Text Image Deblurring Using Text-specific Properties*

Hojin Cho[1], Jue Wang[2], and Seungyong Lee[3]

[1,3]Pohang University of Science and Technology (POSTECH)
[2]Adobe
http://cg.postech.ac.kr/research/text_deblur/

Abstract. State-of-the-art blind image deconvolution approaches have difficulties when dealing with text images, since they rely on natural image statistics which do not respect the special properties of text images. On the other hand, previous document image restoring systems and the recently proposed black-and-white document image deblurring method [1] are limited, and cannot handle large motion blurs and complex background. We propose a novel text image deblurring method which takes into account the specific properties of text images. Our method extends the commonly used optimization framework for image deblurring to allow domain-specific properties to be incorporated in the optimization process. Experimental results show that our method can generate higher quality deblurring results on text images than previous approaches.

1 Introduction

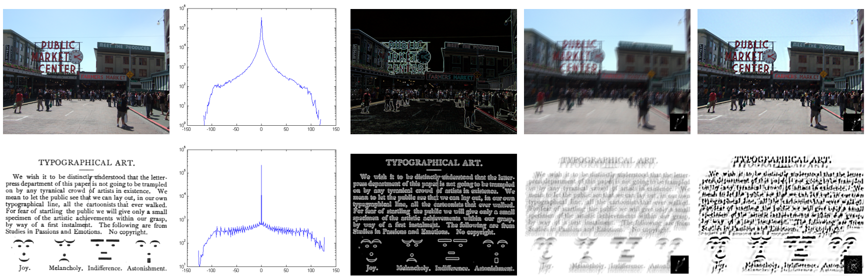
In single image deblurring, image degradation is usually modeled as:

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

where b is the observed blurred image, k is the motion blur kernel, $*$ is the convolution operator, l is the latent image, and n is image noise. The goal of blind deconvolution is to recover both k and l from a single input b, which is an ill-posed problem.

Tremendous progress has been achieved in recent years for solving the blind deconvolution problem, by utilizing natural image statistics as a strong prior to regularize the solution [2,3,4,5,6,7]. Despite their success in natural images, the state-of-the-art approaches are barely suitable for an important category of images: text images, mainly because the natural image statistics that these methods utilize are too weak to constrain text images. On the other hand, previous document restoration methods [8,9,10,11] employ text-specific priors, such as the two-tone property. However, these methods do not consider motion blur, thus are insufficient to deal with text images that have been blurred by large motion kernels. Since restoring text images has important real-world applications, especially in the image forensics domain, there is a significant demand for specialized image deblurring techniques that can effectively restore text images. Recently Chen et al. [1] proposed a text deblurring method, but it depends on text segmentation, which cannot be done robustly for general blurred text images.

* Part of this work was done while the first author was an intern at Adobe.



(a) (b) (c) (d) (e)

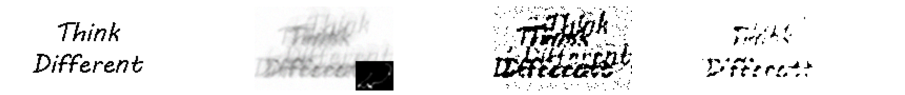
Fig. 1. Natural image deblurring vs. text image deblurring. (a) original sharp images; (b) log-scale gradient histograms of (a); (c) visualization of gradient magnitudes of (a); (d) synthetic blurred images and the input blur kernels; (e) estimated latent image and blur kernel using Levin et al. [7].

In this paper, we propose a novel text image deblurring method that fully utilizes the special properties of text images. We first show previous image deblurring approaches do not work well on text images. We then analyze desirable properties of latent text images and the difficulty of incorporating domain-specific knowledge into the commonly used optimization framework for deblurring. To resolve the difficulty, we extend the optimization framework by introducing an auxiliary image a, which is used to enforce the domain-specific properties on the latent image l. We also apply text-specific properties to the final deconvolution that computes the output latent image l using the estimated kernel k. Experimental results show that our method outperforms both previous general image deblurring algorithms and the recent document deblurring algorithm proposed by Chen et al. [1] in terms of deblurring quality of text images.

2 Related Work and Limitations

Image deblurring has been extensively studied in recent years, and significant breakthrough has been achieved by incorporating various priors for optimization, such as Gaussian prior [5,12], sparsity prior [7], and natural image statistics [13,14,4]. However, these priors do not apply well to text images. As shown in Fig. 1b, the gradient histogram of a natural image is significantly different from a text image. Furthermore, strong gradients of a text image have a highly-regular spatial distribution as they enclose text strokes of similar widths (Fig. 1c). Directly applying a natural image deblurring method to a text image thus will result in an erroneous result (Fig. 1e).

Another problem that general image deblurring approaches face when dealing with text images is the way they handle small objects. Many state-of-the-art methods [15,5,12] predict sharp edges in the image and use them for kernel estimation. However, as discovered in a recent work [12], objects whose scales are smaller than the blur kernel can damage kernel estimation. A solution is to remove small object edges and only rely on large ones for kernel estimation [12]. Although this strategy could work well for natural images, it cannot be directly applied to text images. Text characters are usually small and spatially close to each other, and the edges from them will be removed as small outliers in kernel estimation.



