

Blind Image Deblurring with Modified Richardson-Lucy Deconvolution for Ringing Artifact Suppression

Hao-Liang Yang, Yen-Hao Chiao, Po-Hao Huang, and Shang-Hong Lai

Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
lai@cs.nthu.edu.tw

Abstract. In this paper, we develop a unified image deblurring framework that consists of both blur kernel estimation and non-blind image deconvolution. For blind kernel estimation, we propose a patch selection procedure and integrate it with a coarse-to-fine kernel estimation algorithm to develop a robust blur kernel estimation algorithm. For the non-blind image deconvolution, we modify the traditional Richardson-Lucy (RL) image restoration algorithm to suppress the notorious ringing artifact in the regions around strong edges. Experimental results on some real blurred images are shown to demonstrate the improved efficiency and image restoration by using the proposed algorithm.

1 Introduction

Motion blur is caused by relative motion between the camera and the scene during exposure. The real camera motion is usually too complicated to estimate from a blurred image when it involves camera rotation or large scene depth variations. To simplify the problem formulation, previous researches usually assumed the camera motion is perpendicular to the optical axes and the effect of scene depth variation can be neglected. In other words, the blur kernel is assumed to be spatially invariant. Under this assumption, a blurred image, B , can be modeled as (1), where K is the blur kernel, I is the clear image, N is the noise, and \otimes is the convolution operator.

$$B = I \otimes K + N. \quad (1)$$

The blind image restoration problem in (1) is ill-posed because I and K are highly under-constrained and there are infinitely many possible combinations of I and K such that their convolution is equal to the blurred image B . Fergus et al. [4] proposed to utilize ensemble learning to estimate the blur kernel with a sophisticated variational Bayes inference algorithm, which employs the property of specific distributions of image gradients for natural images to approximate the posterior distribution. Levin [6] also exploited image statistics for estimating blur kernels. Shan et al. [9] proposed two probabilistic models to improve image restoration. One is to model the spatially random distribution of noise, and the other is a smoothness prior model which can reduce the ringing artifacts. Cho and Lee [11] proposed a latent image prediction step, which applied shock filter to recover the sharp edge information for estimating the blur kernel.

Even with a known blur kernel, the restored image may contain some undesirable reconstruction artifacts, such as the ringing artifacts. To overcome this problem, Levin et al. [7] modeled the sparse image derivative distribution as a heavy-tailed function to alleviate the ringing artifacts. Shan et al. [9] proposed a local smoothness prior which assumes the gradients of smooth regions in a blurred image are similar to those in a clear image. Yuan et al. [8, 10] proposed the concept of residual deconvolution and modified the standard Richardson-Lucy (RL) algorithm [1], [2] by incorporating either a gain-control process [7] or a bilateral-filtering-like process [9] for suppressing the ringing artifacts.

In this paper, we propose a simple and efficient method to estimate the blur kernel under the assumption of spatially-invariant case, and modify the Richardson-Lucy algorithm to form a new deconvolution method called GARL, which effectively reduces the ringing artifact. The contributions of this paper are listed as follows:

1. We propose a patch selection scheme to choose a suitable region from the input blurred image for kernel estimation with the purpose of computational efficiency.
2. Combining several popular concepts in blind kernel estimation, we develop our method using a quadratic smoothness prior, bilateral filtering, and a good patch.
3. We exploit the gradient attenuation concept and modify the standard RL algorithm to suppress the ringing artifacts in the RL-based image deconvolution.
4. We propose an iterative detail recovery procedure that can recover missing details due to ringing suppression.

The rest of this paper is organized as follows: The blur kernel estimation algorithm is introduced in section II. Non-blind image deconvolution method is proposed in section III. Experimental results are reported in section IV. Finally, we conclude in section V.

2 Blur Kernel Estimation

Several recent researches have proposed novel and effective ways to estimate the blur kernel. We integrate some of these methods and add other new procedures to form our own kernel estimation method, of both efficiency and accuracy, as depicted in Fig. 2.

In [11], the authors minimized the objective function with a quadratic regularization term using conjugate gradient method. They also adopt bilateral filtering [3] to filter out possible noises in the latent image in their framework. Their algorithm is simple, straightforward, and efficient. In [12], a new metric to measure the usefulness of image edges in motion deblurring is proposed. They found that some regions in an image are good for kernel estimation, while some are not. As a result, they construct a map telling which parts of the image are useful. During the kernel estimation, this map is used as a mask so that only the useful parts are taken into consideration.

