Blur Kernel Estimation using Normalized Color-Line Priors

Wei-Sheng Lai[1]   Jian-Jiun Ding[1]   Yen-Yu Lin[2]   Yung-Yu Chuang[1]

[1]National Taiwan University   [2]Academia Sinica, Taiwan

Abstract

This paper proposes a single-image blur kernel estimation algorithm that utilizes the normalized color-line prior to restore sharp edges without altering edge structures or enhancing noise. The proposed prior is derived from the color-line model, which has been successfully applied to non-blind deconvolution and many computer vision problems. In this paper, we show that the original color-line prior is not effective for blur kernel estimation and propose a normalized color-line prior which can better enhance edge contrasts. By optimizing the proposed prior, our method gradually enhances the sharpness of the intermediate patches without using heuristic filters or external patch priors. The intermediate patches can then guide the estimation of the blur kernel. A comprehensive evaluation on a large image deblurring dataset shows that our algorithm achieves the state-of-the-art results.

1. Introduction

Image blur is a common unwanted photographing artifact caused by camera shake and its removal has been an active research topic for years. Assuming the blur is uniform across the entire image and the captured scene is static without too much depth variation, a motion blurred image y is often modeled as a convolution between a blur kernel k and a sharp image x:

$$y = k \otimes x + n, \tag{1}$$

where $\otimes$ is the convolution operator and n is additive noise.

For the problem of single-image blind deconvolution, the goal is to recover the desired sharp image x and the blur kernel k solely from the given blurry image y. Since there are many pairs of x and k that may result in the same y, many existing algorithms rely on prior knowledge on x and k to avoid trivial solutions. *Maximum a Posterior* (MAP) inference is a popular technique for solving blind deconvolution by jointly optimizing the latent image and the blur kernel. However, as shown by Levin *et al*. [11], the naive MAP would fail because it favors the no-blur solution. As

a result, some state-of-the-art methods use different strategies to maximize marginal distributions [4, 11, 12]. Another group of algorithms [16, 9, 21] incorporate special regularization into MAP to implicitly remove small details and maintain salient structures. Finally, edge-based approaches [3, 20] explicitly select and enhance salient edges.

Most previous methods assume that image gradients of x follow a heavy-tailed distribution [4, 16]. However, gradient priors consider only 2 or 3 neighboring pixels, which are not sufficient for modeling larger image structures. In recent years, patch priors that consider larger neighborhoods (*e.g*., 5×5 or 7×7 image patches) have been developed for image super-resolution [6, 13] and non-blind deconvolution [22, 18]. For blind deconvolution, Sun *et al*. [17] used a patch prior learned from an external image dataset to restore sharp edges, from which the blur kernel is estimated. Michaeli and Irani [14] adopted the internal patch recurrence property for estimation of the blur kernel. Both approaches resulted in a significant improvement in performance of blind deblurring, especially on robustness.

In this paper, we exploit the statistic property of the color-line model [15] on natural images and blur images, and then propose the normalized color-line prior for blur kernel estimation and blind deconvolution. The color-line model has been adopted to non-blind deconvolution by Joshi *et al*. [8]. They used the k-means algorithm to find two color centers for every image patch and built a prior for image reconstruction. Different from their approach, we propose to use *the normalized color-line prior* for *blur kernel estimation* based on two observations. First, the color-line prior is more effective for blur kernel estimation than image reconstruction. Second, the k-means centers are not effective enough for blind deconvolution because the blur process would have shrunk the distance between two centers of an image patch. Figure 1 visualizes the color-line distribution of a sharp patch and its blurred version in the RGB space. As seen from this figure, the centers found by the k-mean algorithm from the blur patch are too close and not effective for restoring contrast of the patch. Our method can find more effective centers.

Based on the normalized color-line prior, we propose a new blur kernel estimation method. Our method belongs
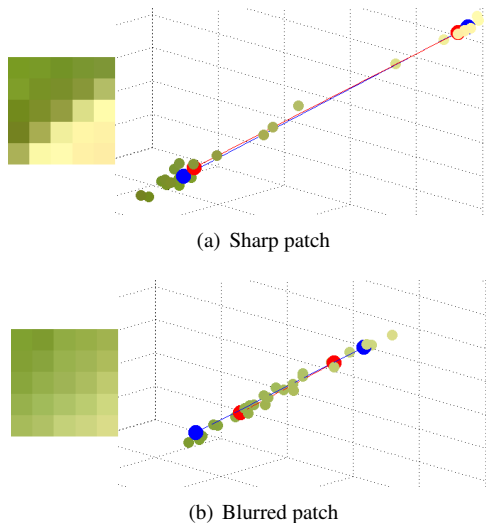
(a) Sharp patch

(b) Blurred patch

Figure 1: The color-line distribution of a sharp patch and its blurred version. The red dots are k-means centers, and the blue dots are centers found using our model. The blur process shrinks the distance between two color clusters and brings close two clusters. Thus, the k-mean centers found from the blurred patch is less effective for deblurring than the ones found by our model. Our method better separates cluster centers and can enhance contrasts better.