(a) Original image (b) Synthetic blurred image (c) Thresholding with different thresholds

Fig. 2. Difficulty of applying the segmentation (text binarization with thresholding) technique to a blurred text image.

Previous document restoration methods [8,9,10,11] have used the two-tone or bi-level property of text. Under this assumption, small blurs and ink-bleeding artifacts can be removed using thresholding. However, such thresholding no longer works when the image is blurred with a large motion kernel. Fig. 2 shows an example. Since the blur is severe in this example, simply applying thresholding would not result in a clean segmentation of the text region, no matter how the threshold value is selected.

The work that is closest to ours is the black-and-white document image deblurring method recently proposed by Chen et al. [1]. This method first detects text from the input blurred image using a text segmentation method [16], then estimates the original intensities of texts using a learned relationship between the intensity histograms of blurred and sharp document images. However, the text segmentation technique used in this method is based on thresholding, which is insufficient for blurred images (Fig. 2). Consequently, using the segmentation result for deblurring will introduce artifacts, as we will show in Sec. 6. Moreover, learning the relationship between the intensity histograms is limited by the characteristics of the training data set, such as the amounts of blurs, types of fonts, and text colors. This approach thus does not generalize very well.

3 Text Image Deblurring

3.1 Desired properties of latent text images

Given that natural image statistics does not work well for text image deblurring, we first identify a set of text-specific properties that should be considered in the deblurring process, as listed below:

Property 1 Text characters usually have high contrasts against nearby background regions.

Property 2 Each character has a near-uniform color. Accordingly, gradient values inside each character should be close to zero.

Property 3 Although many documents have single background colors (e.g., white), advertisements or posters may have more complex background. Thus, assuming a single tone for background as in [1] is too restrictive. We instead assume the background gradient values obey natural image statistics and are sparse [17].

To incorporate these properties into the deblurring process, we could first consider the commonly used optimization framework for deblurring [6,7], which seeks a pair $(\hat{l}, \hat{k})$ by optimizing:

$$\arg\min_{l,k} \|b - k * l\|^2 + \rho_l(l) + \rho_k(k), \tag{2}$$

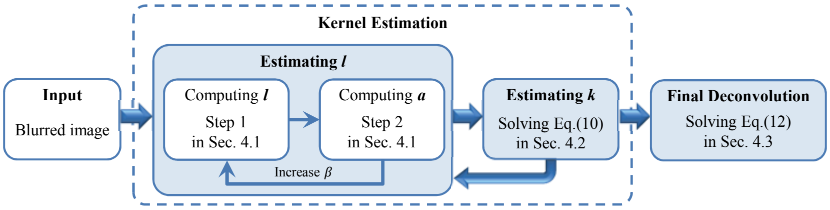


Fig. 3. Process overview of our deblurring algorithm.

where $\|b - k * l\|^2$ is the data fitting term, and $\rho_l(l)$ and $\rho_k(k)$ are priors for the latent image and the motion blur kernel, respectively. Since the text properties play the role of an image prior for l, we need to somehow adjust the prior $\rho_l(l)$, for which previous methods mostly use $\rho_l(l) = \sum_i |\partial_{x,i}(l)|^\alpha + |\partial_{y,i}(l)|^\alpha$ with $0.5 \leq \alpha \leq 2$, so that the sparsity on gradients of l is enforced.

However, it is not straightforward to directly minimize Eq. (2) with the text image properties other than natural image statistics, because solving for the new $\rho_l(l)$ with embedded text image properties involves complicated nonlinear optimization. Our three desirable properties for a latent text image require careful separation between text and background regions, where such spatial segmentation is hardly defined as a filter (e.g., gradients or Laplacian) of l. On the other hand, without explicit segmentation, it is hard to directly encode the three properties as energy functions of l. To resolve this difficulty, we propose a new optimization framework which is capable of incorporating domain-specific properties in the deblurring process.

3.2 Optimization framework

We extend Eq. (2) using the half-quadratic penalty method [18,19] by introducing an auxiliary image a which enforces domain-specific properties on the latent image l as:

$$\arg\min_{l,k,a} \|b - k * l\|^2 + \rho_l(l) + \rho_k(k) + \rho_a(a) + \beta \|l - a\|^2 , \tag{3}$$

where $\rho_a(a)$ is a cost function defined using the domain-specific properties and β is the weight for the similarity penalty between l and a. Note that Eq. (3) approaches Eq. (2) when β becomes large enough and $\rho_a(a) \approx \rho_l(l)$. In this formulation, the domain-specific prior $\rho_a(a)$ is explicitly embedded in the optimization framework, and the parameter β, which automatically adapts in the process, controls when and how strong $\rho_a(a)$ to be reflected in the latent image l.

It is worth mentioning that the auxiliary image a is similar to the *edge prediction* result in some previous kernel estimation methods [5,12]. However, in those approaches, a is computed in a simple way by applying various image filters to l, such as bilateral and shock filters, to produce sharp, step-like edges. In our approach, we can encode more general domain-specific knowledge in a by taking an optimization framework in Eq. (3).

