Supplementary Material: Phase-Only Image Based Kernel Estimation for Single Image Blind Deblurring

1. Method

1.1. Blur Model

Our goal is to find the latent sharp image from a single blurry image. The blurry image can be modeled as a convolution of the latent image with a blur kernel,

$$\mathbf{B} = \mathbf{L} \otimes \mathbf{k}, \tag{1}$$

where $\mathbf{B} \in \mathcal{R}^{m\times n}$, the blurry image is known, $\mathbf{L} \in \mathcal{R}^{m\times n}$ denotes the latent sharp image, and $\mathbf{k}$ is the blur kernel, $\otimes$ is the convolution operator. In the Fourier domain, this corresponds to $\mathcal{F}(\mathbf{B}) = \mathcal{F}(\mathbf{L}) \odot \mathcal{F}(\mathbf{k})$, where $\odot$ represents component-wise multiplication.

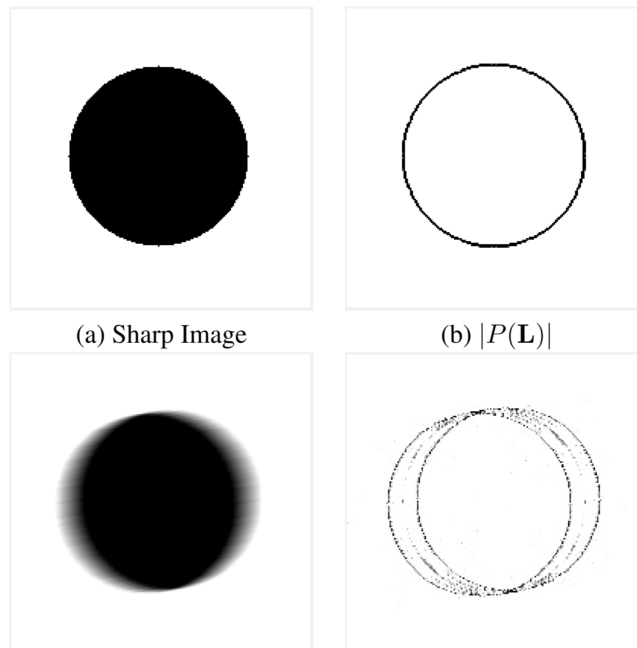
1.2. Fourier theory of phase-only images and deblurring

The phase and amplitude of a complex number $ke^{i\theta}$ are $e^{i\theta}$ and $k \geq 0$ respectively. Applying these component-by-component to a Fourier transformed image $\mathcal{F}(\mathbf{L})$ gives the phase and amplitude components. We denote taking the phase of a complex signal by $\mathcal{P}(\cdot)$. Taking the inverse Fourier transform of the phase-component gives the *phase-only image*, $P(\mathbf{L}) = \mathcal{F}^{-1}(\mathcal{P}(\mathcal{F}(\mathbf{L})))$. It is well known that the phase-only image bears more similarity to the original image than the analogously defined amplitude image. Figures 1 and 4 show phase-only images derived from real and synthetic images. As may be observed, taking a phase-only image acts as a sort of edge-extractor. This is related to the fact, noted by Kovesi ([2]) that the Fourier components of an edge tend to be in-phase with each other. For a real image $\mathbf{L}$, the phase-only image will also be real. Another simple property is *rotation-covariance*: if R represents rotation then $P(R(\mathbf{L})) = R(P(\mathbf{L}))$. It is also shift-covariant.

