Deblurring a Camera-Shake image using a Thinning Kernel

Shwu-Huey Yen*, Pin-An Chen, Hwei-Jen Lin

Tamkang University, 151 Ying-Jun Road, New Taipei City, 23157, Taiwan, R.O.C.

*Corresponding Author: 105390@tku.edu.tw

Abstract

The task of blind deblurring usually consists of estimation of interim images and blur kernels. Due to the lack of information in kernels compared to that in interim images, when only a blurred image is available, most of deblurring methods emphasis the estimation of interim images. However, the resulting kernel is often wider than it should be, thus degrading the quality of the deconvolved image. To remedy the problem of wide kernels, we present a thinning scheme to better estimate a kernel. In this way, a clear image can be recovered from a camera-shake blurred image. To mitigate the insufficient information of blur kernels, we make simple inferences and assumptions for kernels based on the trajectory of the camera shake. Under these inferences and assumptions, we use a three-step approach to estimate the blur kernel. Firstly, we relax the condition to find the shape of the blur kernel. Next, we use a thinning algorithm to obtain the skeleton of the blur kernel. Thirdly, we reweight the blur kernel by Gaussian distribution. By repeating these steps a few times we can get a more accurate blur kernel. Finally, we can reconstruct a high quality deblurred image by using the blur kernel. The proposed method is tested by a public database and our results outperform those of two similar methods.

Keywords: blind deconvolution, blur kernel, camera-shake, deblur, interim image.

1. Introduction

Image deblurring has been widely discussed over the past decades⁽¹⁻³⁾. As far as handheld devices, the most common types of blur are defocus blur and motion blur. The latter is the result of camera-shake and object motion. Camera-shake, which we focus on this paper happens because it is difficult to stabilize a camera for a long exposure time. This often leads to camera-shake blurred

images.

Generally, a motion blur is modeled as

$$B = L \otimes K + n, \quad (1)$$

where B is a blur image, L is a latent image, K is a blur kernel, n is a noise, $\otimes$ is a convolution operator. Here, if the blur kernel K is known, we just estimate the latent image L from B and K . We refer to this as non-blind deconvolution. Otherwise, if the blur kernel K is unknown, we estimate both L and K from B . This we call blind deconvolution.

However, blind deconvolution is an ill-posed problem. It is difficult to estimate an accurate combination of both latent image L and blur kernel K with the only information provided by blurred image B , which has almost infinite combinations. In order to solve this ill-posed problem, we usually impose some extra restrictions (through the statistics or the features of the natural image). This process introduces additional information wherefore the problem is solvable. The general form to describe such a restricted problem is given by an objective function as in (2),

$$\min_{L,K} \|L \otimes K - B\|_2^2 + P(L) + P(K), \quad (2)$$

where the first term is the likelihood term, the second term is the image prior term, which restricts the latent image, and the third term is the kernel prior term, which restricts the blur kernel. Most deblurring methods emphasize the estimation of interim images (the second term). However, the resulting kernel is often wider than it should be and degrades the quality of deconvolved images. To overcome this, we present a thinning scheme to better estimate a kernel. Unlike other methods, our approach focuses on the estimation of a good kernel. In this way, a clear image can be recovered from a camera-shake blurred one.

The outline of the paper is as follows. Section 2 introduces the existing research on blind methods and non-blind methods. Section 3 discusses the proposed method followed by the experimental results. Section 5 concludes our paper.

2. Related work

First, we discuss previous research on non-blind deconvolution, where the blur kernel is known, and only the latent image is unknown. The simple and most common methods are the Richardson-Lucy and Wiener filters⁽⁴⁻⁶⁾. The drawback of these two methods is that when the blur kernel is not accurately estimated, severe ringing artifacts occur, and results are easily affected by noise. Alternatively, the heavy-tailed distribution of gradients in natural scenes has proven to be effective when using priors for deblurring. Krishnan and Fergus used Hyper-Laplacian priors to model the gradient distribution of natural images, and further reconstruct latent images⁽⁷⁾. This method has been adopted by many studies to estimate latent images, when the blur kernel has been already determined⁽⁸⁻¹⁰⁾.