3.3 Process overview

Fig. 3 shows the overall process of our deblurring algorithm, which is capable of incorporating domain-specific properties. To estimate the blur kernel k, our method iterates two main steps: estimating k and estimating l. The estimating l step contains an inner optimization loop, where we iteratively refine the latent image l and the auxiliary image a to impose the domain-specific properties on l.

For kernel estimation, similar to previous approaches [5,12], our system employs a multi-scale approach by building an image pyramid and solving for the kernel starting from the coarsest level. Once optimized on a level, the estimated k and l are bi-linearly interpolated to the next level and are used as initial values for optimization.

4 Algorithm Details

In this section, we present the algorithm details to solve the optimization problem in Eq. (3).

4.1 Estimating *l* with text image properties

Assuming the blur kernel k is fixed, we estimate l which reflects text properties given in Sec. 3.1. To do that, our method involves an iterative minimization strategy to solve:

$$\arg\min_{l,a} \|b - k * l\|^2 + \rho_l(l) + \rho_a(a) + \beta \|l - a\|^2 . \tag{4}$$

In Eq. (4), a and l are computed alternatingly by fixing one of them with the value obtained from the previous iteration. The weight β is set to be a small value β_0 for the first iteration, then is multiplied by δ at each sequential iteration until $\beta > \beta_{\mathrm{MAX}}$. We use $\beta_0 = 0.01$, $\delta = 1.5$, and $\beta_{\mathrm{MAX}} = 5$ in our implementation. The merit of incrementally adjusting β is that in early iterations, a may not be initialized well, thus we rely on $\rho_l(l)$ to obtain good initial estimates of l. In later iterations when a is more accurate, we allows it to play a more dominant role in determining l by using a large β. At the end of the optimization process, since β is large, both l and a converge to the same image satisfying both the data term and text image properties.

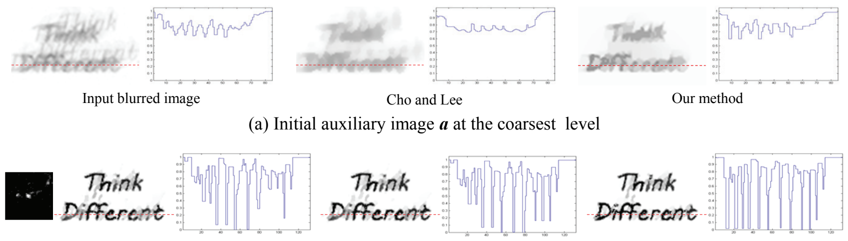
Step 1: computing l Given b, k, and a, we solve:

$$\arg\min_{l} \|b - k * l\|^2 + \rho_l(l) + \beta \|l - a\|^2 , \tag{5}$$

where $\rho_l(l) = \lambda_l \|\nabla l\|^2$ for regularizing the gradients of l. Eq. (5) is solved using Fast Fourier Transform (FFT) as:

$$l = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(k)}\mathcal{F}(b) + \beta\mathcal{F}(a)}{\overline{\mathcal{F}(k)}\mathcal{F}(k) + \beta\mathcal{F}(1) + \lambda_l\overline{\mathcal{F}(\nabla)}\mathcal{F}(\nabla)}\right), \tag{6}$$

where $\mathcal{F}(\cdot)$ and $\overline{\mathcal{F}(\cdot)}$ are the FFT operator and its complex conjugate, respectively, and $\mathcal{F}(1)$ is the FFT of the delta function. Eq. (6) can be solved efficiently since it only consists of component-wise operators except FFTs. We fix $\lambda_l = 0.1$ in our system, and set the initial value of a as l.



Input blurred image

Cho and Lee

Our method

(a) Initial auxiliary image ***a*** at the coarsest level

1st iteration

2nd iteration

11th iteration

(b) Intermediate auxiliary image ***a*** at the iterations for estimating ***l***

Fig. 4. Intermediate results in latent image estimation. (a) The predicted latent image a of Cho and Lee [5] and ours in the first iteration at the coarsest level. (b) The iteratively refined auxiliary image a given an incomplete blur kernel. The graphs show the 1D profiles of the scan lines of image a marked in red. The original latent image is show in Fig. 5.

Step 2: computing a In this step, we first refine l computed from Step 1 to reduce ringing artifacts and noise caused by simple deconvolution in Eq. (6). To achieve this, we adopt the $L0$ gradient minimization method [20] which can effectively remove noise and the edges of small magnitudes. Using $L0$ gradients can provide a good initial for a, which is consistent with the requirements of Property 1 and Property 2. The boundaries of texts have large gradient values and are well preserved in $L0$ gradient minimization, while the gradients of the noise and ring artifacts generally have small magnitudes and are suppressed inside texts.

Then, with the refined l, we compute a by minimizing:

$$\arg\min_{a} \rho_a(a) + \beta \left\| l - a \right\|^2 , \tag{7}$$

where $\rho_a(a)$ is the cost function to measure if a satisfies the text image properties in Sec. 3.1. Mathematically, it is defined as a pixel-wise function as:

$$\rho_a(a_i) = \begin{cases} 0 & \text{if } a_i = a_i^{\mathrm{P}} \\ d_{\mathrm{MAX}} & \text{otherwise} \end{cases} , \tag{8}$$

where i is a pixel index, a^{P} is the ideal image satisfying the text image properties, and d_{MAX} is a constant larger than the maximum pixel value of l. With Eq. (8), $\rho_a(a)$ penalizes the pixels which do not agree with a^{P}.