We now make a basic observation regarding the phase-only image of a convolution.

lemma 1. *The phase-only image of a convolution,* $P(\mathbf{L} \otimes \mathbf{k})$*, is equal to the convolution of the phase-only image and the phase-only kernel.*

$$P(\mathbf{L} \otimes \mathbf{k}) = \mathcal{F}^{-1}(\mathcal{P}(\mathcal{F}(\mathbf{L} \otimes \mathbf{k}))) = P(\mathbf{L}) \otimes P(\mathbf{k})\ . \tag{2}$$



(a) Sharp Image (b) $|P(\mathbf{L})|$

(c) Blurry Image (d) $|P(\mathbf{B})|$

Figure 1. *We use a circle image as an example. The image is blurred by a linear kernel, where the kernel length is 20 pixels and the direction is 10 degree.*

This results from a simple calculation.

$$\begin{aligned}\mathcal{P}(\mathcal{F}(\mathbf{L} \otimes \mathbf{k})) &= \mathcal{F}\big(\mathcal{F}^{-1}(\mathcal{P}(\mathcal{F}(\mathbf{L}))\mathcal{P}(\mathcal{F}(\mathbf{k})))\big) \\ &= \mathcal{F}\big(\mathcal{F}^{-1}(\mathcal{P}(\mathcal{F}(\mathbf{L}))) \otimes \mathcal{F}^{-1}(\mathcal{P}(\mathcal{F}(\mathbf{k})))\big) \\ &= \mathcal{F}\big(P(\mathbf{L}) \otimes P(\mathbf{k})\big)\ .\end{aligned}$$

2. Phase-only image of a linear kernel

For a simple linear (straight-line) blur kernel, the form of $P(\mathbf{k})$ can be computed. By rotation and shift covariance, it may be assumed without loss of generality, that $\mathbf{k}$ is axis-aligned, in which case $\mathbf{k}(x, y) = \delta(y)H(x)$, where $\delta(y)$ is a Dirac delta function and $H(x)$ is a top-hat. The Fourier transform is separable, so it follows that $P(\mathbf{k}) = \delta(y)P(H)$. Hence, we investigate what the $1D$

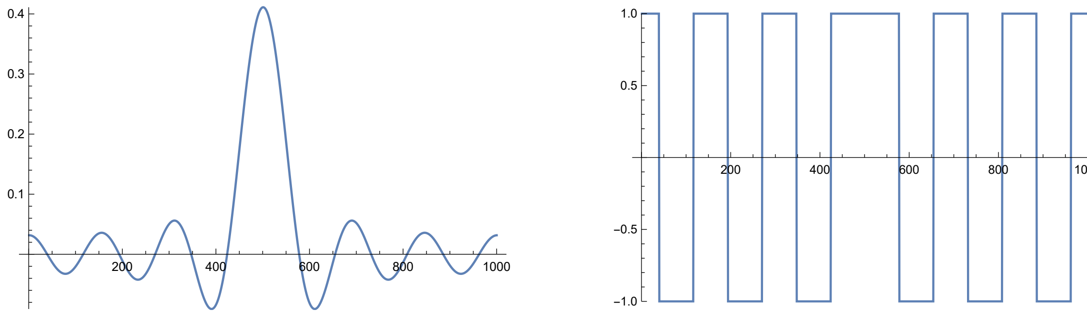


Figure 2. *Fourier transform of a top-hat function is* sinc *(left). The phase of this is shown on right. (The central peak has twice the width of the others. Note that since the top-hat is symmetric, its Fourier transform is real, hence its phase is either* $+1$ *or* -1*.) The phase-only image of the top-hat is obtained by taking the inverse Fourier transform of the function on the right.*

