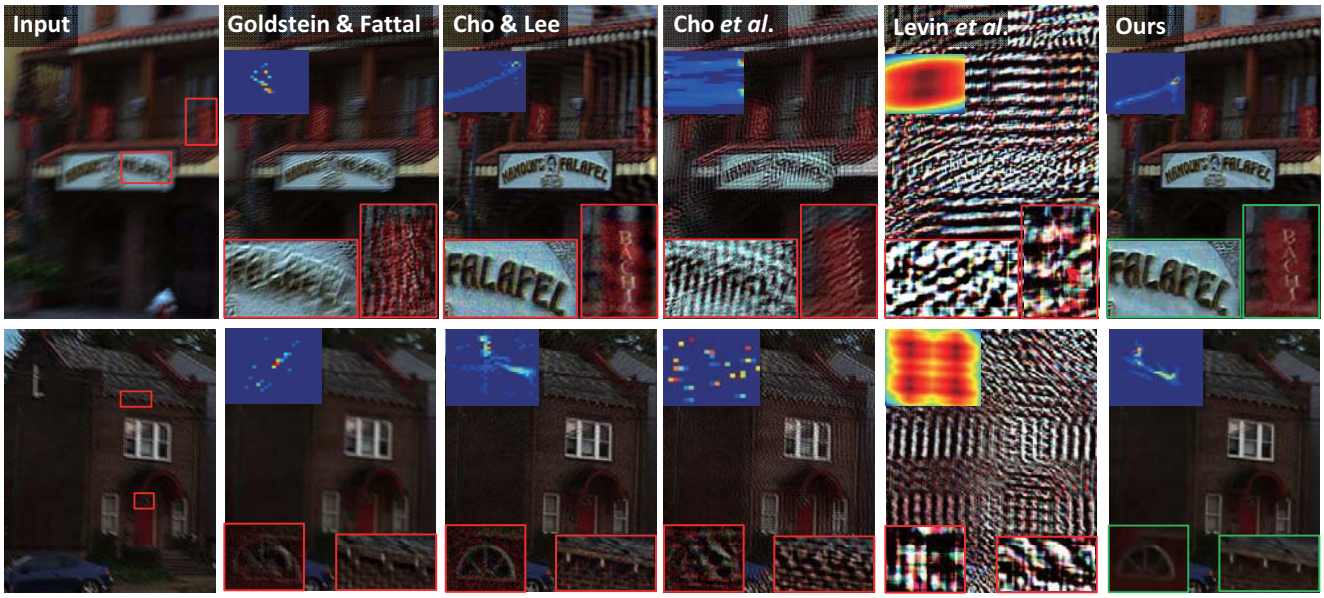


Figure 9. Comparisons on real-world examples, where we compare our results with the results of Goldstein and Fattal [9], Cho and Lee [4], Cho *et al.* [6], Levin *et al.* [15]. More results are in the supplementary material.

[8] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *SIGGRAPH*, 2006. 1

[9] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. *ECCV*, 2012. 1, 7, 8

[10] N. Joshi, R. Szeliski, and D. J. Kriegman. Psf estimation using sharp edge prediction. *CVPR*, 2008. 1, 3

[11] N. Joshi, C. L. Zitnicky, R. Szeliskiy, and D. J. Kriegman. Image deblurring and denoising using color priors. *CVPR*, 2009. 1

[12] B. Kim. *Numerical Optimization Methods for Image Restoration*. PhD thesis, Stanford University, 2002. 2

[13] S. Y. Kim, Y. W. Tai, S. J. Kim, M. S. Brown, and Y. Matsushita. Nonlinear camera response functions and image deblurring. *CVPR*, 2012. 1

[14] A. Levin, Y. Weiss, f. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. *CVPR*, 2009. 1, 6

[15] A. Levin, Y. Weiss, f. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. *CVPR*, 2011. 2, 7, 8

[16] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *SIGGRAPH*, 2008. 1

[17] Y. Tai and S. Lin. Motion-aware noise filtering for deblurring of noisy and blurry images. *CVPR*, 2012. 1, 3, 5, 6, 7, 8

[18] P. Toft. *The Radon Transform - Theory and Implementation*. PhD thesis, Technical University of Denmark, 1996. 3, 4

[19] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *ICCV*, 1998. 3

[20] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. *ECCV*, 2010. 1, 3, 4

[21] L. Yuan, J. Sun, L. Quan, and H. Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graph.*, 2008. 1, 5

[22] L. Zhang, A. Deshpande, and X. Chen. Denoising vs. deblurring: Hdr imaging techniques using moving cameras. *CVPR*, 2010. 1

[23] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. *ICCV*, 2011. 5