Note that the ideal image a^{P} which satisfies the domain-specific properties is unknown. We therefore approximately estimate a^{P} from l using the stroke width transform (SWT) [21], which classifies the image into text and non-text regions. The output of SWT is a stroke width value at each pixel if the pixel is classified as text, and 0 otherwise. Based on the output of SWT, we find connected components to reconstruct the set of characters as explained in the original paper [21]. Then, we force pixels in each character to have the same color value, resulting in the image a^{P}. We will discuss later how to determine the color of each character.

Since $\rho_a(a)$ can be individually computed for each pixel using Eq. (8), the final solution a^* for Eq. (7) is:

$$a_i^* = \begin{cases} a_i^{\mathrm{P}} & \text{if } \beta \left\| l_i - a_i^{\mathrm{P}} \right\|^2 < d_{\mathrm{MAX}} \\ l_i & \text{otherwise} \end{cases} . \tag{9}$$

Recall that β starts from a small value and increases after each iteration. Thus, in early iterations, the penalty for the similarity term in Eq. (7) (the second term) is small, so the first term dominates the solution to impose strong domain-specific priors on a. In later iterations, the penalty for the similarity term becomes significant as β increases, and a is enforced to be close to l. Since l is updated in Step 1 based on the new a in every iteration, l also obeys the text image properties eventually.

The pixels in each character now have the same intensity, thus Property 2 is satisfied. Through $L0$ gradient minimization [20] and re-coloring of text characters, we preserve the large gradient around the boundaries of text characters, thus Property 1 is respected. Note that, however, we do not impose the sparsity prior for background (Property 3) on this step, since the purpose of estimating l here is to restore sharp edges along the boundaries of texts, as salient edges are known to be useful for kernel estimation [5,12]. Instead, the sparsity prior for Property 3 is included in the final deconvolution step that produces the deblurring result (Sec. 4.3).

It is worth mentioning that accurately estimating the expected color of each character is a difficult task. As Chen et al. [1] pointed out, texts are often small in the scene, thus their intensities in the blurred image are shifted. A learned relationship of intensities between sharp and blurred text images could be used [1], but as mentioned in Sec. 2, such a relationship strongly depends on the training data and is hard to be generally applied. In our method, we use the minimum/maximum value of each character for the approximated color. Specifically, if the character is darker than background, we find the minimum intensity of all pixels in the character, and assign it to the pixels. Similarly, we use the maximum intensity if the character is brighter. We found that this method can effectively solve the intensity shift problem mentioned in [1]. The minimum/maximum values quickly converge to the true intensity of a character as the optimization proceeds, thus its inaccuracy in early iterations does not have noticeable effect on the estimated blur kernel.

Fig. 4 illustrates the behavior of our iterative optimization method. In the initial iteration, we set k as a delta function and l as the input blurred image b. Because our method incorporates text-specific priors, our method predicts sharper initial texts than the method of Cho and Lee [5] with less blur artifacts (Fig. 4a). In the intermediate iterations, although the estimated blur kernel is not accurate, our method iteratively refines and predicts a (Fig. 4b) so that pixels in a have almost the same intensities as the original image, which is shown in Fig. 5.

4.2 Computing *k*

With the introduction of a, we estimate the blur kernel by minimizing:

$$E(k) = \sum_{\partial^* \in \Theta} \omega_{(\partial^*)} \left\| \partial^* b - k * \partial^* a \right\|^2 + \gamma \left\| k \right\|^2 , \tag{10}$$

where $\Theta = \{\partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$ are partial derivatives as defined in previous approaches [4,5], and γ is a relative weight for the Tikhonov regularization. The success of gradient-based kernel estimation has been reported in [4,5,7]. Note that we use the image edges of a, instead of l, to update the blur kernel k in Eq. (10). Although both l

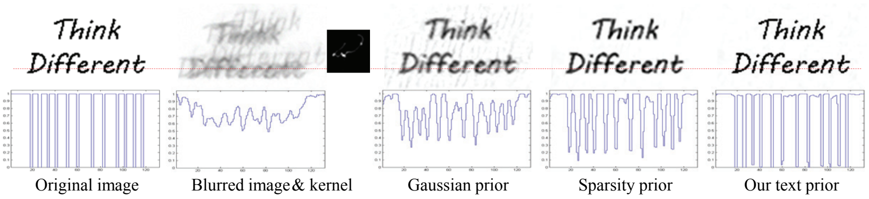


Fig. 5. Comparison of non-blind deconvolution results.

and a are supposed to converge to the same image after the alternating optimization in Sec. 4.1, the l computed by Eq. (6) using FFTs may contain ringing artifacts and noise. Since such artifacts are suppressed in a through Step 2, using a as the latent image in Eq. (10) provided better results in our experiments.

4.3 Final deconvolution

After the blur kernel k has been estimated, a non-blind deconvolution method is applied to the input blurred image b to generate the final output image. Since we have only emphasized reconstruction of salient edges around texts in the kernel estimation process, the latent image generated in the process is not optimal in terms of visual quality.