We take the advantages of the above methods in our kernel estimation algorithm, including quadratic objective function, bilateral filtering, and the map of useful gradients. In addition, we add a new procedure – patch selection. Patch selection has

been commonly used in deblurring to reduce the execution time in blur estimation, but few researchers have put efforts to analyze it. An image of high resolution, say, a million pixels, takes a long time to be processed, so most of the time we choose one or more small patches for the blur kernel estimation instead of using the entire image. If the patch is selected well, an accurate kernel can still be found in a relatively short time. A good patch for deblurring should contain strong edges of various directions. An edge that is parallel to the blur direction provides no information, but an edge perpendicular to the blur direction is the most appropriate one.

To automatically select the patch, we adopt the concept from the Harris corner detector [13], which determines if a pixel is a good corner from the eigenvalues of the gradient covariance matrix for a local neighborhood. For a pixel \mathbf{p} , a 2x2 Harris matrix \mathbf{C} is defined as

$$\mathbf{C}(\mathbf{p}) = \sum_{(i,j) \in W_p} \begin{bmatrix} I_x^2(i,j) & I_x(i,j)I_y(i,j) \\ I_x(i,j)I_y(i,j) & I_y^2(i,j) \end{bmatrix}, \quad (2)$$

where W_p is a local window centered at \mathbf{p} , and I_x and I_y represent the partial derivatives along x and y directions, respectively. The corner response function is then defined by

$$R = \det(\mathbf{C}) - k(\text{trace}(\mathbf{C}))^2, \quad (3)$$

where k is a constant. A high response value at pixel \mathbf{p} means that the region around \mathbf{p} is probably a corner region because it contains a set of image gradients with diverse directions in a local region. In our patch selection, we apply the same strategy, but the window size is set to the patch size, which is an input to our program. Generally, the edge length of the patch is set to be 1/3 to 1/2 of the input blurry image. The patch size is chosen not too small for stability consideration. Nevertheless, very small patch sizes may work well in some cases and it can considerably reduce the execution time.

Fig. 1 shows some examples of synthetic data. We compare the estimated kernels computed from the whole image and some patch selection methods, including the (1) proposed method, (2) point with maximum gradient magnitude (3) central point. Some times (1) and (2) generate similar patches because corners usually have high gradient magnitude. The advantage of using (3) is that the salient part of an image usually lies in the center, and this scheme doesn't need any additional computation. Using the whole image produces the best result most of the time, but the execution time may be too long for a large image. With this effective patch selection scheme, we can find an appropriate window of a pre-selected size for the blur kernel estimation.

2.1 Multi-scale Scheme

To handle large blurs, a multi-scale optimization strategy is imperative. Generally, the initial kernel size is defined by the user. Thus this value would be easy to set if we start from a small scale. Besides, large blurs are more probable to have complex kernels than small ones. A multi-scale scheme improves the robustness and accuracy of kernel estimation.

2.2 Optimization for K

Most kernel estimation methods do not use the whole image information because there are too many noises or redundant regions that are of no use for kernel estimation. Therefore, we usually extract the edges out first.

Given a blur image patch, B_p , we solve for the kernel K by constructing the useful edge map from B_p [12]. As mentioned before, we first compute the r -map defined by

$$r(x) = \frac{\|\sum_{y \in N_h(x)} \nabla B_p(y)\|}{\sum_{y \in N_h(x)} \|\nabla B_p(y)\| + 0.5}, \quad (4)$$

where B_p denotes the blurred image patch and $N_h(x)$ is a $h \times h$ window centered at pixel x , and the constant 0.5 is to prevent producing a large r in flat regions. Then we apply thresholding on the r map to rule out pixels with small r values by

$$M = H(r - \tau_r), \quad (5)$$

where H is the Heaviside step function. The value in map M is 1 if r is higher than the threshold τ_r , and 0 otherwise. The final edge map is determined as

$$\nabla I_p^s = \tilde{\nabla} I_p \cdot H\left(M \|\tilde{\nabla} I_p\|^2 - \tau_s\right), \quad (6)$$

where \tilde{I}_p denotes the shock filtered image patch and τ_s is a threshold of the gradient magnitude. Once the edge map is obtained, we can solve the blur kernel K by minimizing the objective function

$$E(K) = \|\nabla I_p^s \otimes K - \nabla B_p\|^2 + \gamma \|K\|^2, \quad (7)$$

where γ is a weight for the regularization term. Taking the derivatives of $E(k)$ with respect to k and performing FFT on all variables, we obtain the close-form solution for K as follows:

$$K = F^{-1} \left(\frac{\overline{F(\partial_x I_p^s)} F(\partial_x B_p) + \overline{F(\partial_y I_p^s)} F(\partial_y B_p)}{F(\partial_x I_p^s)^2 + F(\partial_y I_p^s)^2 + \gamma} \right), \quad (8)$$

where F means FFT, F^{-1} means inverse FFT, and \bar{F} is the conjugate complex of F .