to the category of edge-based approaches [3, 20, 17] which predict sharp intermediate patches for salient structures and estimate the blur kernel using the blur/sharp pairs. The key of these approaches is to predict the sharp intermediate patches from blur patches. The shock filter was used for this purpose by Cho and Lee [3]. Unfortunately, the shock filter could over-sharpen image edges, and is sensitive to noise. The noise problem is especially obvious when applying the shock filter to color images. Recently, Sun *et al.* used example patches from a set of edge patches as priors to help restore patch contrasts in the intermediate image [17]. This approach could run into problems when there is no appropriate patch for the blur patch in the set of examples. We propose to reconstruct the sharp intermediate patches using the proposed normalized color-line prior. With this prior, we can obtain a pair of patch centers whose distance is longer than k-means centers, as shown in Figure 1(b). The new patch centers will lie on the same color-line and avoid enhancing noise at the same time. In addition, our method does not have the problem with the patch priors using external examples [17] as our contrast restoring process is derived from the blur patch itself and does not rely on external examples. Experiments show that our algorithm produces accurate and stable blur kernels, and outperforms the state-of-the-art methods on a large benchmark for image deblurring.
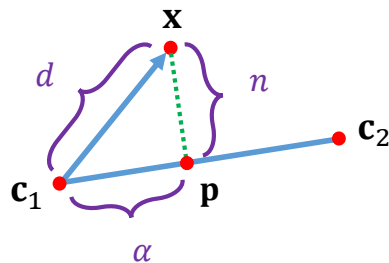

Figure 2: An illustration of the color-line model. The priors of Joshi *et al.* [8] minimize n and fit α to a learning curve. Our prior considers both d and the distance between c_1 and c_2.

2. The color-line model

The color-line model [15] is a local statistic model which claims that pixel colors x within a local image patch can be well represented by linear combinations of two color centroids:

$$x = \alpha c_1 + (1 - \alpha)c_2, \qquad (2)$$

where c_1 and c_2 are centers of two color clusters, and α is the linear mixing parameter. Figure 2 illustrates the color-line model. This model has already been used for several computer vision problems, such as image segmentation [15], alpha matting [10], depth map estimation [2], and image smoothing [7]. Joshi *et al.* [8] adopted the color-line model to non-blind deconvolution. They used k-means algorithm to obtain two centers in a local 5×5 patch. By minimizing the perpendicular distance from a pixel's color value to the 3D line formed by connecting two color centers (the value n in Figure 2) and fitting α to a piece-wise hyper-Laplacian function, their algorithm was shown effective for non-blind deconvolution. However, since the blur kernel is unknown, blind deconvolution algorithms rely on image priors to sharpen edge contrasts. As shown in Figure 1, the distance between c_1 and c_2 has been shorten by the blur process. Therefore, it is not effective enough to enhance edge contrasts by minimizing the distance between image pixels and k-means centers.

To further understand the effectiveness of color centers, we analyze the statistic property of the color-line model between sharp images and blur images using a large set of patches from Berkeley segmentation dataset BSDS500 [1]. We used blur kernels from Levin's dataset [11] and added 1% additive Gaussian noise to synthesize blurred images for each image in BSDS500. For each image, we extract 5×5 patches without overlapping, resulting in a total of 100k patches.

We randomly sample sharp and blur patches to analyze how the color-line model performs. For each sampled patch, we first follow Joshi *et al.*'s method to form the color line model. The k-means algorithm is used to find the two
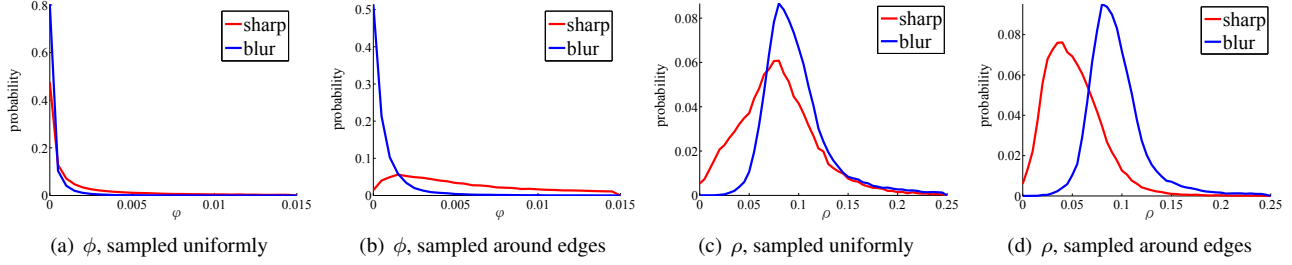
(a) ϕ, sampled uniformly   (b) ϕ, sampled around edges   (c) ρ, sampled uniformly   (d) ρ, sampled around edges

Figure 3: The distributions (normalized histograms) of ϕ and ρ values for blur and sharp patches with different sampling strategies. Our normalized color-line prior ρ separates two modes more effectively than ϕ when sampling around edges.

color centers. In this setting, for a patch P, the k-means algorithm partitions N pixels in P into two sets by minimizing the following objective function:

$$\phi(P) = \sum_{j \in P} \sum_{k=1}^{2} r_{jk} \|\mathbf{x}_j - \mathbf{c}_k\|^2, \qquad (3)$$

where $r_{jk} = 1$ if $\mathbf{x}_j$ is assigned to the k-th cluster, and $r_{jk} = 0$ otherwise. Figure 3(a) shows the distributions of the ϕ values for sharp patches and their blur versions. It can be observed that both distributions almost overlap together. Next, we only sample patches around image edges and perform the same analysis, resulting in the distributions in Figure 3(b). This time, the ϕ distribution of sharp patches slightly shifts towards higher values, but there is still significant overlap between two distributions.