To recover a high quality output image, we propose a non-blind deconvolution prior that considers the text-specific properties in Sec. 3.1 as:

$$\rho_T(l) = \lambda_1 \sum_{i \in T} \|\nabla l_i\|^2 + \lambda_2 \sum_{i \in T} \left\|l_i - a_i^{\mathrm{P}}\right\|^2 + \lambda_3 \sum_{i \notin T} \|\nabla l_i\|^{\alpha}, \tag{11}$$

where T is the set of pixels that belong to the text regions in a^{P} detected using SWT. The first term encourages the gradient values of pixels inside text regions to be zero. The second term makes the color values of pixels consistent with a^{P} inside text regions. The last term encourages the background pixels to have sparse gradients ($\alpha = 0.8$ in our system). Using this prior, the total energy for the final deconvolution is defined as:

$$E(l) = \|b - k * l\|^2 + \rho_T(l), \tag{12}$$

which can be solved using the Iterative Re-weighted Least Square (IRLS) method [17]. At every iteration for re-weighting in IRLS, we update a^{P} and T using the updated l.

Fig. 5 compares the performance of our deconvolution prior against traditional natural image priors using a synthetic example. Since our deconvolution strongly penalizes color variations inside the text region, our method restores sharper texts.

5 Algorithm Analysis

In this section, we provide detailed analysis on how our method can restore sharp texts effectively. In addition, since the $MAP_{l,k}$ framework in Eq. (2) has been known to be vulnerable to the convergence problem [6,7], we explain how our extended framework can overcome it in practice.

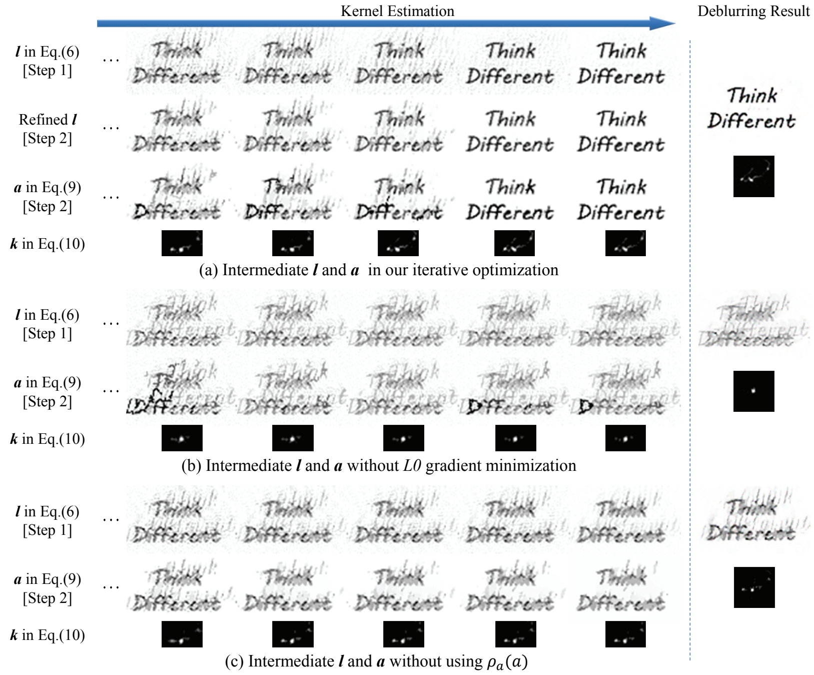


Fig. 6. Effects of $L0$ gradient minimization and $\rho_a(a)$. (a) intermediate l and a in our algorithm at a coarse level; (b) intermediate l and a at the same level without $L0$ gradient minimization; (c) intermediate l and a at the same level without using $\rho_a(a)$ in Eq. (7).

5.1 Effects of $L0$ gradient minimization

In Eq. (6) of Step 1, due to the poor regularization prior and the assumption on data periodicity of FFT, the computed l contains ringing artifacts and noise. Since $L0$ gradient minimization [20] can effectively remove the edges of small magnitudes caused by these artifacts, it leaves only the edges of large magnitudes intact, which are useful for kernel estimation, as demonstrated in [5,12].

To verify this, an experiment is shown in Fig. 6 using the blurred image in Fig. 2b. The results suggest that without $L0$ gradient minimization (Fig. 6b), ringing artifacts and noise in l disturb imposing the text image properties on a, thus the kernel estimation is damaged. In contrast, applying $L0$ gradient minimization ensures the successful kernel estimation.

5.2 The role of the synthetic text image a^{P}

The cost function $\rho_a(a)$ in Eq. (7) plays the most important role in injecting text properties into the final solution. To demonstrate it more clearly, in Fig. 6c, we show intermediate results after removing this term from Eq. (7). It is clear that the intermediate image a no longer obeys the text image properties specified in Sec. 3.1, thus the estimated blur kernel becomes erroneous.