In our experiments, we set γ to be large, from 20 to 100, because we want the kernel to be smooth enough. If γ is too small, the kernel may break into pieces. At the end of each scale, we preserve the maximum component and leave the others out as a denoising step.

2.3 Latent Image Deconvolution

In the quadratic regularization term, the objective function of the latent image patch I_p is defined as

$$E(I_p) = \|I_p \otimes k - B_p\|^2 + \lambda \|\nabla I\|^2, \quad (9)$$

and the latent image can be solved by

$$I_p = F^{-1}\left(\frac{\overline{F(k)}F(B_p)}{F(k)F(k)+\lambda(F(\partial_x)F(\partial_x)+F(\partial_y)F(\partial_y))}\right). \tag{10}$$

It is important to note that the deconvolution here is not our final deconvolution algorithm to restore the blur image. During the kernel estimation procedure, we do not need to create a well-deblurred result with complex algorithms. As long as the important edges are deconvolved well, an accurate kernel is ensured. However, this kind of simple deconvolution method produces noises in the result. We apply bilateral filtering to suppress the noises, and this step is necessary. If no denoising is performed, the noises would be amplified during the iterative procedure, thus affecting the estimated kernel.

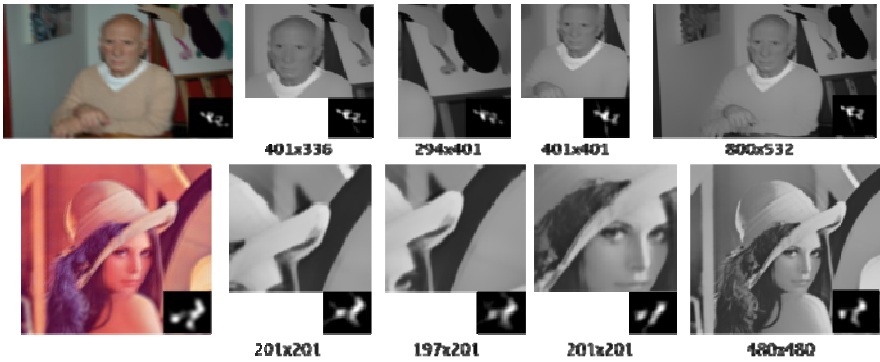


Fig. 1. Different patch selection methods and the corresponding kernel estimation. From left to right: blurred images with the ground-truth kernels, proposed method, max-gradient method, center patch, and whole image. The sizes of the patches are shown below each image.

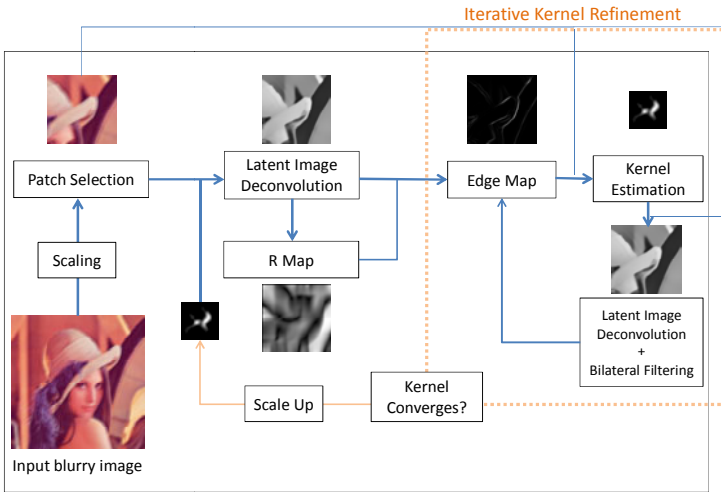


Fig. 2. Proposed framework for kernel estimation

3 Non-blind Image Deconvolution

In this paper, we propose a novel algorithm for non-blind image deconvolution, called GARL, which stands for Gradient Attenuation Richardson-Lucy. It effectively alleviates ringing artifacts by exploiting the gradient attenuation function [4] as the pixel weights to slow down the updating of pixels in the smooth regions and the regions around strong edges, thus suppressing the ringing propagation. The GARL algorithm is based on the classical Richardson-Lucy algorithm.