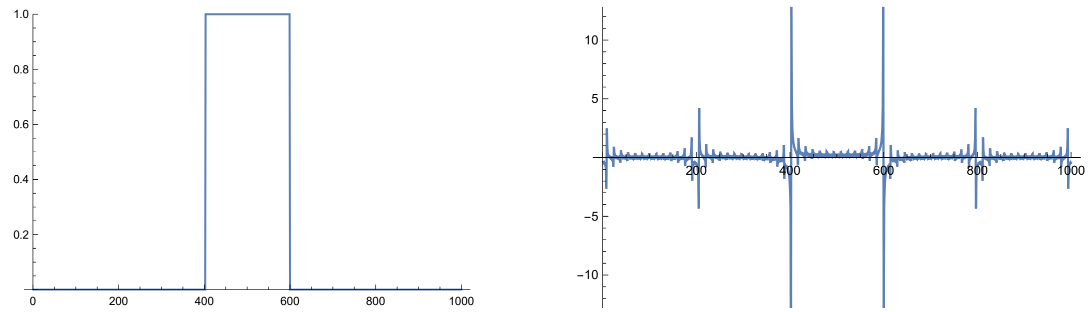


Figure 3. *Given a top-hat function (left), the phase-only signal has peaks separated by the width of the top-hat. The principal (central) peaks have the largest amplitude. (Note: to given most visually clear results, the top-hat functions used here and in Figure 2 are of different width.)*

phase-only signal $P(H)$ is. The result is shown in Figures 2 and 3.

Now, let $\mathbf{B} = \mathbf{L} \otimes \mathbf{k}$, where $\mathbf{k}$ is a straight-line filter $\mathbf{k}(x, y) = H(x)\delta(y)$. For simplicity of notation, we represent this kernel as $H \times \delta$. More generally, $(f \times g)(x, y)$ is defined as $f(x)g(y)$. Then

$$\begin{aligned} P(\mathbf{B}) &= P(\mathbf{L}) \otimes P(\mathbf{k}) \\ &= P(\mathbf{L}) \otimes (P(H) \times \delta) \\ &= P(\mathbf{L}) \otimes (D_x\sigma \times \delta) \\ &= D_x P(\mathbf{L}) \otimes (\sigma \times \delta) \ , \end{aligned} \tag{3}$$

where D_x represents the derivative in the x direction (or more generally the direction of alignment of the filter). We summarize.

theorem 1. *If* $\mathbf{B} = \mathbf{L} \otimes \mathbf{k}$, *where* $\mathbf{k} = H \times \sigma$ *is a straight-line blur kernel, then*

$$P(\mathbf{B}) = D_x P(\mathbf{L}) \otimes (\sigma \times \delta) \ ,$$

where D_x *represents the gradient in the kernel direction.*

In other words, blurring with a straight-line filter results in the creation of multiple copies ("ghosts"), of the derivative of the phase-only image, $D_x P(\mathbf{L})$, separated by the width of the filter. (The copies due to the principal peaks will be the most noticeable.)[1] This is shown in Figure 4.

Autocorrelation. As was shown, doing phase-only to obtain $P(\mathbf{B})$ from a blurry image results in multiple (two principal) shifted copies of $P(\mathbf{L})$. Note that $P(\mathbf{L})$ is not known. However, this suggests the use of autocorrelation of $P(\mathbf{B})$. There should be two peaks (separated by the width of the blur-kernel) in the autocorrelation image.

Autocorrelation of a signal $\mathbf{I}$ (1 or 2-dimensional) is computed using Fourier transform as:

$$\mathcal{A}(\mathbf{I}) = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{I}) \odot \overline{\mathcal{F}(\mathbf{I})}).$$