The key component of $\rho_a(a)$ is the synthetic text image a^{P}, which is computed using SWT. Given a two-color image like Fig. 6, it may be wondered if SWT can be replaced by simple segmentation on top of our framework. For such a two-color image, the performance of SWT and simple segmentation may be similar. However, the main purpose of using SWT is to handle more complex images where the background contains other colors and textures (e.g., Fig. 11). SWT obviously outperforms segmentation-based approaches for text detection in these cases [21], which increases the accuracy, stability, and robustness of our framework. This is evidenced by our comparisons (see Fig. 9 and the supplementary material) with the document deblurring method [1], which is largely based on thresholding-based segmentation.

5.3 Iterative updates of a and l, and its convergence

As shown in Eqs. (5) and (7), a and l affect each other and they are updated in an alternating fashion. A natural question would be if this iterative process will converge. Since β in Eqs. (5) and (7) is increasing after each iteration, the similarity term $\beta \left\| l - a \right\|^2$ eventually dominates both energy functions, and the optimal solutions are $l^* = a^*$. That is, l and a converge to the same image with the increasing parameter β (see Fig. 6a). In practice we found the process converges quickly (usually within 15 iterations) before β reaches a large value.

Since a and l affect each other, it may be wondered whether a could be dominated by l and the text properties would not be reflected in the final solution. Note that in Eq. (4), the prior $\rho_l(l)$ and $\rho_a(a)$ have equal weights, and when computing $\rho_a(a)$, the d_{MAX} value in Eq. (8) is purposely set to be large so that a prefers a^{P} to l at early iterations. Then, the text properties reflected in a affect estimating l in Eq. (5) due to the similarity term with increasing β, thus $\rho_l(l)$ itself cannot dominate in Eq. (5). In this respect, $\rho_a(a)$ plays an important role in the optimization process to ensure the text properties to be reflected in the final images of l and a.

Finally, our framework is extended from, and shares the same convergence limitation with the $MAP_{l,k}$ framework, which was analyzed in [6,7]. In short, converging to a sharp image is not guaranteed in the original $MAP_{l,k}$ framework. However in our approach, since incorporating text properties always imposes a sharper latent image, it can greatly help avoid the no-blur-explanation problem [6,7]. In practice, we found our system converges well under various inputs, as evidenced by the results provided in the paper and the supplementary material.

5.4 Failure cases and limitations

To provide more insights, we further investigate the failure cases of our algorithm (Fig. 7). Although our approach can robustly estimate complex blur kernels in many cases, it is vulnerable to severe noise, which is a common limitation of previous deblurring methods. Specifically, using $L0$ gradient minimization alone cannot refine l from a severe noisy image, so this damages the text intensity estimation on a. Another failure case is the narrow spaces between characters. Overlapping of blurred neighboring characters has negative impact on the performance of the proposed approach. Although our

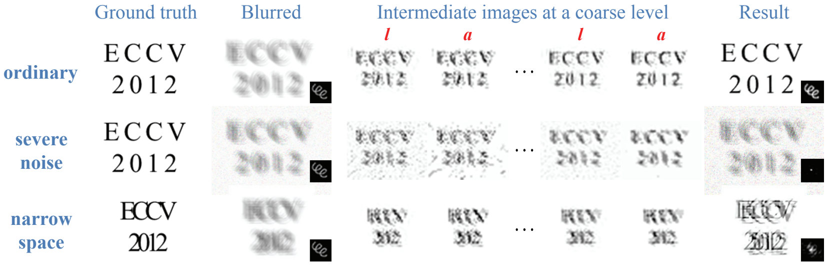
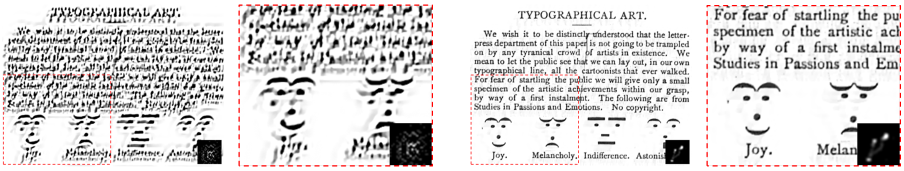


Fig. 7. Failure cases. (top row) our intermediate and final results on a synthetic example; (middle row) results with added severe noise; (bottom row) results with narrow spaces between characters.


TYPOGRAPHICAL ART.
Joy.
Melancholy.
Indifference.
For fear of startling the pu
specimen of the artistic ach
by way of a first instalme
Studies in Passions and Em
Melan

Levin et al. [7] Our result

Fig. 8. Deblurring results on the synthetic blurred image in Fig. 1.