Blind deconvolution, where both the blur kernel and latent image are unknown, is closer to a real situation. Existing methods of blind deconvolution typically use an alternative approach. By iterating the estimation of the interim image and blur kernel, finally, the latent image is reconstructed by using the blur kernel. As mentioned above, due to the limited information of kernels, most blind deblurring methods focus on how to estimate the interim image. These methods usually have different priors. The different properties of images can be observed to better reconstruct latent images. Fergus *et al.* used a mixture of the Gaussians model with a heavy-tailed natural image prior⁽¹¹⁾. Shan *et al.* introduced a probabilistic model with a new local smoothness prior, which can suppress ringing artifacts⁽¹²⁾. Cho and Lee predicted salient edges by using a shock filter and bilateral filter, and used a threshold to remove small structures⁽¹³⁾. The prediction gradient map is effective for estimating blur kernels. Krishnan *et al.* proposed a normalized sparsity measure which gives the lowest cost for a true sharp image⁽⁸⁾. The normalized sparsity measure is used to replace ℓ_0 measure, because ℓ_0 measure is difficult to optimize. Komodakis and Paragios showed an MRF-based prior which utilizes the concept that the blurred images generated from a latent image and the quantized versions constituted by using a MRF prior are visually quite similar⁽¹⁴⁾. Xu *et al.* proposed an ℓ_0 gradient minimization method which can effectively smooth images and optimize the objective function⁽¹⁵⁾; therefore, based on this method, Pan *et al.* and Xu *et al.* showed a prior by using ℓ_0 sparsity⁽⁹⁻¹⁰⁾. All of the above methods focus on

Fig. 1. The influence of the blur kernel on images.

how to estimate the interim image, and provide an easy prior to limit the blur kernel, since the kernel information is completely unknown.

3. Proposed Method

In the proposed method, we explore the model of a blur kernel and make some assumptions. Our approach includes three main parts. The first part is just like other existing methods, i.e., we use a prior to estimate the interim image, and find the shape of the blur kernel in a simple way. In the second part, we use a thinning kernel algorithm to obtain the skeleton of blur kernel. In the third part, we reweight the blur kernel by the Gaussian smoothing function. Finally, we can reconstruct a high quality deblurred image by using the blur kernel.

3.1 Assumptions of blur kernel

It is unknown that the blur kernel looks like, so we make a simple inference. Fig. 1 illustrates the influence of kernel convolution on images. The images on the top show that a sharp image can be expressed as the sharp image convolves with a delta kernel which has a center value of one and zero otherwise. Other words, the delta kernel can be seen as a light which the exposure focuses on. However, as shown on the bottom, when the shutter speed is slow compared to the speed of unintended camera shaking, we get blurry images. The shaking trajectory with light generates a blurring kernel, thus producing a blurred image. Based on this inference, we assume that (a) the shape of kernel is thin and continuous since it is generated by the shaking trajectory, (b) the pixel values of the blur kernel are between 0 and 1, and sum to 1, and (c) the blur kernel is sparse.

3.2 Estimation of interim image

Next, we want to find an interim image, which only

has strong edges. Because the strong edges are likely to exist in the latent image, the edge information can be used to estimate the correct blur kernel. Here, we use an ℓ_0 sparse method like Xu's^(10,15), in which the objective function is

$$\min_x \{ \|x \otimes k - y\|_2^2 + \lambda \|\nabla x\|_0 \}, \quad (3)$$

where x is the interim image, k is the blur kernel, y is the blurred image, $\otimes$ is the convolution operator, and λ is the weight. ∇ denotes the gradient operation. The first term is the likelihood term such that the interim image convolved by the blur kernel should be similar to the blurred image. The second term is the prior term of the image to encourage the interim image to be smooth and only have strong edges. By the loss function in Xu's⁽¹⁰⁾, (3) can be rewritten as

$$\min_{x,l} \left\{ \frac{1}{\lambda} \|x \otimes k - y\|_2^2 + \sum_{i \in \{h,v\}} \sum_i (|l_{*i}|^0 + \frac{1}{\varepsilon^2} (\partial_i x_i - l_{*i})^2) \right\}, \quad (4)$$

where l denotes an auxiliary variable, h and v are the horizontal and vertical directions, $|l_{*i}|^0 = 1$ if $l_{*i} \neq 0$ and zero otherwise, and ε is the weight which controls the optimization of ℓ_0 . The detailed derivations are referred to the supplementary document⁽¹⁶⁾.