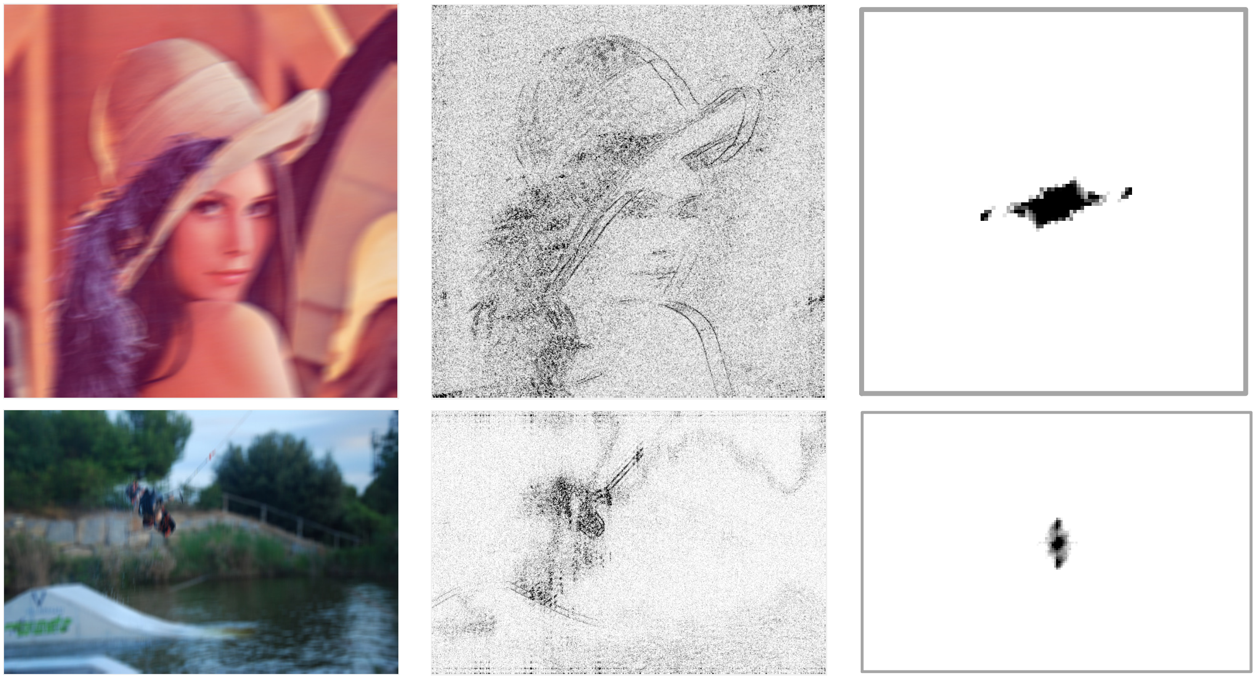
Unfortunately, if $\mathbf{I}$ is itself a phase-only image, derived from $\mathbf{J}$, then

$$\mathcal{F}(\mathbf{I}) = \mathcal{F}(\mathcal{F}^{-1}\mathcal{P}(\mathcal{F}(\mathbf{J}))) = \mathcal{P}(\mathcal{F}(\mathbf{J})).$$

So

$$\mathcal{A}(\mathbf{I}) = \mathcal{F}^{-1}(\mathcal{P}(\mathcal{F}(\mathbf{J})) \odot \overline{\mathcal{P}(\mathcal{F}(\mathbf{J}))}) = \mathcal{F}^{-1}(1) = \delta$$

[1] A more exact statement is that $P(\mathbf{B})$ consists of multiple ghosts, separated by the filter width, of the **gradient** of $P(\mathbf{L})$ in the filter direction. An exact derivation is given in the supplementary material. This includes also an exact derivation of $P(H)$.



(a) Blurry Image (b) $|P(\mathbf{B})|$ (c) $\mathcal{A}(|P(\mathbf{B})|)$

Figure 4. *(a) Input blurry images, the top one is synthetic and the bottom one is real. (b) The phase-only $|P(\mathbf{B})|$ of the blurry images results in two principal copies (others more faint) of $P(\mathbf{L})$. (c) Autocorrelation of phase-only blurred images, showing two distinct peaks (separated by the length of the filter kernel). Distinguishing the two principal peaks of the autocorrelation (apart from the origin) can be used to determine the orientation and width of a linear (straight-line) blur kernel.*

where δ is a Dirac delta function at the origin. In other words, a phase-only image **is completely unselfcorrelated**.

In other words, we cannot derive any information whatever from the autocorrelation of a phase-only image. The solution is to use the absolute value of the phase-only image instead. In other words, we compute $\mathcal{A}(|P(\mathbf{B})|)$, which should show the desired behaviour.

Figure 4 show the autocorrelated phase-only image $\mathcal{A}(|P(\mathbf{B})|)$ for various synthetically and naturally blurred images, blurred by camera shake.