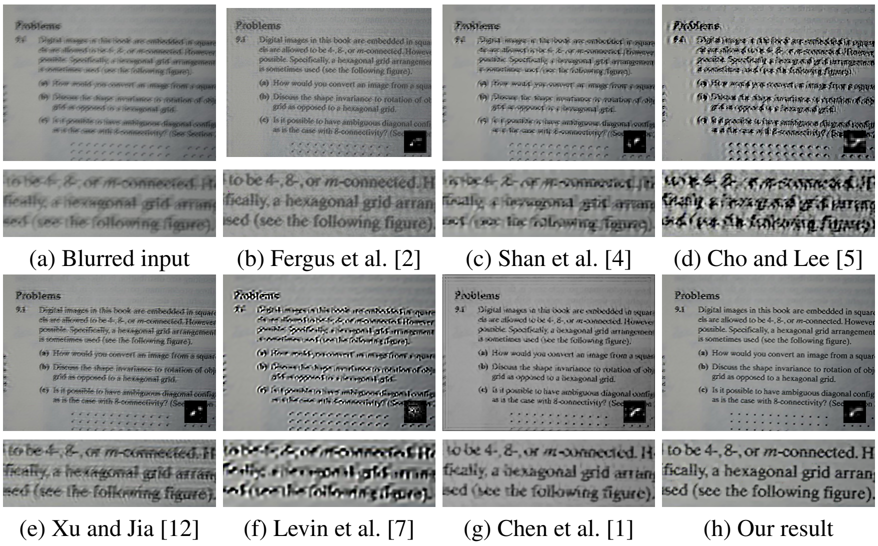
method might fail in extreme cases, it works reliably well on various real-world examples, as demonstrated in the next section.

Our approach also has some limitations in its technical rigorousness. Specifically, the optimization is done through an iterative framework which is not guaranteed to converge to the optimal solution. The different optimization steps are not unified in an integrated cost function, and refining l through $L0$ gradient minimization is accomplished as a separate step prior to computing a. Nevertheless, our approach to solve the optimization problem for image deblurring with domain-specific properties works well in practice, and further analysis and improvements remain as future work.

6 Results

We implemented our method using C++. Our testing environment is a PC running MS Windows 7 64bit version with Intel Core i7 CPU and 12GB RAM. The computation time is a few minutes for an 1M image. For example, it took about one minute for Fig. 9a whose size is 425×313 pixels. Faster response could be achieved in practice by using a small text patch, not the whole image, for kernel estimation. More examples can be found in the supplementary material.

Fig. 8 shows the deblurring results on the synthetically blurred text image in Fig. 1. Our method not only estimates a more accurate blur kernel, but also restores sharper texts. We also measured the restoration errors quantitatively using the synthetic examples in the paper (top row of Fig. 7 and Figs. 8 and 10). The average PSNR value of the blurred images is 15.66, and that of our deblurring results is 28.22. The detailed results are provided in the supplementary material.



(a) Blurred input (b) Fergus et al. [2] (c) Shan et al. [4] (d) Cho and Lee [5]

(e) Xu and Jia [12] (f) Levin et al. [7] (g) Chen et al. [1] (h) Our result

Fig. 9. Comparison with previous methods. The input image (a) was originally provided in [1].

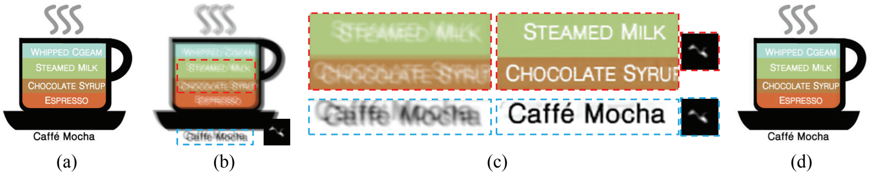
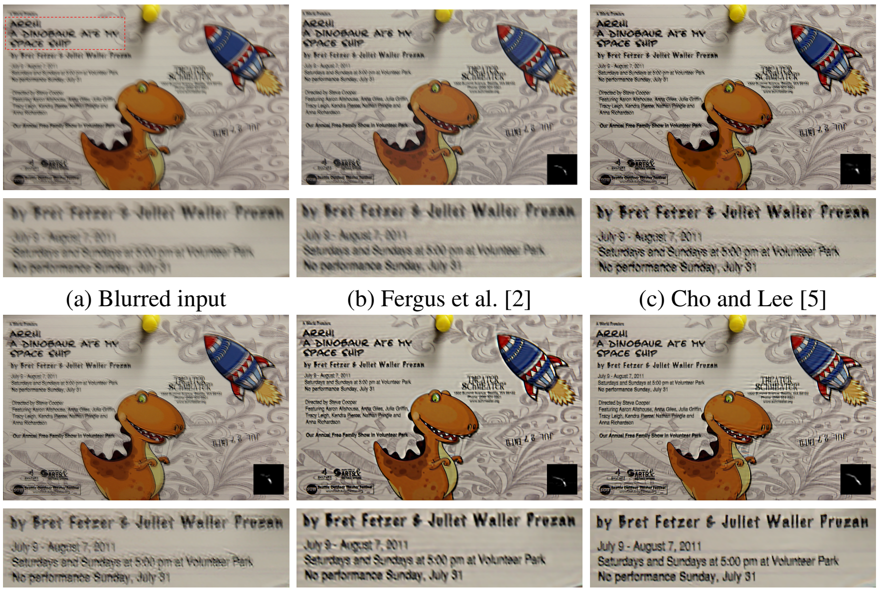


Fig. 10. Deblurring of a blurred image with different text regions. (a) ground truth image, (b) synthetic blurred image and its kernel, (c) cropped images of (b) and our deblurring results, (d) final deblurring result of (b).