To solve this half-quadratic function, we alternate between computing l and x by

$$l_{*i} = \begin{cases} 0, & \text{if } |\partial_{*i} x_i| \leq \varepsilon \\ \partial_{*i} x_i, & \text{otherwise} \end{cases}, \quad (5)$$

$$x = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(k)} \cdot \mathcal{F}(y) + \frac{\lambda}{\varepsilon^2} (\overline{\mathcal{F}(\partial_h)} \cdot \mathcal{F}(l_h) + \overline{\mathcal{F}(\partial_v)} \cdot \mathcal{F}(l_v))}{\overline{\mathcal{F}(k)} \cdot \mathcal{F}(k) + \frac{\lambda}{\varepsilon^2} (|\mathcal{F}(\partial_h)|^2 + |\mathcal{F}(\partial_v)|^2)} \right), \quad (6)$$

where (5) is a fast pixel-wise closed form solution for the ℓ_0 sparse loss function^(10,16). Equation (6) is solved by using a Fast Fourier Transform (FFT) and working directly in the frequency domain. This saves a lot of time but sometimes introduces ringing artifacts.

3.3 Estimation of the primitive blur kernel

We use the interim image obtained above to estimate the blur kernel. However, not all the strong edges are suitable to estimate a blur kernel. Xu and Jia claimed that if the scale of an object is smaller than that of the blur kernel, the edge information could mislead kernel estimation⁽¹⁷⁾. To avoid this ambiguity, we calculate a confident mask to select the informative edges for kernel estimation⁽¹⁷⁾. Firstly, we calculate the confidence values of each pixel by

$$C(x_p) = \frac{\left\| \sum_{n \in N(x_p)} \nabla B(n) \right\|_2}{\sum_{n \in N(x_p)} \|\nabla B(n)\|_2 + 0.5}, \quad (7)$$

where x_p is the index of each pixel in the interim image, $N(x_p)$ is a window centered at x_p with the size similar to the kernel's, and 0.5 is to prevent producing a large $C(x_p)$ in flat regions. After obtaining the confidence values, we use a threshold τ_c to generate a mask with $M = H(C - \tau_c)$, where $H(k)$ is the Heaviside step function, which is zero if k is negative, and one otherwise. Finally, the selected edges for kernel estimation are determined as

$$\nabla x \leftarrow M \odot \nabla x, \quad (8)$$

where $\odot$ is an element wise multiplication operator.

Contrary to the interim image that has some information from the blurred image, the shape of the blur kernel is completely unknown. We impose a sparse prior in estimating the kernel. We define the objective function as

$$\min_k \|\nabla x \otimes k - \nabla y\|_2^2 + \gamma \|k\|_2^2, \quad (9)$$

where γ is a weight. Using the quadratic function, we can quickly get the preliminary shape of the blur kernel. Equation (9) can also be solved very efficiently by using FFT as follows

$$k = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\partial_h x)} \cdot \mathcal{F}(\partial_h y) + \overline{\mathcal{F}(\partial_v x)} \cdot \mathcal{F}(\partial_v y)}{|\mathcal{F}(\partial_h x)|^2 + |\mathcal{F}(\partial_v x)|^2 + \gamma} \right). \quad (10)$$

The estimation of the blur kernel is very sensitive to noise. To relax the sensitivity, based on our assumption given before, we use the following four steps as a post-process to modify the blur kernel: (1) set a pixel value to be zero if its pixel value is negative, let it equal zero; (2) if the pixel value is less than 1/20 of the maximum value of the blur kernel, let it equal zero; (3) clustering pixels via connected components, discard a group if the sum of the pixel values is smaller than 0.1, let the pixels of the group equal zero; and (4) normalize the blur kernel so that the sum to be one.

Figure 2 describes the algorithm of preliminary estimation of the blur kernel and interim image. The Algorithm is carried out in a coarse-to-fine manner using an image pyramid⁽⁹⁻¹¹⁾. The goal is to estimate the overall shape of the kernel by using loose restrictions.