From our observation (as illustrated in Figure 1), for sharp patches, pixel colors tend to be clustered around two ends of the color line. On the other hand, for blur patches, pixel colors stretch over the color line more uniformly. In the other words, in sharp patches, the distance between pixel $\mathbf{x}$ and its associated color center tends to small compared to the distance between two color centers. Thus, we incorporate the term $\|\mathbf{c}_1 - \mathbf{c}_2\|^2$ into Equation (3), and it leads to the following objective value:

$$\rho(P) = \frac{\sum_{j \in P} \sum_{k=1}^{2} r_{jk} \|\mathbf{x}_j - \mathbf{c}_k\|^2}{\|\mathbf{c}_1 - \mathbf{c}_2\|^2}. \qquad (4)$$

Here, we still use the k-means algorithm to find $\mathbf{c}_1$ and $\mathbf{c}_2$, and then evaluate the ρ values on patches sampled randomly (Figure 3(c)) and only around image edges (Figure 3(d)). In Figure 3(c), it is still difficult to distinguish sharp patches from blur patches because the two modes are very close. However, when sampling patches around image edges, the distributions of ρ values for sharp and blur patches become separable in Figure 3(d). The sharp patches usually have lower ρ values than blur patches. From this experiment, we have two observations:

1. The objective function ρ serves as a better image prior for sharp patches than ϕ. By minimizing Equation (4),

pixel colors are pulled close to $\mathbf{c}_1$ and $\mathbf{c}_2$ while the distance between $\mathbf{c}_1$ and $\mathbf{c}_2$ is stretched. Thus, the contrast is enhanced and the patch becomes sharper. We could gradually make blur patches become sharper and sharper by minimizing the ρ value of a patch.

2. The color-line prior is more effective for patches around salient image structures such as edges than other regions such as flat regions, texture regions and noisy patches. Since we assume there are two primary colors in a patch, we have to avoid using our prior on patches that violate the color-line model. Fortunately, these regions are either useless or even harmful to blur kernel estimation [20]. Thus, the color line prior is better suited for kernel estimation than non-blind image deconvolution.

3. Method

Our approach is a MAP-based framework that iteratively solves x and k using a coarse-to-fine scheme. The main difference from other methods is that we use RGB channels together instead of the intensity layer for kernel estimation. Since we only adopt our image prior to patches around useful edges, we do not need to reconstruct the whole latent image during the blur kernel estimation stage. After the blur kernel has been estimated, we restore the final latent image by a state-of-the-art non-blind deconvolution algorithm. Figure 4 illustrates the whole pipeline of our framework.

3.1. Find the edge mask M

For each image scale, we compute the r-map [20] from the blur image y. The r-map selects useful step edges that would benefit the blur kernel estimation. We use the width of the blur kernel at the current scale as the window size of the r-map, and select candidate regions with top 10% of r-values. Then, we adopt the same strategy as Sun *et al.* [17] to filter the blur image with a filter bank consisting of derivatives of Gaussians in eight directions to locate edge pixels. The intersection of these two masks becomes our
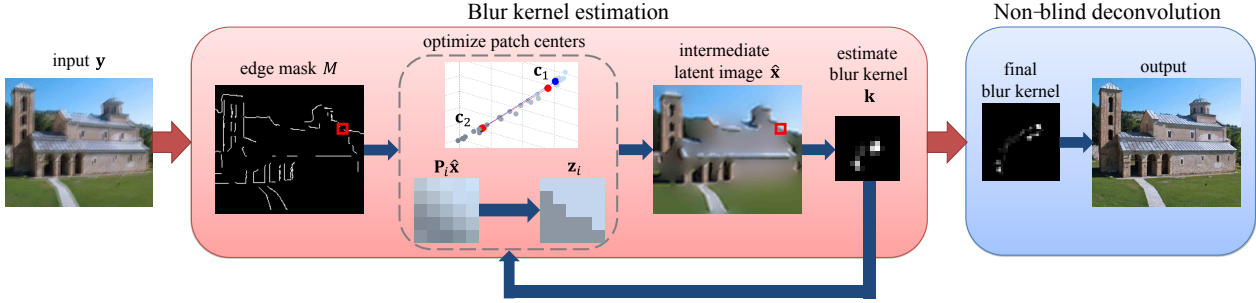
Figure 4: The pipeline of our algorithm. Our algorithm builds on a coarse-to-fine pyramid framework. For each image scale, we iterate between x-step and k-step to restore strong edges and estimate the blur kernel. To be more specific, we optimize the proposed color-line prior ρ in Equation (4) to stretch the distance between patch centers and avoid enhancing noise. After blur kernel estimation, we could apply any state-of-the-art non-blind deconvolution algorithm to restore latent images.

edge mask M. Applying r-map to blur images avoids selecting ringing regions generated from deconvolution, and adopting the filter bank chooses the better locations to apply our color-line prior. Finally, in order to eliminate patches that are not fitted well with the color-line model, we use RANSAC [5] to find the color-line model and further remove outlier pixels from the edge mask M. Note that the edge mask is calculated once for each image scale.

3.2. x-step

Given the current estimation of the blur kernel k, the goal of the x-step is to generate an intermediate image $\hat{x}$, which contains only constructed image patches of the salient edges selected by the mask M. To find $\hat{x}$, we minimize the following objective function:

$$
\begin{aligned}
E_x(\hat{\mathbf{x}}) \;=\; & \|\mathbf{My} - \mathbf{MK}\hat{\mathbf{x}}\|_2^2 + \lambda \sum_{*\in\{h,v\}} \|\mathbf{M}\nabla_*\hat{\mathbf{x}}\|_2^2 \\
+ \; & \frac{\beta}{|M|} \sum_{i\in M} \rho(\mathbf{P}_i\hat{\mathbf{x}}).
\end{aligned}
\tag{5}
$$

Here, the edge mask M is a binary mask indicating pixel locations that we want to apply our color priors; $\mathbf{M}$ is a binary diagonal matrix selecting all pixels in the mask M; $\hat{\mathbf{x}}$ and $\mathbf{y}$ are vector forms of the intermediate image $\hat{x}$ and the blur image y respectively; and $\mathbf{K}$ is the convolution matrix corresponding to the estimated blur kernel k. ∇_h and ∇_v are the matrix forms of the partial derivative operators in horizontal and vertical directions respectively. λ and β are parameters weighting different priors, and $|M|$ is the number of non-zero elements in the mask M. $\mathbf{P}_i$ is a binary extraction operator that extracts the patch centering at the location i with a patch size $w \times w$.