3. More results

We provide additional experimental evaluation of our method on dataset from [1], shown in Fig.5-8.

References

[1] 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 *Proc. Eur. Conf. Comp. Vis.*, pages 27–40. Springer, 2012. 3, 4, 5, 6, 7

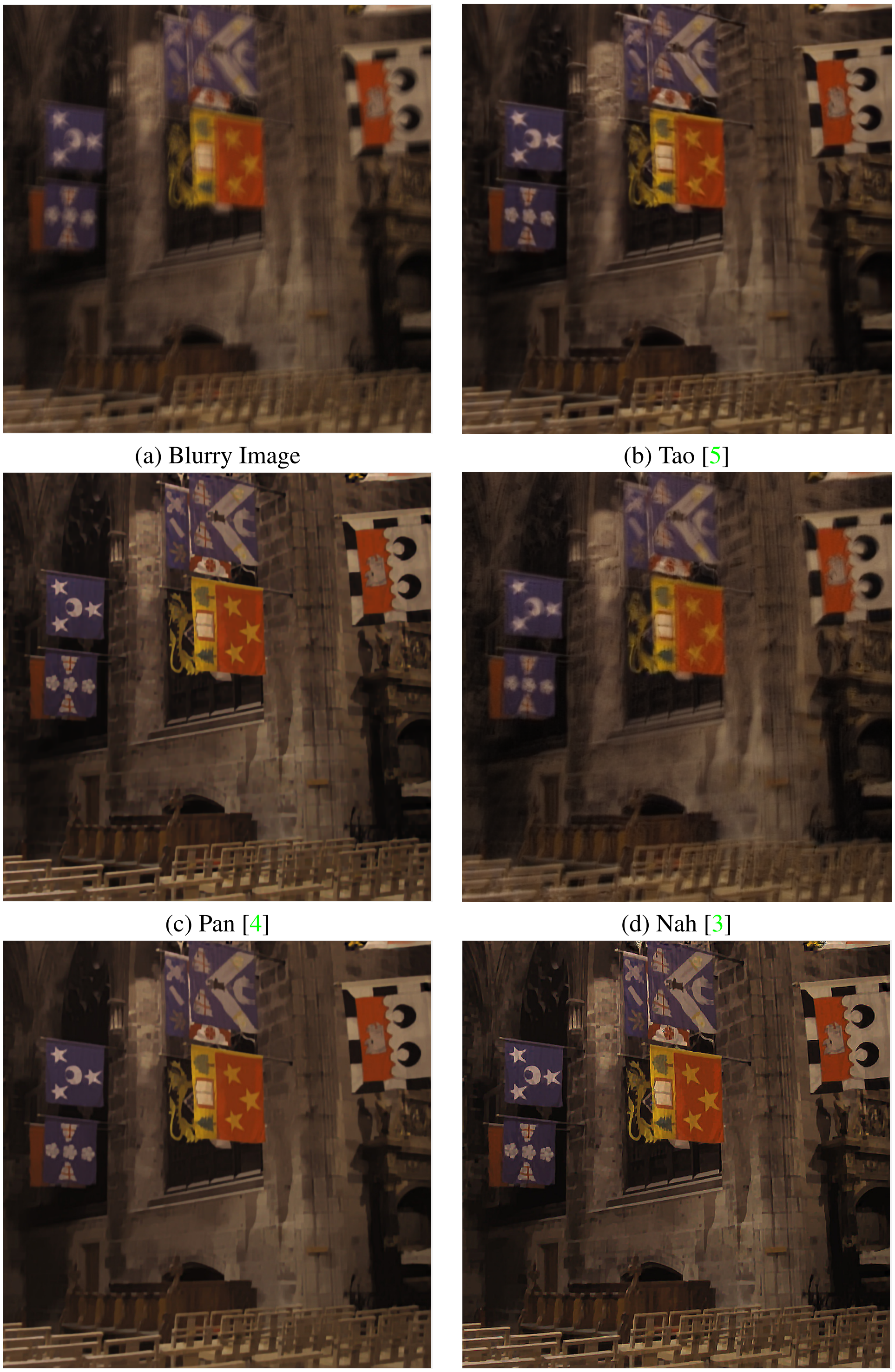
[2] P. Kovesi. Phase congruency detects corners and edges. In *DICTA*, 2003. 1