3.4 Thin kernel estimation

Due to the lack of information of the blur kernel, we use quite loose conditions to estimate the blur kernel in the first part of our system. From the assumption that the trajectory of camera shake is thin and continuous, the

Algorithm 1

Input: blurred image B
Output: blur kernel K , interim image x

1. Initial: $k^1 = \text{delta function}$, $x^1 = y^1 = B \downarrow \sqrt{2^{Scales}}$
2. **for** $S = 1 : Scales$
3.   **for** iteration = 1 : T_1
4.   //update interim image
5. $\varepsilon = 1$
6.   **for** $i = 1 : 4$
7.   **for** $j = 1 : \varepsilon^{-1}$
8.   solve for l by (5)
9.   solve for x^{S+1} by (6)
10.   **end** $\{j\}$
11. $\varepsilon \leftarrow \varepsilon/2$
12.   **end** $\{i\}$
13.   //update blur kernel
14.   solve for ∇x^{S+1} by (8)
15.   solve for k^{S+1} by (10)
16.   **end** $\{\text{iteration}\}$
17.   //up-sample
18. $y^{S+1} \leftarrow B \downarrow \sqrt{2^{Scales - S}}$
19.   up-sample x^{S+1} , k^{S+1} by bilinear interpolation
20. **end** $\{S\}$
21. $x = x^{Scales+1}$, $K = k^{Scales+1}$

Fig. 2. Preliminary estimation of blur kernel K and interim image x .

estimated preliminary kernel should be refined accordingly. We propose a method consisting of thinning and Gaussian reweighting to refine the estimated kernel.

3.4.1 Thinning step

We use a simple thinning algorithm to find the skeleton of the kernel⁽¹⁸⁾. This algorithm applies on the kernel obtained in Algorithm 1. The resulted skeleton of the kernel serves as the information to estimate a weight mask in the next step.

Firstly, a binary kernel map is created such that it would be one if it were non-zero in the estimated kernel and zero otherwise. Then, for those pixel $P(i, j) = 1$, named it as P_1 and its eight neighbors starting from the one right above it as P_2 , followed by P_3 to P_9 clockwise, we use the following four rules to determine whether P_1 should be thinning (deleted). If P_1 fulfills all four rules (a)-(d), it will be deleted. Otherwise, P_1 is a skeleton pixel and will be preserved

$$(a) 2 \leq N(P_1) \leq 6$$

$$(b) A(P_1) = 1$$

$$(c) P_2 * P_4 * P_6 = 0 \quad \text{or} \quad (c') P_2 * P_4 * P_8 = 0$$

$$(d) P_4 * P_6 * P_8 = 0 \quad \text{or} \quad (d') P_2 * P_6 * P_8 = 0$$

where $N(P_1)$ denotes the number of non-zero pixels around P_1 , $A(P_1)$ denotes the number of 01-pattern around P_1 . The

Table 1. Lookup table of the Gaussian mask W
(d: the distance and std is the standard deviation)

std d	2.50	2.00	1.75	1.50	1.25	1.00	0.85
0	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1	0.923	0.882	0.849	0.801	0.726	0.607	0.501
2	0.726	0.607	0.520	0.411	0.278	0.135	0.063
3	0.487	0.325	0.230	0.135	0.056	0.011	0.002
4	0.278	0.135	0.073	0.029	0.006	0.000	0.000
5	0.135	0.044	0.017	0.004	0.000	0.000	0.000

thinning algorithm is implemented alternatively between rules (a)-(d) and rules (a), (b), (c') and (d'). It is because (c) and (d) only remove the south-east boundary points or north-west corner points, and (c') and (d') can remove the north-west boundary points or south-east corner points. The iterations continue until no more points can be removed.

3.4.2 Reweighting step

The weight mask is designed to have large weight for a pixel if it is close to a skeleton pixel. For every skeleton pixel, we take the gradient direction and reweight it using a Gaussian distribution. The weight may be rapidly solved with the lookup table (LUT) shown in Table 1. In this way, we can efficiently obtain a weight mask W and using this weight mask we can reweight the blur kernel as

$$K \leftarrow W \odot K, \quad (11)$$

where $\odot$ is an element wise multiplication operator.