Because of the non-linear term $\rho(\mathbf{P}_i\hat{\mathbf{x}})$, it is difficult to directly optimize Equation (5). Thus, we perform the optimization by alternating the following two steps.

Step 1: *Update* $\mathbf{c}_1$ *and* $\mathbf{c}_2$ *for each patch*

Since we choose to use ρ defined in Equation (4) as the sharp image prior, we cannot use the traditional k-mean algorithm to find $\mathbf{c}_1$ and $\mathbf{c}_2$ as it minimizes ϕ in Equation (3) rather than ρ. For each image patch, by keeping $\hat{\mathbf{x}}$ fixed, we use alternating optimization to iteratively solve the clustering index r_{jk} and the centers $\mathbf{c}_1$ and $\mathbf{c}_2$ to minimize ρ:

1. *Update* r_{jk}: Fix $\mathbf{c}_1$ and $\mathbf{c}_2$, and minimize ρ with respect to r_{jk}. r_{jk} can be solved by finding the closest center to each $\hat{\mathbf{x}}_j$:

$$
r_{jk} = \begin{cases} 1, & \text{if } k = \arg\min_l \|\hat{\mathbf{x}}_j - \mathbf{c}_l\|^2 \\ 0, & \text{otherwise} \end{cases}
\tag{6}
$$

2. *Update* $\mathbf{c}_1$ *and* $\mathbf{c}_2$: Fix r_{jk}, and minimize ρ with respect to $\mathbf{c}_1$ and $\mathbf{c}_2$. Since ρ is non-linear to $\mathbf{c}_1$ and $\mathbf{c}_2$, we use a non-linear solver in MATLAB (`fminunc`) to optimize $\mathbf{c}_1$ and $\mathbf{c}_2$.

We alternate between these two steps until convergence. The solution of non-linear optimization depends on the initial guess. We experimentally found that using the centers found by k-means as the initial solution usually leads to good results. The non-linear minimization in the second step usually converges within two iterations, and the whole process often converges in one iteration. An example of the solution from this step is shown in Figure 1.

Step 2: *Update* $\hat{\mathbf{x}}$

By fixing r_{jk}^i, $\mathbf{c}_1^i$ and $\mathbf{c}_2^i$ for each patch for a pixel i in the edge mask M, we re-write the patch prior $\rho(\mathbf{P}_i\mathbf{x})$ of Equation (5) into the following form:

$$
\rho(\mathbf{P}_i\hat{\mathbf{x}}) = \frac{1}{\left\|\mathbf{c}_1^i - \mathbf{c}_2^i\right\|^2} \left\|\mathbf{P}_i\hat{\mathbf{x}} - \mathbf{z}_i\right\|^2,
\tag{7}
$$

where $\mathbf{z}_i$ is a vector constructed using $\mathbf{c}_1^i$, $\mathbf{c}_2^i$ and r_{jk}^i of the corresponding patch i (see Appendix). As shown in Figure 4, $\mathbf{z}_i$ can be considered as the patch constructed using

only c_1 and c_2, enhancing the contrast of the patch. By writing $\rho(\mathbf{P}_i \mathbf{x})$ in this way, Equation (5) becomes quadratic to $\hat{\mathbf{x}}$. By setting its derivative to zero, $\hat{\mathbf{x}}$ can be updated by solving the following linear system:

$$
\mathbf{M}^T \left(\mathbf{K}^T \mathbf{K} + \lambda \nabla_h^T \nabla_h + \lambda \nabla_v^T \nabla_v \right) \mathbf{M} \hat{\mathbf{x}}
$$
$$
+ \quad \frac{\beta}{|M|} \sum_{i \in M} \left(\frac{1}{\left\| \mathbf{c}_1^i - \mathbf{c}_2^i \right\|^2} \mathbf{P}_i^T \mathbf{P}_i \hat{\mathbf{x}} \right)
$$
$$
= \quad \mathbf{M}^T \mathbf{K}^T \mathbf{M} \mathbf{y} + \frac{\beta}{|M|} \sum_{i \in M} \left(\frac{1}{\left\| \mathbf{c}_1^i - \mathbf{c}_2^i \right\|^2} \mathbf{P}_i^T \mathbf{z}_i \right) \quad (8)
$$

We use the Conjugate Gradient (CG) method to solve the equation.

3.3. k-step

In this step, we optimize the blur kernel k for the given intermediate image $\hat{x}$. To solve k, we minimize the following objective function:

$$
E_k(k) = \sum_{* \in \{h,v\}} \| \nabla_* y - k \otimes (M \odot \nabla_* \hat{x}) \|_2^2 + \gamma \|k\|_1 \quad (9)
$$

where $\odot$ is the component-wise multiply operator. The same as the x-step, we only allow edges in the mask M to participate in the kernel estimation by setting the gradient $\nabla_* \hat{x}$ outside M to zero. We choose Laplacian prior to regularize the kernel k because L_1 prior leads to sparse solutions, and thus avoids using heuristic thresholding to remove noise in the kernel. The L_1 regularized optimization problem can be solved efficiently using Iteratively Reweighted Least Squares (IRLS) method.

3.4. Implementation details

Similar to previous methods, we construct an image pyramid to speed up the convergence of the algorithm and avoid trivial solutions. We down-scale the input blurred image y with a scale factor of $\alpha = 1/\sqrt{2}$ until the corresponding blur kernel size becomes 3×3. For each image scale, we apply four iterations between the x-step and the k-step. For pixel values in the range $[0, 1]$, we set $\gamma = 0.05$ in Equation (9) and linearly decrease λ from 0.2 to 0.1 for Equation (5) at each image scale. As for β, we empirically set it as 0.01. Finally, we choose 5×5 as the patch size.