Fig. 9 shows a comparison with previous deblurring methods. Figs. 9b-f were produced using the authors' implementations with intensive manual parameter tuning. The result of Chen et al. [1] was taken from the original paper. In Fig. 9, most natural image deblurring methods fail to generate a sharp, clean text image as output. Fergus et al.'s approach [2] worked well, but the result still contains ringing artifacts and noise. The result genereated by Chen et al.'s approach [1] is less noisy, but the recovered texts are still blurry. This may have resulted from that their histogram-based training method does not apply well for general text images beyond their training data set. In contrast, our method recovered a much better latent image in terms of the sharpness of the texts as well as noise level. Our method achieved this without using any training data or hard-coded statistics.

Since our method does not depend on the training data, we can estimate the blur kernel from any text region in the given image (Fig. 10). To illustrate this merit, we selected and cropped two different regions from the input blurred image and estimated the blur kernel individually (Fig. 10c). The estimated kernels of two regions are almost same, and the deconvolution results show clear texts regardless of the text and background



(a) Blurred input (b) Fergus et al. [2] (c) Cho and Lee [5]

(d) Xu and Jia [12] (e) Levin et al. [7] (f) Our result

Fig. 11. Deblurring results of a real image which consists of texts and complex background.

colors, the sizes of fonts, and the amounts of texts. The final deconvolution result using one of the estimated kernels is shown in Fig. 10d.

Fig. 11 shows a deblurring result of a postcard image which contains complex background. We used a small region (shown as the red box in Fig. 11a) to estimate a blur kernel, and then computed the latent image using the estimated kernel. Although the image contains color figures and textures as well as texts, our system can restore different types of regions well. Our result shows sharper texts compared to the previous deblurring method [12] which utilizes only large objects for kernel estimation.

7 Conclusion and Future Work

Previous natural image deblurring methods do not work well for text images due to the lack of consideration of text-specific properties. We analyzed the desired properties for latent text images and the difficulty of applying the properties to text image deblurring. Based on the analysis, we proposed a novel text image deblurring algorithm which explicitly incorporates the text-specific properties into the optimization framework. Experimental results show that our system can generate higher quality results on deblurring text images than previous methods.

In this work, we define the cost function $\rho_a(\cdot)$ in Eq. (8) based on text-related properties. However, we believe that as a general framework, our approach can be extended and applied to other domains, by defining the cost function according to other domain-specific priors. As future work, we plan to explore along this direction to develop specialized deblurring algorithms for specific types of images, such as medical images.

Acknowledgments. We thank the anonymous reviewers for their valuable comments, and Xiaogang Chen for sharing the images used in his paper [1]. This work was supported in part by Industrial Strategic Technology Development Program of KEIT (KI001820) and Basic Science Research Program of NRF (2010-0019523).

References

1. Chen, X., He, X., Yang, J., Wu, Q.: An effective document image deblurring algorithm. In: Proc. CVPR 2011. (2011) 369–376
2. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graphics **25** (2006) 787–794
3. Levin, A.: Blind motion deblurring using image statistics. In: Proc. Advances in Neural Information Processing Systems. (2006) 841–848
4. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graphics **27** (2008) 73:1–73:10
5. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graphics **28** (2009) 145:1–145:8
6. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Understanding and evaluating blind deconvolution algorithms. In: Proc. CVPR 2009. (2009) 1964–1971
7. Levin, A., Weiss, Y., Freeman, B., Durand, F.: Efficient approximations to the marginal likelihood in blind deconvolution. In: Proc. CVPR 2011. (2011)
8. Banerjee, J., Namboodiri, A., Jawahar, C.: Contextual restoration of severely degraded document images. In: Proc. CVPR 2009. (2009) 517–524
9. Li, T.H., Lii, K.S.: A joint estimation approach for two-tone image deblurring by blind deconvolution. IEEE Trans. Image Processing **11** (2002) 847–858
10. Donaldson, K., Myers, G.: Bayesian super-resolution of text in video with a text-specific bimodal prior. In: Proc. CVPR 2005. Volume 1. (2005) 1188–1195
11. Lam, E.: Blind bi-level image restoration with iterated quadratic programming. IEEE Trans. Circuits and Systems II: Express Briefs **54** (2007) 52–56
12. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: Proc. ECCV 2010. (2010) 157–170
13. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: Proc. CVPR 2005. (2005) 860–867
14. Weiss, Y., Freeman, W.T.: What makes a good model of natural images? In: Proc. CVPR 2007. (2007) 1–8
15. Joshi, N., Szeliski, R., Kreigman, D.: PSF estimation using sharp edge prediction. In: Proc. CVPR 2008. (2008) 1–8
16. Kasar, T., Kumar, J., Ramakrishnan, A.G.: Font and background color independent text binarization. In: Proc. Intl. Workshop on Camera Based Document Analysis and Recognition (ICDAR). (2007) 3–9
17. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Trans. Graphics **26** (2007) 70:1–70:9
18. Geman, D., Yang, C.: Nonlinear image recovery with half-quadratic regularization. IEEE Trans. Image Processing **4** (1995) 932–946
19. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Img. Sci. **1** (2008) 248–272
20. Xu, L., Lu, C., Xu, Y., Jia, J.: Image smoothing via L0 gradient minimization. ACM Trans. Graphics (2011)
21. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proc. CVPR 2010. (2010) 2963–2970