3.4.3 Estimating step

Finally, using the reweight kernel of (11) to estimate a more accurate blur kernel, our objective function is

$$\min_K \|\nabla L \otimes K - \nabla B\|_2^2 + \tau \|K\|_1, \quad (12)$$

where τ is a weight. We apply the well-known Alternating Direction Method of Multipliers (ADMM) method⁽¹⁴⁾, which is essentially a variant of the Augmented Lagrangian method and allows us to efficiently optimize the half-quadratic problem through an auxiliary variable K' . By using the auxiliary variable K' , (12) can be rewritten as

$$\min_{K, K'} \|\nabla L \otimes K' - \nabla B\|_2^2 + \tau \|K'\|_1 + \frac{b}{2} \|K' - K - a_i\|_2^2, \quad (13)$$

$$a_{i+1} = a_i - (K'_{i+1} - K_{i+1}), \quad (14)$$

where a is a transformed vector of Lagrange multipliers, and b is a penalty weight. We use FFT to solve K'_{i+1} then a soft shrinkage operator to solve K_{i+1} :

$$K'_{i+1} = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\partial_h x)} \cdot \mathcal{F}(\partial_h y) + \overline{\mathcal{F}(\partial_v x)} \cdot \mathcal{F}(\partial_v y) + \frac{b}{2} (\mathcal{F}(K) + \mathcal{F}(a))}{|\mathcal{F}(\partial_h x)|^2 + |\mathcal{F}(\partial_v x)|^2 + \frac{b}{2}} \right), \quad (15)$$

$$K_{i+1} = \text{soft}_{\tau/b}(K'_{i+1} - a_i), \quad (16)$$

where $\text{soft}_{\tau}(x) = \max(|x| - \tau, 0) \cdot \text{sign}(x)$. Therefore, we can achieve optimization via an iterative way. An example of the thin kernel estimation is shown in Fig. 3.

3.5 Latent image estimation

We use the blur kernel obtained via the thinning kernel step to estimate the latent image. Here, if we use the ℓ_0 regularization method as in Subsection 3.2, it will lead to a lack of texture and detail in the image. Due to the property of sparseness, it only has strong edges and other details are ignored. Instead, we use the method adopted in Krishnan *et al.* with the objective function of (17) where λ is a weight⁽⁷⁾. Likewise, it is a half-quadratic function, so we solve it by using a variable w , where the equation is rewritten as (18). To solve variable w , we also use a soft shrinkage operator, which is efficient and simple. To solve latent image L , we use FFT as follows in (19).

$$\min_L (L \otimes K - B)^2 + \lambda \|\nabla L\|_1, \quad (17)$$

$$\min_{L, w} (L \otimes K - B)^2 + \beta (\|\partial_h L - w_h\|^2 + \|\partial_v L - w_v\|^2) + \lambda (|w_h|_1 + |w_v|_1) \quad (18)$$

$$L = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(K)} \cdot \mathcal{F}(B) + \frac{\lambda}{\beta} (\overline{\mathcal{F}(\partial_h)} \cdot \mathcal{F}(w_h) + \overline{\mathcal{F}(\partial_v)} \cdot \mathcal{F}(w_v))}{\overline{\mathcal{F}(K)} \cdot \mathcal{F}(K) + \frac{\lambda}{\beta} (|\mathcal{F}(\partial_h)|^2 + |\mathcal{F}(\partial_v)|^2)} \right). \quad (19)$$

Algorithm 2 summaries the steps for image deblurring via a thinning kernel process. The goal is to estimate a more accurate kernel by adding new restrictions and reconstruct a high quality deblurred image by using the blur kernel.

4. Experimental Results

The proposed method is tested by a public database,

Fig. 3. (a) The kernel shape from Alg.1. (b) Thinning algorithm is applied. (c) The weight mask obtained by Gaussian distribution. (d) The reweight kernel obtained by (a) and (c). (e) The final kernel obtained by the proposed method. (f) The ground truth kernel.