Figure 5 shows comparisons of the intermediate images recovered by the shock filter [3], the patch prior [17] and the normalized color-line prior. The shock filter over-sharpens edges and generates noise in the intermediate patch. In this example, the best example found by the patch prior [17] cannot delineate the underline edges very well. Our prior enhances edge contrasts of the patches without introducing noise and alternating edge structures.

(a) Intermediate image   (b) Blur kernel

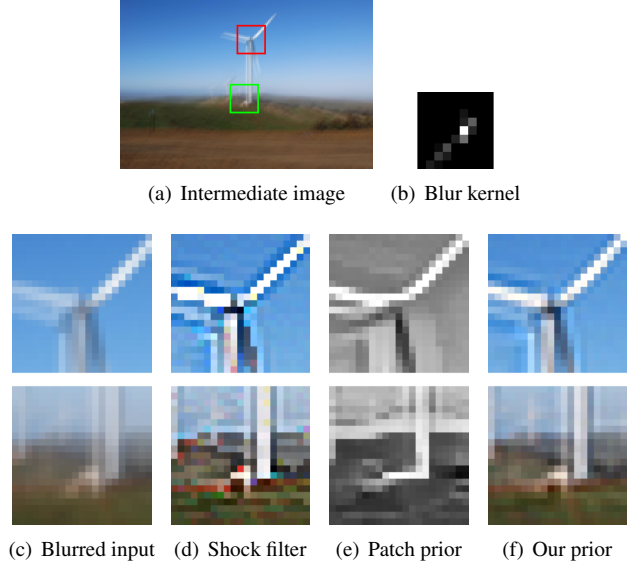(c) Blurred input   (d) Shock filter   (e) Patch prior   (f) Our prior

Figure 5: The intermediate patches constructed using the shock filter [3], the patch prior [17], and our normalized color-line prior. Our color-line prior restores sharp edges without enhancing noise or altering structures.

4. Experiments

In this section, we present comparisons with state-of-the-art blind deconvolution methods [3, 20, 12, 9, 17, 14] and results with real-world photos.

4.1. Quantitative evaluation

To fairly compare with the state-of-the-art algorithms, we tested our algorithm on the synthetic dataset provided by Sun et al. [17] on their website[1]. There are totally 640 blurred images synthesized from the dataset of Sun et al. [19] with 80 images and 8 blur kernels provided by Levin et al. [11]. 1% of Gaussian noise is added to model noise.

However, the dataset [17] only provides gray-scale blurred images, while our algorithm requires color images to estimate blur kernels. Therefore, we synthesize a color version of Sun et al.'s dataset as our input images. Then, we use our blur kernels estimated from color images to deconvolute the gray-scale images of Sun et al.'s dataset. Following the same setting in previous work [17, 14], we assume the size of the blur kernel is 51×51 and apply the final non-blind deconvolution method of Zoran and Weiss [22] to recover latent images.

We obtain the result of Michaeli and Irani [14] from their website[2]. The results of all other competing methods are obtained from website of Sun et al. [17]. Table 1 reports the
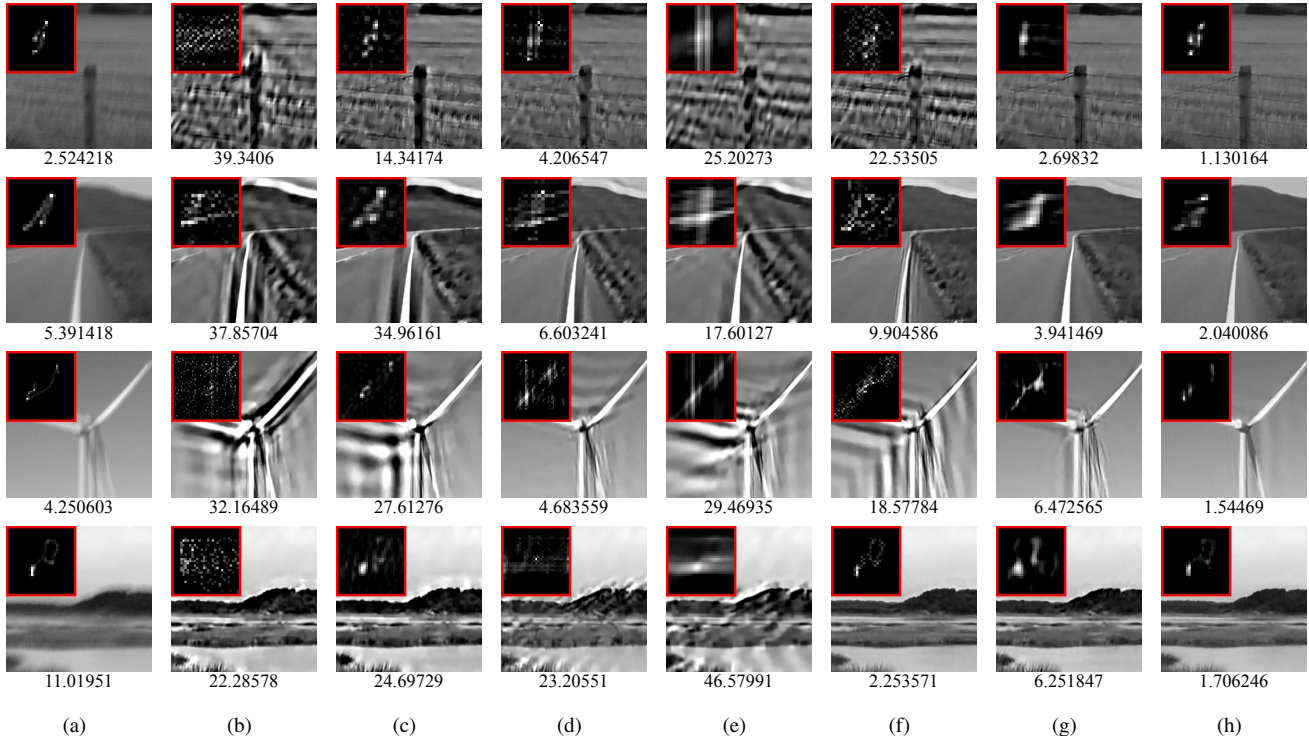

Figure 6: Visual comparisons for four examples with different methods. We show both the estimated kernels and close-ups of the recovered images for each method. The number below each result is its error-ratio r. (a) Blurred input. (b) Cho & Lee [3]. (c) Xu & Jia [20]. (d) Levin *et al*. [12]. (e) Krishnan *et al*. [9]. (f) Sun *et al*. [17]. (g) Michaeli & Irani [14]. (h) Ours.