The Richardson-Lucy algorithm iteratively updates the image according to

$$I^{t+1} = I^t \cdot \left(F * \frac{B}{I^t \otimes F} \right), \quad (11)$$

where $*$ is the correlation operator, and t indicates the iteration number. In [7], the residual RL is proposed, which performs the RL algorithm on residual image to reduce the absolute amplitudes of the signals, hence reducing the ringing artifact. The iterative updating formula on the residual image becomes

$$\Delta I^{t+1} = (\Delta I^t + 1) \cdot \left(F * \frac{\Delta B + 1}{(\Delta I^t + 1) \otimes F} \right) - 1, \quad (12)$$

where ΔI denotes the residual image and ΔB is the residual blurred image: $\Delta B = \Delta I \otimes F + N$.

3.1 GARL

As described in [7], [9], the frequency of ringing artifact is lower than that of image details and ringing artifacts are negligible by human perception in highly textured regions. Therefore, to suppress ringing artifact, we force the smoothness constraint on the middle range of frequencies and the iterative update equation becomes

$$\Delta I^{t+1} = \frac{1}{1 + \mu W} \left\{ (\Delta I^t + 1) \cdot \left(F * \frac{\Delta B + 1}{(\Delta I^t + 1) \otimes F} \right) - 1 \right\}, \quad (13)$$

Since we want to suppress the contrast of ringing in the smooth regions while avoiding the suppression of sharp edges, the weight matrix should be large in smooth regions and small in edge and textured regions. We modify the gradient attenuation function [5] to determine the weight W for each pixel in (13). In the hierarchical restoration scheme, our modified gradient attenuation function, defined by W , is re-computed for each pixel at each scale s by propagating the scaling factor φ_s as follows:

$$\varphi_s(x, y) = \left(\frac{\alpha_s}{\|\nabla I_s(x, y)\|} \right)^\beta \cdot \left(\frac{\|\nabla I_s(x, y)\|}{\alpha_s} \right)^{\gamma M_s(x, y)}, \quad (14)$$

$$W_0 = \varphi_0, W_s = (W_{s-1})_\uparrow \cdot \varphi_s, \quad (15)$$

$$M_s = \langle \|\nabla^2 I_s\| \rangle \otimes F \cdot (1 - \langle \|\nabla I_s\| \rangle), \quad (16)$$

where $(\cdot)_\uparrow$ defines the up-sampling operator with linear interpolation, x and y denote the position in an image, α_s determines which gradient magnitude defines the smooth

regions, β and γ control the attenuating scale globally and locally, respectively, and they are set between 0.5 and 0.6 ($\beta > \gamma$) in our experiments. M_s indicates the influence range of strong edges according to the estimated blur kernel, and $\langle \cdot \rangle$ is the normalization operator. The values of M_s are between 0 and 1 so γ only effects the scaling factor at the positions where $M_s(x,y) \neq 0$. Because details are recovered more at finer scales, the gradient magnitudes of smooth area are larger than those at the same positions of the coarser scales.

3.2 Detail Recovery

The GARL can suppress most ringing artifacts with superior results compared to the other results, but it also suppresses some details around the strong edges. Therefore we propose a detail recovery process to further recover the lost details.

If we obtain two restored images from the GARL and the standard RL, the difference between them would contain the details and ringings, thus it can be expressed as follows:

$$I_{Diff} = I_{RL} - I_{GARL} = I_D + I_R \quad (17)$$

where I_{GARL} and I_{RL} denote the restored images by GARL and the standard RL, respectively, their difference is denoted by I_{Diff} , and I_D and I_R indicate the detail and ringing layer, respectively.