Algorithm 2

Input: blur kernel K and interim image x (from Alg. 1)
Output: Latent image L , blur kernel K

1. Initial: $L^1 = x, K^1 = K$
2. **for** $t = 1 : T_2$
3. // estimate a better and thinner blur kernel K
4. thin K^t by steps in Section 3.4.1
5. reweight K^t by steps in Section 3.4.2
6. estimate K^{t+1} by steps in Section 3.4.3
7. // update Latent image L
7. solve for L^{t+1} by (19)
8. // check for stop criteria
9. **if** $\frac{\|K^{t+1} - K^t\|}{\|K^t\|} \leq \epsilon_K$ **then** break
10. **end** {t}
11. $L = L^{t+1}, K = K^{t+1}$

Fig. 4. Image deblurring by thinning kernel.

provided by Levin *et al.*⁽³⁾, of 32 blurred images produced by 4 images and 8 different blur kernels shown in Fig. 5, where the size of the images is 255 by 255, and the various sizes of the blur kernel range from 13 by 13 to 27 by 27.

To demonstrate the effectiveness of the estimated blur kernels, we use error ratios to measure it,

$$\text{error ratios} = \|I_r - I_g\|^2 / \|I_{k_g} - I_g\|^2, \quad (20)$$

where I with subscripts g, r and k_g are the ground truth image, the reconstructed image by the proposed method and by the ground truth kernel, respectively. The smaller the error ratios the better. Empirically, error ratios above 2 are already visually implausible⁽³⁾. The following results, Fig. 6, are obtained from the average of 32 blurred images.

Shown in Fig. 6(a), 68.75% of the results our method takes fall under the error ratio of 1.5. Compared to other state-of-the-art methods, shown in Fig. 6(b), our method is the best, with 93.75% of the results falling under the error ratio of 2. Due to incomplete accessibility of the source codes, we take the error ratios of other methods of Xu's⁽¹⁰⁾.

Next, we compare our results with Pan *et al.*⁽⁹⁾ and Xu *et al.*⁽¹⁰⁾, who used a similar ℓ_0 regularization method to estimate the interim image. However, they only used the

Fig. 5. Ground truth images (top) and blur kernels (bottom) in the database⁽³⁾.

Fig. 6. (a) Error ratios of the proposed method. (b) Comparison with other famous methods (modified from Xu's⁽¹⁰⁾). The horizontal axis shows error ratios and the vertical axis shows the percentage of testing images falling under the indicated error ratio.

relaxed restrictions for blur kernel estimation. In Fig. 7, we can see that because Pan *et al.*⁽⁹⁾ and Xu *et al.*⁽¹⁰⁾ used a similar method for estimation of both the interim image and blur kernel, their reconstructed images and blur kernels are very similar. Comparing the reconstructed kernels in (b) and (c) to the ground truth in (e), the estimated blur kernels are wider than the ground truth kernel. The proposed method, in contrast, can recover the blur kernel correctly using the thinning algorithm as indicated in (d). Examining the second row of Fig. 7, although the blur kernel estimation of the proposed method is more similar to the ground truth kernel, more serious ringing artifacts appear in the boundary of the image. The main point is that when using FFT to estimate the latent image, it is necessary to extend and fill the image in order to meet the period in the frequency domain. The proposed method uses a simple mirroring method to pad it, while other approaches use optimization methods. In the third row, we show a failure example. The ground truth kernel in (e) indicates that the main light of exposure is at the lower-left corner. However, we see an unnatural circular trajectory on the upper-right side. In the proposed algorithm, under the assumption of natural shake trajectory, we take the delta kernel as the initial kernel. However, with the unnatural shake trajectory,

(a) Input (b) Pan^[9] (c) Xu^[10] (d) Ours (e) G. T.
Fig. 7. Comparison with others using ℓ_0 regularization to estimate the interim image. The insets on the lower left of the images indicate the blur kernels. Compare estimated kernels of (b), (c), (d), and the results indicate that our method can resolve the wide kernel problem and get better estimation of blur kernels. However, the proposed method may cause ringing artifacts appear in the image border, as in (d) of the second row when implementing FFT.

the delta function is not an appropriate initial kernel, because it produces error estimation for the overall structure of the kernel. When the recovered kernel is not correct, the reconstructed image is doomed to fail.

5. Conclusions

Blind deblurring methods usually consist of estimation of interim images and estimation of blur kernels. Since there is little information in kernels compared to that in interim images, when only a blurred image is available, most deblurring methods emphasize the estimation of interim images. Sometimes, these methods obtain a wider blur kernel, thus affecting the quality of the reconstructed deblurred image. In this study, to deblur camera-shake blurring images, we focus on how to estimate a good kernel.