[3] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, July 2017. 4, 5, 6, 7

[4] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Deblurring images via dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017. 4, 5, 6, 7

[5] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2018. 4, 5, 6, 7

[6] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao. Image deblurring via extreme channels prior. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6978–6986, 2017. 4, 5, 6, 7

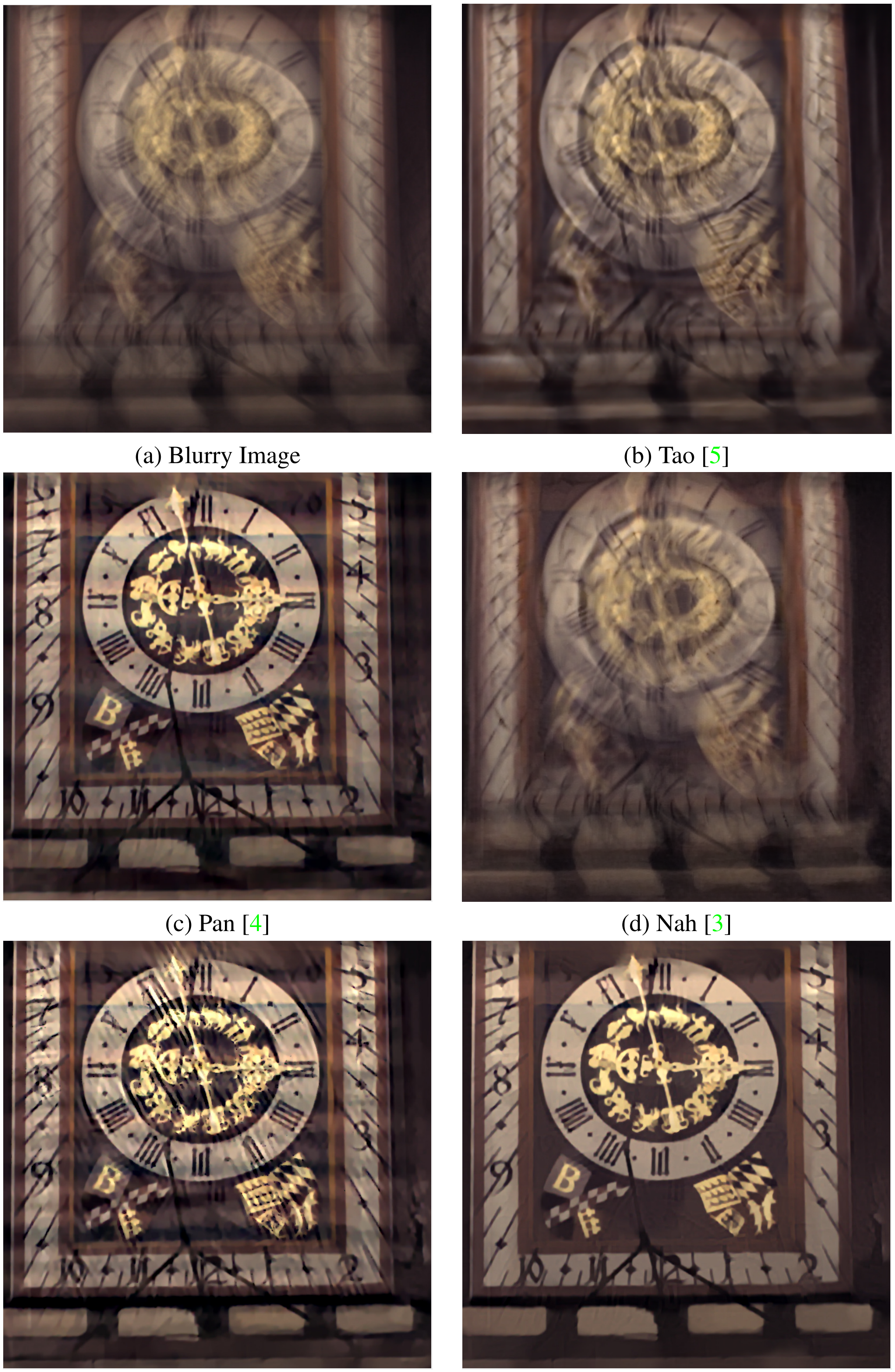


(a) Blurry Image (b) Tao [5]

(c) Pan [4] (d) Nah [3]

(e) Yan [6] (f) Ours