For each iteration, the I_R^t is obtained by applying a bilateral filter on I_{Diff}^t and then we determine a scaling factor, λ^t , to obtain a more accurate ringing layer by minimizing the following equation:

$$\lambda^t = \operatorname{argmin}_\lambda \|I_{Diff}^t - \lambda I_R^t\|^2 \quad (18)$$

$$\hat{I}_R^t = \lambda^t I_R^t \quad (19)$$

The detail layer and the difference layer for the next iteration are updated as:

$$I_D^t = \left(1 + \frac{|I_R^t|}{\max(|I_R^t|)}\right)^{-1} \cdot (1 - M) \cdot (I_{Diff}^t - \hat{I}_R^t) \quad (20)$$

$$I_{Diff}^{t+1} = I_{Diff}^t - \hat{I}_R^t - I_D^t \quad (21)$$

where M was defined in eq. (5) in the kernel estimation method.

The M map is a zero-one mask, with the pixels of value 1 representing the regions around strong edges and value 0 for smooth regions or regions containing many textures. It is inevitably to carefully select appropriate parameters to produce good restoration results. We set τ_s to be inside the range from 0.1 to 0.3, depending on the characteristics of the images.

The final restored image, I_F , is determined by:

$$I_F = I_{GARL} + \sum_t I_D^t \quad (22)$$

Fig.3 is an example of GARL method. We also compare the results with other methods [7], [8]. The GARL result apparently contains less ringing artifact. Fig.4 shows the procedure of the detail recovery procedure.

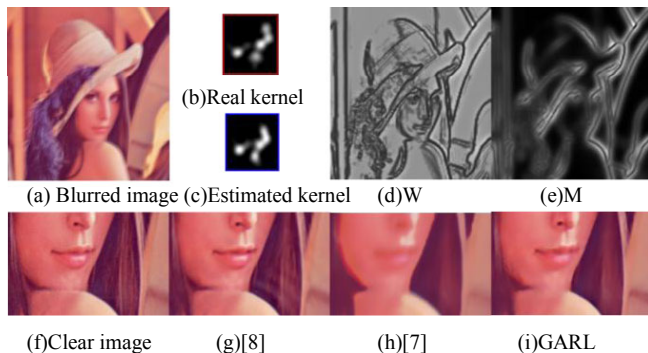


Fig. 3. An example of GEARL algorithm compared with other methods

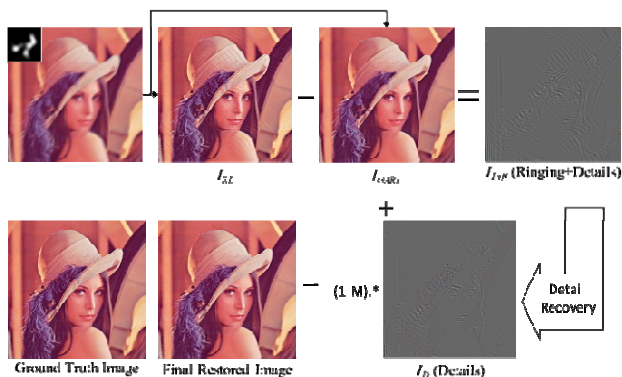




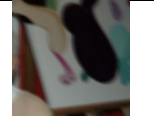



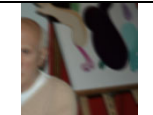
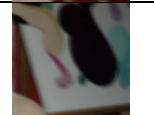





Fig. 4. Flowchart of the proposed detail recovery procedure

4 Experimental Results

In this section, we show some quantitative performance of our patch selection method and some experimental results to demonstrate the ringing suppression by using the proposed algorithm. Also, we provide some deblurred results on real images. Real images contain many unexpected factors, such as spatial varying blur or noises, which may lead to failed cases of kernel estimation or deconvolution. The proposed method, however, can restore most real blurred images well with the ringings suppressed and the details recovered. The computing platform for our experiments is a notebook running MS Vista 32-bit version with Intel Core2 CPU P7450 2.13GHz, and 4GB RAM. The program was implemented in MATLAB.

We measure the performance of our patch selection method by computing the minimum SAD between the estimated kernels and the ground truth. From Table 1, our Harris corner patch selection method produces the kernel which is most similar to the ground truth. Although the patches of scale 3 in method (1) and (2) look similar, the

Table 1. Performance of the patch selection methods. (1)proposed Harris corner patch method, (2)patch with maximum gradient magnitude, (3)central patch. The number below each estimated kernel is the minimum SAD value with the ground truth kernel.

Method	Scale 1	Scale 2	Scale 3	kernel	GT
(1)				 0.4218	
(2)				 0.6851	
(3)				 0.4254	

estimated kernels differs because (2)'s kernel is not estimated well in scale 2, affecting the result. Therefore, finding an appropriate patch is very important in every scale.