We make simple inferences and assumptions for blur kernels based on the trajectory of the camera shake and propose a thinning kernel method for better estimation. The method has three main steps. Firstly, we relax the condition to find the overall shape of the blur kernel. Next, we use a thinning algorithm to obtain a skeleton of the blur kernel. Thirdly, we reweight the blur kernel through Gaussian distribution. By repeating these steps a few times we can get a more accurate blur kernel. Finally, we can reconstruct a high quality deblurred image by using the blur kernel. In addition, our method can be used for post-processing in the kernel estimation of other existing methods.

However, our method is not suitable when the shake trajectory is unnatural. The boundary problem also influences the reconstruction quality of latent images. In the future, we will study the problem of kernel size estimation. This is an important that has received little attention.

Acknowledgment

The authors would like to thank the Ministry of Science and Technology (MOST) of R.O.C. financially supporting this research under the grand number MOST 106-2221-E-032-053.

References

- (1) R. Wang, and D. Tao. "Recent Progress in Image Deblurring." arXiv preprint arXiv:1409.6838, 2014.
- (2) I. M. El-Henawy, A. E. Amin, K. Ahmed and H. Adel. "A comparative study on image deblurring techniques," *Int. J. Adv. in Comput. Sci. Technol. (IJACST)*, vol.3, no.12, pp. 1-8. Special Issue of ICCEET 2014.
- (3) A. Levin, Y. Weiss, F. Durand and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. Comput. Vision Pattern Recognit. (CVPR 2009)*, pp. 1964-1971, 2009.
- (4) W. H. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Am. (JOSA)*, vol. 62, no. 1, pp. 55-59, 1972.
- (5) L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.*, vol. 79, no. 6, pp. 745-754, 1974.
- (6) N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, Cambridge, MA: Technol. Press of MIT, and New York, Wiley, 1949.
- (7) D. Krishnan and R. Fergus. "Fast image deconvolution using hyper-Laplacian priors," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS 2009)*, pp. 1033-1041, 2009.
- (8) D. Krishnan, T. Tay and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. Comput. Vision Pattern Recognit. (CVPR 2011)*, 2011.
- (9) J. Pan, Z. Hu, Z. Su and M. Yang, "Deblurring text images via L_0 -regularized intensity and gradient prior," in *Proc. Comput. Vision Pattern Recognit. (CVPR 2014)*, 2014.
- (10) L. Xu, S. Zheng and J. Jia, "Unnatural L_0 sparse representation for natural image deblurring," in *Proc. Comput. Vision Pattern Recognit. (CVPR 2013)*, pp. 1107-1114, 2013.
- (11) R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graphics (Proceedings of ACM SIGGRAPH 2006)*, vol. 25, no. 3, pp 787-794, 2006. doi>[10.1145/1141911.1141956](https://doi.org/10.1145/1141911.1141956)
- (12) Q. Shan, J. Jia and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graphics (Proceedings of ACM SIGGRAPH 2008)*, vol. 27, no. 3, 2008. doi>[10.1145/1360612.1360672](https://doi.org/10.1145/1360612.1360672)
- (13) Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graphics (Proceedings of SIGGRAPH Asia 2009)*, vol. 28, no. 5, 2009. doi>[10.1145/1618452.1618491](https://doi.org/10.1145/1618452.1618491)
- (14) N. Komodakis and N. Paragios, "MRF-Based blind image deconvolution," in *Proc. 11th Asian Conf. Comput. Vision (ACCV 2012)*.
- (15) L. Xu, C. Lu, Y. Xu and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graphics (Proceedings of ACM SIGGRAPH Asia 2011)*, vol. 30, no. 6, 2011. doi>[10.1145/2070781.2024208](https://doi.org/10.1145/2070781.2024208).
- (16) L. Xu, S. Zheng and J. Jia, "Unnatural L_0 sparse representation for natural image deblurring supplementary material," available in http://www.cse.cuhk.edu.hk/~leojia/projects/10deblur/10deblur_supp.pdf.
- (17) L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vision (ECCV 2010)*, pp. 157-170, 2010.
- (18) T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 239-242, 1984.