	PSNR	SSIM	Error Ratio	Success Rate
Blurred	24.7821	0.7888	6.2568	0.4016
Known PSF	32.3409	0.9478	1.0000	-
Cho & Lee [3]	26.2319	0.8824	8.7481	0.6579
Xu & Jia [20]	28.3086	0.9316	3.6102	0.8547
Levin *et al*. [12]	24.9386	0.8706	6.4207	0.4906
Krishnan *et al*. [9]	23.2649	0.8232	11.4853	0.2797
Sun *et al*. [17]	29.5216	0.9400	2.3371	0.9406
Michaeli & Irani [14]	28.6210	0.9249	2.5096	0.9719
Ours	29.6142	0.9376	2.0738	0.9781

Table 1: Quantitative comparisons on Sun *et al*.'s dataset with 640 images [17]. The success rate is the percentage of images with the error ratio ≤ 5.
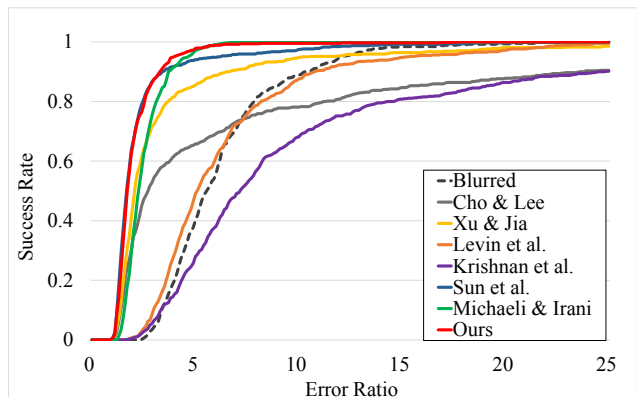

Figure 7: The cumulative distributions of error ratios with different methods. Our method has slightly better performance than Sun *et al*. [17] and Michaeli and Irani [14], but significantly outperforms the other state-of-the-art methods.

average PSNR, average SSIM, average error ratio and the success rate over 640 images for each method. The *success rate* is the percent of images that have an error ratio below a threshold. Following the discussion by Michaeli and Irani [14], we choose the error ratio $r = 5$ as the threshold to define the success rate. Table 1 shows that our method achieves the lowest average error ratio and also the best performance in PSNR and the success rate. It ranks the second in terms of SSIM, only next to Sun *et al*. [17]. Figure 7 shows the cumulative error ratio over the entire dataset for

each algorithm. Our method is the most robust. Figure 6 presents a few visual comparisons of the estimated blur kernels, close-ups for some recovered images, and their corresponding error ratios using different algorithms. Compared with other methods, our method usually suffers from much less ringing artifact and reveals sharper details.
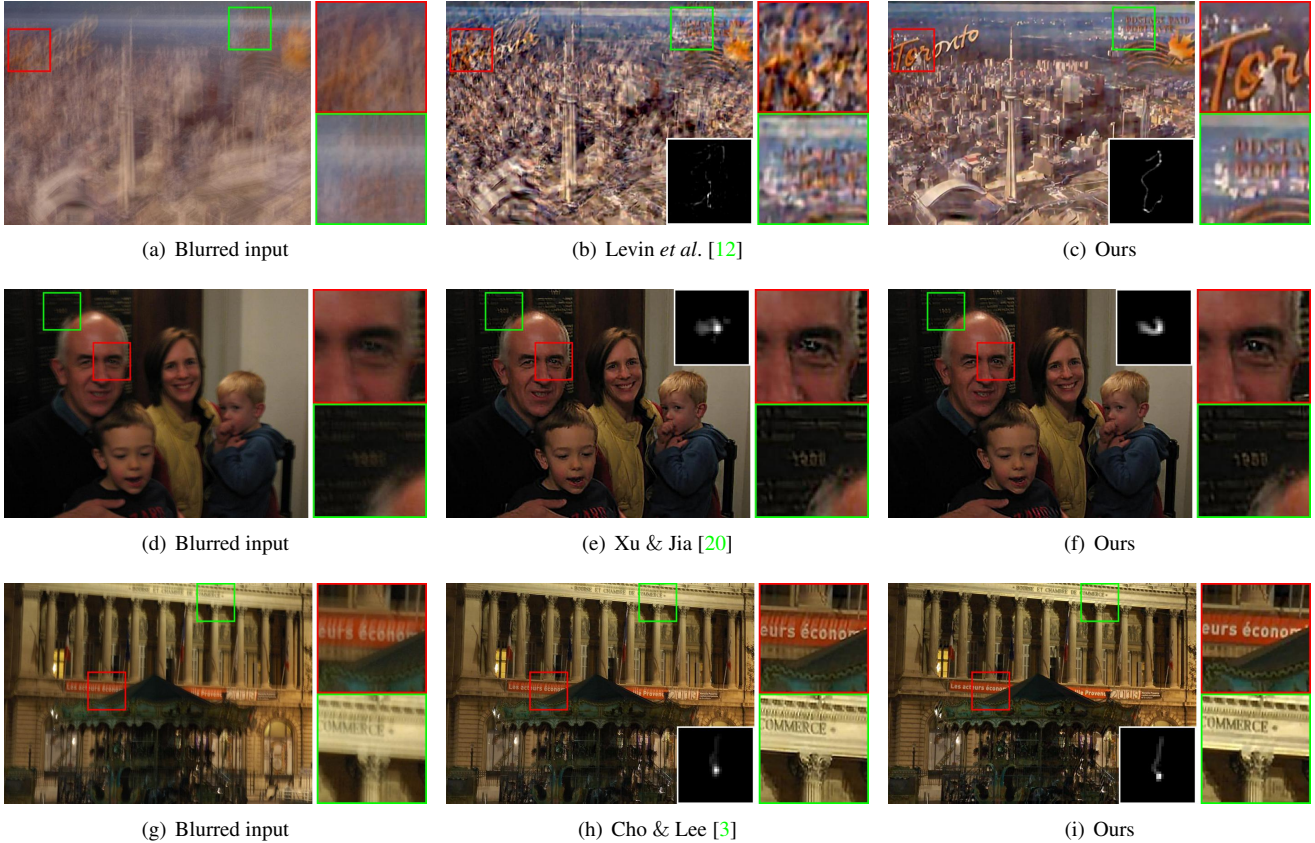
Figure 8: Visual comparisons with Levin *et al*. [12], Cho & Lee [3] and Xu & Jia [20] on real-world photos with unknown camera shake. In general, our method recovers sharper latent images with more details.