Some detailed deblurred results are shown in Fig. 5 and Fig. 6. These experimental comparisons demonstrate the improved image restoration results by using the proposed algorithm, especially in the ringing suppression and detail recovery. The revised detail recovery results contain more texture than GARL and less ringing than the previous Richardson-Lucy based deconvolution methods. We compare our results with some other representative previous works [9][14][15] in Fig. 7. More well-deblurred results by using the proposed algorithm on real images are shown in Fig. 8 and Fig. 9.

As for the execution time, our program also performs well. We have tried two methods to solve the objective functions given in eq. (7) and (9), as in Table.2. Using conjugate gradient method for solving eq. (7) and RL method for eq. (9) generates more controllable and stable results. However, the FFT division method considerably decreases the execution time. This experiment is tested on a 900x900 image (statue).

Table 2. Execution time of the proposed image deblurring algorithm

Objective function (7)	Objective function (9)	Execution time((7)+(9))
Conjugate gradient	Richardson-Lucy	330s
FFT division	FFT division	46s

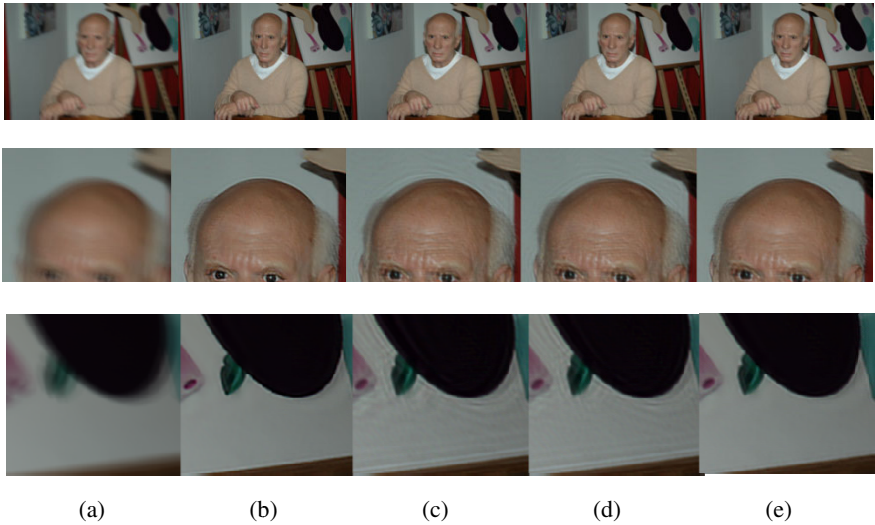


Fig. 5. Detail recovery example 1. (a) blurred image, restored images by using (b) GEARL, (c) standard RL, (d) detail recovery, and (e) revised detail recovery.

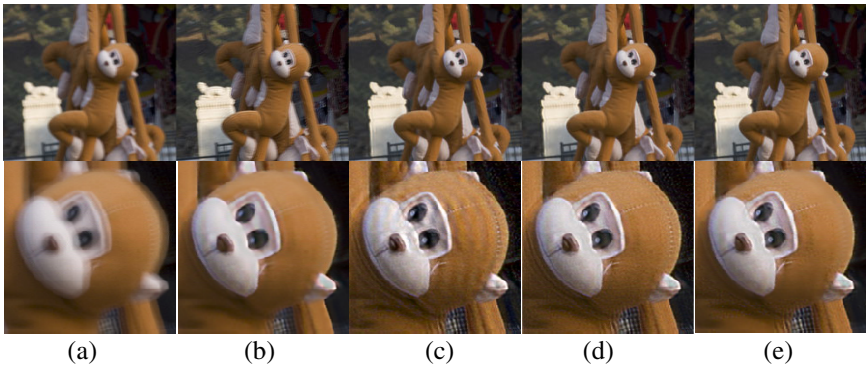


Fig. 6. Detail recovery example 2. (a) Blurred image. (b) GEARL. (c) standard RL. (d) Detail recovery. (e) Revised detail recovery.