Figure 5. *Example of deblurring result on [1] dataset with kernel estimated by our method. (a) Input blurry images. (b) Deblurring results of [5]. (c) Deblurring results of [4]. (d) Deblurring results of [3]. (e) Deblurring results of [6]. (f) Our deblurring result. (Best viewed on screen).*



(a) Blurry Image (b) Tao [5]

(c) Pan [4] (d) Nah [3]

(e) Yan [6] (f) Ours

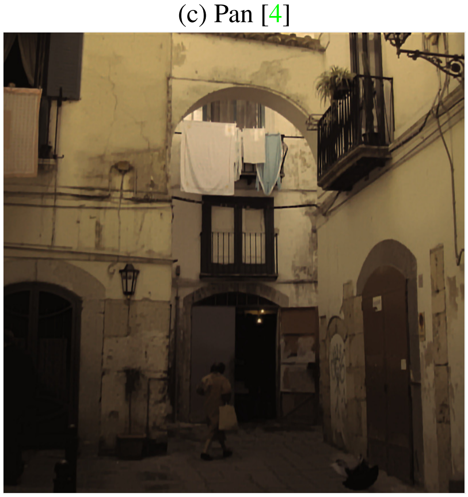
Figure 6. *Example of deblurring result on [1] dataset with kernel estimated by our method. (a) Input blurry images. (b) Deblurring results of [5]. (c) Deblurring results of [4]. (d) Deblurring results of [3]. (e) Deblurring results of [6]. (f) Our deblurring result. (Best viewed on screen).*

(a) Blurry Image

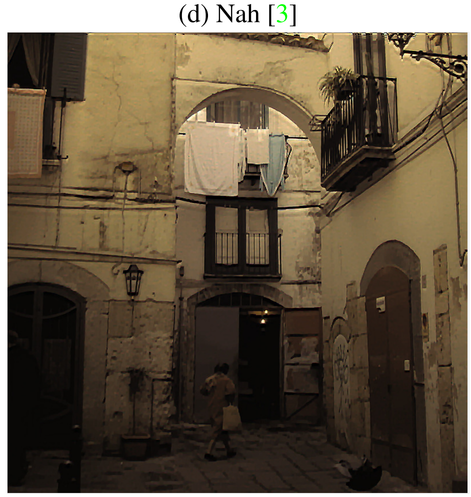
(b) Tao [5]

(c) Pan [4]

(d) Nah [3]