4.2. Comparisons on real-world photos

Figure 8 and Figure 9 show deblurring results for some real-world photos. In this part, we also use Zoran and Weiss [22]'s non-blind deconvolution to recover latent images. Since we do not have the ground truth, it is impossible to present quantitative comparisons. Instead, we show visual comparisons with several representative methods with the codes publicly available from authors' websites, including Sun *et al*. [17], Xu & Jia [20], Cho & Lee [3], Michaeli & Irani [14] and Krishnan *et al*. [9]. In general, our method obtains robust blur kernels and suppresses noise, thus revealing more details and resulting in less ringing artifacts in the recovered images.

4.3. Limitations

The main limitation of the proposed method is that it is more suitable for patches containing two primary colors. For regions with more than two dominant colors, such as corners or texture regions, the method cannot generate good results. Figure 10 gives a failure example. Our method fails to obtain a good blur kernel because the input image is full of colorful textures. Although the problem can be mitigated by removing patches that do not fit the model well, in this particular example, too many patches were eliminated and the insufficient number of patches leads to unstable kernel estimation. In addition, our method shares with other edge-based methods [3, 20, 17] the limitation that the input image needs to have enough step edges.

5. Conclusion

In this paper, we have introduced a single-image blind deconvolution method that utilizes the proposed normalized color-line prior for blur kernel estimation. By optimizing the proposed prior, our method gradually enhances the sharpness of image patches without using heuristic filters or external patch priors. Similar to Michaeli and Irani's method [14], our prior is an evolving image-specific prior that changes from iteration to iteration. Experiments shows more robust deblurring results. It would be interesting to extend the color-line model to larger patches and patches with more than two primary colors. We leave this problem for future study.
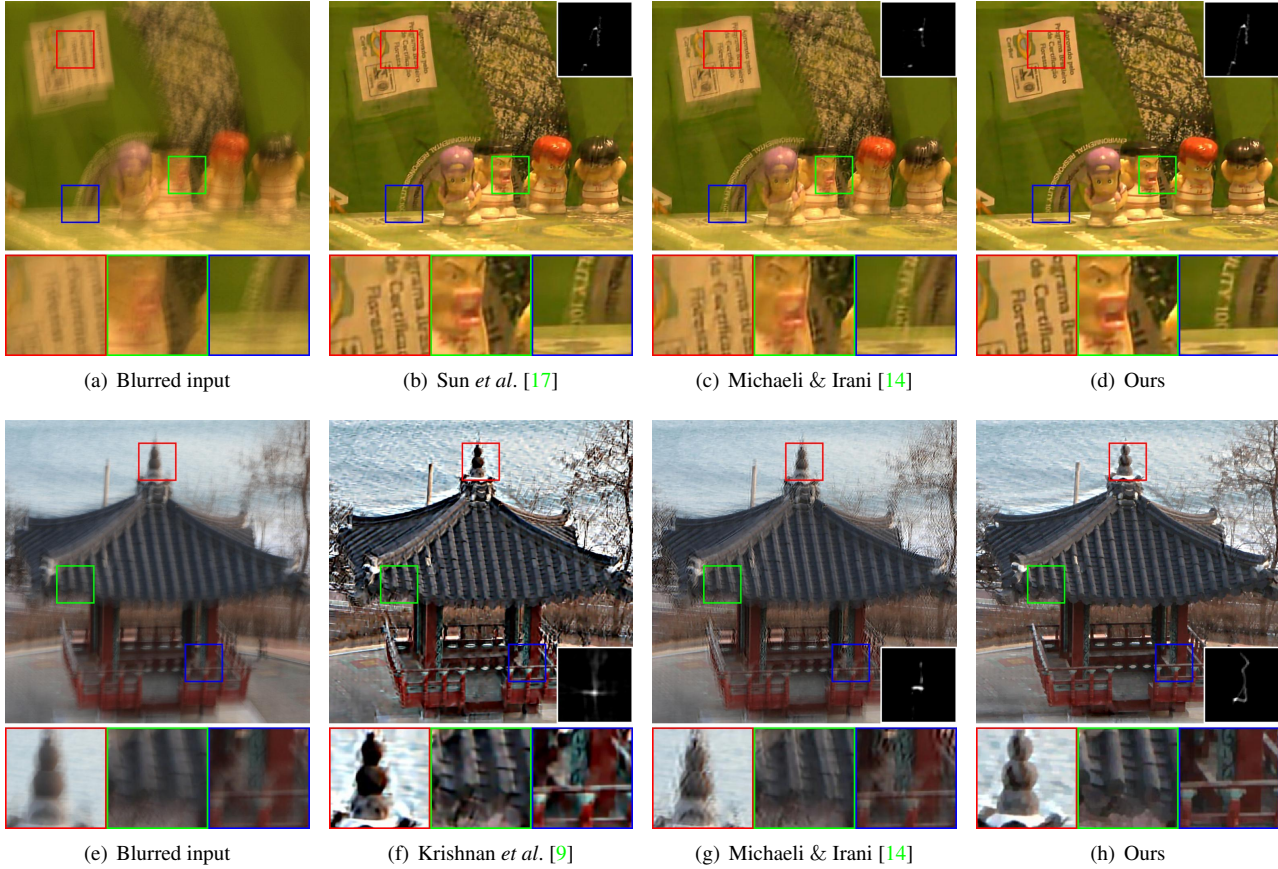
(a) Blurred input   (b) Sun *et al*. [17]   (c) Michaeli & Irani [14]   (d) Ours