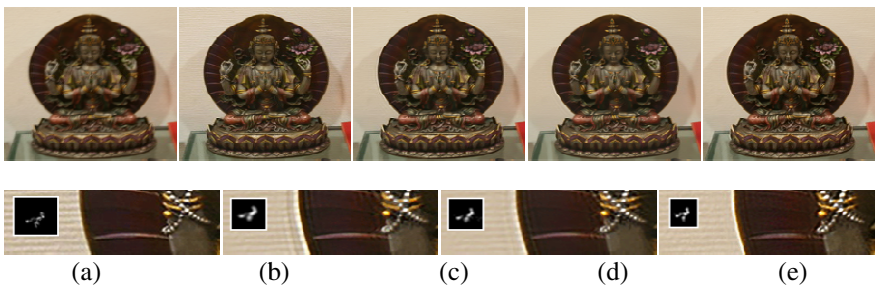


Fig. 7. (Top) (a) Blurred image, deblurred images by using (b) Shan et al. [9], (c) Joshi et al. [14], (d) Levin et al. [15], and (e) the proposed deblurring algorithm. (Down) Enlarged parts of (b)~(e).



Fig. 8. (a) A real blurred image and (b) the deblurred image by using the proposed deblurring algorithm

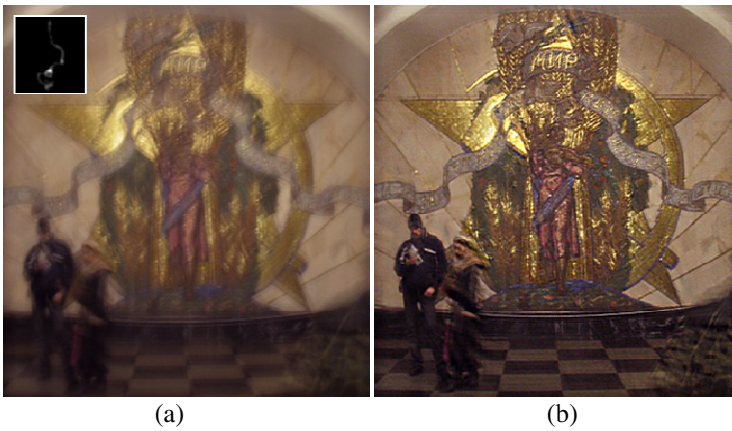


Fig. 9. (a) A real blurred image and (b) the deblurred image by using the proposed deblurring algorithm

5 Conclusion

In this paper, we proposed a framework for image deblurring from a single blurred image. The proposed blur kernel estimation can effectively suppress the ringing artifact and recover the image details compared to the previous methods. However, the proposed image deblurring algorithm has the following limitations. First, using FFT to speed up the computation in the kernel estimation procedure may produce some problem near the image boundary. In addition, like many other deblurring methods, the proposed algorithm also requires the user to carefully tune the parameters to obtain the optimized deblurred image. In the future, developing a fully automatic image deblurring system is the direction of our research.

References

1. Richardson, W.H.: Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America* 62, 55–59 (1972)
2. Lucy, L.B.: An iterative technique for the rectification of observed distributions. *Astronomical Journal* 79, 745–765 (1974)
3. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *ICCV*, pp. 839–847 (1998)
4. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. *ACM Trans. Graphics* 25, 787–794 (2006)
5. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. *ACM Trans. Graphics*, 249–256 (2002)
6. Levin, A.: Blind motion deblurring using image statistics. In: *NIPS*, pp. 841–848 (2006)
7. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graphics* 26, 70 (2007)
8. Yuan, L., Sun, J., Quan, L., Shum, H.-Y.: Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graphics* 27, 1–10 (2008)
9. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. *ACM Trans. Graphics* 27, 73–83 (2008)
10. Yuan, L., Sun, J., Quan, L., Shum, H.-Y.: Image deblurring with blurred/noisy image pairs. *ACM Trans. Graphics* 26 (2007)
11. Cho, S., Lee, S.: Fast motion deblurring. *ACM Trans. Graphics (SIGGRAPH ASIA)* (2009)
12. Xu, L., Jia, J.: Two-Phase Kernel Estimation For Robust Motion Deblurring. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010*. LNCS, vol. 6311, pp. 157–170. Springer, Heidelberg (2010)
13. Harris, C., Stephens, M.: A Combined corner and edge detector. In: *Alvey Vision Conference* (1988)
14. Joshi, N., Szeliski, R., Kriegman, D.: PSF estimation using sharp edge prediction. In: *CVPR* (2008)
15. Levin, A., Sand, P., Cho, T.S., Durand, F., Freeman, W.T.: Motion-invariant photography. In: *SIGGRAPH* (2008)