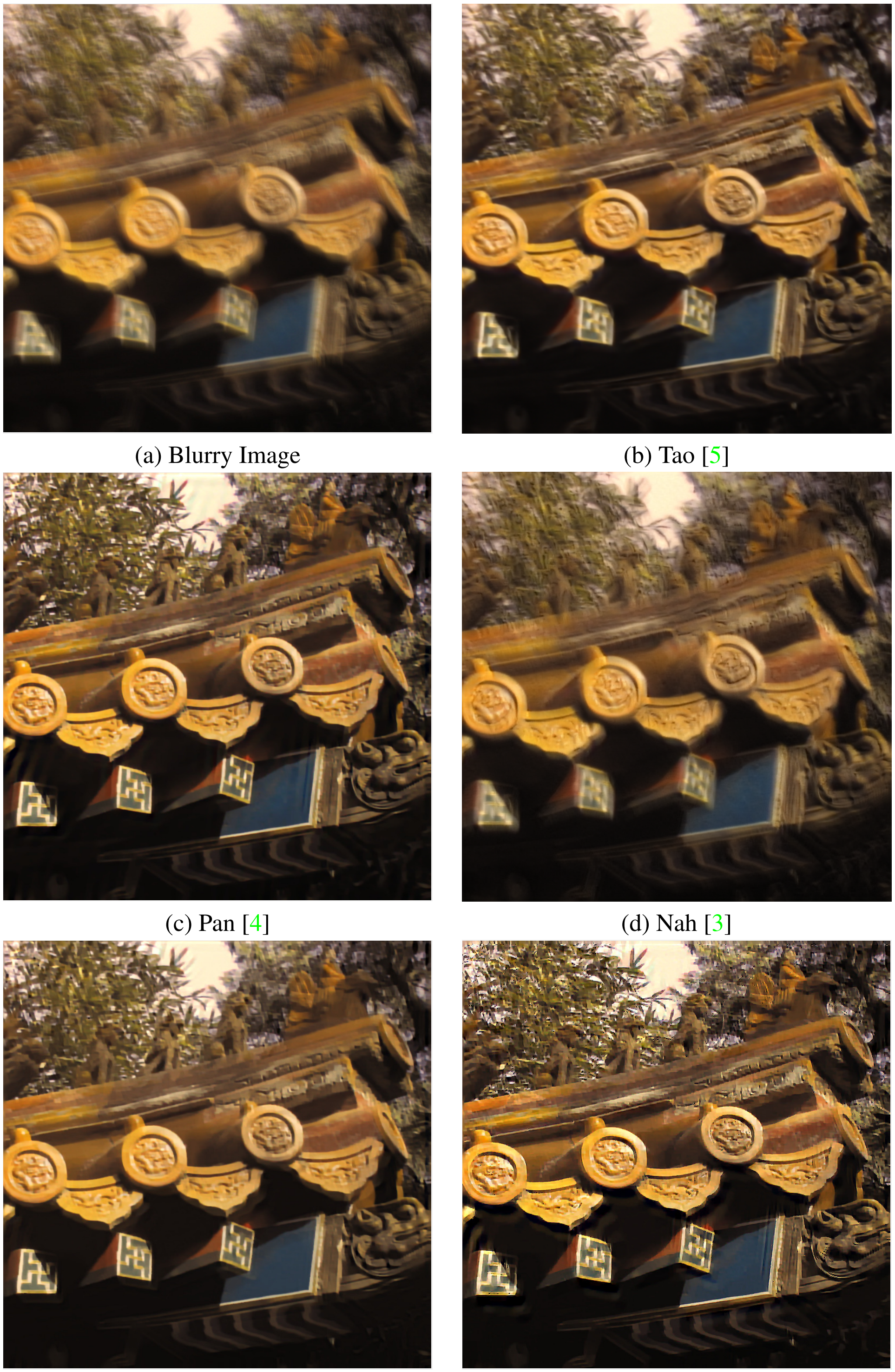


(e) Yan [6]



(f) Ours

Figure 7. *Example of deblurring result on [1] dataset with kernel estimated by our method. (a) Input blurry images. (b) Deblurring results of [5]. (c) Deblurring results of [4]. (d) Deblurring results of [3]. (e) Deblurring results of [6]. (f) Our deblurring result. (Best viewed on screen).*



(a) Blurry Image (b) Tao [5]

(c) Pan [4] (d) Nah [3]

(e) Yan [6] (f) Ours

Figure 8. *Example of deblurring result on [1] dataset with kernel estimated by our method. (a) Input blurry images. (b) Deblurring results of [5]. (c) Deblurring results of [4]. (d) Deblurring results of [3]. (e) Deblurring results of [6]. (f) Our deblurring result. (Best viewed on screen).*