(e) Blurred input   (f) Krishnan *et al*. [9]   (g) Michaeli & Irani [14]   (h) Ours

Figure 9: Visual comparisons with Sun *et al*. [17], Krishnan *et al*. [9] and Michaeli & Irani [14] on real-world photos with unknown camera shake. In general, our method is more robust and suffers from less artifact.
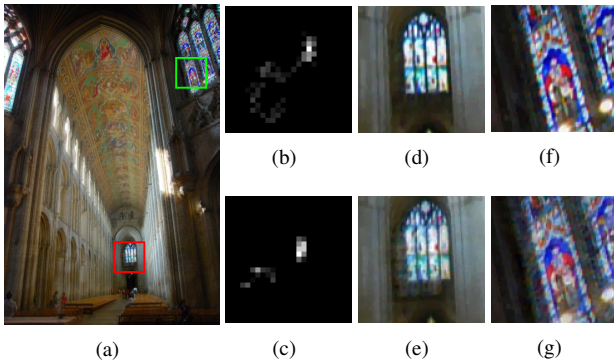

Figure 10: A failure example of our method. Most of the input image contains textured regions. Thus, there are not a sufficient number of patches for estimating blur kernel estimation and the result with our method is worse than Sun *et al*. [17]. (a) Our deblurred result. (b) Sun *et al*.'s kernel. (c) Our kernel. (d)(f) Close-ups of Sun *et al*.'s result. (e)(g) Close-ups of our result.

Appendix

The contrast-enhanced patch $\mathbf{z}_i$ in Equation (7) can be composed in the following way.

$$
\mathbf{z}_i = \begin{pmatrix} 0 \\ \vdots \\ r_{11}^i \mathbf{c}_1^i \\ \vdots \\ r_{N1}^i \mathbf{c}_1^i \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ r_{12}^i \mathbf{c}_2^i \\ \vdots \\ r_{N2}^i \mathbf{c}_2^i \\ \vdots \\ 0 \end{pmatrix}, \quad (10)
$$

where N is the number of pixels in the patch. Each non-zero element in $\mathbf{z}_i$ corresponds to the pixel in the patch $\mathbf{P}_i \hat{\mathbf{x}}$.

Acknowledgement

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. 2

[2] Y. Bando, B.-Y. Chen, and T. Nishita. Extracting depth and matte using a color-filtered aperture. *ACM Transation on Graphics*, 27(5):134:1–134:9, 2008. 2

[3] S. Cho and S. Lee. Fast motion deblurring. *ACM Transation on Graphics*, 28(5):145:1–145:8, 2009. 1, 2, 5, 6, 7

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transation on Graphics*, 25(3):787–794, 2006. 1

[5] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 4

[6] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 349–356. IEEE, 2009. 1

[7] K. He, J. Sun, and X. Tang. Guided image filtering. In *Proceedings of the European Conference on Computer Vision*, pages 1–14, 2010. 2

[8] N. Joshi, C. L. Zitnick, R. Szeliski, and D. Kriegman. Image deblurring and denoising using color priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1550–1557, 2009. 1, 2

[9] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 233–240, 2011. 1, 5, 6, 7, 8

[10] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008. 2

[11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2009. 1, 2, 5

[12] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2657–2664, 2011. 1, 5, 6, 7

[13] T. Michaeli and M. Irani. Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–952. IEEE, 2013. 1

[14] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *Proceedings of the European Conference on Computer Vision*, pages 783–798. Springer, 2014. 1, 5, 6, 7, 8

[15] I. Omer and M. Werman. Color lines: Image specific color representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 946–953, 2004. 1, 2

[16] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3):73, 2008. 1

[17] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *Proceedings of the IEEE International Conference on Computational Photography*, pages 1–8, 2013. 1, 2, 3, 5, 6, 7, 8

[18] L. Sun, S. Cho, J. Wang, and J. Hays. Good image priors for non-blind deconvolution. In *Proceedings of the European Conference on Computer Vision*, pages 231–246. Springer, 2014. 1

[19] L. Sun and J. Hays. Super-resolution from internet-scale scene matching. In *Proceedings of the IEEE International Conference on Computational Photography*, 2012. 5

[20] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proceedings of the European Conference on Computer Vision*, pages 157–170, 2010. 1, 2, 3, 5, 6, 7

[21] L. Xu, S. Zheng, and J. Jia. Unnatural L_0 sparse representation for natural image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1114, 2013. 1

[22] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 479–486, 2011. 1, 5, 7